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

PATRICK COUSOT, New York University, USA

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

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

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

ACM Reference Format:
Patrick Cousot. 2024. Calculational Design of [In]Correctness Transformational Program Logics by Abstract Interpretation. *Proc. ACM Program. Lang.* 8, POPL, Article 7 (January 2024), 34 pages. https://doi.org/10.1145/3632849

1 INTRODUCTION

In verification, the focus is on which program properties can be expressed and proved. We discuss transformational (or Hoare's style) logics characterized by formulas expressing program properties that relate initial/input values of variables to their final/output values, nontermination, or runtime errors (or inversely final to initial) and a Hilbert-style proof system [Hilbert and Ackermann 1959, §10] to prove that a program has a property expressed by a formula of the logic (but not that a given program does not have a property expressed by a formula of the logic or that no program can have this property [Kim et al. 2023]). Examples are Hoare's logic [Hoare 1969] and the reverse Hoare logic [de Vries and Koutavas 2011] aka incorrectness logic [O'Hearn 2020].

1.1 The Classic Proof-Theoretic Approach

The "classic approach" to the design of a Hoare style logic follows the proof-theoretic semantics in logic originated by Hilbert, Gentzen, Prawitz, and others [Piecha and Schroeder-Heister 2019]. The true program properties are the provable ones, which is also the idea of "axiomatic semantics" [Winskel 1993], that is, Floyd's idea that a program proof method is "Assigning Meaning to Programs" [Floyd 1967]. First the syntax of program properties is defined (e.g. $P\{C\}Q$, $\{P\}C\{Q\}$, $[P]C[Q]$). Then proof rules are postulated (e.g. "If $\vdash P\{Q\}R$ and $\vdash R \supset S$ then $\vdash P\{Q\}S$" [Hoare

Author's address: Patrick Cousot, pcousot@cims.nyu.edu, New York University, New York, USA.

ACM 2475-1421/2024/1-ART7
https://doi.org/10.1145/3632849

1969, page 578]). Finally, soundness and completeness theorems are proved to relate the logic properties to a more concrete/refined semantics (e.g. years after its design, Hoare logic [Hoare 1969] was proved sound by Donahue [Donahue 1976] (with respect to a denotational semantics) and sound and relatively complete by Pratt [Pratt 1976] (with respect to a relational semantics excluding nontermination) and Cook [Cook 1978, 1981] (with respect to an operational trace semantics)). This design method has perdured over time, even if, nowadays, soundness and completeness proofs are often published together with the logic (e.g. [Bruni et al. 2023; Dardinier 2023; de Vries and Koutavas 2011; Gotsman et al. 2011; Möller et al. 2021; O'Hearn 2020; Vanegue 2022; Zhang et al. 2022; Zhang and Kaminski 2022; Zilberstein et al. 2023] a.o.). Therefore, in this "classic approach" the program properties of interest (partial correctness, total correctness, incorrectness, etc) are the one provable by the proof system, while soundness and completeness theorems aims at connecting the provable properties to the program semantics.

1.2 The Model-Theoretic Semantic Abstraction Approach

In this paper, we consider an alternative "semantic abstraction approach" which is based on Tarski's truth paradigm [Tarski 1933] in model theory and the abstract interpretation of the semantics of languages [Cousot 2021; Cousot and Cousot 1977]. First, a formal semantics is specified for the language (preferable using structural fixpoints or deductive proof systems). This induces a collecting semantics defining the strongest (hyper) property of programs. Then the program properties of interest for the logic are specified by a Galois connection abstracting the collecting (hyper) properties. The abstraction is usually the composition of several primitive ones, in the spirit of [Cousot and Cousot 2014]. Varying the primitives and their composition yields different logics. At this point, the logic is precisely and fully determined since all expressible properties of all programs have been formally specified. For example, the logic can be compared and combined with other logics (see e.g. Figs. 1, 2, 3 and the taxonomy in Sect. I.3.14). Finally the rules of the proof system are designed by calculus using fixpoint abstraction (Sect. II.2), fixpoint induction principles (Sect. II.3), and Peter Aczel [Aczel 1977] construction of deductive rule-based systems from fixpoints, or conversely (Sect. II.5).

The advantage is that reasoning on abstractions of program properties is much more concise and easy than reasoning on proof systems. This clearly appears e.g. in Fig. 2 comparing 40 logics by combining only 8 abstractions (plus one, common to all logics defining "transformational"). Fig. 2 is itself part of the lattice of abstract interpretations of [Cousot and Cousot 1977, section 8] including many logics whose abstraction is given in this paper. Another advantage is that the proof system is derived by calculus so sound and complete by construction.

1.3 The Structure of the Paper

The paper has two main parts. In the first part, we characterize the semantics of a transformational logics, i.e. the true formulas (a theory in logic), as an abstract interpretation of the program (collecting) semantics. This allows us to provide a taxonomy of transformational semantics by comparing their abstractions, without referring to their proof systems.

After showing that theories of logics are set abstractions of the program (collecting) semantics in the first part, we have to design the corresponding proof systems in the second part.

Aczel has shown that deductive rule-based systems and set-theoretic fixpoint definitions are equivalent [Aczel 1977]. Therefore we first define the program semantics in fixpoint form, then abstract this semantics to get a fixpoint definition of the theory of the logic, and finally apply Aczel's method to derive the equivalent proof system. The proof system is then sound and complete by construction.

Hyper references Ⓐ refer to the full paper on Zenodo [Cousot 2024]. 10.5281/zenodo.10439108

Part I: Design of the Theory of Logics by Abstraction of the Program Semantics

In part I, we show that the theory (or semantics) of transformational logics are abstractions of the relational semantics of programs, which leads to a taxonomy of transformational logics, as well as, to their combinations. The meaning or semantics of a logic is the set of true formulas of that logic which is also called the theory of the logic. So we use "theory" for the meaning of a logic and "semantics" for the meaning of a program or a programming language.

I.1 RELATIONAL SEMANTICS

"Relational" means that the semantics defines a relation between initial states of executions and final states or $\bot$ to denote nontermination (as conventional in denotational semantics [Scott and Strachey 1971]). Our notations on relations are classic and defined in the appendix Ⓐ.

I.1.1 Structural Deductive Definition of the Natural Relational Semantics

We consider an imperative language $\mathbb{S}$ with assignments, sequential composition, conditionals, and conditional iteration with breaks. The syntax is $\mathtt{S} \in \mathbb{S} ::= \mathtt{x = A} \mid \mathtt{x = [}a\mathtt{,}b\mathtt{]} \mid \mathtt{skip} \mid \mathtt{S;S} \mid \mathtt{if\ (B)\ S\ else\ S} \mid \mathtt{while\ (B)\ S} \mid \mathtt{break}$. The nondeterministic assignment $\mathtt{x = [}a\mathtt{,}\ b\mathtt{]}$ with $a \in \mathbb{Z} \cup \{-\infty\}$ and $b \in \mathbb{Z} \cup \{\infty\}$, $-\infty - 1 = -\infty$, $\infty + 1 = \infty$ may be unbounded. break is a simple form of exception (to answer a question on exceptions by Matthias Felleisen at POPL 2014 [Cousot and Cousot 2014]).

States $\sigma \in \Sigma \triangleq \mathbb{X} \to \mathbb{V}$ (also called environments) map variables $\mathtt{x} \in \mathbb{X}$ to their values $\sigma(\mathtt{x})$ in $\mathbb{V}$ including integers, $\mathbb{Z} \subseteq \mathbb{V}$. We let $\bot \notin \Sigma$ denote nontermination with $\Sigma_\bot \triangleq \Sigma \cup \{\bot\}$.

We deliberately leave unspecified the syntax and semantics of arithmetic expressions $\mathcal{A}[\![\mathtt{A}]\!] \in \Sigma \to \mathbb{V}$ and Boolean expressions $\mathcal{B}[\![\mathtt{B}]\!] \in \wp(\Sigma) \simeq \Sigma \to \{\text{true}, \text{false}\}$. The only assumption on expressions is the absence of side effects.

The relational semantics $[\![\mathtt{S}]\!]_\bot$ of a command $\mathtt{S} \in \mathbb{S}$ is an element of $\wp(\Sigma \times \Sigma_\bot)$. Formally, $\langle\sigma, \sigma'\rangle \in [\![\mathtt{S}]\!]_\bot$ means that an execution of the nondeterministic command S from initial state $\sigma \in \Sigma$ may terminate in final state $\sigma' \in \Sigma$ or may not terminate when $\sigma' = \bot$. (The relational semantics could have been proven to be the abstraction of a finite and infinite trace semantics [Cousot 2021].) The right-image $\boldsymbol{\lambda}\sigma \bullet \{\sigma' \in \Sigma_\bot \mid \langle\sigma, \sigma'\rangle \in [\![\mathtt{S}]\!]_\bot\}$ of the natural relational semantics $[\![\mathtt{S}]\!]_\bot$ is isomorphic to Plotkin's natural denotational semantics [Plotkin 1976]. Such natural relational semantics have been originated by Park [Park 1979].

We partition the relational natural semantics into the semantics $[\![\mathtt{S}]\!]^e \in \wp(\Sigma \times \Sigma)$ of statement S terminating/ending normally, the semantics $[\![\mathtt{S}]\!]^b \in \wp(\Sigma \times \Sigma)$ of statement S terminating by a break statement, and the semantics $[\![\mathtt{S}]\!]^\bot \in \wp(\Sigma \times \{\bot\})$ of nontermination $\bot$. Therefore $[\![\mathtt{S}]\!]_\bot \triangleq [\![\mathtt{S}]\!]^e \cup [\![\mathtt{S}]\!]^b \cup [\![\mathtt{S}]\!]^\bot$. The angelic semantics

$$[\![\mathtt{S}]\!] \quad \triangleq \quad [\![\mathtt{S}]\!]^e \cup [\![\mathtt{S}]\!]^b \quad = \quad [\![\mathtt{S}]\!]_\bot \cap (\Sigma \times \Sigma) \tag{1}$$

ignores non termination.

We follow the tradition established by Plotkin [Plotkin 2004a,b] to define the program semantics by structural induction (i.e. by induction on the program syntax) using a deductive system of rules. We extend the semantics of the deductive system using bi-induction combining induction for terminating executions and co-induction for nonterminating ones [Cousot and Cousot 1992, 1995, 2009].

