Patrick Cousot & Jeffery Wang Courant Institute, New York University

Calculational Design of Hyperlogics by Abstract Interpretation

POPL 2024, London © P. Cousot 1

POPL 2024, London © P. Cousot 2

Conceive a method to design program transformational hyperlogics

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

- What is the program semantics? SIPI
- What is the strongest program semantic property (collecting semantics)? {S[[P]]}
- What is the strongest program property of interest? α_s S [P] R
- The properties of interest derive by implication (consequence rule) αc o αs{S⟦P⟧} (theory of the logic)
- What are the proof rules?

Understanding a program logic

Proof rules of the logic + $\alpha = \alpha_c$ o α_s S.{S} Aczel+Park & …

POPL 2024, London © P. Cousot 5

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 αs *(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 Denotational semantics, Hoare logic, predicate transformers, and the abstract semantics of sect. **3.4 Algebraic semantics** (called "compositions" in definitions (called "composition" in density of \mathbb{R} " in denoted "contract "composition" in denoted "contract "composition" in denoted "contract "contract "contract ")⟩ ∣ ^S′ [⊲] ^S}) where ^𝐹^S [∈] {⟨S′ *,* ′ ⟩ ∣ ^S′ [⊲] ^S [∧] ′ [∈] V} [→] ^V is a total function. **Denotational semantics**, Hospital semantics semantics semantics semantics semantics semantics semantics semantics semantics of semantics semantics semantics semantics semantics semantics semantics semantics semantics sema Δ for basic commanding Δ Algebraic semantics Denotational semantics, Hoare logic, predicate transformers, and the abstract semantics of sect.

• Parameterized by an abstract semantic domain providing the model of executions and effect of primitives Parameterized by an abstract semantic domain the forimitives and infinitive computations and partial lines. tational orderings [⊑][♯] providing the model of executions and
Providently the We consider computational domains **D**[♯]

$$
\mathbb{D}_{+}^{\sharp} \triangleq \langle \mathbb{L}_{+}^{\sharp}, \Xi_{+}^{\sharp}, \bot_{+}^{\sharp}, \sqcup_{+}^{\sharp}, \text{init}^{\sharp}, \text{assign}^{\sharp}[\mathsf{x}, \mathsf{A}],
$$

assign ^{\sharp} [x, a, b], test ^{\sharp} [B], break ^{\sharp} , skip ^{\sharp} , § ^{\sharp}

$$
\mathbb{D}_{\infty}^{\sharp} \triangleq \langle \mathbb{L}_{\infty}^{\sharp}, \Xi_{\infty}^{\sharp}, \top_{\infty}^{\sharp}, \top_{\infty}^{\sharp}, \top_{\infty}^{\sharp}, \S^{\sharp} \rangle
$$

fect of

^𝑏 ≜ #[♯]

Test as 2.2.D.in as 2.2.D.in as 3.2.D.in and 3.2.D.in and 3.2.D.in the 3.2.D.in the 3.2.D.in the 3.2.D.in the RemaRK 3.4. Hypotheses 3.2.B, 3.2.D.d.i and 3.2.D.d.ii determine the precision of the semantic of basic composition, composition, composition, and iteration, and it 3.4 Definition of the Algebraic Semantics

- · Structural fixpoint definition of the effect of commands commands assignment, and the immediately enclosing loop, and the following loop, and the following α Te algebraic semantics of statements S is an abstract property of executions. Te basic communities of executions. Te basic communities of executions. The basic communities of executions. Te basic contract of executions. T T_{S} S S S S S S is an absorption of S S ructural fixpoint definition of the effect of • Structural fixpoint definition of the effect of commands states are assigned, random assignment, r_{max}
- E.g. assignment E.g. break $s = 100$ **E.g. assignment** $m₁$ are assignment, random assignment, break out of the immediately encoded in $m₂$ -49 • E.g. assignment mands S are assignment, random assignment, break out of the immediately enclosing loop, and infnite/nonterminating "S#[♯] 3. assignment \bullet S# \bullet E.a. assignr ^𝑏 fnite/ending/terminating semantics in **L**[♯] i nfite $\frac{1}{n}$ inf n inating "S# $\frac{1}{n}$ inating" $\frac{1}{n}$ L_{\cdot} 9. absignmond

 \sim ⁺ "break#[♯] For the assignment x=A, the abstract semantics assign♯"x*,* ^A# is specifed by the abstract domain, ⁺ "x=A#[♯] ⁺ "x=[,]#[♯]

$$
\begin{aligned}\n\begin{bmatrix}\n x &= A\n \end{bmatrix}\n \begin{bmatrix}\n x &= A\n \end{bmatrix
$$

$i \in \mathbb{Z}$ ing "S# \mathbb{Z} " S# \mathbb{Z}

• E.g. break as ignition in endingle-minimation in F.g. break $E.g.$ bi \mathbf{I} x^*
 x^* x^* y^* y^*

