

Abstract Interpretation and (Hyper)-Logics

Patrick Cousot

Courant Institute, New York University

Abstract Interpretation

- **Abstract interpretation** is a theory formalizing the **abstraction of discrete systems properties** (such as the semantics of programming languages)

Abstract Interpretation

- **Abstract interpretation** has been **used to**
 - formalize the **hierarchy of program semantics** (e.g. operational, denotational, axiomatic, ...)
 - formalize **program refinement** techniques
 - design **sound program analysis** methods (including model-checking, runtime and static analysis, typing, ...)
- We show that it **can also be used to design program logics**

Program logics

- **Program logics** formally define what must be proved to ensure that **the semantics** of programs of a language **has a specified property**
e.g. Hoare logic $\{P\} C \{Q\}$
- Program logics must be **sound** (and **complete**)
- So program logics define the soundness of static analyzers

Content

- **Part I:** logics to prove properties of **any execution** (e.g. safety, termination)
- **Part II:** logics to prove properties of **any set of executions** (e.g. security, privacy)

Part I:

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

Patrick Cousot:

Calculational Design of [In]Correctness Transformational Program Logics by Abstract Interpretation.
Proc. ACM Program. Lang. 8(POPL): 175-208 (2024)

Objective

Method to design program transformational logics

Transformational logic = Hoare style logics $\{P\} S \{Q\}$

Method to design a program transformational logics

- I. Define the **natural relational semantics** $\llbracket S \rrbracket_{\perp}$ 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** $\llbracket S \rrbracket_{\perp}$ 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)
3. Calculate the theory $\alpha(\{\llbracket S \rrbracket_{\perp}\})$ 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** $\llbracket S \rrbracket_{\perp}$ 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)
3. Calculate the theory $\alpha(\{\llbracket S \rrbracket_{\perp}\})$ 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

Two simple examples*:

(1) Hoare (HL)

(2) incorrectness logic (IL. aka
reverse Hoare logic)

* in ``On the Design of Program Logics'' to appear in Proc. Festschrift Podelski 65th Birthday. Springer (2024).

General Idea

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

} theory
}

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
- } theory
- } proof system
- } theory
- } proof system

I. Angelic relational semantics $\llbracket S \rrbracket^e$

- Syntax^{*}:

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

- States: Σ

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

ends



* plus unbounded nondeterminism, breaks, and nontermination \perp in the POPL24 paper.

I. Angelic relational semantics $\llbracket S \rrbracket$ (in deductive form)

- Notations using judgements:

- $\sigma \vdash S \xRightarrow{e} \sigma'$ for $\langle \sigma, \sigma' \rangle \in \llbracket S \rrbracket^e$

- $\sigma \vdash \text{while}(B) \ S \xRightarrow{i} \sigma'$ for σ leads to σ' after 0 or more iterations

I. Angelic relational semantics $\llbracket S \rrbracket$ (in deductive form)

- Notations using judgements:

- $\sigma \vdash S \xRightarrow{e} \sigma'$ for $\langle \sigma, \sigma' \rangle \in \llbracket S \rrbracket^e$

- $\sigma \vdash \text{while}(B) S \xRightarrow{i} \sigma'$ for σ leads to σ' after 0 or more iterations

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

$$\begin{array}{ll} \text{(a)} & \sigma \vdash W \xRightarrow{i} \sigma \\ \text{(b)} & \frac{\mathcal{B}[\![B]\!]\sigma, \quad \sigma \vdash S \xRightarrow{e} \sigma', \quad \sigma' \vdash W \xRightarrow{i} \sigma''}{\sigma \vdash W \xRightarrow{i} \sigma''} \end{array} \quad (2)$$

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

*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* $W = \text{while}(B) \ S :$

$$F^e(X) \triangleq \text{id} \cup (\llbracket B \rrbracket \circ \llbracket S \rrbracket^e \circ X), \quad X \in \wp(\Sigma \times \Sigma) \quad (49)$$

$$\llbracket \text{while } (B) \ S \rrbracket^e \triangleq \text{lfp}^{\subseteq} F^e \circ \llbracket \neg B \rrbracket \quad (\text{no break}) \quad (51)$$

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

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

Aczel correspondence between deductive systems and fixpoints

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

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

- Subset of the universe \mathcal{U} defined by R :

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

$$= \text{lfp}^{\sqsubseteq} F(R)$$

$$F(R)X \triangleq \left\{ c \mid \exists \frac{P}{c} \in R . P \subseteq X \right\}$$

← model theoretic (gfp for coinduction)

← consequence operator

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, $\frac{\emptyset}{c}$ axiom)

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

- Subset of the universe \mathcal{U} defined by R :

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

proof theoretic \downarrow

$$= \text{lfp}^{\sqsubseteq} F(R)$$

\leftarrow model theoretic (gfp for coinduction)

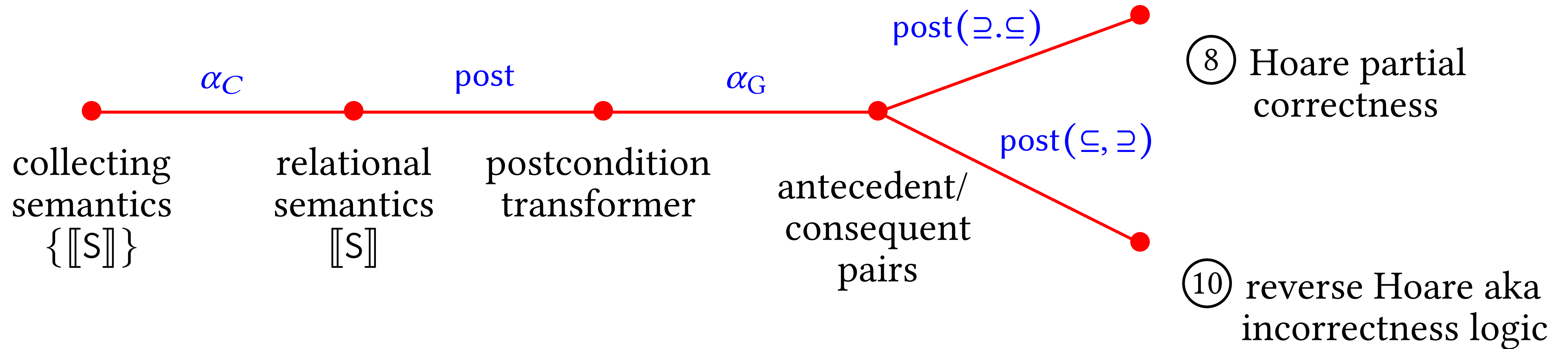
$$F(R)X \triangleq \left\{ c \mid \exists \frac{P}{c} \in R . P \subseteq X \right\}$$

\leftarrow consequence operator

- Deductive system defining $\text{lfp}^{\sqsubseteq} F$: $R_F \triangleq \left\{ \frac{P}{c} \mid P \subseteq \mathcal{U} \wedge c \in F(P) \right\}$

2. Abstraction (much simplified)

- The composition of these abstractions is



- This is an oversimplification of Fig. I of the POPL24 paper, forgetting about nontermination including total correctness and relational predicates

2. Abstraction (much simplified)

- Hyper properties to properties abstraction:

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

2. Abstraction (much simplified)

- Hyper properties to properties abstraction:

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

- Post-image isomorphism:

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

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

2. Abstraction (much simplified)

- Hyper properties to properties abstraction:

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

- Post-image isomorphism:

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

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

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

$$\langle \wp(\Sigma) \rightarrow \wp(\Sigma), = \rangle \xrightleftharpoons[\alpha_G]{\gamma_G} \langle \wp_{\text{fun}}(\wp(\Sigma) \times \wp(\Sigma)), = \rangle \quad f \in \wp(\Sigma) \rightarrow \wp(\Sigma)$$

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

$$\gamma_G(R) \triangleq \lambda P \cdot (Q \text{ such that } \langle P, S \rangle \in R)$$

2. Abstraction (much simplified)

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

$$\begin{aligned}\mathcal{T}(s) &\triangleq \alpha_G \circ \text{post} \circ \alpha_C(\{[s]\}) \\ &= \{ \langle P, \text{post}[s]P \rangle \mid P \in \wp(\Sigma) \}\end{aligned}$$

2. Abstraction (much simplified)

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

$$\begin{aligned}\mathcal{T}(s) &\triangleq \alpha_G \circ \text{post} \circ \alpha_C(\{[s]\}) \\ &= \{ \langle P, \text{post}[s]P \rangle \mid P \in \wp(\Sigma) \}\end{aligned}$$

- 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 $\langle C, \sqsubseteq \rangle \xrightleftharpoons[\alpha]{\gamma} \langle A, \preceq \rangle$ 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 $\alpha(\text{lfp}^{\sqsubseteq} f) = \text{lfp}^{\preceq} \bar{f}$ (while semi-commutation $\alpha \circ f \preceq \bar{f} \circ \alpha$ implies $\alpha(\text{lfp}^{\sqsubseteq} f) \preceq \text{lfp}^{\preceq} \bar{f}$).

2. Abstraction (much simplified)

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

THEOREM II.2.1 (FIXPOINT ABSTRACTION). If $\langle C, \sqsubseteq \rangle \xleftrightarrow[\alpha]{\gamma} \langle A, \preceq \rangle$ 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 $\alpha(\text{lfp}^{\sqsubseteq} f) = \text{lfp}^{\preceq} \bar{f}$ (while semi-commutation $\alpha \circ f \preceq \bar{f} \circ \alpha$ implies $\alpha(\text{lfp}^{\sqsubseteq} f) \preceq \text{lfp}^{\preceq} \bar{f}$).

- 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 $W = \text{while } (B) \ S :$

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

1 PROPERTIES OF STRONGEST POSTCONDITIONS

LEMMA 1.1 (COMPOSITION). $\text{post}(X \mathbin{\text{;}} Y) = \text{post}(Y) \circ \text{post}(X)$.

PROOF OF LEM. 1.1.

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

LEMMA 1.2 (TEST). $\text{post}[\![\mathbf{B}]\!]P = P \cap \mathcal{B}[\![\mathbf{B}]\!]$.

PROOF OF LEM. 1.2.

$$\begin{aligned}
& \text{post}[\![\mathbf{B}]\!]P \\
&= \{\sigma' \mid \exists \sigma \in P . \langle \sigma, \sigma' \rangle \in [\![\mathbf{B}]\!]\} \quad \text{\textit{\text{def. post}}\text{\textit{}}} \\
&= \{\sigma \mid \sigma \in P \wedge \sigma \in \mathcal{B}[\![\mathbf{B}]\!]\} \quad \text{\textit{\text{def. } [\![\mathbf{B}]\!] \triangleq \{\langle \sigma, \sigma \rangle \mid \sigma \in \mathcal{B}[\![\mathbf{B}]\!]\}\text{\textit{}}}\text{\textit{}}} \\
&= P \cap \mathcal{B}[\![\mathbf{B}]\!] \quad \text{\textit{\text{def. intersection } \cup}\text{\textit{}}} \quad \square
\end{aligned}$$

LEMMA 1.3 (STRONGEST POSTCONDITION). $\mathcal{T}(S) = \alpha_G \circ \text{post}[\![S]\!] = \{\langle P, \text{post}[\![S]\!]P \rangle \mid P \in \wp(\Sigma)\}$.