Let us write judgements $\sigma \vdash \mathtt{S} \overset{e}{\Rightarrow} \sigma'$ for $\langle\sigma, \sigma'\rangle \in [\![\mathtt{S}]\!]^e$, $\sigma \vdash \mathtt{S} \overset{b}{\Rightarrow} \sigma'$ for $\langle\sigma, \sigma'\rangle \in [\![\mathtt{S}]\!]^b$, and $\sigma \vdash \mathtt{S} \overset{\infty}{\Rightarrow}$ for $\langle\sigma, \bot\rangle \in [\![\mathtt{S}]\!]^\bot$. Moreover, for the conditional iteration statement $\mathtt{W} \triangleq \mathtt{while\ (B)\ S}$, we write $\sigma \vdash \mathtt{W} \stackrel{i}{\Rightarrow} \sigma'$ to mean that if σ is a state before executing W, then σ' is reachable after 0 or more iterations of the loop body (so $\sigma = \sigma'$ for 0 iterations, before entering the loop in case (2.a)). We have the axiom and inductive rule for iterations W

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

The following axioms define termination (these are axioms since the precondition has been previously established either by $\stackrel{i}{\Rightarrow}$ or by structural induction). (3.b) is for termination by a `break`.

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

The following axiom and co-inductive rule define nontermination (the left rule is an axiom since the precondition has already been defined either by $\stackrel{i}{\Rightarrow}$ or by structural induction). Rule (4.b) right-marked ∞ is co-inductive.

$$\text{(a)}\quad \frac{\sigma \vdash \mathtt{W} \stackrel{i}{\Rightarrow} \sigma', \quad \mathcal{B}[\![\mathtt{B}]\!]\sigma', \quad \sigma' \vdash \mathtt{S} \stackrel{\infty}{\Rightarrow}}{\sigma \vdash \mathtt{W} \stackrel{\infty}{\Rightarrow}} \qquad\qquad \text{(b)}\quad \frac{\mathcal{B}[\![\mathtt{B}]\!]\sigma, \quad \sigma \vdash \mathtt{S} \stackrel{e}{\Rightarrow} \sigma', \quad \sigma' \vdash \mathtt{W} \stackrel{\infty}{\Rightarrow}}{\sigma \vdash \mathtt{W} \stackrel{\infty}{\Rightarrow}}\infty \tag{4}$$

I.1.2 State Properties, Semantics Properties, and Collecting Semantics

We define properties in extension as the set of elements of a universe $\mathbb{U}$ that have this property. So false is $\varnothing$, true is $\mathbb{U}$, logical implication is $\subseteq$, disjunction is $\cup$, conjunction is $\cap$, negation is $\neg P \triangleq \mathbb{U} \setminus P$ and $\langle \wp(\mathbb{U}),\ \varnothing,\ \mathbb{U},\ \cup,\ \cap,\ \neg\rangle$ is a complete Boolean lattice [Grätzer 1998].

For example, properties of states $\sigma \in \Sigma_\perp$ (considered to be the universe) belong to $\wp(\Sigma_\perp)$. The singleton $\{\perp\}$ is the property "not to terminate", $\varnothing$ is "false", $\{\sigma_1, \ldots, \sigma_n\} \subseteq \Sigma$ is "to terminate with any one of the states $\sigma_1, \ldots, \sigma_n \in \Sigma$", $\{\sigma_1, \ldots, \sigma_n, \perp\}$ is " "to terminate with any one of the states $\sigma_1, \ldots, \sigma_n \in \Sigma$ or not to terminate", Σ is to terminate, $\Sigma_\perp$ is "true" i.e. "to terminate with any state in Σ or not to terminate" (the common alternative to terminate with an error is assumed to be encoded with some specific values in the set Σ of states).

Let $[\![\mathtt{S}]\!]_\perp \in \wp(\Sigma \times \Sigma_\perp)$ be the natural relational semantics of programs $\mathtt{S} \in \mathbb{S}$ in Sect. I.1.1. When defined in extension, semantic properties belong to $\wp(\wp(\Sigma \times \Sigma_\perp))$. The program collecting semantics $\{\!|\mathtt{S}_\perp|\!\} \triangleq \{[\![\mathtt{S}_\perp]\!]\} \in \wp(\wp(\Sigma \times \Sigma_\perp))$ is the strongest (hyper) property of program S.

I.2 GALOIS CONNECTIONS

Galois connections [Cousot 2021, Ch. 11] are used throughout the paper. They formalize correspondences between program properties which preserve implication and one is less precise/expressive than the other. The interest is that proofs in the abstract are valid in the concrete (or equivalent in case of Galois isomorphisms). Moreover, there is a most precise way to abstract any concrete property or logic, which provides a guideline for calculational design of logics from a program semantics. The definition and properties of Galois connections are recalled in the appendix Ⓐ.

I.3 THE DESIGN OF A NATURAL TRANSFORMATIONAL LOGIC THEORY BY COMPOSING ABSTRACTIONS OF THE NATURAL RELATIONAL SEMANTICS

A program logic consists of formal statements some of which are true and constitute the theory of the logic. Our objective in this section is to characterize the theory of transformational logics by abstraction of the natural relational collecting semantics. This abstraction is obtained by composition of basic Galois connections and functors introduced in this section.

Example I.3.1. The body `fact` $\equiv$ `while (n!=0) { f = f * n; n = n - 1;}` of the factorial `f = 1; fact` can be specified as $\{n = \underline{n} \wedge f = 1\}$ `fact` $\{(\underline{n} \geqslant 0 \wedge f = !\underline{n}) \vee (\underline{n} < 0 \wedge n = f = \perp)\}$ where, following

Manna [Manna 1971], $\underline{\mathtt{x}}$ or $\mathtt{x}_0$ denotes the initial value of variable x in the postcondition. When later incorporating break statements, the specification will be $\{n = \underline{n} \wedge f = 1\}\,\mathtt{fact}\,\{ok : (\underline{n} \geqslant 0 \wedge f = !\underline{n}) \vee (\underline{n} < 0 \wedge n = f = \bot),\ br : \mathsf{false}\}$.

Assuming $P \neq \varnothing$ (false) and $\bot \notin Q$, $\{P\}\mathtt{S}\{Q\}$ specifies total correctness, as does Manna and Pnueli logic [Manna and Pnueli 1974]. $\{P\}\mathtt{S}\{\bot\}$ specifies definite non termination. Otherwise, when $\bot \in Q$, $\{P\}\mathtt{S}\{Q\}$ expresses partial correctness, as does Hoare logic [Hoare 1969].

By adding auxiliary variables (see Sect. E.1 in the appendix), this specification can also be partially formulated by two Hoare triples $\{n = \underline{n} \geqslant 0 \wedge f = 1\}\,\mathtt{fact}\,\{f = !\underline{n}\}$ (although not ensuring termination) and $\{n < 0 \wedge f = 1\}\,\mathtt{fact}\,\{\mathsf{false}\}$ (ensuring nontermination) but the conjunction of Hoare triples is not a Hoare triple and anyway the partial specification cannot preclude nontermination when $\underline{n} \geqslant 0$.

This specification cannot be expressed by Manna and Pnueli [Manna and Pnueli 1974] logic since the program is not totally correct.

The theory of the adequate logic (that we call the natural transformational over approximation logic) will be formally specified in (13) as $\{P\}\mathtt{S}\{Q\}$, $P \in \wp(\Sigma\times\Sigma)$, $Q \in \wp(\Sigma\times\Sigma_\bot)$ if and only if $\forall\langle\underline{\sigma}, \sigma\rangle \in P\ .\ \forall\sigma'\ .\ \langle\sigma, \sigma'\rangle \in [\![\mathtt{S}]\!]_\bot \Rightarrow \langle\underline{\sigma}, \sigma'\rangle \in Q$. The proof system of this logic is designed in Sect. II.8.1. ■

I.3.1 Collecting Semantics to Semantics Abstraction

The collecting semantics of a program component is its strongest property, so transformational logic statements are weaker abstract properties that we specify by composition of Galois connections. The first abstraction α_C abstracts hyper properties into properties.

Let $\mathcal{D}$ be a set (e.g. $\mathcal{D} = \Sigma\times\Sigma_\bot$ for the natural relational semantics of Sect. I.1.1). There is a Galois connection

$$\langle\wp(\wp(\mathcal{D})), \subseteq\rangle \xleftarrow[\alpha_C]{\gamma_C} \langle\wp(\mathcal{D}), \subseteq\rangle \tag{5}$$

where $\alpha_C(P) \triangleq \bigcup P$ is surjective and $\gamma_C(S) \triangleq \wp(S)$ is injective (since $\alpha_C(P) \subseteq S \Leftrightarrow \bigcup P \subseteq S \Leftrightarrow P \subseteq \wp(S) \Leftrightarrow P \subseteq \gamma_C(S)$).

Example I.3.2. If $\mathcal{D}$ is a set of finite or infinite traces, $[\![\mathtt{S}]\!]_\bot$ defines the finite or infinite execution traces of S, $\{\!|\mathtt{S}|\!\}_\bot$ is the strongest hyper property of program S [Clarkson and Schneider 2010], and $\alpha_C(\{\!|\mathtt{S}|\!\}_\bot) = [\![\mathtt{S}]\!]_\bot$ is the strongest semantic property of S (called a trace property in [Clarkson and Schneider 2010]). ■

Our first abstraction is therefore $\alpha_C(\{\!|\mathtt{S}|\!\}_\bot) = \alpha_C(\{[\![\mathtt{S}]\!]_\bot\}) = [\![\mathtt{S}]\!]_\bot$ where this natural relational semantics defines in Sect. I.1.1 specifies the program properties of interest.

I.3.2 Semantics to Relational Postcondition Transformer Post Abstraction

While the natural relational semantics establishes a relation between initial and final states or nontermination, the postcondition transformers establish a relation between properties of initial states and properties of final states or nontermination. The postcondition may be an assertion on final states only (as in Hoare partial correctness logic [Hoare 1969]) or a relation between initial and final states (as in Manna partial correctness [Manna 1971]). The postcondition may also include nontermination. Although Hoare logic is assertional, the initial values of variables can be recorded into auxiliary variables (see Sect. E.1 in the appendix). We start with the relational case since assertional property transformers are abstractions of relational ones (as shown in Sect. I.3.6).

The relational postcondition transformer Post is also called the relational forward/right-image/post-image/strongest consequent/strongest post condition.

$$\begin{array}{rcl}\mathsf{Post} & \in & \wp(\mathcal{X}\times\mathcal{Y}) \to \wp(\mathcal{Z}\times\mathcal{X}) \to \wp(\mathcal{Z}\times\mathcal{X}) \\ \mathsf{Post}(r)P & \triangleq & \{\langle\sigma_0, \sigma'\rangle \mid \exists\sigma\ .\ \langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in r\}\end{array} \tag{6}$$

$\mathrm{Post}(\alpha_C(\{\!|\mathtt{S}|\!\}_\perp))P = \mathrm{Post}[\![\mathtt{S}]\!]_\perp P$ is a relation between initial states σ related to σ_0 satisfying the precondition P and final states σ' related to σ_0 upon termination of $\mathtt{S}$ or $\sigma' = \perp$ in case of nontermination. This is the basis for the natural relational transformational logic (as in example I.3.1 and Sect. II.8.1), except for the use of a transformer instead of logic triples. We will later prove in (35) that Post is the lower adjoint of a Galois connection.