 $\llbracket x, A \rrbracket$ $\llbracket \text{break} \rrbracket^{\sharp}$ \triangleq \triangleq \perp $\#$ $\llbracket \mathsf{break} \rrbracket^{\sharp}_b$ ≜ break[#] \mathbf{k} \mathbb{I}^{\sharp} \triangleq \mathbb{I}^{\sharp} ∞ $[\![\text{break}]\!]_b^{\sharp}$ $\overset{''}{e}$ $[{\small \texttt{break}}]$ $\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$ |Jore \cdot \overline{A} T
 \overline{A} T \overline{A} T \overline{A} T \overline{A} T \overline{A} T \overline{A} T \overline{A} ^𝑏 ≜ #[♯] \uparrow *∆,* \uparrow \uparrow $=$ \perp^{π}_{τ} $\frac{4}{b}$ $\stackrel{\triangle}{=}$ break[#] $\leq \frac{1}{4}$ \cong $\perp \frac{4}{\infty}$ $\begin{array}{c} \|\mathsf{UI}\ \mathsf{dK}\|_e^r\\ \mathsf{f}\|\mathsf{v}\|\mathsf{dK}\|_e^r\end{array}$ \$ ≜ #[♯] ∞ $\lfloor \textsf{break} \rfloor \rfloor_{b}^{*}$ $=$ pre ∞ $\frac{4}{1}$ $\frac{4}{1}$ $\frac{4}{1}$

 $\binom{\sharp}{b}$ $\frac{1}{l}$

Algebraic semantics (cont'd) Δ lachraic coronations **Algebraic semantics** Δ loohraic somantics (cont'd) Algebraic semantics (cont d) nates. ∎nates. ∎nates.
∎nates. ∎nates. ∎nates We now show that gfp⊑♯ [∞] ♯ # coinductively characterizes the infnite executions of the iteration Algebraic semantics (contid) $\begin{array}{c}\n\hline\n\text{MSEU (d) C} & \text{SCH (d) UCS}\n\end{array}$ [∞] then [♯] # satisfes the same property and gfp⊑♯ WE NOW SHOW SHIM! $\frac{1}{2}$ coinciductively characterizes the informations of the informations of the informations of the informations of the interacterizes the interacterizes the interacterizes of the interacterizes the interacterizes of