PROOF OF LEM. 1.3.

$$\begin{aligned}
& \mathcal{T}(S) \\
&= \alpha_G \circ \text{post} \circ \alpha_I \circ \alpha_C(\{[\![S]\!]_{\perp}\}) \quad \text{\textit{\text{def. } \mathcal{T}\text{\textit{}}}\text{\textit{}}} \\
&= \alpha_G \circ \text{post} \circ \alpha_I([\![S]\!]_{\perp}) \quad \text{\textit{\text{def. } \alpha_C}\text{\textit{}}} \\
&= \alpha_G \circ \text{post}([\![S]\!]_{\perp} \cap (\Sigma \times \Sigma)) \quad \text{\textit{\text{def. } \alpha_I}\text{\textit{}}} \\
&= \alpha_G \circ \text{post}[\![S]\!] \quad \text{\textit{\text{def. (1) of the angelic semantics } [\![S]\!]}\text{\textit{}}} \\
&= \{\langle P, \text{post}[\![S]\!]P \rangle \mid P \in \wp(\Sigma)\} \quad \text{\textit{\text{def. } \alpha_G}\text{\textit{}}} \quad \square
\end{aligned}$$

LEMMA 1.4 (STRONGEST POSTCONDITION OVER APPROXIMATION).

$$\mathcal{T}_{\text{HL}}(S) \triangleq \text{post}(\supseteq, \subseteq) \circ \mathcal{T}(S) = \{\langle P, Q \rangle \mid \text{post}[\![S]\!]P \subseteq Q\} = \text{post}(=, \subseteq) \circ \mathcal{T}(S)$$

PROOF OF LEM. 1.4.

$$\begin{aligned}
& \text{post}(\supseteq, \subseteq) \circ \mathcal{T}(S) \\
&= \text{post}(\supseteq, \subseteq)(\mathcal{T}(S)) \quad \text{\textit{\text{def. function composition } \circ}\text{\textit{}}} \\
&= \text{post}(\supseteq, \subseteq)(\{\langle P, \text{post}[\![S]\!]P \rangle \mid P \in \wp(\Sigma)\}) \quad \text{\textit{\text{Lem. 1.3}}\text{\textit{}}} \\
&= \{\langle P', Q' \rangle \mid \exists \langle P, Q \rangle \in \{\langle P, \text{post}[\![S]\!]P \rangle \mid P \in \wp(\Sigma)\} . \langle \langle P, Q \rangle, \langle P', Q' \rangle \rangle \in \supseteq, \subseteq\} \quad \text{\textit{\text{def. (10) of post}}\text{\textit{}}} \\
&= \{\langle P', Q' \rangle \mid \exists P . \langle \langle P, \text{post}[\![S]\!]P \rangle, \langle P', Q' \rangle \rangle \in \supseteq, \subseteq\} \quad \text{\textit{\text{def. } \in}\text{\textit{}}} \\
&= \{\langle P', Q' \rangle \mid \exists P . \langle P, \text{post}[\![S]\!]P \rangle \supseteq \langle P', Q' \rangle\} \quad \text{\textit{\text{def. } \in}\text{\textit{}}} \\
&= \{\langle P', Q' \rangle \mid \exists P . P \supseteq P' \wedge \text{post}[\![S]\!]P \subseteq Q'\} \quad \text{\textit{\text{def. } \supseteq, \subseteq}\text{\textit{}}} \\
&= \{\langle P', Q' \rangle \mid \exists P . P' \subseteq P \wedge \text{post}[\![S]\!]P \subseteq Q'\} \quad \text{\textit{\text{def. } \supseteq}\text{\textit{}}}
\end{aligned}$$

$$\begin{aligned}
&= \{\langle P', Q' \rangle \mid \text{post}[\![S]\!]P' \subseteq Q'\} \\
&\quad \text{\textit{\text{(\subseteq) by Galois connection (12), post is increasing so that } } P' \subseteq P \wedge \text{post}[\![S]\!]P \subseteq Q' \text{ implies } \\
&\quad \text{post}[\![S]\!]P' \subseteq \text{post}[\![S]\!]P \wedge \text{post}[\![S]\!]P \subseteq Q' \text{ hence } \text{post}[\![S]\!]P' \subseteq Q' \text{ by transitivity;}} \\
&\quad \text{\textit{(\supseteq) take } } P = P' \text{\textit{}}\text{\textit{}}} \\
&= \{\langle P', Q' \rangle \mid \exists P . P' = P \wedge \text{post}[\![S]\!]P \subseteq Q'\} \quad \text{\textit{\text{def. } =}\text{\textit{}}} \\
&= \{\langle P', Q' \rangle \mid \exists P . \langle P, \text{post}[\![S]\!]P \rangle =, \subseteq \langle P', Q' \rangle\} \quad \text{\textit{\text{def. } =, \subseteq}\text{\textit{}}} \\
&= \{\langle P', Q' \rangle \mid \exists P . \langle \langle P, \text{post}[\![S]\!]P \rangle, \langle P', Q' \rangle \rangle \in =, \subseteq\} \quad \text{\textit{\text{def. } \in}\text{\textit{}}} \\
&= \{\langle P', Q' \rangle \mid \exists \langle P, Q \rangle \in \{\langle P, \text{post}[\![S]\!]P \rangle \mid P \in \wp(\Sigma)\} . \langle \langle P, Q \rangle, \langle P', Q' \rangle \rangle \in =, \subseteq\} \quad \text{\textit{\text{def. } \in}\text{\textit{}}} \\
&= \{\langle P', Q' \rangle \mid \exists \langle P, Q \rangle \in \mathcal{T}(S) . \langle \langle P, Q \rangle, \langle P', Q' \rangle \rangle \in =, \subseteq\} \quad \text{\textit{\text{Lem. 1.3}}\text{\textit{}}} \\
&= \text{post}(=, \subseteq)(\mathcal{T}(S)) \quad \text{\textit{\text{def. (10) of post}}\text{\textit{}}} \\
&= \text{post}(=, \subseteq) \circ \mathcal{T}(S) \quad \text{\textit{\text{def. function composition } \circ}\text{\textit{}}} \quad \square
\end{aligned}$$

For simplicity, we consider conditional iteration $\mathbf{W} = \text{while } (\mathbf{B}) \ S$ with no break.

LEMMA 1.5 (COMMUTATION). $\text{post} \circ F'^e = \bar{F}^e \circ \text{post}$ where $\bar{F}^e(X) \triangleq \text{id} \dot{\cup} (\text{post}([\![\mathbf{B}]\!] \mathbin{\text{;}} [\![S]\!]^e) \circ X)$ and $F'^e \triangleq \lambda X \bullet \text{id} \cup (X \mathbin{\text{;}} [\![\mathbf{B}]\!] \mathbin{\text{;}} [\![S]\!]^e)$, $X \in \wp(\Sigma \times \Sigma)$ by (70).

PROOF OF LEM. 1.5.

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

LEMMA 1.6 (POINTWISE COMMUTATION). $\forall X \in \wp(\Sigma) \rightarrow \wp(\Sigma) . \forall P \in \wp(\Sigma) . \bar{F}^e(X)P \triangleq \bar{\bar{F}}_P^e(X(P))$ where $\bar{\bar{F}}_P^e(X) \triangleq P \cup \text{post}([\![\mathbf{B}]\!] \mathbin{\text{;}} [\![S]\!]^e)X$.

PROOF OF LEM. 1.6.

$$\begin{aligned}
& \bar{F}^e(X)P \\
&= (\text{id} \dot{\cup} (\text{post}([\![\mathbf{B}]\!] \mathbin{\text{;}} [\![S]\!]^e) \circ X))P \quad \text{\textit{\text{def. } } } \bar{F}^e \text{\textit{}}} \\
&= \text{id}(P) \cup (\text{post}([\![\mathbf{B}]\!] \mathbin{\text{;}} [\![S]\!]^e) \circ X)(P) \quad \text{\textit{\text{pointwise def. } \dot{\cup} \text{ and function composition } \circ}\text{\textit{}}} \\
&= P \cup \text{post}([\![\mathbf{B}]\!] \mathbin{\text{;}} [\![S]\!]^e)(X(P)) \quad \text{\textit{\text{def. identity id and function application}}\text{\textit{}}} \\
&= \bar{\bar{F}}_P^e(X(P)) \quad \text{\textit{\text{def. } } } \bar{\bar{F}}_P^e(X) \triangleq P \cup \text{post}([\![\mathbf{B}]\!] \mathbin{\text{;}} [\![S]\!]^e)X \text{\textit{}}} \quad \square
\end{aligned}$$

THEOREM 1.7 (ITERATION STRONGEST POSTCONDITION). $\text{post}[\![\mathbf{W}]\!]P = \text{post}[\![\neg \mathbf{B}]\!](\text{lfp}^{\subseteq} \bar{\bar{F}}_P^e)$ where $\bar{\bar{F}}_P^e(X) \triangleq P \cup \text{post}([\![\mathbf{B}]\!] \mathbin{\text{;}} [\![S]\!]^e)X$.

PROOF OF TH. 1.7.

$$\begin{aligned}
& \text{post}[\![\mathbf{W}]\!] \\
&= \text{post}(\text{lfp}^{\subseteq} F^e \mathbin{\text{;}} [\![\neg \mathbf{B}]\!]) \quad \text{\textit{\text{def. (49) of } } } [\![\mathbf{W}]\!] \text{ in absence of break}\text{\textit{}}} \\
&= \text{post}[\![\neg \mathbf{B}]\!] \circ \text{post}(\text{lfp}^{\subseteq} F^e) \quad \text{\textit{\text{composition Lem. 1.1}}\text{\textit{}}} \\
&= \text{post}[\![\neg \mathbf{B}]\!] \circ \text{post}(\text{lfp}^{\subseteq} F'^e) \quad \text{\textit{\text{since } } } \text{lfp}^{\subseteq} F^e = \text{lfp}^{\subseteq} F'^e \text{ in (70)}\text{\textit{}}} \\
&= \text{post}[\![\neg \mathbf{B}]\!](\text{lfp}^{\subseteq} \bar{F}^e) \quad \text{\textit{\text{commutation Lem. 1.5 and fixpoint abstraction Th. II.2.2}}\text{\textit{}}}
\end{aligned}$$

$$\begin{aligned}
&= \text{post}[\![\neg \mathbf{B}]\!] \circ \lambda P \bullet \text{lfp}^{\subseteq} \bar{\bar{F}}_P^e \\
&\quad \text{\textit{\text{pointwise commutation Lem. 1.6 and pointwise abstraction Cor. II.2.2}}\text{\textit{}}} \quad \square
\end{aligned}$$

COROLLARY 1.8 (CONDITIONAL ITERATION STRONGEST POSTCONDITION GRAPH). $\mathcal{T}(\mathbf{W}) = \{\langle P, \text{post}[\![\neg \mathbf{B}]\!](\text{lfp}^{\subseteq} \bar{\bar{F}}_P^e) \rangle \mid P \in \wp(\Sigma)\}$ where $\bar{\bar{F}}_P^e(X) \triangleq P \cup \text{post}([\![\mathbf{B}]\!] \mathbin{\text{;}} [\![S]\!]^e)X$.