Example I.3.3. Incrementation is characterized by $\mathrm{Post}(\mathtt{x = x+1})(\mathtt{x = x_0}) = (\mathtt{x = x_0 + 1})$ which, representing the semantics and relational properties as sets, is $\mathrm{Post}(\{\langle x,\ x+1\rangle \mid x \in \mathbb{Z}\})\{\langle x_0, x\rangle \mid x = x_0 \in \mathbb{Z}\} = \{\langle x_0, x'\rangle \mid \exists x\ .\ \langle x_0, x\rangle \in \{\langle x_0, x\rangle \mid x = x_0 \in \mathbb{Z}\} \wedge \langle x, x'\rangle \in \{\langle x, x{+}1\rangle \mid x \in \mathbb{Z}\}\} = \{\langle x_0, x_0 + 1\rangle \mid x_0 \in \mathbb{Z}\}$. Here, σ_0 is the initial value of the variables before the assignment but, in general, this initial relation can be arbitrary. More generally, Floyd's strongest postcondition for assignment $\mathtt{x\ :=\ A}$ [Floyd 1967] is $\mathrm{Post}[\![\mathtt{x\ :=\ A}]\!]P = \{\langle\sigma_0,\ \sigma'\rangle \mid \exists\sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in P \wedge \sigma' = \sigma[\mathtt{x} \leftarrow \mathcal{A}[\![\mathtt{A}]\!]\sigma]\}$. ■

I.3.3 Relational Postcondition Transformer to Antecedent/Consequent Pairs

Transformational logic triples $\{P\}\mathtt{S}\{Q\}$ associate pairs $\langle P,\ Q\rangle$ of predicates to each program command $\mathtt{S} \in \mathbb{S}$. So the theory of the logic is the set $\{\langle P,\ Q\rangle \mid \{P\}\mathtt{S}\{Q\}\}$ for each statement $\mathtt{S}$. For the natural transformational logic, this theory contains the graph of the $\mathrm{Post}([\![\mathtt{S}]\!]_\perp)$ function. Conversely, from this graph, we can recover the strongest valid triples $\{P\}\mathtt{S}\{Q\}$. (Notice that we say "contains" not "is" and "strongest" since, in absence of a consequence rule, the graph does not contain all valid triples, only the $\{P\}\mathtt{S}\{\mathrm{Post}([\![\mathtt{S}]\!]_\perp)P\}$ ones. Consequence rules will be introduced thanks to another abstraction discussed in next Sect. I.3.4.)

More generally, a function $f \in \mathcal{X} \to \mathcal{Y}$ is isomorphic to its graph $\alpha_{\mathrm{G}}(f) = \{\langle x,\ f(x)\rangle \mid x \in \mathcal{X}\}$. This graph $\alpha_{\mathrm{G}}(f)$ is a functional relation. We have the Galois isomorphism Ⓐ

$$\langle \mathcal{X} \to \mathcal{Y},\ \dot{=}\rangle \underset{\alpha_{\mathrm{G}}}{\overset{\gamma_{\mathrm{G}}}{\leftrightarrows}} \langle \wp_{\mathsf{fun}}(\mathcal{X} \times \mathcal{Y}),\ =\rangle \tag{7}$$

where $\gamma_{\mathrm{G}}(r) \triangleq \boldsymbol{\lambda} x \bullet (y \textit{ such that } \langle x,\ y\rangle \in r)$ is uniquely well-defined since r is a functional relation. We have Ⓐ

$$\alpha_{\mathrm{G}}(\mathrm{Post}(\alpha_C(\{\!|\mathtt{S}|\!\}_\perp))) = \{\langle P,\ \{\langle\sigma_0,\ \sigma'\rangle \mid \exists\sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in P \wedge \langle\sigma,\ \sigma'\rangle \in [\![\mathtt{S}]\!]_\perp\}\rangle \mid P \in \wp(\Sigma \times \Sigma)\} \tag{8}$$

So $\alpha_{\mathrm{G}}(\mathrm{Post}(\alpha_C(\{\!|\mathtt{S}|\!\}_\perp)))$ is the set of pairs $\langle P,\ Q\rangle$ such that Q is the strongest relational postcondition of P for the natural relational semantics $[\![\mathtt{S}]\!]_\perp$. It is not a program logic since, as was the case for transformers, it is missing a consequence rule.

Example I.3.4. Floyd/Hoare logic rules [Hoare 1978] provide the strongest assertional post-condition except for the iteration and consequence rule, e.g., $\{P\}\mathtt{skip}\{P\}$ is $\{P\}\mathtt{skip}\{\mathrm{post}([\![\mathtt{skip}]\!])P\}$ (see (10) below for the classic definition of post). But excluding the consequence rule and using the following iteration rule (for bounded nondeterminism)

$$\frac{I^0 = P, \quad \forall n \in \mathbb{N}\ .\ \{I^n \wedge \mathtt{B}\}\,\mathtt{S}\,\{I^{n+1}\}}{\{P\}\,\mathtt{while\ (B)\ S}\,\{\exists n \in \mathbb{N}\ .\ I^n \wedge \neg B\}} \tag{9}$$

would yield the strongest post condition in all cases. ■

I.3.4 Weakening and Strengthening Abstractions

Following [Burstall 1969] to make program proofs using the natural relational semantics proof rules 2–4, or, by (8), the transformer $\mathrm{Post}([\![\mathtt{S}]\!]_\perp)$ or, isomorphically by (7), its graph $\{\langle P,\ \{\langle\sigma_0,\ \sigma'\rangle \mid \exists\sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in P \wedge \langle\sigma,\ \sigma'\rangle \in [\![\mathtt{S}]\!]_\perp\}\rangle \mid P \in \wp(\Sigma \times \Sigma)\} \in \wp(\wp(\Sigma \times \Sigma) \times \wp(\Sigma \times \Sigma_\perp))$ is inadequate since the semantics describes executions exactly, without any possibility of approximation.

In contrast, as first shown by Turing [Morris and Jones 1984; Turing 1950], using executions properties is the basis for elegant and concise program correctness proofs since it allows for approximations.

This is even implicitly acknowledged by the most enthusiastic supporter of transformers. Edsger W. D. Dijkstra in [Dijkstra 1976] has chapters 0 to 4 defining predicate transformers until chapter 5 introducing properties weakening by implication (i.e. one form of approximation) as well as the "Fundamental Invariance Theorem for Loops" (i.e. fixpoint induction Th. II.3.1 replacing the strongest loop invariant by weaker ones). Moreover, in chapter 6, it is explained how "to choose an appropriate proof for termination" (for bounded nondeterminism). Iterative program design and proofs are only considered after over approximation (invariance) and under approximation (for termination) have been introduced, from chapter 7 on. We have to do the same, but for any transformer (including $\mathsf{Post}[\![\mathsf{S}]\!]_\perp$).

For that purpose, we introduce weakening and strengthening abstractions. Consequence rules, understood as an abstraction losing precision on program properties, will be a specific instance for a specific transformer. We also need compatible general induction principles to handle loops (of which invariance and (non)termination will be specific instances). Such induction principles are not relative to expressivity but to proofs, and so will be considered in part 2 of the paper.

I.3.4.1 The Over Approximation Abstraction. Pairs of properties $\langle P, Q\rangle \in R \in \wp(\wp(\mathcal{X})\times\wp(\mathcal{Y}))$ can be approximated by weakening or strengthening P and/or Q. For Hoare logic [Hoare 1969], we can strengthen P by $P' \subseteq P$ and weaken Q by Q' such that $Q \subseteq Q'$. This is the over approximation abstraction $\mathsf{post}(\supseteq,\subseteq)R = \{\langle P', Q'\rangle \mid \exists\langle P, Q\rangle \in R \,.\, \langle P, Q\rangle \supseteq,\subseteq \langle P', Q'\rangle\} = \{\langle P', Q'\rangle \mid \exists\langle P, Q\rangle \in R \,.\, P \supseteq P' \wedge Q \subseteq Q'\} = \{\langle P', Q'\rangle \mid \exists\langle P, Q\rangle \in R \,.\, P' \subseteq P \wedge Q \subseteq Q'\}$ by defining the classic assertional right image transformer (denoted Xr in [Pratt 1976])

$$\mathsf{post}(r)X \quad \triangleq \quad \{y \mid \exists x \in X \,.\, \langle x, y\rangle \in r\} \tag{10}$$

and the component wise ordering $\sqsubseteq,\preceq$ on pairs

$$\langle x, y\rangle \sqsubseteq,\preceq \langle x', y'\rangle \quad \triangleq \quad x \sqsubseteq x' \wedge y \preceq y' \tag{11}$$

If $r \in \wp(\mathcal{X}\times\mathcal{Y})$, we have the classic Galois connection

$$\langle\wp(\mathcal{X}), \subseteq\rangle \underset{\mathsf{post}(r)}{\overset{\widetilde{\mathsf{pre}}(r)}{\leftrightarrows}} \langle\wp(\mathcal{Y}), \subseteq\rangle \tag{12}$$

where $\widetilde{\mathsf{pre}}(r)Q = \{x \mid \forall y \,.\, \langle x, y\rangle \in r \Rightarrow y \in Q\}$ (see example C.1 in the appendix). The theory of the *natural transformational over approximation logic* is therefore Ⓐ

$$\begin{aligned}\mathsf{post}(\supseteq,\subseteq)(\alpha_{\mathrm{G}}(\mathsf{Post}[\![\mathsf{S}]\!]_\perp)) &= \{\langle P, Q\rangle \mid \mathsf{Post}[\![\mathsf{S}]\!]_\perp P \subseteq Q\} \\ &= \{\langle P, Q\rangle \mid \forall\langle\sigma_0, \sigma\rangle \in P \,.\, \forall\sigma' \,.\, \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp \Rightarrow \langle\sigma_0, \sigma'\rangle \in Q\}\end{aligned} \tag{13}$$

that is, for any initial state σ related to σ_0 by the precondition P and any final state σ' of S, possibly $\perp$, the pair $\langle\sigma_0, \sigma'\rangle$ satisfies the postcondition Q, as considered in example I.3.1. The difference with the interpretation of Manna and Pnueli total correctness logic [Manna and Pnueli 1974] is that we may have $\langle\sigma_0, \perp\rangle \in Q$ thus allowing possible nontermination for some initial pair of states $\langle\sigma_0, \sigma\rangle$ of P. Therefore we can both express both total and partial correctness plus nontermination when $Q = \Sigma \times \{\perp\}$. With this convention, only one of Dijkstra's weakest preconditions transformers [Dijkstra 1975, 1976; Dijkstra and Scholten 1990] is needed since $\mathsf{wlp}(\mathsf{S}, Q) = \mathsf{wp}(\mathsf{S}, Q \cup \{\perp\})$. This is similar to the classic characterization of Hoare logic by a forward transformer, $\{P\}\mathsf{S}\{Q\}$ if and only if $\mathsf{post}[\![\mathsf{S}]\!]P \Rightarrow Q$ given by [Pratt 1976, equation (S), p. 110] or, equivalently by (12), $P \Rightarrow \widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]Q$ [Pratt 1976, equation (w), p. 110] (except that in (13), P and Q are relational and take nontermination into account). By (12), the abstraction $\mathsf{post}(\supseteq,\subseteq)$ is the lower adjoint of a Galois connection.