- E.g. iteration while (B) S iteration while (B) S \bullet E.g. iteration while (B) S ration while (B true. \bullet E.g. iteration while (B) S g. Iteration while (B) S
	- \bar{F}_e^{\sharp} \triangleq $\lambda X \in \mathbb{L}_+^{\sharp} \cdot \text{init}^{\sharp} \cup_+^{\sharp} (\llbracket B; S \rrbracket_e^{\sharp} \circ^{\sharp} X)$ \tilde{F}_e^{\sharp} $\overset{''}{e}$ ≜ 𝝀𝑋 ∈ **^L**[♯] + . init[♯] [⊔][♯] $+$ ($[B; S]$ # $\frac{9}{e}$ $^{\sharp}$ X) \mathbf{r} \tilde{F} F^{\sharp}_e ≜ 𝝀𝑋 ∈ **^L**[♯] + . init[♯] [⊔][♯] $(\llbracket B; S \rrbracket^{\sharp}$ of X) \tilde{F}_e^{\sharp} \triangleq $\lambda X \in \mathbb{L}_+^{\sharp} \cdot \text{init}^{\sharp} \cup_+^{\sharp} (\llbracket B ; S \rrbracket_e^{\sharp} \circ^{\sharp} X)$
	- \mathcal{A}^{\sharp} and $\mathcal{A}X \in \mathbb{L}^{\sharp}_{\infty}$. $\left[\mathbb{B}; \mathsf{S}\right]_{e}^{\sharp}$ of X F_{\perp}^{\sharp} \triangleq $\lambda X \in \mathbb{L}_{\infty}^{\sharp}$ \cdot \mathbb{E} ; $S \mathbb{I}_{e}^{\sharp}$ \circ ^{\sharp} forward F_{\perp}^{\sharp} \triangleq $\lambda X \in \mathbb{L}^{\sharp}_{\infty} \cdot [\mathbb{B} ; S]$ F_{\perp}^{\sharp} \triangleq $\lambda X \in \mathbb{L}_{\infty}^{\sharp}$ \cdot $\llbracket B; S \rrbracket_{e}^{\sharp}$ \circ $\sharp X$ F_{\perp}^{\sharp} \triangleq $\lambda X \in \mathbb{L}_{\infty}^{\sharp}$ • $[\mathbb{B}; \mathbb{S}]_{e}^{\sharp}$ $\circ^{\sharp} X$ $\mathcal{A} \subseteq \mathbb{L}^*_{\infty} \cdot \mathbb{B}$; $\mathcal{S} \parallel_e^* \circ^* X$
		- $\begin{bmatrix} \text{while} & \text{(B)} & \text{S} \end{bmatrix}^{\sharp}_{e}$ $\overset{''}{e}$ \triangleq (lfp^{\equiv #} \overline{F} [#]_e \leftarrow $\begin{bmatrix} \sharp \\ e \end{bmatrix}$ $\begin{bmatrix} \circ \\ e \end{bmatrix}$ $\begin{bmatrix} \left[-B \right] \right] \stackrel{\sharp}{e} \cup \stackrel{\sharp}{e} \left[B; S \right] \stackrel{\sharp}{b}$ $\prod_{\mathbf{A}}$ +
while (B) S^{∥#} ≜ (Ifp^{=#} $\frac{1}{2}$ $\llbracket \text{while (B) } S \rrbracket_b^{\sharp} \triangleq \bot^{\sharp}_+$ $\llbracket \text{while (B) } \mathsf{S} \rrbracket_e^{\sharp} \triangleq (\mathsf{lfp}^{\equiv \sharp} \ F_e^{\sharp}) \ ^{\circ \sharp} \ (\llbracket \neg \mathsf{B} \rrbracket_e^{\sharp} \sqcup_e^{\sharp} \llbracket \mathsf{B} ; \mathsf{S} \rrbracket_b^{\sharp})$ while (B) S $\parallel^{\sharp}_{\rho}\ \triangleq$ (Ifp $^{\Xi^{\sharp}_{+}}$ $\tilde{F}^{\sharp}_{\rho}$) $\,^{\sharp}_{\vartheta}\parallel$ (\parallel ¬B $\parallel^{\sharp}_{\rho}\parallel$ B;S $\parallel^{\sharp}_{\hbar}$ \llbracket while (B) $\text{S} \rrbracket_e^{\sharp} \triangleq (\text{Ifp}^{\equiv \frac{\pi}{l}} \tilde{F})$ \llbracket while (B) S $\rrbracket_e^{\sharp} \triangleq (\mathsf{lfp}^{\sqsubseteq \sharp} \tilde{F}_e^{\sharp}) \; \circ^{\sharp} \left(\llbracket \neg \mathsf{B} \rrbracket_e^{\sharp} \sqcup^{\sharp}_e \llbracket \mathsf{B} ; \mathsf{S} \rrbracket_b^{\sharp} \right)$
			- \llbracket while (B) S $\rrbracket_b^{\sharp} \triangleq \bot^{\sharp}_+$ $\frac{H}{\perp}$ $\overline{}$ $\sum_{\mu=1}^{\infty}$ $\frac{4}{1}$ $\begin{array}{ccc} \hbox{``\quad~} & \hbox$ Γ satisfy the same property Γ is least Γ in Γ in Γ in Γ is least Γ in Γ is least Γ in Γ is least Γ is le $[$ while (B) S $]_t^{\sharp}$ \ddot{l} $\stackrel{\triangle}{=} \perp$ $\mathbf{F} = \begin{bmatrix} \mathbf{F} & \mathbf{$ \leftarrow #
⊥
⊥ \mathbf{r} while (B) $S\Vert_{b}^{\sharp} \triangleq \bot_{+}^{\sharp}$ \mathcal{L} \triangleq (lfp^{$=$ †} F_e^{\sharp} ⃗ $\binom{4}{e}$ $\frac{6}{9}$ $\left[\mathsf{B}\,;\mathsf{S}\right] \right\}$ $\frac{1}{1}$ $\overline{}$ ╋
┸
═┉╙╷┈╝ $^{\text{th}}$ \sim $^{\text{th}}$ \sim $^{\text{th}}$
				-
		- $\llbracket \text{while (B) } \mathsf{S} \rrbracket_{bi}^{\sharp} \triangleq (\mathsf{lfp}^{\equiv \sharp} \tilde{F}_e^{\sharp}) \, \S^{\sharp} \llbracket \mathsf{B}; \mathsf{S} \rrbracket^{\sharp}$ $\lfloor \text{while (B) S} \rfloor \rfloor_i^* \triangleq \text{gfp}^{-\circ}$ $\sqrt{\frac{1}{2}}$ Interations. The finite extending of the form of the form of the local $\sin \frac{1}{2}$ or \sin t_{t} is false, or a break is executed in the body which exists the body which exists the loop. By (9) and $\mathcal{L}(\mathcal{U})$ $\llbracket \text{while (B) S} \rrbracket_{li}^{\sharp} \triangleq \text{gfp}^{\equiv \sharp} F_{\perp}^{\sharp}$ $\llbracket \textsf{while (B) S} \rrbracket_\bot^* \triangleq \llbracket \textsf{while (B) S} \rrbracket_\ell^*$ \llbracket while (B) $S \rrbracket_{bi}^* = (\llbracket \text{tp}^{-+} F_e^*)$ of $\llbracket \text{B}; S \rrbracket_{\perp}^*$ \llbracket while (B) S $]\rrbracket^{\sharp} \triangleq \llbracket$ while (B) S $]\rrbracket^{\sharp}_{bi} \sqcup_{\infty}^{\sharp} \llbracket$ while (B) S $]\rrbracket$ entry of the iteration while(B) S afer zero or more terminating body iterations. To see that, we have that, we
To see that, we have the international international international international international international i $\begin{array}{ccccccc}\n\mathbb{I}^{\text{WILLLC}} & \mathbb{I}^{\text{U}} & \$ \mathbb{I}
 \mathbb{I} inductions reaching the set of finite executions reaching the set of \mathbb{I} \llbracket while (B) S $\rrbracket^{\texttt{\#}}_1 \triangleq \llbracket$ while (B) S $\rrbracket^{\texttt{\#}}_{bi} \sqcup^{\texttt{\#}}_{\infty} \llbracket$ while (B) S $\rrbracket^{\texttt{\#}}_{li}$ $\llbracket \text{while (B) } \mathsf{S} \rrbracket_{bi}^{\sharp} \triangleq (\mathsf{lfp}^{\equiv \frac{\pi}{4}} \tilde{F}_{e}^{\sharp})$ $\llbracket \textsf{while (B) S} \rrbracket_{bi}^{\sharp} \triangleq (\textsf{lfp}^{\textsf{H}} + \bar{F}_{e}^{\sharp}) \, \textsf{s}^{\sharp} \, \llbracket \textsf{B} ; \textsf{s} \rrbracket_{\perp}^{\sharp}$ $\llbracket \text{while (B) } \mathsf{S} \rrbracket_{li}^{\sharp} \triangleq \mathsf{gfp}^{\equiv \sharp_{\infty}} F_{\perp}^{\sharp}$ $\texttt{le (B) } \texttt{S} \parallel_{bi}^{\texttt{H}} \sqcup_{\infty}^{\texttt{t}} \parallel$ while (B) $\texttt{S} \parallel_{li}^{\texttt{H}}$ $\llbracket \text{while (B) } \mathsf{S} \rrbracket_{li}^{\sharp} \triangleq \mathsf{gfp}^{\equiv \frac{\mathsf{d}}{\infty}} F_{\perp}^{\sharp}$ \llbracket while (B) S \rrbracket^\sharp \triangleq \llbracket while (B) S \rrbracket^\sharp_b $\lfloor \textsf{while (B) S} \rfloor \rfloor_b^{\mu}$ \parallel while (B) S $\parallel_{li}^{\sharp} \triangleq$ g f $p = \infty$ F_{\perp}^{\sharp} $\llbracket \text{while (B) } \mathsf{S} \rrbracket^{\sharp}_{\perp} \triangleq \llbracket \text{while (B) } \mathsf{S} \rrbracket^{\sharp}_{bi} \sqcup^{\sharp}_{\infty} \llbracket \text{while (B) } \mathsf{S} \rrbracket^{\sharp}_{b}$
- 3) S
"B"S
- $^{\sharp}X$
- The classic postulated presentation by equational axioms $\tilde{ }$ can be calculated by (\star) and (\star) and (\star)
- structural induction \bullet structural induction [51] Reinhold Heckmann. 1993. Power Domains and Second-Order Predicates. Teor. Comput. Sci. 111, 1&2 (1993), 59–88.
	- deductive systems (see POPL 2024)