PROOF OF COR. 1.8.

$$\begin{aligned}
& \mathcal{T}(\mathbf{W}) \\
&= \alpha_G \circ \text{post}([\![\mathbf{W}]\!]) \quad \text{\textit{\text{Lem. 1.3}}\text{\textit{}}} \\
&= \alpha_G \circ \text{post}[\![\neg \mathbf{B}]\!] \circ \lambda P \bullet \text{lfp}^{\subseteq} \bar{\bar{F}}_P^e \quad \text{\textit{\text{Th. 1.7}}\text{\textit{}}} \\
&= \{\langle P, \text{post}[\![\neg \mathbf{B}]\!](\text{lfp}^{\subseteq} \bar{\bar{F}}_P^e) \rangle \mid P \in \wp(\Sigma)\} \quad \text{\textit{\text{def. (7) of } } } \alpha_G \text{\textit{}}} \quad \square
\end{aligned}$$

3. Approximation

- The component wise approximation:

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

3. Approximation

- The component wise approximation:

$$\langle x, y \rangle \sqsubseteq, \leq \langle x', y' \rangle \triangleq x \sqsubseteq x' \wedge y \leq y'$$

- The over approximation abstraction for HL:

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

$$\mathcal{T}_{\text{HL}}(S) \triangleq \text{post}(\supseteq, \sqsubseteq) \circ \mathcal{T}(S)$$

3. Approximation

- The component wise approximation:

$$\langle x, y \rangle \sqsubseteq, \leq \langle x', y' \rangle \triangleq x \sqsubseteq x' \wedge y \leq y'$$

- The **over** approximation abstraction for HL:

$$\text{post}(\sqsubseteq, \supseteq) = \lambda R \cdot \{ \langle P, Q \rangle \mid \exists \langle P', Q' \rangle \in R . P \sqsubseteq P' \wedge Q' \sqsubseteq Q \}$$

$$\mathcal{T}_{\text{HL}}(S) \triangleq \text{post}(\supseteq, \sqsubseteq) \circ \mathcal{T}(S)$$

- The (order dual) **under** approximation abstraction for IL:

$$\text{post}(\supseteq, \sqsubseteq) = \lambda R \cdot \{ \langle P, Q \rangle \mid \exists \langle P', Q' \rangle \in R . P' \sqsubseteq P \wedge Q \sqsubseteq Q' \}$$

$$\mathcal{T}_{\text{RL}}(S) \triangleq \text{post}(\sqsubseteq, \supseteq) \circ \mathcal{T}(S)$$

- Shows what is 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 (\gggggggg 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 (\gggggggg 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 $\langle L, \sqsubseteq, \perp, \top, \sqcup, \sqcap \rangle$ be a complete lattice, $f \in L \xrightarrow{i} L$ be increasing, and $p \in L$. Then $\text{lfp}^\sqsubseteq f \sqsubseteq p$ if and only if $\exists i \in L . f(i) \sqsubseteq i \wedge i \sqsubseteq p$.

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 $\langle L, \sqsubseteq, \perp, \sqcup \rangle$. $P \sqsubseteq \text{lfp}^\sqsubseteq f$, if and only if there exists an increasing transfinite sequence $\langle X^\delta, \delta \in \mathbb{O} \rangle$ such that

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

δ bounded by ω for continuous f .

5. Computational design of HL

- Theory of HL (for iteration):

$$\begin{aligned}\mathcal{T}_{HL}(W) &\triangleq \text{post}(\exists.\sqsubseteq) \circ \mathcal{T}(W) \\ &= \{ \langle P, Q \rangle \mid \exists I . P \sqsubseteq I \wedge \langle I \cap \mathcal{B}[\![B]\!], I \rangle \in T_{HL}(S) \wedge (I \cap \neg \mathcal{B}[\![B]\!]) \sqsubseteq Q \}\end{aligned}$$

5. Computational design of HL

- Theory of HL (for iteration):

$$\begin{aligned}\mathcal{T}_{HL}(W) &\triangleq \text{post}(\exists.\sqsubseteq) \circ \mathcal{T}(W) \\ &= \{ \langle P, Q \rangle \mid \exists I . P \sqsubseteq I \wedge \langle I \cap \mathcal{B}[[B]], I \rangle \in T_{HL}(S) \wedge (I \cap \neg \mathcal{B}[[B]]) \sqsubseteq Q \}\end{aligned}$$

- HL proof system:

THEOREM 3 (HOARE RULES FOR CONDITIONAL ITERATION).

$$\frac{P \sqsubseteq I, \{I \cap \mathcal{B}[[B]]\} S \{I\}, (I \cap \neg \mathcal{B}[[B]]) \sqsubseteq Q}{\{P\} \text{ while } (B) S \{Q\}}$$

2 CALCULATIONAL DESIGN OF HOARE LOGIC HL

2.1 Calculational Design of Hoare Logic Theory

THEOREM 2.1 (THEORY OF HOARE LOGIC HL).

$$\begin{aligned}\mathcal{T}_{HL}(\mathbb{W}) &\triangleq \text{post}(\exists.\subseteq) \circ \mathcal{T}(\mathbb{W}) \\ &= \{ \langle P, Q \rangle \mid \exists I . P \subseteq I \wedge \langle I \cap \mathcal{B}[\mathbb{B}], I \rangle \in T_{HL}(S) \wedge (I \cap \neg \mathcal{B}[\mathbb{B}]) \subseteq Q \}\end{aligned}$$

PROOF OF TH. 2.1 .

$$\begin{aligned}&\mathcal{T}_{HL}(\mathbb{W}) \\&= \text{post}(\exists.\subseteq) \circ \mathcal{T}(\mathbb{W}) && \text{\{def. } \mathcal{T}_{HL}\}} \\&= \text{post}(=\, \subseteq) \circ \mathcal{T}(\mathbb{W}) && \text{\{Lem. 1.4\}} \\&= \{ \langle P', Q' \rangle \mid \langle P, Q \rangle \in \mathcal{T}(\mathbb{W}) . \langle P, Q \rangle =, \subseteq \langle P', Q' \rangle \} && \text{\{def. post\}} \\&= \{ \langle P', Q' \rangle \mid \langle P, Q \rangle \in \mathcal{T}(\mathbb{W}) . P = P' \wedge Q \subseteq Q' \} && \text{\{component wise def. =, \subseteq\}} \\&= \{ \langle P, Q' \rangle \mid \exists Q . \langle P, Q \rangle \in \mathcal{T}(\mathbb{W}) . Q \subseteq Q' \} && \text{\{def. =\}} \\&= \{ \langle P, Q' \rangle \mid \exists Q . \text{post}[\neg \mathbb{B}](\text{lfp}^e \bar{F}_P^e) \subseteq Q \wedge Q \subseteq Q' \} && \text{\{Th. 1.7\}} \\&= \{ \langle P, Q' \rangle \mid \exists Q . \text{post}[\neg \mathbb{B}](\text{lfp}^e \bar{F}_P^e) \subseteq Q' \} \\&\quad \text{\{(\subseteq) } \exists Q . \text{post}[\neg \mathbb{B}](\text{lfp}^e \bar{F}_P^e) \subseteq Q \wedge Q \subseteq Q' \text{ and transitivity;} \\&\quad \text{\{(\supseteq) take } Q = Q' \}} \\&= \{ \langle P, Q' \rangle \mid \exists Q . \text{lfp}^e \bar{F}_P^e \subseteq Q \wedge \text{post}[\neg \mathbb{B}](Q) \subseteq Q' \} \\&\quad \text{\{(\subseteq) take } Q = \text{lfp}^e \bar{F}_P^e; \text{\{(\supseteq) post}[\neg \mathbb{B}] \text{ is increasing by (12)\}}\}} \\&= \{ \langle P, Q' \rangle \mid \exists Q . \exists I . \bar{F}_P^e(I) \subseteq I \wedge I \subseteq Q \wedge \text{post}[\neg \mathbb{B}](Q) \subseteq Q' \} && \text{\{Park fixpoint induction Th. II.3.1\}} \\&= \{ \langle P, Q' \rangle \mid \exists I . \bar{F}_P^e(I) \subseteq I \wedge \text{post}[\neg \mathbb{B}](I) \subseteq Q' \} \\&\quad \text{\{(\subseteq) } I \subseteq Q \text{ implies } \text{post}[\neg \mathbb{B}](I) \subseteq \text{post}[\neg \mathbb{B}](Q) \text{ since } \text{post}[\neg \mathbb{B}] \text{ is increasing by (12) hence} \\&\quad \text{post}[\neg \mathbb{B}](I) \subseteq Q' \text{ by transitivity;} \\&\quad \text{\{(\supseteq) take } Q = I \}} \\&= \{ \langle P, Q \rangle \mid \exists I . P \cup \text{post}([\mathbb{B}] \circ [\mathbb{S}]^e)(I) \subseteq I \wedge \text{post}[\neg \mathbb{B}](I) \subseteq Q \} && \text{\{renaming, def. } \bar{F}_P^e\}} \\&= \{ \langle P, Q \rangle \mid \exists I . P \cup \text{post}([\mathbb{B}] \circ [\mathbb{S}])(I) \subseteq I \wedge \text{post}[\neg \mathbb{B}](I) \subseteq Q \} && \text{\{[\mathbb{S}]^e = [\mathbb{S}] in absence of breaks\}} \\&= \{ \langle P, Q \rangle \mid \exists I . P \subseteq I \wedge \text{post}([\mathbb{B}] \circ [\mathbb{S}])I \subseteq I \wedge \text{post}[\neg \mathbb{B}](I) \subseteq Q \} && \text{\{def. \subseteq and \cup\}} \\&= \{ \langle P, Q \rangle \mid \exists I . P \subseteq I \wedge \text{post}[\mathbb{S}](\text{post}[\mathbb{B}]I) \subseteq I \wedge \text{post}[\neg \mathbb{B}](I) \subseteq Q \} && \text{\{composition Lem. 1.1\}} \\&= \{ \langle P, Q \rangle \mid \exists I . P \subseteq I \wedge \text{post}[\mathbb{S}](I \cap \mathcal{B}[\mathbb{B}]) \subseteq I \wedge (I \cap \neg \mathcal{B}[\mathbb{B}]) \subseteq Q \} && \text{\{test Lem. 1.2\}} \\&= \{ \langle P, Q \rangle \mid \exists I . P \subseteq I \wedge \langle I \cap \mathcal{B}[\mathbb{B}], I \rangle \in \{ \langle P, Q \rangle \mid \text{post}[\mathbb{S}]P \subseteq Q \} \wedge (I \cap \neg \mathcal{B}[\mathbb{B}]) \subseteq Q \} && \text{\{def. \in\}} \\&= \{ \langle P, Q \rangle \mid \exists I . P \subseteq I \wedge \langle I \cap \mathcal{B}[\mathbb{B}], I \rangle \in \text{post}(=\, \subseteq) \circ \mathcal{T}(S) \wedge (I \cap \neg \mathcal{B}[\mathbb{B}]) \subseteq Q \} && \text{\{Lem. 1.4\}} \\&= \{ \langle P, Q \rangle \mid \exists I . P \subseteq I \wedge \langle I \cap \mathcal{B}[\mathbb{B}], I \rangle \in T_{HL}(S) \wedge (I \cap \neg \mathcal{B}[\mathbb{B}]) \subseteq Q \} && \text{\{Lem. 1.4\}} \quad \square\end{aligned}$$

2.2 Hoare logic rules

THEOREM 2.2 (HOARE RULES FOR CONDITIONAL ITERATION).

$$\frac{P \subseteq I, \{I \cap \mathcal{B}[\mathbb{B}]\} S \{I\}, (I \cap \neg \mathcal{B}[\mathbb{B}]) \subseteq Q}{\{P\} \text{while } (\mathbb{B}) S \{Q\}} \quad (1)$$

PROOF OF TH. 2.2. We write $\{P\} S \{Q\} \triangleq \langle P, Q \rangle \in \mathcal{T}_{HL}(S)$;

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 $\text{lfp}^e \lambda X . S$ is defined by $\{\frac{\emptyset}{c} \mid c \in S\}$;

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

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):

$$\mathcal{T}_{\text{HL}}(S) = \text{post}(=, \sqsubseteq) \circ \mathcal{T}(S)$$

yields the sound and complete proof system:

\sqsubseteq comes from \longrightarrow Th. II.3.1

$$\frac{P \sqsubseteq I, \quad \{I \cap \mathcal{B}[\![B]\!]\} \text{ s } \{I\}}{\{P\} \text{ while } (B) \text{ s } \{I \cap \neg \mathcal{B}[\![B]\!]\}}$$

$$\frac{\{P\} \text{ s } \{Q\}, \quad Q \sqsubseteq Q'}{\{P\} \text{ s } \{Q'\}}$$


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 $\longrightarrow P \subseteq I, \quad \{I \cap \mathcal{B}[\![B]\!]\} \text{ s } \{I\}$
Th. II.3.1

$$\frac{\{I \cap \mathcal{B}[\![B]\!]\} \text{ s } \{I\}}{\{P\} \text{ while } (B) \text{ s } \{I \cap \neg \mathcal{B}[\![B]\!]\}}$$
$$\frac{\{P\} \text{ s } \{Q\}, \quad Q \subseteq Q'}{\{P\} \text{ s } \{Q'\}}$$


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

~~If $\vdash P\{Q\}R$ and $\vdash S \supset P$ then $\vdash S\{Q\}R$~~

5. Computational design of Incorrectness Logic IL

- Theory of IL (for iteration):

$$\begin{aligned}\mathcal{T}_{IL}(W) &\triangleq \text{post}(\sqsubseteq.\exists) \circ \mathcal{T}(W) \\ &= \{ \langle P, Q \rangle \mid \exists \langle J^n, n \in \mathbb{N} \rangle . J^0 = P \wedge \langle J^n \cap \mathcal{B}[[B]], J^{n+1} \rangle \in \mathcal{T}_{IL}(S) \wedge Q \sqsubseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap \mathcal{B}[[\neg B]] \}\end{aligned}$$

5. Computational design of IL

- Theory of IL (for iteration):

$$\begin{aligned}\mathcal{T}_{IL}(W) &\triangleq \text{post}(\sqsubseteq.\exists) \circ \mathcal{T}(W) \\ &= \{ \langle P, Q \rangle \mid \exists \langle J^n, n \in \mathbb{N} \rangle . J^0 = P \wedge \langle J^n \cap \mathcal{B}[[B]], J^{n+1} \rangle \in \mathcal{T}_{IL}(S) \wedge Q \sqsubseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap \mathcal{B}[[\neg B]] \}\end{aligned}$$

- IL proof system:

THEOREM 5 (IL RULES FOR CONDITIONAL ITERATION).

$$\frac{J^0 = P, [J^n \cap \mathcal{B}[[B]]] S [J^{n+1}], Q \sqsubseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap \mathcal{B}[[\neg B]]}{[P]_{\text{while } (B)} S [Q]}$$

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

Computational design of IL

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).

$$\begin{aligned}\mathcal{T}_{\text{IL}}(\mathbb{W}) &\triangleq \text{post}(\subseteq, \supseteq) \circ \mathcal{T}(\mathbb{W}) \\ &= \{ \langle P, Q \rangle \mid \exists \langle J^n, n \in \mathbb{N} \rangle . J^0 = P \wedge \langle J^n \cap \mathcal{B}[\![\mathbb{B}]\!], J^{n+1} \rangle \in \mathcal{T}_{\text{IL}}(\mathbb{S}) \wedge Q \subseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap \mathcal{B}[\![\neg \mathbb{B}]\!]\} \end{aligned}$$

PROOF OF TH. 3.1.

$$\begin{aligned}\mathcal{T}_{\text{IL}}(\mathbb{W}) &= \text{post}(\subseteq, \supseteq) \circ \mathcal{T}(\mathbb{W}) \quad \{\text{def. } \mathcal{T}_{\text{IL}}\} \\ &= \{ \langle P, Q \rangle \mid Q \subseteq \text{post}[\![\mathbb{W}]\!]P \} \quad \{\subseteq\text{-order dual of Lem. 1.4}\} \\ &= \{ \langle P, Q \rangle \mid Q \subseteq \text{post}[\![\neg \mathbb{B}]\!](\text{lfp}^{\subseteq} \bar{F}_P^e) \} \quad \{\text{Th. 1.7 where } \bar{F}_P^e(X) \triangleq P \cup \text{post}(\llbracket \mathbb{B} \rrbracket ; \llbracket \mathbb{S} \rrbracket^e)X\} \\ &= \{ \langle P, Q \rangle \mid \exists I . Q \subseteq \text{post}[\![\neg \mathbb{B}]\!](I) \wedge I \subseteq \text{lfp}^{\subseteq} \bar{F}_P^e \} \\ &\quad \{\subseteq\} \quad \text{Take } I = \text{lfp}^{\subseteq} \bar{F}_P^e \text{ and reflexivity;} \\ &\quad \{\supseteq\} \quad \text{By Galois connection (12), } \text{post}[\![\neg \mathbb{B}]\!] \text{ is increasing so } Q \subseteq \text{post}[\![\neg \mathbb{B}]\!](I) \subseteq \text{post}[\![\neg \mathbb{B}]\!](\text{lfp}^{\subseteq} \bar{F}_P^e) \text{ and transitivity} \} \\ &= \{ \langle P, Q \rangle \mid \exists I . Q \subseteq \text{post}[\![\neg \mathbb{B}]\!](I) \wedge \exists \langle J^n, n < \omega \rangle . J^0 = \emptyset \wedge J^{n+1} \subseteq \bar{F}_P^e(J^n) \wedge I \subseteq \bigcup_{n < \omega} J^n \} \\ &\quad \{\text{fixpoint underapproximation Th. II.3.6}\} \\ &= \{ \langle P, Q \rangle \mid \exists \langle J^n, n < \omega \rangle . J^0 = \emptyset \wedge J^{n+1} \subseteq \bar{F}_P^e(J^n) \wedge Q \subseteq \text{post}[\![\neg \mathbb{B}]\!](\bigcup_{n < \omega} J^n) \} \\ &\quad \{\subseteq\} \quad \text{By Galois connection (12), } \text{post}[\![\neg \mathbb{B}]\!] \text{ is increasing so } Q \subseteq \text{post}[\![\neg \mathbb{B}]\!](I) \subseteq \text{post}[\![\neg \mathbb{B}]\!](\bigcup_{n < \omega} J^n) \text{ and transitivity;} \\ &\quad \{\supseteq\} \quad \text{take } I = \bigcup_{n < \omega} J^n \} \\ &= \{ \langle P, Q \rangle \mid \exists \langle J^n, n < \omega \rangle . J^0 = \emptyset \wedge J^{n+1} \subseteq (P \cup \text{post}(\llbracket \mathbb{B} \rrbracket ; \llbracket \mathbb{S} \rrbracket^e)(J^n)) \wedge Q \subseteq \text{post}[\![\neg \mathbb{B}]\!](\bigcup_{n < \omega} J^n) \} \\ &\quad \{\text{def. } \bar{F}_P^e\} \\ &= \{ \langle P, Q \rangle \mid \exists \langle J^n, 1 \leq n < \omega \rangle . J^1 = P \wedge J^{n+1} \subseteq \text{post}(\llbracket \mathbb{B} \rrbracket ; \llbracket \mathbb{S} \rrbracket^e)(J^n) \wedge Q \subseteq \text{post}[\![\neg \mathbb{B}]\!](\bigcup_{1 \leq n < \omega} J^n) \} \\ &\quad \{\text{getting rid of } J^0 = \emptyset\} \\ &= \{ \langle P, Q \rangle \mid \exists \langle J^n, n \in \mathbb{N} \rangle . J^0 = P \wedge J^{n+1} \subseteq \text{post}(\llbracket \mathbb{B} \rrbracket ; \llbracket \mathbb{S} \rrbracket^e)(J^n) \wedge Q \subseteq \text{post}[\![\neg \mathbb{B}]\!](\bigcup_{n \in \mathbb{N}} J^n) \} \\ &\quad \{\text{changing } n+1 \text{ to } n\} \\ &= \{ \langle P, Q \rangle \mid \exists \langle J^n, n \in \mathbb{N} \rangle . J^0 = P \wedge J^{n+1} \subseteq \text{post}[\![\mathbb{S}]\!](J^n \cap \mathcal{B}[\![\mathbb{B}]\!]) \wedge Q \subseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap \mathcal{B}[\![\neg \mathbb{B}]\!]\} \\ &\quad \{\text{Lem. 1.2}\} \\ &= \{ \langle P, Q \rangle \mid \exists \langle J^n, n \in \mathbb{N} \rangle . J^0 = P \wedge \langle J^n \cap \mathcal{B}[\![\mathbb{B}]\!], J^{n+1} \rangle \in \{ \langle P', Q' \rangle \mid Q' \subseteq \text{post}[\![\mathbb{S}]\!](P') \} \wedge Q \subseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap \mathcal{B}[\![\neg \mathbb{B}]\!]\} \\ &\quad \{\text{def. } \in\} \\ &= \{ \langle P, Q \rangle \mid \exists \langle J^n, n \in \mathbb{N} \rangle . J^0 = P \wedge \langle J^n \cap \mathcal{B}[\![\mathbb{B}]\!], J^{n+1} \rangle \in \mathcal{T}_{\text{IL}}(\mathbb{S}) \wedge Q \subseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap \mathcal{B}[\![\neg \mathbb{B}]\!]\} \quad \{\text{def. } \mathcal{T}_{\text{IL}}\} \end{aligned}$$

□

3.2 Calculational design of IL rules

$$\frac{J^0 = P, [J^n \cap \mathcal{B}[\![\mathbb{B}]\!]] \mathbb{S} [J^{n+1}], Q \subseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap \mathcal{B}[\![\neg \mathbb{B}]\!]}{[P] \text{ while } (\mathbb{B}) \mathbb{S} [Q]} \quad (2)$$

PROOF. We write $[P] \mathbb{S} [Q] \triangleq \langle P, Q \rangle \in \mathcal{T}_{\text{IL}}(\mathbb{S})$;

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

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

So for $\text{while } (\mathbb{B}) \mathbb{S}$ we have an axiom $\frac{\emptyset}{\{P\} \text{ while } (\mathbb{B}) \mathbb{S} \{Q\}}$ with side condition $J^0 = P, [J^n \cap$

$\mathcal{B}[\![\mathbb{B}]\!]] \mathbb{S} [J^{n+1}], Q \subseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap \mathcal{B}[\![\neg \mathbb{B}]\!]$;

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

Much more in the POPL24 paper

Much more in the POPL24 paper

- Bi-inductive relational semantics with break and non termination (\perp), for **termination and nontermination proofs**