I.3.4.2 The Under Approximation Abstraction. For the *natural transformational under approximation logic*, as well as reverse Hoare logic [de Vries and Koutavas 2011] aka incorrectness logic [O'Hearn 2020], we can weaken P by $P' \supseteq P$ and strengthen Q by Q' such that $Q \supseteq Q'$. This is the under approximation abstraction $\mathsf{post}(\subseteq,\supseteq)R = \{\langle P',\ Q'\rangle \mid \exists\langle P,\ Q\rangle \in R\ .\ \langle P,\ Q\rangle \subseteq,\supseteq \langle P',\ Q'\rangle\} = \{\langle P',\ Q'\rangle \mid \exists\langle P,\ Q\rangle \in R\ .\ P \subseteq P' \wedge Q \supseteq Q'\} = \{\langle P',\ Q'\rangle \mid \exists\langle P,\ Q\rangle \in R\ .\ P \subseteq P' \wedge Q' \subseteq Q\}$ which is the consequence rule called *Symmetry* in [O'Hearn 2020, Fig. 1] and *Consequence* in [O'Hearn 2020, Fig. 2].

The theory of the natural transformational under approximation logic is therefore Ⓐ

$$\begin{aligned}\mathsf{post}(\subseteq,\supseteq)(\alpha_{\mathrm{G}}(\mathsf{Post}[\![\mathtt{S}]\!]_{\perp})) &= \{\langle P,\ Q\rangle \mid Q \subseteq \mathsf{Post}[\![\mathtt{S}]\!]_{\perp}P\} \\ &= \{\langle P,\ Q\rangle \mid \forall\langle\sigma_0,\ \sigma\rangle \in P\ .\ \forall\sigma'\ .\ \langle\sigma_0,\ \sigma'\rangle \in Q \Rightarrow \langle\sigma,\ \sigma'\rangle \in [\![\mathtt{S}]\!]_{\perp}\}\end{aligned} \tag{14}$$

that is, for any initial state σ related to σ_0 satisfying the precondition P and any final state σ' related to σ_0, possibly $\perp$, if the pair $\langle\sigma_0,\ \sigma'\rangle$ satisfies the postcondition Q then there exists an execution of $\mathtt{S}$ from σ to σ' (possibly non termination). The difference with reverse Hoare logic [de Vries and Koutavas 2011] aka incorrectness logic [O'Hearn 2020] is that we may have $\langle\sigma,\ \perp\rangle \in Q$ thus allowing possible nontermination for some initial states $\langle\sigma_0,\ \sigma\rangle$ of P so we can both express total and partial correctness plus nontermination when $Q = \Sigma \times \{\perp\}$.

Up to the use of relations instead of assertions and the consideration of nontermination $\perp$, this is similar to the classic characterization of reverse Hoare logic aka incorrectness logic by a forward transformer, $\{P\}\mathtt{S}\{Q\}$ if and only if $Q \Rightarrow \mathsf{post}([\![\mathtt{S}]\!])P$ given by [de Vries and Koutavas 2011, section 5] and [O'Hearn 2020, Lemma 3.(2)], showing that both logics have the same semantics/theory (again up to nontermination and relational postconditions). By (12), the abstraction $\mathsf{post}(\subseteq,\supseteq)$ is the lower adjoint of a Galois connection.

I.3.4.3 The Incorrectness Logic is Insufficient to Prove That All Alarms in Static Analysis Are True or False Alarms. Incorrectness logic [O'Hearn 2020] "was motivated in large part by the aim of providing a logical foundation for bug-catching program analyses" [Le et al. 2022]. In particular incorrectness logic is useful to prove that alarms in static analyzers are true alarms. This consists in showing that the alarm is definitely reachable from some input. However, not all alarms are reachable from initial states since static analyses are over approximating reachable states so that unreachable code under the precondition may produce false alarms.

Example I.3.5. Consider the factorial of example I.3.1 specified by $\{f = 1\}\,\mathtt{fact}\,\{f > 0\}$. This contract is obviously satisfied since on exit $f = !\underline{n} > 0$. However, an interval analysis of this program with initially $\underline{\mathtt{n}} \in \mathbb{Z}$ is totally imprecise and will produce an alarm on program exit with postcondition $Q = \mathtt{f} \leqslant 0$. This is a false alarm since the loop exit is unreachable. This unreachability is not provable by incorrectness logic. This is provable by Hoare logic as $\{\underline{n} < 0 \wedge f = 1\}\,\mathtt{fact}\,\{\mathrm{false}\}$ but then we don't want to use two different logics to prove incorrectness, the main motivation for recent work on combining logics (e.g. [Bruni et al. 2023; Maksimovic et al. 2023; Milanese and Ranzato 2022; Zilberstein et al. 2023], etc). This is also provable by the natural transformational under approximation logic which extends incorrectness logic to nontermination, that is, in the assertional form of Sect. I.3.6, $\{\perp\} \subseteq \mathsf{Post}[\![\mathtt{fact}]\!]_{\perp}\{\underline{n} < 0 \wedge f = 1\}$, see example II.8.2. ∎

I.3.5 To Terminate or Not to Terminate Abstraction for Properties

Total correctness excludes nontermination while partial correctness allows it. This corresponds to different abstractions of the natural relational semantics.

I.3.5.1 The Termination Exclusion Abstraction. We can exclude the possibility of nontermination by the abstraction

$$\alpha^2_{\not\perp}(R) \quad\triangleq\quad \{\langle P,\ Q\rangle \mid \langle P,\ Q\rangle \in R \wedge Q \cap (\Sigma \times \{\perp\}) = \varnothing\} \tag{15}$$

excluding $\perp$ from the postcondition. This is an abstraction by the Galois connection

$$\langle\wp(\wp(\Sigma\times\Sigma)\times\wp(\Sigma\times\Sigma_\perp)),\ \subseteq\rangle \xleftarrow[\alpha^2_{\not\perp}]{\gamma^2_{\not\perp}} \langle\wp(\wp(\Sigma\times\Sigma)\times\wp(\Sigma\times\Sigma)),\ \subseteq\rangle \tag{16}$$

with $\gamma^2_{\not\perp}(R') \triangleq R' \cup \{\langle P,\ Q\rangle \mid Q\cap(\Sigma\times\{\perp\})\neq\varnothing\}$ Ⓐ.

Example I.3.6 (Manna and Pnueli total correctness logic). By eliminating the nontermination possibility from the postcondition of the natural transformational over approximation logic (13), we get Manna and Pnueli logic [Manna and Pnueli 1974] with theory Ⓐ

$$\alpha^2_{\not\perp}(\mathsf{post}(\supseteq,\subseteq)(\alpha_{\mathrm{G}}(\mathsf{Post}[\![\mathtt{S}]\!]_\perp))) \quad=\quad \{\langle P,\ Q\smallsetminus(\Sigma\times\{\perp\})\rangle \mid \mathsf{Post}[\![\mathtt{S}]\!]_\perp P\subseteq Q\smallsetminus(\Sigma\times\{\perp\})\} \tag{17}$$

that is, for any initial state $\langle\sigma_0,\ \sigma\rangle$ satisfying the precondition P, execution terminates in a final state σ' such that the pair $\langle\sigma_0,\ \sigma'\rangle$ satisfies the postcondition $Q\cap\Sigma\times\Sigma$). This is relational total correctness since nontermination is excluded. ■

Another abstraction to specify total correctness is to consider a transformer for a modified semantics $[\![\mathtt{S}]\!]\cup\{\langle\sigma_0,\ \sigma'\rangle \mid \langle\sigma,\ \perp\rangle\in[\![\mathtt{S}]\!]_\perp\wedge\sigma'\in\Sigma\}$ returning any possible result in case of nontermination [Plotkin 1979] using Smyth powerdomain [Smyth 1978] so that it is impossible to make any conclusion on final values in case of possible nontermination for an initial state. However, this is an impractical basis for static analysis since the abstraction introduces great imprecision.

I.3.5.2 The Termination Inclusion Abstraction. We can include the possibility of nontermination by the abstraction

$$\alpha^2_\perp(R) \quad\triangleq\quad \{\langle P,\ Q\cup(\Sigma\times\{\perp\})\rangle \mid \langle P,\ Q\rangle\in R\} \tag{18}$$

allowing the possibility of nontermination for all input states by adding $\perp$ to the postcondition. This is an abstraction by a Galois connection Ⓐ

$$\langle\wp(\wp(\Sigma\times\Sigma)\times\wp(\Sigma\times\Sigma_\perp)),\ \subseteq\rangle \xleftarrow[\alpha^2_\perp]{\gamma^2_\perp} \langle\wp(\wp(\Sigma\times\Sigma)\times\wp(\Sigma\times\Sigma_\perp)),\ \subseteq\rangle \tag{19}$$

with $\gamma^2_\perp(R') \triangleq \{\langle P,\ Q\rangle \mid \langle P,\ Q\cup(\Sigma\times\{\perp\})\rangle\in R'\}$.