Algebraic semantics (cont'd) 16:32 P. Cousot and J. Wang

• I he classic postulated presentation by equational and \blacksquare axioms Y can be calculated by 1978, Proceedings, 7th Symposium, Zakopane, Poland, September 4-8, 1978 (Lecture Notes in Computer Science, Vol. 64),

• Aczel correspondence between fixpoints and [52] Eric C. R. Hehner. 1990. A Practical Teory of Programming. Sci. Comput. Program. 14, 2-3 (1990), 133–158. https: [53] Eric C. R. Hehner. 1993. A Practical Teory of Programming. Springer. https://doi.org/10.1007/978-1-4419-8596-5

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! Continuous Algebras. J. ACM 24, 1 (1977), 68–95. https://doi.org/10.1145/321992.321997 [50] Irène Guessarian. 1978. Some Applications of Algebraic Semantics. In Mathematical Foundations of Computer Science 1978, Proceedings, 7th Symposium, Zakopane, Poland, September 4-8, 1978 (Lecture Notes in Computer Science, Vol. 64),
- Other encodings of properties are mere abstractions. [51] Reinhold Heckmann. 1993. Power Domains and Second-Order Predicates. Teor. Comput. Sci. 111, 1&2 (1993), 59–88.
- 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 (*)

Property transformer

Algebraic property transformer \mathbf{A} culus post[♯] on programs. \mathbf{A}

 $\sharp S$

• Forward property transformer: • Forward property transformer:

Let us defne the transformer post[♯] [∈] **^L**[♯] "→↗ **^L**[♯] "→↗ **^L**[♯] such that Let us defne the transformer post[♯] [∈] **^L**[♯] "→↗ **^L**[♯] "→↗ **^L**[♯] such that $\text{post}^{\sharp}(S)P \triangleq P^{\circ}_{\varphi}$

$$
\rightarrow \mathbb{L}^{\sharp} \xrightarrow{\mathcal{A}} \mathbb{L}^{\sharp}
$$

$$
\triangleq P g^{\sharp} S
$$

A structural fixpoint characterization of the property transformer

Te following Galois connection shows the equivalence of forward/deductive and backward/abdigebraic executivir proper thes A calculus of algebraic execution properties \blacksquare