Much more in the POPL24 paper

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

Much more in the POPL24 paper

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

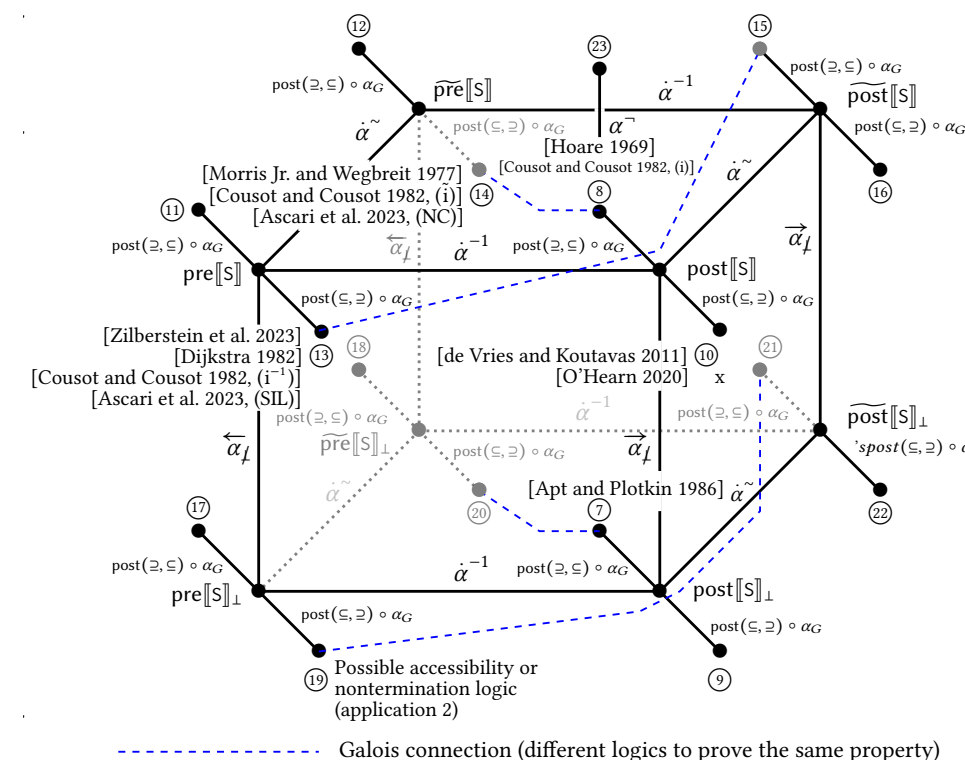


Fig. 3. Taxonomy of assertional logics

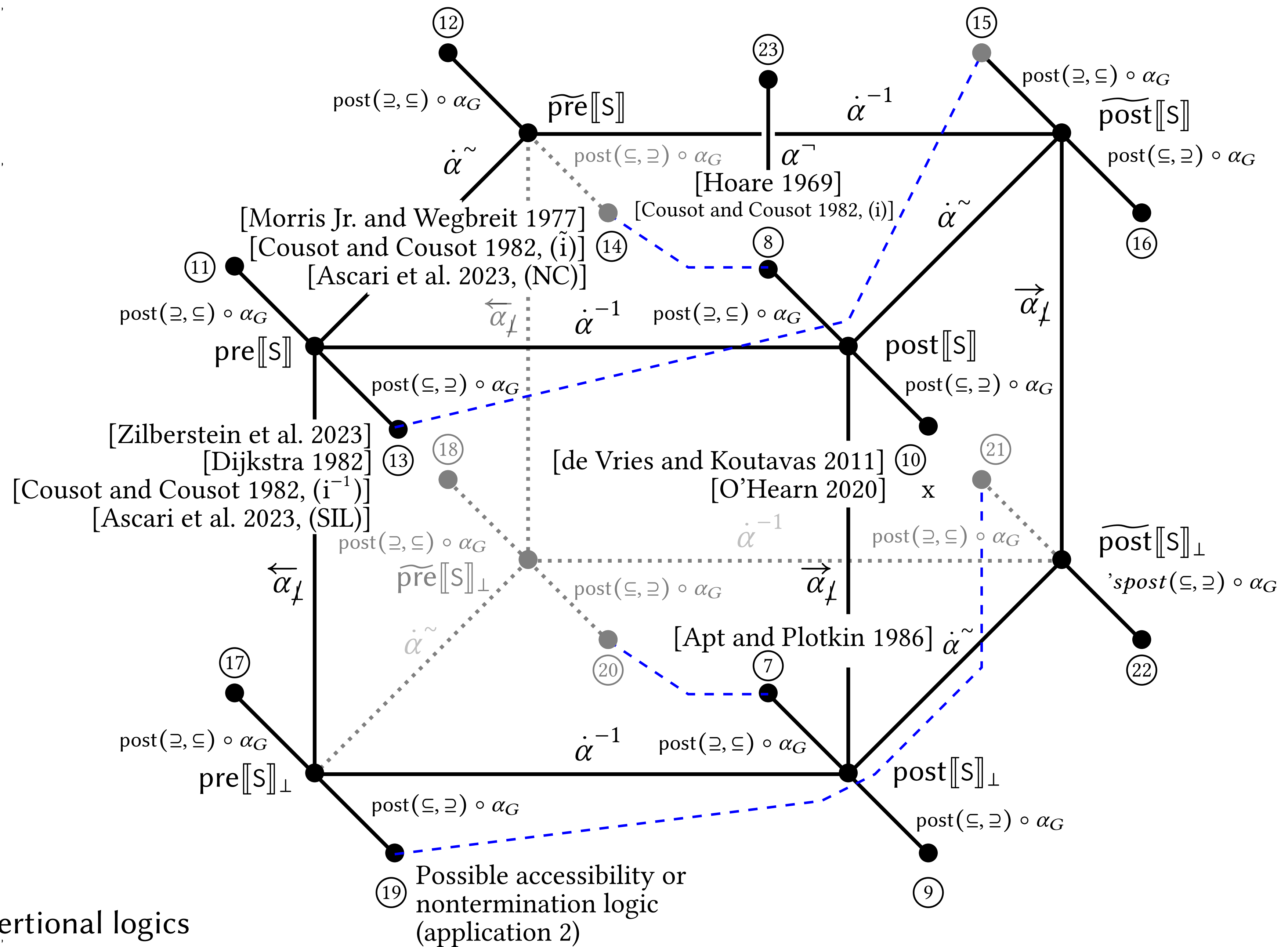


Fig. 3. Taxonomy of assertional logics

Possible accessibility or nontermination logic (application 2)

----- Galois connection (different logics to prove the same property)

Much more in the POPL24 paper

- Many more fixpoint induction principles (including $P \sqsubseteq \text{lfp}^\sqsubseteq F$, $\text{lfp}^\sqsubseteq F \sqsubseteq P$, $P \sqsubseteq \text{gfp}^\sqsubseteq F$, $\text{gfp}^\sqsubseteq F \sqsubseteq P$, $\text{lfp}^\sqsubseteq F \sqcap P \neq \emptyset$, $\text{gfp}^\sqsubseteq F \sqcap P \neq \emptyset$, etc)

Much more in the POPL24 paper

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

$$\{ n = \underline{n} \wedge f = 1 \}$$

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

$$\{ (\underline{n} \geq 0 \wedge f = !\underline{n}) \vee (\underline{n} < 0 \wedge n = f = \perp) \}$$

Much more in the POPL24 paper

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

Much more in the POPL24 paper

- Example II: calculational design of an incorrectness logic including non termination
- A specification for factorial:
$$\{ n \in [-\infty, \infty] \wedge f \in [1, 1] \}$$
$$\text{while } (n \neq 0) \{ f = f * n; n = n - 1; \}$$
$$\{ f \in [1, \infty] \}$$
- 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:
$$\{ n \in [-\infty, \infty] \wedge f \in [1, 1] \}$$
$$\text{while } (n \neq 0) \{ f = f * n; n = n - 1; \}$$
$$\{ f \in [1, \infty] \}$$
- 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)

$$\begin{aligned}\neg(\{P\} s \{Q\}) & \not\Rightarrow [P] s [\neg Q] \\ & \Leftrightarrow \exists R \in \wp(\Sigma) . [P] s [R] \wedge R \cap \neg Q \neq \emptyset \\ & \Leftrightarrow \exists \sigma \in \Sigma . [P] s [\{\sigma\}] \wedge \sigma \notin Q\end{aligned}$$

- The logic $\mathcal{T}_{\overline{HL}}(W) \triangleq \text{post}(\sqsubseteq, \supseteq) \circ \alpha^{-1} \circ \mathcal{T}_{HL}(W) = \alpha^{-1} \circ \mathcal{T}_{HL}(W)$ can be calculated by the design method (and does not need a consequence rule)

Computational design of Hoare incorrectness logic $\overline{\text{HL}}$

4 CALCULATIONAL DESIGN OF HOARE INCORRECTNESS LOGIC

4.1 Calculational Design of Hoare Incorrectness Logic Theory

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

$$\mathcal{T}_{\overline{\text{HL}}}(\mathbb{W}) \triangleq \text{post}(\sqsubseteq, \supseteq) \circ \alpha^- \circ \mathcal{T}_{\text{HL}}(\mathbb{W}) = \alpha^- \circ \mathcal{T}_{\text{HL}}(\mathbb{W}) \quad \text{W = while (B) S}$$

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

PROOF OF TH. 4.1.

$$\begin{aligned} \mathcal{T}_{\overline{\text{HL}}}(\mathbb{W}) &= \text{post}(\sqsubseteq, \supseteq) \circ \alpha^- \circ \mathcal{T}_{\text{HL}}(\mathbb{W}) && \text{\{def. } \mathcal{T}_{\overline{\text{HL}}}\} \\ &= \text{post}((\sqsubseteq, \supseteq)(\neg\{\langle P, Q \rangle \mid \text{post}[\![\mathbb{W}]\!]P \subseteq Q\})) && \text{\{Lem. 1.4 and def. (30) of } \alpha^-\}} \\ &= \text{post}(\sqsubseteq, \supseteq)(\{\langle P, Q \rangle \mid \neg(\text{post}[\![\mathbb{W}]\!]P \subseteq Q)\}) && \text{\{def. } \neg\}} \\ &= \text{post}(\sqsubseteq, \supseteq)(\{\langle P, Q \rangle \mid \text{post}[\![\mathbb{W}]\!]P \cap \neg Q \neq \emptyset\}) && \text{\{def. } \sqsubseteq \text{ and } \neg\}} \\ &= \{\langle P', Q' \rangle \mid \exists \langle P, Q \rangle \in \{\langle P, Q \rangle \mid \text{post}[\![\mathbb{W}]\!]P \cap \neg Q \neq \emptyset\} . \langle P, Q \rangle \sqsubseteq, \supseteq \langle P', Q' \rangle\} && \text{\{def. post}\}} \\ &= \{\langle P', Q' \rangle \mid \exists \langle P, Q \rangle . \text{post}[\![\mathbb{W}]\!]P \cap \neg Q \neq \emptyset \wedge \langle P, Q \rangle \sqsubseteq, \supseteq \langle P', Q' \rangle\} && \text{\{def. } \in\}} \\ &= \{\langle P', Q' \rangle \mid \exists \langle P, Q \rangle . \text{post}[\![\mathbb{W}]\!]P \cap \neg Q \neq \emptyset \wedge P \subseteq P' \wedge Q \supseteq Q'\} && \text{\{component wise def. of } \sqsubseteq, \supseteq\}} \\ &= \{\langle P', Q' \rangle \mid \exists Q . \text{post}[\![\mathbb{W}]\!]P' \cap \neg Q \neq \emptyset \wedge Q \supseteq Q'\} \\ &\quad \text{\{(\sqsubseteq) if } P \subseteq P' \text{ then } \text{post}[\![\mathbb{W}]\!]P \subseteq \text{post}[\![\mathbb{W}]\!]P' \text{ by (12) so that } \text{post}[\![\mathbb{W}]\!]P \cap \neg Q \neq \emptyset \text{ implies } \\ &\quad \text{post}[\![\mathbb{W}]\!]P' \cap \neg Q \neq \emptyset; \\ &\quad \text{(\supseteq) conversely, if } \exists Q . \text{post}[\![\mathbb{W}]\!]P', \text{ then } \exists P . \text{post}[\![\mathbb{W}]\!]P \cap \neg Q \neq \emptyset \wedge P \subseteq P' \text{ by choosing } \\ &\quad P = P'. \}} \\ &= \{\langle P', Q' \rangle \mid \text{post}[\![\mathbb{W}]\!]P' \cap \neg Q' \neq \emptyset\} \\ &\quad \text{\{(\sqsubseteq) if } Q \supseteq Q' \text{ then } \neg Q' \supseteq \neg Q \text{ so } \text{post}[\![\mathbb{W}]\!]P' \cap \neg Q \neq \emptyset \text{ implies } \text{post}[\![\mathbb{W}]\!]P' \cap \neg Q' \neq \emptyset; \\ &\quad \text{(\supseteq) conversely } \text{post}[\![\mathbb{W}]\!]P' \cap \neg Q' \neq \emptyset \text{ implies } \exists Q . \text{post}[\![\mathbb{W}]\!]P' \cap \neg Q \neq \emptyset \wedge Q \supseteq Q' \text{ by choosing } \\ &\quad Q = Q'. \}} \\ &= \{\langle P, Q \rangle \mid \neg(\text{post}[\![\mathbb{W}]\!]P \subseteq Q)\} && \text{\{def. } \sqsubseteq \text{ and } \neg\}} \\ &= \alpha^- \circ \mathcal{T}_{\text{HL}}(\mathbb{W}) && \text{\{def. } \alpha^- \text{ and } \mathcal{T}_{\text{HL}} \text{ for Hoare logic}\}} \quad \square \end{aligned}$$

THEOREM 4.2 (THEORY OF $\overline{\text{HL}}$).