Example I.3.7 (Manna relational partial correctness logic). Manna's relational partial correctness logic [Manna 1971] includes the nontermination possibility for all input states. Its theory is Ⓐ

$$\alpha^2_\perp(\mathsf{post}(\supseteq,\subseteq)(\alpha_{\mathrm{G}}(\mathsf{Post}[\![\mathtt{S}]\!]_\perp))) \quad=\quad \{\langle P,\ Q\cup(\Sigma\times\{\perp\})\rangle \mid \mathsf{Post}[\![\mathtt{S}]\!]_\perp P\subseteq Q\} \tag{20}$$

which is $\{\langle P,\ Q\rangle\in\wp(\Sigma\times\Sigma)\times\wp(\Sigma\times\Sigma) \mid \forall\langle\sigma_0,\ \sigma\rangle\in P\ .\ \forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle\in[\![\mathtt{S}]\!]\Rightarrow\langle\sigma_0,\ \sigma'\rangle\in Q\}$ when using the angelic semantics $[\![\mathtt{S}]\!]$ i.e. any terminating execution started within P satisfies Q. ■

So to prove partial correctness, we essentially add the possibility of nontermination to postconditions in $\wp(\Sigma\times\Sigma_\perp)$. However, for partial correctness, postconditions are traditionally chosen in $\wp(\Sigma\times\Sigma)$ not $\wp(\Sigma\times\Sigma_\perp)$. This equivalent alternative uses the Galois connection Ⓐ

$$\langle\wp(\wp(\Sigma\times\Sigma)\times\wp(\Sigma\times\Sigma_\perp)),\ \subseteq\rangle \xleftarrow[\alpha^{2\prime}_\perp]{\gamma^{2\prime}_\perp} \langle\wp(\wp(\Sigma\times\Sigma)\times\wp(\Sigma\times\Sigma)),\ \subseteq\rangle \tag{21}$$

with

$$\alpha^{2\prime}_\perp(R) \ \triangleq\ \{\langle P,\ Q\cap(\Sigma\times\Sigma)\rangle \mid \langle P,\ Q\rangle\in R\} \qquad \gamma^{2\prime}_\perp(R') \ \triangleq\ \{\langle P,\ Q\rangle \mid \langle P,\ Q\cap(\Sigma\times\Sigma)\rangle\in R'\}$$

Example I.3.8 (Manna relational partial correctness logic, continuing example I.3.7). In that case, the theory of Manna's logic is Ⓐ

$$\alpha^{2\prime}_\perp(\mathsf{post}(\supseteq,\subseteq)(\alpha_{\mathrm{G}}(\mathsf{Post}[\![\mathtt{S}]\!]_\perp))) \quad=\quad \{\langle P,\ Q\cap(\Sigma\times\Sigma)\rangle \mid \mathsf{Post}[\![\mathtt{S}]\!]_\perp P\subseteq Q\} \tag{22}$$ ■

I.3.6 Relational to Assertional Abstraction

Since they relate initial pairs $\langle\sigma_0,\ \sigma\rangle$ to final pairs $\langle\sigma_0,\ \sigma'\rangle$, $\sigma_0\in\mathcal{X}$, $\sigma\in\mathcal{Y}$, and $\sigma'\in\mathcal{Z}$, relational logics have their theory in a set $\wp(\wp(\mathcal{X}\times\mathcal{Y})\times\wp(\mathcal{X}\times\mathcal{Z}))$ while assertional logic theories are in

$\wp(\wp(\mathcal{Y}) \times \wp(\mathcal{Z}))$ where e.g. the postcondition is on final states and unrelated to the initial ones. This is an abstraction by projection on the second component Ⓐ

$$\langle\wp(\mathcal{X}\times\mathcal{Y}),\,\subseteq\rangle \underset{\alpha_{\downarrow^2}}{\overset{\gamma_{\downarrow^2}}{\leftrightarrows}} \langle\wp(\mathcal{Y}),\,\subseteq\rangle,\ \langle\wp(\wp(\mathcal{X}\times\mathcal{Y})\times\wp(\mathcal{X}\times\mathcal{Z})),\,\subseteq\rangle \underset{\dot{\alpha}_{\downarrow^2}}{\overset{\dot{\gamma}_{\downarrow^2}}{\leftrightarrows}} \langle\wp(\wp(\mathcal{Y})\times\wp(\mathcal{Z})),\,\subseteq\rangle \tag{23}$$

with

$$\begin{array}{rclrcl} \alpha_{\downarrow^2}(P) & \triangleq & \{\sigma \mid \exists\sigma_0 \,.\, \langle\sigma_0,\,\sigma\rangle \in P\} & \gamma_{\downarrow^2}(Q) & \triangleq & \mathcal{X}\times Q \\ \dot{\alpha}_{\downarrow^2}(R) & \triangleq & \{\langle\alpha_{\downarrow^2}(P),\,\alpha_{\downarrow^2}(Q)\rangle \mid \langle P,\,Q\rangle \in R\} & \dot{\gamma}_{\downarrow^2}(R') & \triangleq & \{\langle\gamma_{\downarrow^2}(P'),\,\gamma_{\downarrow^2}(Q')\rangle \mid \langle P',\,Q'\rangle \in P\} \end{array} \tag{24}$$

Example I.3.9. At this point we have got the theory of Hoare logic as the abstraction

$$\begin{array}{rcl} \alpha_{\mathsf{H}}(\{\!|\mathsf{S}|\!\}_{\perp}) & \triangleq & \underset{\text{assertional}}{\alpha_{\downarrow^2}} \circ \underset{\text{nontermination}}{\alpha^2_{\perp}} \circ \underset{\text{consequence}}{\mathsf{post}(\supseteq,\subseteq)} \circ \underset{\text{graph}}{\alpha_{\mathrm{G}}} \circ \underset{\text{transformer}}{\mathsf{Post}} \circ \underset{\text{relational semantics}}{\alpha_{\mathrm{C}}}(\underset{\text{collecting semantics}}{\{\!|\mathsf{S}|\!\}_{\perp}}) \\ & = & \{\langle P,\,Q\rangle \mid \forall\sigma\in P \,.\, \forall\sigma' \,.\, \langle\sigma,\,\sigma'\rangle \in [\![\mathsf{S}]\!]_{\perp} \Rightarrow \sigma' \in Q\cup\{\perp\}\} \end{array} \tag{25}$$

The set of valid Hoare triples $\{P\}\mathsf{S}\{Q\}$ is the set of pairs $\langle P, Q\rangle$ in $\alpha_{\mathsf{H}}([\![\mathsf{S}]\!]_{\perp})$ such that any execution started in a state σ of P, that terminates, if ever, does terminate in a state σ' of Q. ■

Example I.3.10. Similarly the assertional abstraction $\alpha_{\downarrow^2}$ of Manna and Pnueli logic (17) yields Apt and Plotkin generalization of Hoare logic to total correctness [Apt and Plotkin 1986, equation (6), page 749] (generalizing [Harel 1979] using naturals to unbounded nondeterminism using ordinals, equivalently a variant function in well-founded sets, as first considered by Turing [Turing 1950] and Floyd [Floyd 1967]). ■

Similarly, we can define an abstraction by projection on the first component

$$\alpha^{-1}(r) \ \triangleq\ r^{-1} \qquad \alpha_{\downarrow^1} \ \triangleq\ \alpha_{\downarrow^2}\circ\alpha^{-1} \qquad \gamma_{\downarrow^1} \ \triangleq\ \alpha^{-1}\circ\gamma_{\downarrow^2} \tag{26}$$

so that by composition of Galois connections and isomorphisms (proposition B.1) and by the forthcoming (27), we have Galois connection similar to (23) for $\langle\alpha_{\downarrow^1},\,\gamma_{\downarrow^1}\rangle$.

One may wonder why, for such a well-known result, we have considered so many successive abstractions (six when including the abstraction (5) of the collecting semantics into the relational semantics). There are three main reasons.

(1) The composition of Galois connections and isomorphisms is a Galois connection (Prop. B.1 in the appendix). Since abstractions preserves existing joins and concretizations preserve existing meets, we get “healthiness conditions” (such as [Hoare 1978, (H2), page 469]) as theorems, not hypotheses. In absence of a Galois connection, there would be no unique, most precise approximation, of the collecting semantics by a formula of the logic (e.g. [Gotsman et al. 2011]);
(2) By varying slightly the abstractions, we get a hierarchy of transformational logics (which extends the hierarchy of semantics in [Cousot 2002]), that we can compare without even knowing their proof systems. This is the objective for the rest of this part I on the theories of logics;
(3) Knowing the program semantics and its abstraction to the theory of a logic, we can constructively design, by calculus, a sound and complete proof system for this logic. This will be developed in part II.

I.3.7 The Forward Transformational Logics Hierarchy

We have built the theories of logics in Fig. 1 by composition of abstractions. The relational and assertional logics are considered equivalent in practice by using an auxiliary program with phantom variables recording the values of the initial or final variables (see Sect. E.1 in the appendix). By allowing the explicit use of nontermination $\perp$ in the postcondition, the over/under approximating antecedent/consequent logics subsume their approximations by $\alpha_\perp^2$ or $\alpha_{\not\perp}^2$ and $\alpha_{\downarrow^2}$ (including the logics marked by circled numbers that do not look to have been considered in the literature).

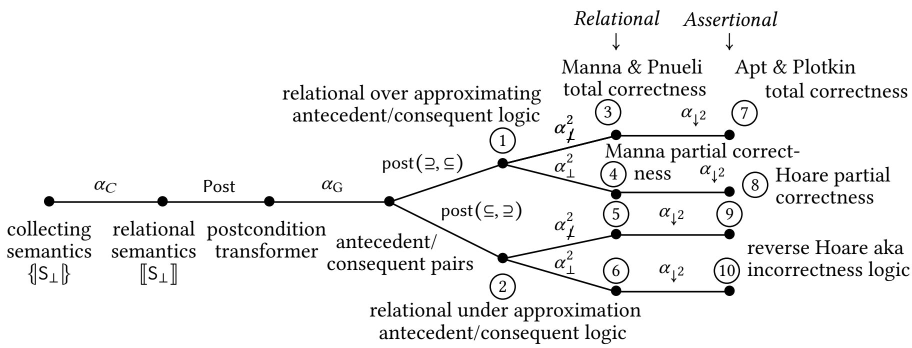


Fig. 1. Forward semantics and logics

I.3.8 Singularities of Logics

I.3.8.1 Emptiness Versus Universality. The same way that false is satisfied by no element of the universe in logic, some transformational logics have this emptiness property, meaning that some programs satisfy no formula of the logic. This is the case of a nonterminating program for Manna and Pnueli total correctness logic [Manna and Pnueli 1974]. Emptiness may look awkward since using the deductive system to prove any specification will always fail.

The same way that true is satisfied by all elements of the universe in logic, transformational logics may have the universality property, meaning that there exist programs for which any pair $\langle P, Q\rangle$ for that program is in the logic (i.e. is satisfied in logical terms). For example, in Hoare logic, $\{P\}$ `while (true) skip` $\{Q\}$ is satisfied for all P and Q. $[P]$ `S` $[$`false`$]$ is always true in incorrectness logic [O'Hearn 2020]. Universality may look awkward since using the deductive system to prove this obvious fact may be very complicated.