[∀]^𝑆 [∈] **^L** *.* ⟨**L***,* ⊑⟩ −−−−−−−− ←−−−−−−−−→ • (and the lattice of algebraic transformational logics) ⟨**L***,* ⊑⟩ where pre

- generalize POPL 2024 to
- properties • a structural fixpoint algebraic calculus of execution composition on . If r is a properties any of the properties of definition 2.2 or the posterior α
- · (and the lattice

• Galois connection

$\overline{\text{Cyl}}$ (L, ⊑) ((L, ⊑, ⊔) is a poset) posity)
tion methodology of POPL 2024 • Using the abstraction methodology of POPL 2024, we **L,**

• a structural fixpoint algebraic calculus of execution

$$
\forall S \in \mathbb{L}. \langle \mathbb{L}, \Xi \rangle \xrightarrow{\text{pre}(S)}
$$
\nUsing the abstraction met

Hyperproperties

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

Algebraic hyperproperties

Hyperproperty transformer

Algebraic hyperproperty transformer Algebraic hyperproperty transformer

- Transformer **the Transformer**
	- Post[#] ∈
- Galois connection \bullet Galois connection backward reasonings, we defne Pre such that for all [∈] **^L**♯, we have ◯^A
	- $\langle \wp(\mathbb{L}^{\sharp}), \subseteq \rangle \longrightarrow \longrightarrow$

 \mathcal{L} $\text{Pre}(S)$ $Post^{\mathcal{F}}(S)$ Pre(S) ⟨℘(**L**[♯])*,* ⊆⟩ (36)

$$
Post^{\sharp} \in L^{\sharp} \rightarrow \wp(L^{\sharp}) \longrightarrow \wp(L^{\sharp})
$$

$$
Post^{\sharp}(S) \mathcal{P} \triangleq \{post^{\sharp}(S)P | P \in \mathcal{P}\}
$$

Structural fixpoint characterization of the hyperproperty transformer

er Hoare Logic: (Dis-)Proving Program Hyperproperties. *P*
MPL) 8, Issue PLDI, Article No.: 207 (June 2024), 1485–150 roceedings
19. https: Thibault Dardinier and Peter Müller. 2024. Hyper Hoare Logic: (Dis-)Proving Program Hyperproperties. Proceedings conditional, this link is located to the ACMP in the ACMP of the ACM on Program Hyperproperties. Proceedings
of the ACM on Programming Languages (PACMPL) 8, Issue PLDI, Article No.: 207 (June 2024), 1485–1509. https: [30] Tibault Dardinier and Peter Müller. 2024. Hyper Hoare Logic: (Dis-)Proving Program Hyperproperties. Proceedings

Incomplete structural characterization of Post[#](S) t_{current} <u>bilipiele</u> structural characterization of rost"(5). <u>ipiece</u> su uctural crial traditional [⊆]. But the classic structural defnition (see sect. 3.2) of the transformer Post[♯] fails (unless <u>icomplete</u> structural characterization of Post[#](3)

- Counter-example Counter-example. For the considered hyperproperties on the constant of the constant of the conditions of the condi Counter-example • Counter-example 82, 1 (1979), 43–57. https://doi.org/10.2140/pjm.1979.82.43
	- $Post^{\sharp}$ [if (B) S_1 else S_2][#] $Post^{\sharp}$ [if (B) S_1 else S_2][#] \mathcal{P} $= \{post^{\sharp}[[B;S_{1}]]^{\sharp}P \sqcup^{\sharp}post$ [#] $[-B;S_{2}]]^{\sharp}$ $Post$ [#][if (B) S₁ else S₂][#]7 Post^Hlif (R) S, else \mathbb{C} $\mathbb{$ https://doi.org/10.1016/J.IC.2008.03.025 Ω Ω Ω Ω
- This structural collecting semantics (*) is incomplete This structural collecting semantics (*) is incomplete 1 his struct [29] Tibault Dardinier. 2024. Formalization of Hyper Hoare Logic: A Logic to (Dis-)Prove Program Hyperproperties. Arch.
- Thibault Dardinier and Peter Müller. 2024. Hyper Hoare Logic: (Dis-)Proving Program H
of the ACM on Programming Languages (PACMPL) 8, Issue PLDI, Article No.: 207 (Jun $\frac{1}{\frac{1}{2}}$ the two possible executions of the two possible executions of the conditions of of the ACM on Programming Languages (PACMPL) 8, Issue PLDI, Article No.: 207 (June 2024), 1485–1509. https:
//doi.org/10.1145/3656437 $t/doi.org/10.1145/3656437$ a solution to preserve strictly is to observe that the preserve that $t/doi.org/10.1145/3656437$ (*) of the ACM on Programming Languages (PACMPL) 8, Issue PLDI, Article No.: 207 (June 2024), 1485–1509. https:
//doi.org/10.1145/3656437 ϵ of ϵ //doi.org/10.1145/3656437