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

PROOF OF TH. 4.2.

$$\begin{aligned} \mathcal{T}_{\overline{\text{HL}}}(\mathbb{W}) &= \{\langle P, Q \rangle \mid \text{post}[\![\neg \mathbb{B}]\!](\text{lfp}^\sqsubseteq \bar{F}_P^e) \cap \neg Q \neq \emptyset\} && \text{\{Lem. 1.3, where } \bar{F}_P^e(X) \triangleq P \cup \text{post}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^e)X\}} \\ &= \{\langle P, Q \rangle \mid \text{lfp}^\sqsubseteq \bar{F}_P^e \cap \text{pre}[\![\neg \mathbb{B}]\!](\neg Q) \neq \emptyset\} && \text{\{ (39.d) \}} \\ &= \{\langle P, Q \rangle \mid \exists I \in \wp(\Sigma) . \bar{F}_P^e(I) \subseteq I \wedge \exists \langle W, \leq \rangle \in \mathfrak{W}\mathfrak{f} . \exists \nu \in I \rightarrow W . \exists \langle \sigma_i \in I, i \in [1, \infty] \rangle . \sigma_1 \in \bar{F}_P^e(\emptyset) \wedge \forall i \in [1, \infty] . \sigma_{i+1} \in \bar{F}_P^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\prec \nu(\sigma_{i+1}) \Rightarrow \{\sigma_i\} \cap \text{pre}[\![\neg \mathbb{B}]\!](\neg Q) \neq \emptyset)\} && \text{\{induction principle Th. H.3\}} \\ &= \{\langle P, Q \rangle \mid \exists I \in \wp(\Sigma) . P \subseteq I \wedge \text{post}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^e)I \subseteq I \wedge \exists \langle W, \leq \rangle \in \mathfrak{W}\mathfrak{f} . \exists \nu \in I \rightarrow 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}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^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\prec \nu(\sigma_{i+1}) \Rightarrow \sigma_i \in \text{pre}[\![\neg \mathbb{B}]\!](\neg Q))\} \end{aligned}$$

$$\begin{aligned} &\text{\{def. } \bar{F}_P^e(X) \triangleq P \cup \text{post}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^e)X, \sqsubseteq, \text{ and post, which is } \emptyset\text{-strict}\}} \\ &= \{\langle P, Q \rangle \mid \exists I \in \wp(\Sigma) . P \subseteq I \wedge \text{post}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^e)I \subseteq I \wedge \exists \langle W, \leq \rangle \in \mathfrak{W}\mathfrak{f} . \exists \nu \in I \rightarrow 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}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^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\prec \nu(\sigma_{i+1}) \Rightarrow \sigma_i \in \text{pre}[\![\neg \mathbb{B}]\!](\neg Q))\} && \text{\{since if } \sigma_{i+1} \in P, \text{ 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}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^e)I \subseteq I \wedge \exists n \geq 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}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^e)\{\sigma_i\} \wedge \sigma_n \in \text{pre}[\![\neg \mathbb{B}]\!](\neg Q)\} && \text{\{(\sqsubseteq) By } \langle W, \leq \rangle \in \mathfrak{W}\mathfrak{f}, \nu \in I \rightarrow W, \forall i \in [1, \infty] . (\sigma_i \neq \sigma_{i+1}) \Rightarrow (\nu(\sigma_i) > \nu(\sigma_{i+1})), \text{ the sequence is ultimately stationary at some rank } n. \text{ For then on, } \sigma_{i+1} = \sigma_i, i \geq n \text{ and so } \nu(\sigma_i) = \nu(\sigma_{i+1}). \text{ Therefore } \forall i \in [1, \infty] . (\nu(\sigma_i) \not\prec \nu(\sigma_{i+1}) \Rightarrow \sigma_i \notin Q \text{ implies that } \sigma_n \in \text{pre}[\![\neg \mathbb{B}]\!](\neg Q); \\ &\quad \text{(\supseteq) Conversely, from } \langle \sigma_i \in I, i \in [1, n] \rangle \text{ we can define } W = \{\sigma_i \mid i \in [1, n]\} \cup \{-\infty\} \text{ with } -\infty < \sigma_i < \sigma_{i+1} \text{ and } \nu(x) = (\!| x \in \{\sigma_i \mid i \in [1, n]\} \text{ ? } x \text{ : } -\infty \!) \text{ and the sequence } \langle \sigma_j \in I, j \in [1, \infty] \rangle \text{ repeats } \sigma_n \text{ ad infimum for } j \geq n.\}} \\ &= \{\langle P, Q \rangle \mid \exists I \in \wp(\Sigma) . P \subseteq I \wedge \text{post}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^e)I \subseteq I \wedge \exists n \geq 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}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^e)\{\sigma_i\} \wedge \sigma_n \notin \mathcal{B}[\![\mathbb{B}]\!] \wedge \sigma_n \notin Q\} && \text{\{def. pre}\}} \\ &= \{\langle P, Q \rangle \mid \exists n \geq 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}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^e)\{\sigma_i\} \wedge \sigma_n \notin \mathcal{B}[\![\mathbb{B}]\!] \wedge \sigma_n \notin Q\} && \text{\{I is not used and can always be chosen to be } \Sigma\}} \\ &= \{\langle P, Q \rangle \mid \exists n \geq 1 . \exists \langle \sigma_i \in I, i \in [1, n] \rangle . \sigma_1 \in P \wedge \forall i \in [1, n[. \text{post}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^e)\{\sigma_i\} \cap \{\sigma_{i+1}\} \neq \emptyset \wedge \sigma_n \notin \mathcal{B}[\![\mathbb{B}]\!] \wedge \sigma_n \notin Q\} && \text{\{since } x \in X \Leftrightarrow X \cap \{x\} \neq \emptyset\}} \\ &= \{\langle P, Q \rangle \mid \exists n \geq 1 . \exists \langle \sigma_i \in I, i \in [1, n] \rangle . \sigma_1 \in P \wedge \forall i \in [1, n[. \text{post}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^e)\{\sigma_i\} \cap \neg(\neg\{\sigma_{i+1}\}) \neq \emptyset \wedge \sigma_n \notin \mathcal{B}[\![\mathbb{B}]\!] \wedge \sigma_n \notin Q\} && \text{\{def. } \neg X = \Sigma \setminus X\}} \\ &= \{\langle P, Q \rangle \mid \exists n \geq 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}(\llbracket \mathbb{B} \rrbracket \circ \llbracket \text{S} \rrbracket^e)\{\sigma_i\} \subseteq (\neg\{\sigma_{i+1}\})) \wedge \sigma_n \notin \mathcal{B}[\![\mathbb{B}]\!] \wedge \sigma_n \notin Q\} && \text{\{ } \neg(X \subseteq Y) \Leftrightarrow (X \cap \neg Y \neq \emptyset)\}} \\ &= \{\langle P, Q \rangle \mid \exists n \geq 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}(\llbracket \text{S} \rrbracket^e)(\mathcal{B}[\![\mathbb{B}]\!] \cap \{\sigma_i\})) \subseteq (\neg\{\sigma_{i+1}\})) \wedge \sigma_n \notin \mathcal{B}[\![\mathbb{B}]\!] \wedge \sigma_n \notin Q\} && \text{\{def. post, } \llbracket \mathbb{B} \rrbracket, \text{ and } \circ\}} \\ &= \{\langle P, Q \rangle \mid \exists n \geq 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}[\![\mathbb{B}]\!] \cap \{\sigma_i\}, \neg\{\sigma_{i+1}\} \rangle \in \{\langle P, Q \rangle \mid \neg(\text{post}(\llbracket \text{S} \rrbracket^e)P \subseteq Q)\} \wedge \sigma_n \notin \mathcal{B}[\![\mathbb{B}]\!] \wedge \sigma_n \notin Q\} && \text{\{def. } \in\}} \\ &= \{\langle P, Q \rangle \mid \exists n \geq 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}[\![\mathbb{B}]\!] \cap \{\sigma_i\}, \neg\{\sigma_{i+1}\} \rangle \in \mathcal{T}_{\overline{\text{HL}}}(\text{S}) \wedge \sigma_n \notin \mathcal{B}[\![\mathbb{B}]\!] \wedge \sigma_n \notin Q\} && \text{\{def. } \mathcal{T}_{\overline{\text{HL}}}(\text{S})\}} \quad \square \end{aligned}$$

4.2 Calculational Design of $\overline{\text{HL}}$ Proof Rules

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

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

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

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

By [Aczel method](#), the (constant) fixpoint $\text{lfp}^\sqsubseteq \lambda X . \text{S}$ is defined by $\{\frac{\emptyset}{c} \mid c \in \text{S}\}$;

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

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

Conclusion of part I

A transformational logic is
an abstract interpretation of
a natural relational semantics

Part II:

Calculational Design of Hyperlogics by Abstract Interpretation

Patrick Cousot, Jeffery Wang:
Calculational Design of Hyperlogics by Abstract Interpretation. Proc. ACM Program. Lang. 9(POPL):
446-478 (2025)