These phenomena have been criticized (e.g. emptiness for necessary preconditions [Cousot et al. 2013, 2011] in [O'Hearn 2020, section, page 10:28]) but are inherent to semantic approximation.

I.3.8.2 Correctness Versus Incorrectness. The use of a logic to prove correctness or incorrectness is not intrinsic but depending upon the application domain. For example, termination is required for most programs so that Manna and Pnueli logic is a correctness logic [Manna and Pnueli 1974]. However, operating systems should not terminate, and proving the contrary by Manna and Pnueli logic [Manna and Pnueli 1974] would make it an incorrectness logic. Another example is the incorrectness logic [O'Hearn 2020] which has the same theory as the reverse Hoare logic used by [de Vries and Koutavas 2011] to prove correctness. The qualification of under or over approximation instead of correctness or incorrectness logics looks more independent of specific applications, as suggested by [Maksimovic et al. 2023].

I.3.9 Backward Logics

Backward logics originates from the inversion abstraction (using the inverse program semantics $([\![\mathtt{S}]\!]_\perp)^{-1}$) or the dual complement abstraction (stating the impossibility of the negation of a property, which is called the *duality principle for programs* by Pratt [Pratt 1976, p. 110]) and the *conjugate* in [Dijkstra and Scholten 1990, equation (2) page 82]. They correspond to the commutative diagram of [Cousot and Cousot 1977, page 241], also found on [Cousot and Cousot 1982, page 98] (where inversion is $^{-1}$ and complement is ~), diagrams which are extended to Fig. 2.

I.3.9.1 The Inversion Abstraction. As noticed by [Pratt 1976, section 1.2], the inversion isomorphism transforms forward antecedent-consequent logics into backward consequent-antecedent logics. For that purpose, let us define the relation isomorphic abstraction α^{-1}, its pointwise extension $\dot{\alpha}^{-1}$, and the inverse transformer abstraction $\vec{\alpha}^{-1}$.

$$\alpha^{-1}(r) \triangleq r^{-1} \qquad \dot{\alpha}^{-1}(f) \triangleq \alpha^{-1} \circ f \circ \alpha^{-1} \qquad \vec{\alpha}^{-1}(T) \triangleq \dot{\alpha}^{-1} \circ T \circ \alpha^{-1} \tag{27}$$

so that we have the following Galois isomorphisms Ⓐ

$$\begin{gathered}
\langle \wp(\mathcal{X} \times \mathcal{Y}),\ \subseteq\rangle \underset{\alpha^{-1}}{\overset{\alpha^{-1}}{\leftrightarrows}} \langle \wp(\mathcal{Y} \times \mathcal{X}),\ \subseteq\rangle \\
\langle \wp(\mathcal{Z} \times \mathcal{X}) \to \wp(\mathcal{Z} \times \mathcal{Y}),\ \dot{\subseteq}\rangle \underset{\dot{\alpha}^{-1}}{\overset{\dot{\alpha}^{-1}}{\leftrightarrows}} \langle \wp(\mathcal{X} \times \mathcal{Z}) \to \wp(\mathcal{Y} \times \mathcal{Z}),\ \dot{\subseteq}\rangle \\
\langle \wp(\mathcal{X} \times \mathcal{Y}) \to \wp(\mathcal{Z} \times \mathcal{X}) \to \wp(\mathcal{Z} \times \mathcal{Y}),\ \ddot{\subseteq}\rangle \underset{\vec{\alpha}^{-1}}{\overset{\vec{\alpha}^{-1}}{\leftrightarrows}} \langle \wp(\mathcal{Y} \times \mathcal{X}) \to \wp(\mathcal{X} \times \mathcal{Z}) \to \wp(\mathcal{Y} \times \mathcal{Z}),\ \ddot{\subseteq}\rangle
\end{gathered} \tag{28}$$

Using these Galois isomorphisms (28), we define the precondition transformer Ⓐ

$$\mathsf{Pre} \quad \triangleq \quad \vec{\alpha}^{-1}(\mathsf{Post}) \quad = \quad \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \{\langle \sigma,\ \sigma_f\rangle \mid \exists \sigma' \,.\, \langle \sigma,\ \sigma'\rangle \in r \wedge \langle \sigma',\ \sigma_f\rangle \in Q)\} \tag{29}$$

so that $\mathsf{Pre}(r)Q$ is the set of initial states σ related to σ_f from which it is possible to reach a final state σ' related to σ_f satisfying the consequent Q through a transition by r.

I.3.9.2 The Complement Abstraction. The complement abstraction is useful to express that a program property does not hold (e.g. to contradict a Hoare triple).

Let $\mathcal{X}$ be a set and $X \in \wp(\mathcal{X})$. The complement abstraction is $\alpha^{\neg}(X) \triangleq \neg X$ (where $\neg X \triangleq \mathcal{X} \setminus X$ when $X \in \wp(\mathcal{X})$). We have the Galois isomorphisms

$$\langle \wp(\mathcal{X}),\ \subseteq\rangle \underset{\alpha^{\neg}}{\overset{\alpha^{\neg}}{\leftrightarrows}} \langle \wp(\mathcal{X}),\ \supseteq\rangle \qquad \text{and} \qquad \langle \wp(\mathcal{X}),\ \supseteq\rangle \underset{\alpha^{\neg}}{\overset{\alpha^{\neg}}{\leftrightarrows}} \langle \wp(\mathcal{X}),\ \subseteq\rangle \tag{30}$$

(which follow from $X \subseteq Y \Leftrightarrow \neg X \supseteq \neg Y$ and $\neg\neg X = X$ and implies De Morgan laws $\alpha^{\neg}(\bigcup X) = \bigcap \alpha^{\neg}(X)$ and $\alpha^{\neg}(\bigcap X) = \bigcup \alpha^{\neg}(X)$ since, in a Galois connection, α preserves existing joins and γ preserves existing meets).

I.3.9.3 The Emptiness and Non-Emptiness Abstraction. Negation is sometimes equivalent to an emptiness or non-emptiness check. For example, $\neg(A \subseteq B) \Leftrightarrow A \cap \neg B \neq \varnothing$. These are abstractions.

$$\begin{array}{ll}
\text{Emptiness} & \text{Non-emptiness} \\
\alpha^{\vec{\varnothing}}(\tau) \triangleq \{\langle P,\ Q\rangle \mid Q \cap \tau(P) = \varnothing\} & \alpha^{\vec{\otimes}}(\tau) \triangleq \alpha^{\neg} \circ \alpha^{\vec{\varnothing}}(\tau) = \{\langle P,\ Q\rangle \mid Q \cap \tau(P) \neq \varnothing\} \\
\alpha^{\overleftarrow{\varnothing}}(\tau) \triangleq \alpha^{-1}(\alpha^{\vec{\varnothing}}(\tau)) = \{\langle P,\ Q\rangle \mid P \cap \tau(Q) = \varnothing\} & \alpha^{\overleftarrow{\otimes}}(\tau) \triangleq \alpha^{-1}(\alpha^{\vec{\otimes}}(\tau)) = \{\langle P,\ Q\rangle \mid P \cap \tau(Q) \neq \varnothing\}
\end{array} \tag{31}$$

We have Ⓐ

$$\langle \wp(\mathcal{X}) \to \wp(\mathcal{Y}),\ \dot{\subseteq}\rangle \underset{\alpha^{\vec{\varnothing}}}{\overset{\gamma^{\vec{\varnothing}}}{\leftrightarrows}} \langle \wp(\mathcal{X} \times \mathcal{Y}),\ \supseteq\rangle \tag{32}$$

and similarly Galois connections for the other cases $\alpha^{\overleftarrow{\varnothing}}$, $\alpha^{\vec{\otimes}}$, and $\alpha^{\overleftarrow{\otimes}}$.

I.3.9.4 The Complement Dual Abstractions. Pratt's "Duality Principle for Programs" [Pratt 1976, section 1.2], is similar the complement duality in classical logic i.e. something not false is true.

This can be stated for functions f by defining the complement dual abstraction $\alpha^{\sim}$ of functions and its pointwise extension $\dot{\alpha}^{\sim}$ below, which yields the Galois connections as follows Ⓐ

$$\alpha^{\sim}(f) \quad \triangleq \quad \neg \circ f \circ \neg \qquad\qquad \dot{\alpha}^{\sim}(F) \quad \triangleq \quad \boldsymbol{\lambda} x \bullet \alpha^{\sim}(F(x)) \tag{33}$$

with connections Ⓐ

$$\begin{gathered}
\langle \wp(\mathcal{X}) \to \wp(\mathcal{Y}),\ \dot{\subseteq}\rangle \underset{\alpha^{\sim}}{\overset{\alpha^{\sim}}{\leftrightarrows}} \langle \wp(\mathcal{X}) \to \wp(\mathcal{Y}),\ \dot{\supseteq}\rangle \\
\langle \wp(\mathcal{Z}) \to \wp(\mathcal{X}) \to \wp(\mathcal{Y}),\ \ddot{\subseteq}\rangle \underset{\dot{\alpha}^{\sim}}{\overset{\dot{\alpha}^{\sim}}{\leftrightarrows}} \langle \wp(\mathcal{Z}) \to \wp(\mathcal{X}) \to \wp(\mathcal{Y}),\ \ddot{\supseteq}\rangle
\end{gathered}$$

where $\dot{\subseteq}$ is the pointwise extension of $\subseteq$, that is, $f \mathrel{\dot{\subseteq}} g \Leftrightarrow \forall X \in \mathcal{X} \,.\, f(X) \subseteq g(X)$, $\ddot{\subseteq}$ is the pointwise extension of $\dot{\subseteq}$, etc.