P $= \{post^* \llbracket B; S_1 \rrbracket^* P \sqcup^* post^* \llbracket \neg B; S_2 \rrbracket^* P \mid P \in \mathcal{P} \}$ \subseteq {post^{*} \parallel B; S₁ \parallel ⁺ $P_1 \sqcup$ [#] post^{*} \parallel ¬B; S₂ \parallel ⁺ $P_2 \mid P_1 \in \mathcal{P} \land P_2 \in \mathcal{P}$ } $=\{Q_1 \sqcup^{\#} Q_2 \mid Q_1 \in \text{Post}^{\#}[\mathbb{B}; \mathsf{S}_1] \cup^{\#} \mathcal{P} \wedge Q_2 \in \text{Post}^{\#}[\mathbb{B}; \mathsf{S}_2] \cup^{\#} \mathcal{P}\}$ **⊃**
Definition of Postal extensive postal extensive postal extensive postal extensive postal extensive postal exten
Definition of Postel extensive postal extensive postal extensive postal extensive postal extensive postal $P | P \in \mathcal{P} \}$ \subseteq {post[‡][[B; S₁] $\sharp P_1 \sqcup \sharp$ post^{\sharp}[-B; S₂] $\sharp P_2 \mid P_1 \in \mathcal{P} \wedge P_2 \in \mathcal{P}$ } $= \{Q_1 \sqcup^{\sharp} Q_2 \mid Q_1 \in \text{Post}^{\sharp}[\mathbb{B}; \mathsf{S}_1] \sharp \mathcal{P} \land Q_2 \in \text{Post}^{\sharp}[\neg \mathsf{B}; \mathsf{S}_2] \sharp \mathcal{P}\}$ $\mathcal{L} = \mathcal{L} = \mathcal$ $P \mid P \in \mathcal{P} \}$ P } ^𝑃 [∣] ^𝑃 [∈] P} #def. (31) of Post[♯] $= \{post^{\sharp}[[B;S_{1}]]^{\sharp}P \sqcup^{\sharp}post{\sharp}[[\neg B;S_{2}]]^{\sharp}P \mid P \in \mathcal{P}\}\$ $P | P \in \mathcal{P}$ = $\{Q_1 \sqcup^{\sharp} Q_2 \mid Q_1 \in \text{Post}^{\sharp}[\mathbb{B}; S_1] \sharp \mathcal{P} \wedge Q_2 \in \text{Post}^{\sharp}[\mathbb{B}]; S_2] \sharp \mathcal{P}\}$ [24] Patrick Cousot and Radhia Cousot. 1992. Inductive Defnitions, Semantics and Abstract Interpretation. In POPL. ACM $\frac{1}{2}$ Patrick Coussouries Coussouries Semantic Semantic Definitions in Fixpoint, Equations in Fixpoint, Equation [26] Patrick Cousot and Radhia Cousot. 2009. Bi-inductive structural semantics. Inf. Comput. 207, 2 (2009), 258–283. [27] Patrick Cousot and Radhia Cousot. 2012. An abstract interpretation framework for termination. In POPL. ACM, 245– $\mathcal{L}_1 = \{ \mathcal{L}_2 \mid \mathcal{L}_2 \in \mathbb{R}^d \mid \mathcal{L}_2 = \mathcal{L}_2 \cup \{ \mathcal{L}_1 \mid \mathcal{L}_2 \mid \mathcal{L}_3 \mid \mathcal{L}_4 \mid \mathcal{L}_5 \}$ for refactoring with application to extract methods with contracts. In OOPSLA. ACM, 213–232. https://doi.org/10.

This structural collecting semantics (*) is incomplete ϵ \bullet

Complete structural characterization of Post#(*S*) Complete structural characterization of Post#(S) $\frac{1}{2}$. But the classic structural definition (see sect. 3.2) of the transformer Post $\frac{1}{2}$ Te problem is that in (32) that in (32) the two possible executions of the conditions of the co C omplete structural, characterization of $D_{\text{coeff}}(S)$ <u>Complete</u> stricted are sharacterization of 1936 (3)