Objective

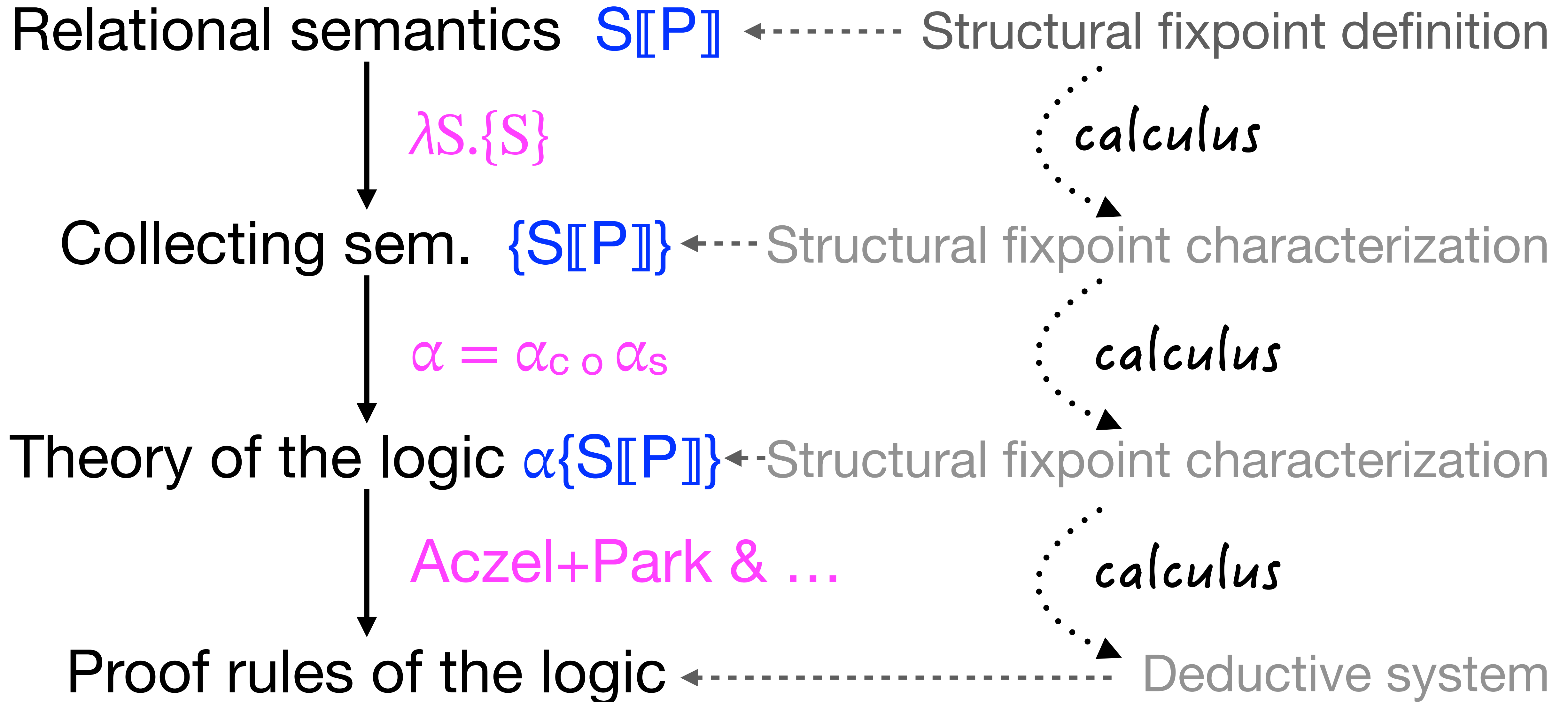
Conceive a method to design program
transformational hyperlogics

Transformational logic = Hoare style logics $\{P\} S \{Q\}$

Understanding a program logic in Part I

- What is the program semantics? $S[P]$
- What is the strongest program semantic property (collecting semantics)? $\{S[P]\}$
- What is the strongest program property of interest? $\alpha_s\{S[P]\}$
- The properties of interest derive by implication (consequence rule) $\alpha_c \circ \alpha_s\{S[P]\}$ (theory of the logic)
- What are the proof rules?

Reminder (of Part I, POPL 2024)



Methodology

Can we calculate hyperlogics proof systems by structural abstractions of the program semantics?

We will conclude that ``Yes'', but

- For hyperlogics, the strongest program property of interest is the collecting semantics itself $\{S[P]\}$
- There is no abstraction α_s (in general)
- Any proof of a *general* hyperproperty must characterize the program semantics exactly!
- Unmanageable in practice!
- The only workaround is to consider only *abstract* hyperproperties!

Which semantics?

Which semantics?

- Hoare logic soundness/completeness for **invariants** is with respect to a **relational semantics**
- The logic would be essentially the same with **execution traces** (but for primitives)
- Is there **a semantics covering both cases** (and even many others)?

Algebraic semantics: a structural fixpoint definition

Algebraic semantics

- Parameterized by an **abstract semantic domain** providing the model of executions and effect of primitives

$$\mathbb{D}_+^\# \triangleq \langle \sqsubseteq_+^\#, \sqsupseteq_+^\#, \perp_+^\#, \sqcup_+^\#, \text{init}^\#, \text{assign}^\# \llbracket x, A \rrbracket, \\ \text{rassign}^\# \llbracket x, a, b \rrbracket, \text{test}^\# \llbracket B \rrbracket, \text{break}^\#, \text{skip}^\#, \circ^\# \rangle$$

$$\mathbb{D}_\infty^\# \triangleq \langle \sqsubseteq_\infty^\#, \sqsupseteq_\infty^\#, \top_\infty^\#, \sqcap_\infty^\#, \circ^\# \rangle$$

Algebraic semantics (cont'd)

- Structural fixpoint definition of the effect of commands
- E.g. assignment
- E.g. break

$$\llbracket x = A \rrbracket_e^\# \triangleq \text{assign}^\# \llbracket x, A \rrbracket$$

$$\llbracket x = A \rrbracket_b^\# \triangleq \perp_+^\#$$

$$\llbracket x = A \rrbracket_\perp^\# \triangleq \perp_\infty^\#$$

$$\llbracket \text{break} \rrbracket_e^\# \triangleq \perp_+^\#$$

$$\llbracket \text{break} \rrbracket_b^\# \triangleq \text{break}^\#$$

$$\llbracket \text{break} \rrbracket_\perp^\# \triangleq \perp_\infty^\#$$

Algebraic semantics (cont'd)

- E.g. iteration `while (B) S`

$$\tilde{F}_e^\# \triangleq \lambda X \in \mathbb{L}_+^\# \cdot \text{init}^\# \sqcup_+^\# (\llbracket B; S \rrbracket_e^\# \circ^\# X)$$

$$F_\perp^\# \triangleq \lambda X \in \mathbb{L}_\infty^\# \cdot \llbracket B; S \rrbracket_e^\# \circ^\# X$$

$$\llbracket \text{while } (B) \text{ } S \rrbracket_e^\# \triangleq (\text{lfp}^{\Xi_+^\#} \tilde{F}_e^\#) \circ^\# (\llbracket \neg B \rrbracket_e^\# \sqcup_e^\# \llbracket B; S \rrbracket_b^\#)$$

$$\llbracket \text{while } (B) \text{ } S \rrbracket_b^\# \triangleq \perp_+^\#$$

$$\llbracket \text{while } (B) \text{ } S \rrbracket_{bi}^\# \triangleq (\text{lfp}^{\Xi_+^\#} \tilde{F}_e^\#) \circ^\# \llbracket B; S \rrbracket_\perp^\#$$

$$\llbracket \text{while } (B) \text{ } S \rrbracket_{li}^\# \triangleq \text{gfp}^{\Xi_\infty^\#} F_\perp^\#$$

$$\llbracket \text{while } (B) \text{ } S \rrbracket_\perp^\# \triangleq \llbracket \text{while } (B) \text{ } S \rrbracket_{bi}^\# \sqcup_\infty^\# \llbracket \text{while } (B) \text{ } S \rrbracket_{li}^\#$$

Algebraic semantics (cont'd)

- The classic postulated presentation by equational axioms ^(*) can be calculated by
 - structural induction
 - Aczel correspondence between fixpoints and deductive systems (see Part I on POPL 2024)

(*) C. A. R. Hoare, Ian J. Hayes, Jifeng He, Carroll Morgan, A. W. Roscoe, Jeff W. Sanders, Ib Holm Sørensen, J. Michael Spivey, and Bernard Sufrin. 1987. Laws of Programming. *Commun. ACM* 30, 8 (1987), 672–686. <https://doi.org/10.1145/27651.27653>

How to express
program properties?

“Programs are predicates” (*)

- We are only interested in properties of programs (not in arbitrary properties)
- A program encodes a program execution property defined by its semantics
- So defining properties as programs, we don't need a language for programs + another language for predicates!
- Other encodings of properties are mere abstractions.

(*) Eric C. R. Hehner. 1990. A Practical Theory of Programming. *Sci. Comput. Program.* 14, 2-3 (1990), 133–158. [https://doi.org/10.1016/0167-6423\(90\)90018-9](https://doi.org/10.1016/0167-6423(90)90018-9)

Property transformer

Algebraic property transformer

- Forward property transformer:

$$\text{post}^\# \in \mathbb{L}^\# \xrightarrow{\nearrow} \mathbb{L}^\# \xrightarrow{\nearrow} \mathbb{L}^\#$$

$$\text{post}^\#(S)P \triangleq P \circ^\# S$$

A structural fixpoint characterization of the property transformer

A calculus of algebraic execution properties

- Galois connection

$$\forall S \in \mathbb{L} . \langle \mathbb{L}, \sqsubseteq \rangle \xrightleftharpoons[\text{post}(S)]{\widetilde{\text{pre}}(S)} \langle \mathbb{L}, \sqsubseteq \rangle \quad (\langle \mathbb{L}, \sqsubseteq, \sqcup \rangle \text{ is a poset})$$

- Using the abstraction methodology of POPL 2024, we generalize POPL 2024 to
 - a structural fixpoint algebraic calculus of execution properties
 - (and the lattice of algebraic transformational logics)

Hyperproperties

Algebraic hyperproperties

- \mathbb{L} is the semantic domain (e.g. set of finite and infinite traces, input-output relation)
- $\wp(\mathbb{L})$ is the set of hyperproperties (defined in extension)
- \subseteq is logical implication

Hyperproperty transformer

Algebraic hyperproperty transformer

- Transformer

$$\begin{aligned}\text{Post}^\# &\in \mathbb{L}^\# \rightarrow \wp(\mathbb{L}^\#) \xrightarrow{\quad} \wp(\mathbb{L}^\#) \\ \text{Post}^\#(S)\mathcal{P} &\triangleq \{\text{post}^\#(S)P \mid P \in \mathcal{P}\}\end{aligned}$$

- Galois connection

$$\langle \wp(\mathbb{L}^\#), \subseteq \rangle \xrightleftharpoons[\text{Post}^\#(S)]{\text{Pre}(S)} \langle \wp(\mathbb{L}^\#), \subseteq \rangle$$

Structural fixpoint characterization of the hyperproperty transformer

Incomplete structural characterization of $\text{Post}^\#(S)$

- Counter-example

$$\begin{aligned} & \text{Post}^\# [\text{if } (B) \ S_1 \ \text{else } S_2]^\# \mathcal{P} \\ &= \{ \text{post}^\# [B; S_1]^\# P \sqcup^\# \text{post}^\# [\neg B; S_2]^\# P \mid P \in \mathcal{P} \} \\ & \subseteq \{ \text{post}^\# [B; S_1]^\# P_1 \sqcup^\# \text{post}^\# [\neg B; S_2]^\# P_2 \mid P_1 \in \mathcal{P} \wedge P_2 \in \mathcal{P} \} \\ &= \{ Q_1 \sqcup^\# Q_2 \mid Q_1 \in \text{Post}^\# [B; S_1]^\# \mathcal{P} \wedge Q_2 \in \text{Post}^\# [\neg B; S_2]^\# \mathcal{P} \} \end{aligned}$$

- This structural collecting semantics (*) is incomplete

(*) Thibault Dardinier and Peter Müller. 2024. Hyper Hoare Logic: (Dis-)Proving Program Hyperproperties. *Proceedings of the ACM on Programming Languages (PACMPL)* 8, Issue PLDI, Article No.: 207 (June 2024), 1485–1509. <https://doi.org/10.1145/3656437>