Using this Galois connection (33), we define the dual complement transformers Ⓐ

$$\begin{array}{rcccl}\widetilde{\mathsf{Post}} & \triangleq & \dot{\alpha}^{\sim}(\mathsf{Post}) & = & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \{\langle \sigma_0,\, \sigma' \rangle \mid \forall \sigma \,.\, \langle \sigma,\, \sigma' \rangle \in r \Rightarrow \langle \sigma_0,\, \sigma \rangle \in P\} \\ \widetilde{\mathsf{Pre}} & \triangleq & \dot{\alpha}^{\sim}(\mathsf{Pre}) & = & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \{\langle \sigma,\, \sigma_f \rangle \mid \forall \sigma' \,.\, \langle \sigma,\, \sigma' \rangle \in r \Rightarrow \langle \sigma',\, \sigma_f \rangle \in Q\}\end{array} \tag{34}$$

If $r \in \wp(\mathcal{X} \times \mathcal{Y})$ then Ⓐ

$$\langle \wp(\mathcal{Z} \times \mathcal{X}),\, \subseteq \rangle \xrightleftharpoons[\mathsf{Post}(r)]{\dot{\alpha}^{-1}(\widetilde{\mathsf{Pre}}(r))} \langle \wp(\mathcal{Z} \times \mathcal{Y}),\, \subseteq \rangle \quad \langle \wp(\mathcal{X} \times \mathcal{Z}),\, \subseteq \rangle \xrightleftharpoons[\mathsf{Pre}(r)]{\dot{\alpha}^{-1}(\widetilde{\mathsf{Post}}(r))} \langle \wp(\mathcal{Y} \times \mathcal{Z}),\, \subseteq \rangle \tag{35}$$

I.3.10 The Hierarchical Taxonomy of Forward and Backward Transformational Logics

The composition of abstractions applied to $\mathsf{Post}[\![\mathtt{S}]\!]_\perp$ of Fig. 1 can also be applied to $\widetilde{\mathsf{Post}}[\![\mathtt{S}]\!]_\perp$, $\mathsf{Pre}[\![\mathtt{S}]\!]_\perp$, and $\widetilde{\mathsf{Pre}}[\![\mathtt{S}]\!]_\perp$ to get Fig. 2. Fig. 1 can be recognized at the bottom right of Fig. 2. We get

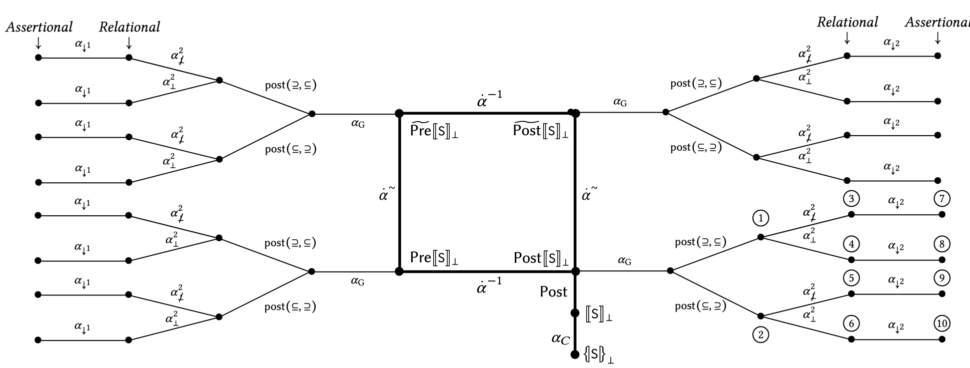


Fig. 2. Hierarchical taxonomy of transformational logics

40 transformational logics with 40 different proof systems which understanding is reduced to the composition of 9 abstractions (plus 2 to get $\mathsf{Post}[\![\mathtt{S}]\!]_\perp$ by abstraction of the collecting semantics). Adding the negation abstraction (30), we obtain 40 more logics to disprove program properties (see sections I.3.14.6 to I.3.14.6 and D.1 to D.6 for assertional logics), 160 logics with symbolic inversion in Sect. I.3.16, etc.

I.3.11 Abstraction for Assertional Logics

Theories of forward assertional logics in Fig. 1 are abstractions of theories of relational logics by $\alpha_{\downarrow^2}$ (and backward ones by $\alpha_{\downarrow^1}$). The more classic view [Pratt 1976] and recent followers a.o. [de Vries and Koutavas 2011; O'Hearn 2020; Zhang and Kaminski 2022] directly abstract the program semantics by the assertional transformer post (10) which is the abstraction of the relational transformer post (6), as follows

$$\begin{array}{llll}\alpha_2^A(\theta) \triangleq \alpha_{\downarrow^2} \circ \theta \circ \gamma_{\downarrow^2} & \gamma_2^A \triangleq \boldsymbol{\lambda}\theta \bullet \gamma_{\downarrow^2} \circ \theta \circ \alpha_{\downarrow^2}(Q) & \dot{\alpha}_2^A(\Theta) \triangleq \boldsymbol{\lambda} r \bullet \alpha_2^A(\Theta(r)) & \dot{\gamma}_2^A \triangleq \boldsymbol{\lambda} r \bullet \gamma_2^A(r) \\ \alpha_1^A(\theta) \triangleq \alpha_{\downarrow^1} \circ \theta \circ \gamma_{\downarrow^1} & \gamma_1^A \triangleq \boldsymbol{\lambda}\theta \bullet \gamma_{\downarrow^1} \circ \theta \circ \alpha_{\downarrow^1}(Q) & \dot{\alpha}_1^A(\Theta) \triangleq \boldsymbol{\lambda} r \bullet \alpha_1^A(\Theta(r)) & \dot{\gamma}_1^A \triangleq \boldsymbol{\lambda} r \bullet \gamma_1^A(r)\end{array} \tag{36}$$

we have Galois connections[1] Ⓐ

[1]We write $\xrightarrow{i}$, $\xrightarrow{c}$, $\xrightarrow{\sqcup}$, $\xrightarrow{\dot{\sqcup}}$, $\xrightarrow{\sqcap}$, and $\xrightarrow{\dot{\sqcap}}$ respectively for increasing, continuous, non-empty join, arbitrary join (including empty), non-empty meet, and arbitrary meet preserving functions.

$$\langle \wp(\mathcal{Z}\times\mathcal{X}) \xrightarrow{i} \wp(\mathcal{Z}\times\mathcal{Y}),\ \dot{\subseteq}\rangle \xleftrightarrows[\alpha_2^A]{\gamma_2^A} \langle \wp(\mathcal{X}) \xrightarrow{i} \wp(\mathcal{Y}),\ \dot{\subseteq}\rangle \tag{37}$$

$$\langle \wp(\mathcal{X}\times\mathcal{Y}) \to \wp(\mathcal{Z}\times\mathcal{X}) \xrightarrow{i} \wp(\mathcal{Z}\times\mathcal{Y}),\ \ddot{\subseteq}\rangle \xleftrightarrows[\dot{\alpha}_2^A]{\dot{\gamma}_2^A} \langle \wp(\mathcal{X}\times\mathcal{Y}) \to \wp(\mathcal{X}) \xrightarrow{i} \wp(\mathcal{Y}),\ \ddot{\subseteq}\rangle$$

These abstractions of the relational transformers yield the following generalization of the classic predicate transformers $\wp(\Sigma) \to \wp(\Sigma)$ [Pratt 1976], by extension to nontermination $\bot$. Ⓐ

$$\begin{array}{rclclcl}
\mathsf{post} & = & \dot{\alpha}_2^A(\mathsf{post}) & = & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \{\sigma' \mid \exists \sigma \,.\, \sigma \in P \wedge \langle\sigma,\,\sigma'\rangle \in r\}, & & \text{as in (10)} \\
\widetilde{\mathsf{post}} & \triangleq & \dot{\alpha}_2^A(\widetilde{\mathsf{Post}}) & = & \dot{\alpha}^{\sim} \circ \mathsf{post} & = & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \{\sigma' \mid \forall \sigma \,.\, \langle\sigma,\,\sigma'\rangle \in r \Rightarrow \sigma \in P\} \\
\mathsf{pre} & \triangleq & \dot{\alpha}_1^A(\mathsf{Pre}) & = & \dot{\alpha}^{-1} \circ \mathsf{post} & = & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \{\sigma \mid \exists \sigma' \,.\, \langle\sigma,\,\sigma'\rangle \in r \wedge \sigma' \in Q\} \\
\widetilde{\mathsf{pre}} & \triangleq & \dot{\alpha}_1^A(\widetilde{\mathsf{Pre}}) & = & \dot{\alpha}^{\sim} \circ \mathsf{pre} & = & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \{\sigma \mid \forall \sigma' \,.\, \langle\sigma,\,\sigma'\rangle \in r \Rightarrow \sigma' \in Q\}
\end{array} \tag{38}$$

The classic transformers (38) are illustrated by Fig. 4 in the appendix Ⓐ.

Given a relation $r \in \wp(\mathcal{X}\times\mathcal{Y})$, in addition to (12), these classic transformers are also connected as follows [Cousot 2021, Chapter 12], (d) is proved in sect. C of the appendix Ⓐ

$$\text{(a)}\ \langle \wp(\mathcal{X}\times\mathcal{Y}),\ \subseteq\rangle \xleftrightarrows[\mathsf{post}]{\mathsf{post}^{-1}} \langle \wp(\mathcal{X}) \xrightarrow{\cup} \wp(\mathcal{Y}),\ \dot{\subseteq}\rangle \quad \text{(b)}\quad \langle \wp(\mathcal{X}),\ \subseteq\rangle \xleftrightarrows[\mathsf{pre}(r)]{\widetilde{\mathsf{post}}(r)} \langle \wp(\mathcal{Y}),\ \subseteq\rangle \tag{39}$$

$$\text{(c)}\ \langle \wp(\mathcal{X}\times\mathcal{Y}),\ \subseteq\rangle \xleftrightarrows[\mathsf{pre}]{\mathsf{pre}^{-1}} \langle \wp(\mathcal{X}) \xrightarrow{\cup} \wp(\mathcal{Y}),\ \dot{\subseteq}\rangle \quad \text{(d)}\ \ \mathsf{post}(R)P \cap Q \neq \varnothing \Leftrightarrow P \cap \mathsf{pre}(R)Q \neq \varnothing$$

Example I.3.11. Hoare incorrectness logic is $\neg(\{P\}\,\mathsf{S}\{Q\}) \Leftrightarrow \neg(\mathsf{post}[\![\mathsf{S}]\!]P \subseteq Q) \Leftrightarrow \mathsf{post}[\![\mathsf{S}]\!]P \cap \neg Q \neq \varnothing \Leftrightarrow \exists \sigma \in P \,.\, \exists \sigma' \notin Q \,.\, \langle\sigma,\,\sigma'\rangle \in [\![\mathsf{S}]\!] \Leftrightarrow P \cap \mathsf{pre}[\![\mathsf{S}]\!]\neg Q \neq \varnothing$ by. This is different from incorrectness logic [O'Hearn 2020], that is $[P]\,\mathsf{S}\,[Q] \Leftrightarrow \neg Q \subseteq \mathsf{post}[\![\mathsf{S}]\!]P \Leftrightarrow \forall \sigma' \notin Q \,.\, \exists \sigma \in P \,.\, \langle\sigma,\,\sigma'\rangle \in [\![\mathsf{S}]\!]$. The incorrectness Hoare logic is designed in Sect. J.1 in the appendix. ■

All transformers in (35), (12), and (39) inherit the properties of Galois connections. For example, the lower adjoint preserves arbitrary joins and dually the upper adjoint preserves arbitrary meets. This implies, for example, the healthiness conditions postulated for transformers [Dijkstra and Scholten 1990; Hoare 1978].