 ${post[#](S)P} = Post[#]$

P

 $= \{post^* \|B; S_1\|^* P \sqcup^{\sharp} post^* \| \neg B; S_2\|^* P \mid P \in \mathcal{P}\}\$ $= \{Q_1 \sqcup^* Q_2 \mid Q_1 \in \{\text{post*} \parallel B; S_1 \parallel^* P\} \land Q_2 \in \{\text{post*} \parallel \neg B; S_2 \parallel^* P\} \land P \in \mathcal{P}\}$ $= \{Q_1 \sqcup^r Q_2 \mid Q_1 \in \text{PSL}^r \llbracket B; S_1 \rrbracket^r \{P\} \wedge Q_2 \in \text{PSL}^r \llbracket \neg B; S_2 \rrbracket^r \{P\} \wedge P \in P \}$ ^𝑃 [∣] ^𝑃 [∈] P} #def. (31) of Post[♯] $P | P \in \mathcal{P}$ $=\{Q_1 \sqcup^{\sharp} Q_2 \mid Q_1 \in \{\text{post}^{\sharp}[\mathbb{B}; \mathsf{S}_1] \sharp P\} \wedge Q_2 \in \{\text{post}^{\sharp}[\neg \mathsf{B}; \mathsf{S}_2] \sharp P\} \wedge P \in \mathcal{P}\}\$ $= \{Q_1 \sqcup^* Q_2 \mid Q_1 \in \text{Post}^* \llbracket B; S_1 \rrbracket^* \{P\} \wedge Q_2 \in \text{Post}^* \llbracket \neg B; S_2 \rrbracket^* \{P\} \wedge P \in \mathcal{P}\}$ $= \{Q_1 \sqcup^{\sharp} Q_2 \mid Q_1 \in \text{Post}^{\sharp}[\mathbb{B};S_1]^{\sharp} \{P\} \wedge Q_2 \in \text{Post}^{\sharp}[\mathbb{B}^{\sharp};S_2]^{\sharp} \{P\} \wedge P \in \mathcal{P}\}$ \mathcal{D} } $= {P^{0.5t}} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$ $P^{0.5t} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, $S_2 \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$ $I \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$
 $= {Q_1 \cup \begin{bmatrix} 1 & 0 \end{bmatrix} Q_2 \begin{bmatrix} Q_1 \in {\text{post}}^{\sharp} | B; S_1 | \end{bmatrix} | P \} \land Q_2 \in {\text{post}}^{\sharp} | A, S_2 | \]$ $= \{Q_1 \sqcup^{\sharp} Q_2 \mid Q_1 \in \text{Post}^{\sharp}[\mathbb{B};S_1]^{\sharp} \{P\} \wedge Q_2 \in \text{Post}^{\sharp}[\neg B;S_2]^{\sharp} \{P\} \wedge P \in \mathcal{P}\}\$

• We get a complete elementwise characterization of Post#(S)

- Example: **For the construction are placed on the considered on the considered hyperproperties** Post^{\sharp} [if (B) S_1 else S_2][#] ple: $Post$ [#][if (B) S₁ else S₂][#] \mathcal{P} $= \{post^{\sharp}[\mathbb{B}; \mathsf{S}_1]^{\sharp}P \sqcup^{\sharp} post^{\sharp}[\neg \mathsf{B}; \mathsf{S}_2]^{\sharp}$ • Example: \bullet that the calculation to preserve that the calculation goes on at (32) \bullet calculation goes on at (32) \bullet = Post
France $\begin{array}{ll} \texttt{H}\parallel 1\texttt{I} & \texttt{(B)} & \texttt{S}_1 & \texttt{else} & \texttt{S}_2 \parallel^* P' \\ \texttt{H}\parallel \texttt{R}\cdot \texttt{S}_1 \parallel^* p & \texttt{H} & \texttt{next}^{\sharp} \llbracket -\texttt{R}\cdot \texttt{S}_2 \parallel^{\sharp} p \mid p \in \mathcal{B} \end{array}$ $\left[\begin{array}{ccc} 0 & 0 & 1 \ 0 & 1 & 0 \end{array} \right]$.
	- We get a complete elementwise characterization of Post#(S) • We get a complete elementwise characterization of Post#(S) e vve get a complete digition. $\overline{\mathbf{B}}$!¬B;S2"[♯] • We get a complete elementwise characterization of Post#(*S*)

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

Calculational design of the algebraic hyperlogic rules

Upper and lower algebraic hyperlogics \mathbf{r} and ♯ lower algeb Defning the upper and lower logic triples Joper a Opper and lower arged

- The proof system is derived by calculational design (as in POPL 2024) (where for symmetry, we can write a symmetry, we can write $\frac{1}{2}$ $\$ generalizations of Hoare logic $[5,5]$ and incorrectness logic $[3,5]$ from execution to semantic \mathcal{S}
- $\overline{\mathcal{L}}$ \mathcal{Q} = Post^{\sharp} $[\mathsf{S}]^{\sharp} \mathcal{P} \subseteq \mathcal{Q}$

• Definition

$\{P\}\$ S $\{Q\}$ = $Post^{\sharp}[S]^{\sharp}$ $\{\mathcal{P}\}\$ S $\{\mathcal{Q}\}\$ = $Post^{\sharp}$ $[$ S $]$ $\sharp \mathcal{P} \subseteq \mathcal{Q}$ ${\mathcal{P}} \mathop{\mathcal{P}} S {\mathop{\mathcal{Q}}\mathop{\mathcal{Q}}\} = {\mathcal{Q}} \subseteq \text{Post}^{\sharp}[\![S]\!]^{\sharp} \mathcal{P}$ $\overline{\mathfrak{h}}$

P = ∀𝑄 ∈ Q *.* ∃𝑃 ∈ P *.* post[♯]

 $\left(P_e = \text{Ifp}^{\equiv \frac{1}{2}F}\right)$ $F^{\scriptscriptstyle\rm \parallel}_p$

- Requires an *EXACT* characterization of the program semantics of the semantics. Tis is because, for complete the complete series and in full generality, hyper-ality, hyperlogics cannot make any approximation of the program semantics definition of the program semantics definition of
Definition of the program semantics definition of the program semantics definition of the program semantics
- Unmanageable in practice