Complete structural characterization of $\text{Post}^\#(S)$

$$\{\text{post}^\#(S)P\} = \text{Post}^\#(S)\{P\}$$

- Example:

$$\text{Post}^\#[\text{if } (B) \ S_1 \ \text{else } S_2]^\# \mathcal{P}$$

$$= \{\text{post}^\# [B; S_1]^\# P \sqcup^\# \text{post}^\# [\neg B; S_2]^\# P \mid P \in \mathcal{P}\}$$

$$= \{Q_1 \sqcup^\# Q_2 \mid Q_1 \in \{\text{post}^\# [B; S_1]^\# P\} \wedge Q_2 \in \{\text{post}^\# [\neg B; S_2]^\# P\} \wedge P \in \mathcal{P}\}$$

$$= \{Q_1 \sqcup^\# Q_2 \mid Q_1 \in \text{Post}^\# [B; S_1]^\# \{P\} \wedge Q_2 \in \text{Post}^\# [\neg B; S_2]^\# \{P\} \wedge P \in \mathcal{P}\}$$

- We get a complete **elementwise** characterization of $\text{Post}^\#(S)$

Computational design of the algebraic hyperlogic rules

Upper and lower algebraic hyperlogics

- Definition

$$\begin{aligned}\overline{\{\mathcal{P}\}} s \overline{\{\mathcal{Q}\}} &= \text{Post}^\# \llbracket S \rrbracket^\# \mathcal{P} \subseteq \mathcal{Q} \\ \underline{\{\mathcal{P}\}} s \underline{\{\mathcal{Q}\}} &= \mathcal{Q} \subseteq \text{Post}^\# \llbracket S \rrbracket^\# \mathcal{P}\end{aligned}$$

- The proof system is derived by calculational design (as in POPL 2024)

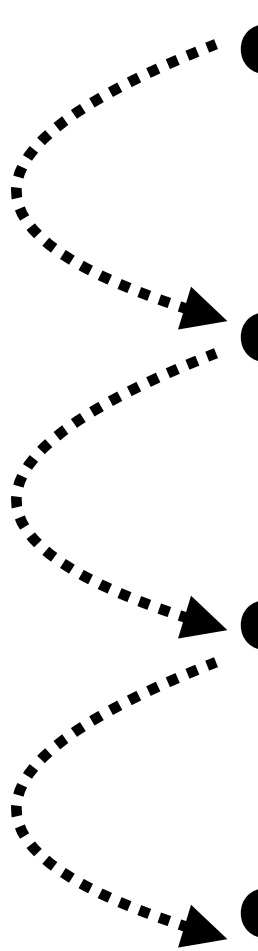
Upper algebraic hyperlogic for iteration

$$\begin{aligned}
 & \left(P_e = \text{lfp}^{\Xi^{\#}_{+}} \vec{F}_{pe}^{\#}(P') \wedge \overline{\{\{P_e\}\}} \neg B \overline{\{\{Q_e\}\}} \wedge \overline{\{\{P_e\}\}} B; S \overline{\{\{Q_b\}\}} \wedge \right. \\
 & \quad \left. \overline{\{\{P_e\}\}} B; S \overline{\{\{Q_{\perp\ell}\}\}} \wedge Q_{\perp b} = \text{gfp}^{\Xi^{\#}_{\infty}} F_{p\perp}^{\#} \wedge P' \in \mathcal{P} \right) \Rightarrow \\
 & \quad \left(\langle e : Q_e \sqcup_e^{\#} Q_b, \perp : Q_{\perp\ell} \sqcup_{\infty}^{\#} Q_{\perp b}, br : P_{br} \rangle \in \mathcal{Q} \right) \\
 \hline
 & \overline{\{\mathcal{I}\}} \text{ while } (B) \text{ } S \overline{\{\mathcal{Q}\}}
 \end{aligned}$$

- Requires an *EXACT* characterization of the program semantics
- *Unmanageable* in practice

Abstractions

Abstractions

- Since proofs of general hyperproperties are unmanageable, we consider abstractions of
 - the algebraic semantics
 - program properties
 - program hyperproperties
 - program logics
- 
- A diagram consisting of four dotted arrows. The first arrow starts at the bullet point for 'the algebraic semantics' and points to the bullet point for 'program properties'. The second arrow starts at the bullet point for 'program properties' and points to the bullet point for 'program hyperproperties'. The third arrow starts at the bullet point for 'program hyperproperties' and points to the bullet point for 'program logics'. The fourth arrow starts at the bullet point for 'program logics' and points to the bullet point for 'program hyperproperties'.

Algebraic semantics abstraction

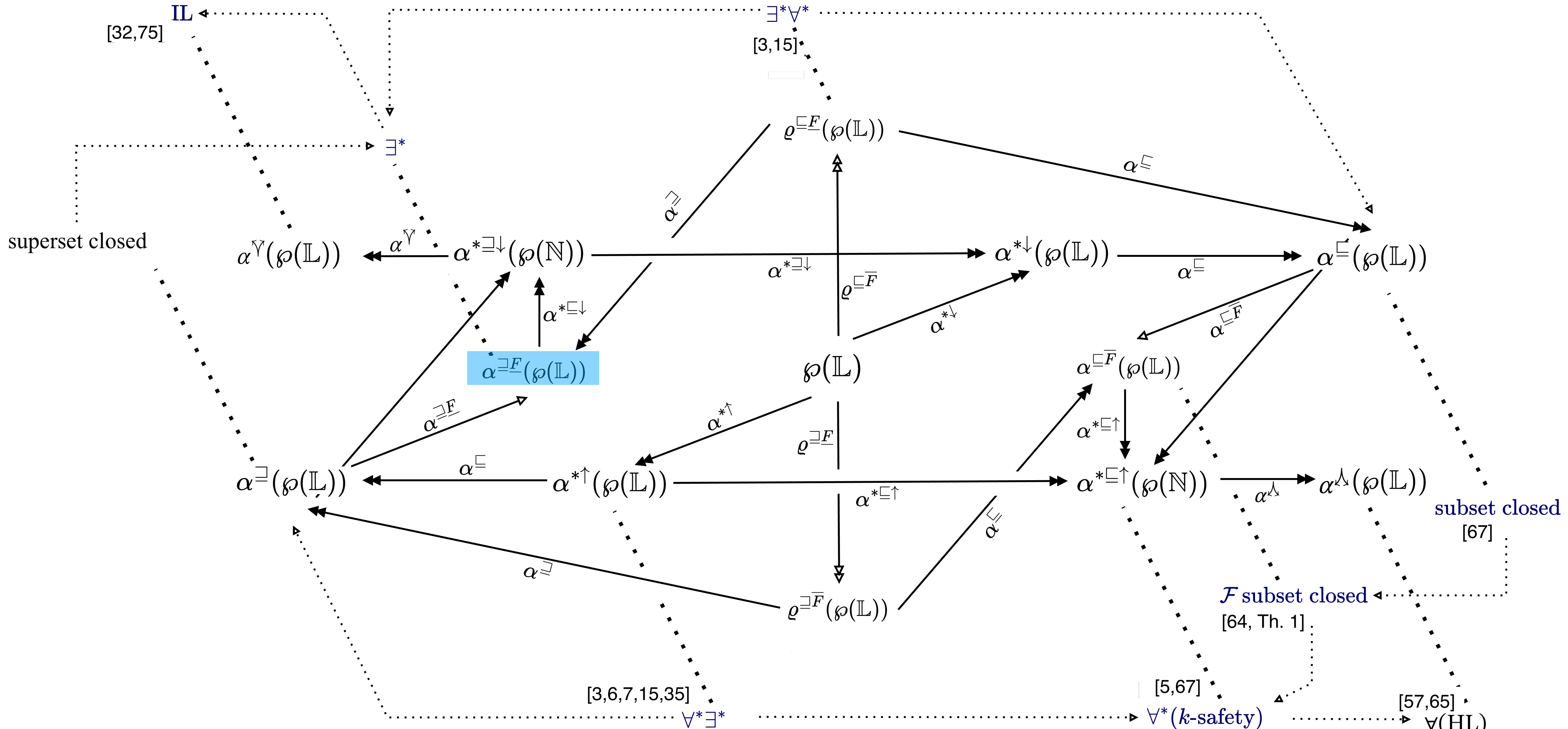
- An abstraction of the algebraic semantics is another instance of the algebraic semantics
 - e.g. trace semantics \rightarrow relational semantics
- This extends to logics and hyperlogics
- But still proofs require exact characterizations of the (abstract) semantics

Hyperproperty abstraction

Hyperproperty abstraction

- A dozen abstractions are considered in the paper
- This leads to a lattice of hyperlogics

Hierarchy of hyperlogics



Chain limit order ideal abstraction

Chain limit order ideal abstraction (cont'd)

- The **chain limit order ideal abstraction** of algebraic hyperproperties is an algebraic generalization of the abstraction to $\forall^*\exists^*$ hyperproperties

- $\forall^*\exists^*$ hyperproperties (for traces in Π) $\mathcal{AEH} \triangleq$

$$\{ \{ P \in \wp(\Pi) \mid \forall \pi_1 \in P . \exists \pi_2 \in P . \langle \pi_1, \pi_2 \rangle \in A \} \mid A \in \wp(\Pi \times \Pi) \}$$

Chain limit order ideal abstraction

$$\alpha^{\uparrow}(\mathcal{P}) \triangleq \left\{ \bigsqcup_{i \in \mathbb{N}} P_i \mid \langle P_i, i \in \mathbb{N} \rangle \in \mathcal{P} \text{ is an increasing chain with existing lub} \right\}$$

$$\alpha^{\sqsubseteq}(\mathcal{P}) \triangleq \{ P' \in \mathbb{L} \mid \exists P \in \mathcal{P} . P' \sqsubseteq P \}$$

$$\alpha^{\sqsubseteq\uparrow} \triangleq \alpha^{\sqsubseteq} \circ \alpha^{\uparrow} \quad (\text{extensive, increasing, not idempotent})$$

$$\check{\alpha}^{\sqsubseteq\uparrow}(\mathcal{P}) \triangleq \text{Ifp}^{\sqsubseteq} \lambda X . \mathcal{P} \cup \alpha^{\sqsubseteq\uparrow}(X) \quad (\text{upper closure operator hence G.C.})$$

- in particular for traces:

$$\mathcal{AEH} \subseteq \check{\alpha}^{\uparrow}(\wp(\wp(\Pi)))$$

Conclusion of Part II

Conclusion of Part II

- We have introduced a **new algebraic semantics** (instantiable to any classic semantics)
- We have considered **programs** (i.e. their semantics) as **properties**
- We have designed by calculus a **general algebraic logic** (sound & complete and generalizing POPL 2024)
- We have designed by calculus a **general algebraic hyperlogic** (sound & complete but **unmanageable** in practice)
- All this for terminating and nonterminating executions

Conclusion of Part II (cont'd)

- We have considered **abstractions of algebraic hyperproperties** :
 - less expressive than general hyperproperties
 - but with sound and complete hyperlogics using only approximations of the program semantics
- This was illustrated by an **algebraic generalization of $\forall^*\exists^*$ hyperproperties**

More in the POPL25 paper

- Various **instanciations** of the algebraic semantics
- **Abstractions** of the algebraic semantics leading to complete hyperlogics
- A dozen of other **abstractions** of hyperproperties
- Including algebraic **generalizations** of $\exists^* \forall^*$ as well as $\forall^* \forall^*$ **hyperproperties**
- **Correction** of errors and **generalizations** of results in the literature
- etc

Conclusion of the conclusion

A transformational [hyper]logic
is
an abstract interpretation
of
an [hyper]transformer
of
an instantiation
of
an algebraic semantics.

(Conclusion of the conclusion)-I

A [hyper]logic is
another (complicated) way
of defining
an abstract interpretation
of
an instantiation
of
an algebraic semantics.

The End, Thank You