Upper algebraic hyperlogic for iteration \Box P, P, ({v} also krais by portagis for iteration ebraic hyperlogic for iteration

 $\{P_e(P') \wedge \{P_e\} \}$ ¬B $\{\{Q_e\}\}\wedge \{P_e\}$ $\}$ B; S $\{\{Q_b\}\}\wedge \{P_e\}$ $\{P_e\}\}\$ B; S $\{Q_{\perp e}\}\}\wedge Q_{\perp b} = gfp^{\equiv \frac{\pi}{6}} F_{p\perp}^{\sharp} \wedge P' \in \mathcal{P}\} \Rightarrow$ $((e: Q_e \sqcup_e^{\sharp} Q_b, \perp: Q_{\perp}e \sqcup_{\infty}^{\sharp} Q_{\perp}b, br:P_{br}) \in \mathcal{Q})$ $\overline{2}$ $\overline{2}$ While (B) S $\overline{3}$ Q $\overline{8}$

Notice that no consequence rule is required for completeness, although they are sound ◯^A .

Abstractions

Abstractions

• Since proofs of general hyperproperties are unmanageable, we consider abstractions of • the algebraic semantics • program properties • program hyperproperties • program logics

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 ides

- abstraction to ∀*∃* hyper Lemma 18.4. The chain limit order ideal aboth 18.2 Forall Existence Hyperproperties
	- ∀*∃* hyperproperties (for traces in ∏) $\{P \in \wp(\Pi) \mid \forall \pi_1 \in P : \exists \pi_2 \in P : \langle \pi_1, \pi_2 \rangle \in A\} \mid A \in \wp_{\text{fwhler}}(P)$ is $\exists \pi_2 \in P : \langle \pi_1, \pi_2 \rangle \in A\}$

LEMMA 1.6. $\mathcal{O}[\![\alpha]\!]$

, Vol. 1, No. 1, Article . Publ $n = 37$
37, N_{el} for N_{el} for a low variable and N_{el} and N_{el}

Chain limit order ideal abstraction \hat{A} **CHAIR INITER** 1126 1127

- $\alpha^{\uparrow}(\mathcal{P})$ \triangleq { $\bigsqcup P_i | \langle P_i, i \in \mathbb{N} \rangle \in \mathcal{P}$ is an increasi α^{\perp} 𝑖∈**^N** \uparrow \uparrow $\bigcup_{i\in\mathbb{N}}$ $\bigcup_{i\in\mathbb{N}}$ $\bigcup_{i\in\mathbb{N}}$ $\bigcup_{i\in\mathbb{N}}$ $\bigcup_{i\in\mathbb{N}}$ $\alpha^{\uparrow}(P) \triangleq \{ | \, |P_i| \, \langle P_i, i \in \mathbb{N} \rangle \in \mathcal{P} \text{ is an i } \}$
	- $\begin{array}{ccc} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{array}$
- $\alpha^{\perp\uparrow}$ $\stackrel{\scriptscriptstyle\triangle}{=}$ α^{\perp} $\begin{array}{ccc} -\uparrow & & \uparrow & \uparrow \end{array}$ $\frac{1}{2}$
	- $\overset{*}{\alpha}$ $(\mathcal{P}) \triangleq \text{ If } p^{\epsilon} \lambda X \cdot \mathcal{P} \cup \alpha^{\epsilon} (X) \text{ (upper cl } \epsilon)$ $\alpha^{\pm \uparrow}(\mathcal{P}) \triangleq$ lfp^{$\in \lambda X \cdot \mathcal{P} \cup \alpha^{\pm \uparrow}(X)$ (u} (\mathcal{S}) = 1 fp^{\subseteq} $\lambda X \cdot \mathcal{P} \cup \alpha^{\equiv 1}(X)$ (u
	- in particular for traces: ↑(℘(℘(Π))) subsume ∀∃ hyperproperties in AEH in that ◯^A
- Λ Ω = $\frac{*}{\alpha}$ \int Ω \int Ω $A\Sigma H \subseteq \alpha'(\wp(\wp(\Pi)))$

∗ ↓

1096

 777

Conclusion

Conclusion

- 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)
- (sound & complete but unmanageable in practice)
- All this for terminating and nonterminating executions

• We have introduced a new algebraic semantics (instantiable

• We have designed by calculus a general algebraic hyperlogic

Conclusion (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

- 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

More in the paper

POPL 2024, London © P. Cousot 28

Conclusion of the conclusion

- A transformational hyperlogic is
	- an abstract interpretation of
		- an hypertransformer
			- of
			- an instantiation
				- of
		- an algebraic semantics.

(Conclusion of the conclusion)-1

A (hyper)logic is another (complicated) way of defining

- an abstract interpretation
	- of

of

- an instantiation
- an algebraic semantics.

- Online full version of the clickable paper + appendix:
	- auxiliary material of the ACM digital library
	- my web page (https://cs.nyu.edu/~pcousot/) + slides
	- arXiv<https://arxiv.org/abs/2411.11113>
	- Zenodo <https://zenodo.org/records/14173478>

The End, Thank You