Remark I.3.12. By (12), pre preserves joins ($\cup$) but maybe not meets ($\cap$). Same for post. Ⓐ ■

I.3.12 To Terminate or Not to Terminate Abstraction for Transformers

We have shown in Sect. I.3.5 that we can abstract antecedant-consequence pairs by (15) or (18) to take nontermination into account (e.g. total correctness) or not (partial correctness). An equivalent alternative uses the natural semantics $[\![\mathsf{S}]\!]_\bot$ or the angelic one $[\![\mathsf{S}]\!]$ in (1). We can also abstract transformers, which we do in the assertional case, by

$$\alpha_{\not\bot}(P) \triangleq P \setminus \{\bot\} \qquad \overrightarrow{\alpha}_{\not\bot}(\theta) \triangleq \alpha_{\not\bot} \circ \theta \qquad \overleftarrow{\alpha}_{\not\bot}(\theta) \triangleq \theta \circ \gamma_{\not\bot} \tag{40}$$

$$\gamma_{\not\bot}(Q) \triangleq Q \cup \{\bot\} \qquad \overrightarrow{\gamma}_{\not\bot}(\bar\theta) \triangleq \gamma_{\not\bot} \circ \bar\theta \qquad \overleftarrow{\gamma}_{\not\bot}(\bar\theta) \triangleq \bar\theta \circ \alpha_{\not\bot} \tag{41}$$

which yield Galois connections Ⓐ

$$\langle \wp(\Sigma_\bot),\ \subseteq\rangle \xleftrightarrows[\alpha_{\not\bot}]{\gamma_{\not\bot}} \langle \wp(\Sigma),\ \subseteq\rangle \qquad \langle \mathcal{X} \to \wp(\Sigma_\bot),\ \dot{\subseteq}\rangle \xleftrightarrows[\overrightarrow{\alpha}_{\not\bot}]{\overrightarrow{\gamma}_{\not\bot}} \langle \mathcal{X} \to \wp(\Sigma),\ \dot{\subseteq}\rangle \tag{42}$$

$$\langle \wp(\Sigma_\bot) \xrightarrow{i} \wp(\Sigma),\ \dot{\subseteq}\rangle \xleftrightarrows[\overleftarrow{\alpha}_{\not\bot}]{\overleftarrow{\gamma}_{\not\bot}} \langle \wp(\Sigma) \xrightarrow{i} \wp(\Sigma),\ \dot{\subseteq}\rangle$$

I.3.13 Abstract Logics

Finally logics may refer to any abstraction of the antecedents and consequents of a transformational logics. For example, [Cousot et al. 2012] is an abstraction of Hoare logic such that $\{\bar P\}\,\mathsf{S}\,\{\bar Q\}$

means Hoare triple $\{\gamma_1(\bar{P})\}\,\mathsf{S}\,\{\gamma_2(\bar{Q})\}$. Without appropriate hypotheses on the abstraction, some rules of Hoare logic like disjunction and conjunction may be invalid in the abstract, see counter-examples and sufficient hypotheses in [Cousot et al. 2012, pages 219–221]. Similarly, [Gotsman et al. 2011] provides a counterexample showing the unsoundness of the conjunction rule. This is an argument for the use of a principled method for designing logics.

Another abstract logic [Bruni et al. 2023] combines an over approximation (for correctness) and an under approximation (for incorrectness) in the same abstract domain. The "(relax)" rule requires that the under approximation uses abstract properties $\alpha(P)$ that exactly represent concrete properties P by requiring that $\gamma \circ \alpha(P) = P$. This restricts the concrete points that can be used in the under approximation, and will be a source of incompleteness and imprecision for most static analyses.

Under approximation is the order semidual of an over approximation, with abstraction $\langle\wp(\Sigma_\perp), \subseteq\rangle \xleftrightarrow[\alpha]{\gamma} \langle\mathcal{A}, \sqsupseteq\rangle$ exploited e.g. in [Ball et al. 2005]. The study by [Ascari et al. 2022] provides a number of classic abstract domain examples showing the imprecision of such under approximation static analyses, but for few exceptions like [Asadi et al. 2021; Miné 2014].

These under approximation approaches are based on Th. II.3.6 for fixpoint under approximation by transfinite iterates. Termination proofs do not use an under approximation but instead an over approximation and a variant function as, e.g., in Th. II.3.8. Alternatively, over approximating static analysis is classic and variant functions can also be inferred by abstract interpretation [D'Silva and Urban 2015; Urban 2013, 2015; Urban et al. 2016; Urban and Miné 2014a,b, 2015].

I.3.14 The Subhierarchy of Assertional Logics

Comparing logics means comparing their theories, that is their expressivity, through their respective abstractions of the collecting semantics (as formalized by fixpoint abstraction in Sect. II.2), and comparing the induction principles induced by their abstractions (as formalized in Sect. II.3 by fixpoint induction). For example, figure 3 shows that Hoare logic and subgoal induction are different but equivalent abstractions of the collecting semantics so have the same theory and equivalent but different proof systems.

These abstractions yield the hierarchical taxonomy of assertional transformational logics of Fig. 3, which is a subset of Fig. 2. Fig. 3, with a larger instance in the appendix Ⓐ, is commented thereafter.

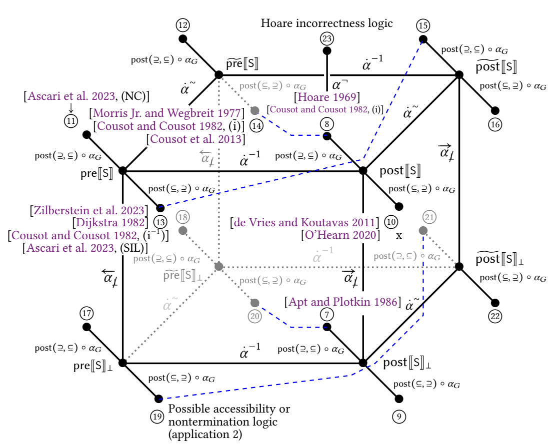


Fig. 3. Taxonomy of assertional logics

We use *universal* to mean for all initial or final states and *existential* to mean there exists at least one initial or final state. We use *reachability* (often forward) for initial to final states and *accessibility* (often backward) for final to initial states. We use *definite* to mean "for all executions" and *possible* to mean "for some execution" (maybe none). In both cases, the qualification does not exclude possible nontermination or blocking states, which is emphasized by *partial*. We use *total* to mean that all executions must be finite. We use *blocking* to mean a state, which is not final, but from which execution cannot go on. No such blocking states exist in the semantics $[\![\mathsf{S}]\!]_\perp$ of statements S in Sect. I.1.1 and II.1 but would correspond e.g. to an aborted execution after a runtime error (like a division by zero).

The taxonomy for **direct proofs** (the hypothesis implies the conclusion) is illustrated in Fig. 3.

I.3.14.1 Partial Definite Accessibility of Some Final State From All Initial States $\mathsf{post}[\![\mathsf{S}]\!]P \subseteq Q \Leftrightarrow P \subseteq \widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]Q$, $P, Q \in \wp(\Sigma)$. Partial correctness, allowing blocking states, characterizes executions starting from any initial state in P, which, if terminating normally, do terminate in a state of Q and no other one. So blocking states are not excluded. This is Naur [Naur 1966], Hoare [Hoare 1969] partial correctness, and Dijkstra weakest liberal preconditions [Dijkstra 1976] and the partial correctness part of Turing [Turing 1950] and Floyd [Floyd 1967] total correctness.

(8) $\mathsf{post}(\supseteq, \subseteq) \circ \alpha_G(\mathsf{post}[\![\mathsf{S}]\!]) \triangleq \{\langle P,\ Q\rangle \in \wp(\Sigma) \times \wp(\Sigma) \mid \mathsf{post}[\![\mathsf{S}]\!]P \subseteq Q\}$ yields the theory of Hoare logic [Hoare 1969]. This claim can be substantiated by (re)constructing Hoare logic by abstracting the angelic relational semantics $[\![\mathsf{S}]\!] = \alpha^{\perp}([\![\mathsf{S}]\!]_{\perp})$ by $\mathsf{post}(\supseteq, \subseteq) \circ \alpha_G \circ \mathsf{post}$. This has been done, e.g., in [Cook 1978, Theorem 1, page 79] (modulo a later correction in [Cook 1981]) as well as in [Cousot 2021, Chapter 26], although using the intermediate abstractions into an equational semantics and then verification conditions to explain Turing-Floyd's transition based invariance proof method.

(14) By Galois connection (12), $\mathsf{post}(\subseteq, \supseteq) \circ \alpha_G(\widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]) \triangleq \{\langle P,\ Q\rangle \in \wp(\Sigma) \times \wp(\Sigma) \mid P \subseteq \widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]Q\}$ is equivalent and yields the theory of a logic axiomatizing subgoal induction [Morris Jr. and Wegbreit 1977] or necessary preconditions [Cousot et al. 2013, 2011].

Hoare and subgoal induction logics can be used to prove universal partial correctness (Q is good, as in static accessibility analysis [Cousot and Cousot 1977]) and universal partial incorrectness (Q is bad, as in necessary preconditions analyses [Cousot et al. 2013, 2011]). Both logics can be also used to prove bounded termination, by introducing a counter incremented in loops and proved to be bounded [Luckham and Suzuki 1977]. However, this is incomplete for unbounded nondeterminism. $\mathsf{post}[\![\mathsf{S}]\!]P \subseteq \varnothing \Leftrightarrow P \subseteq \widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]\varnothing \Leftrightarrow P \subseteq \neg\mathsf{pre}[\![\mathsf{S}]\!]\Sigma \Leftrightarrow \mathsf{pre}[\![\mathsf{S}]\!]\Sigma \subseteq \neg P$ is *definite nontermination from all initial states* (executions from any initial state of P do not terminate).

Subgoal induction is exploited in necessary preconditions analyses [Cousot et al. 2013, 2011]. Finding P such that $\mathsf{post}[\![\mathsf{S}]\!]P \subseteq Q$ is equivalent to finding P such that $P \subseteq \widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]Q$ for the given error postcondition Q, which the necessary precondition analysis does by under approximating $\widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]$ defined structurally on the programming language and using fixpoint under approximation to handle iteration and recursion.

I.3.14.2 Total Definite Accessibility of Some Final States From All Initial States $\mathsf{post}[\![\mathsf{S}]\!]_{\perp}P \subseteq Q \Leftrightarrow P \subseteq \widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]_{\perp}Q$, $P, Q \in \wp(\Sigma)$. Total correctness, allowing blocking states, characterizes executions from any initial state in P that do terminate normally in a final state satisfying Q or block. Taking $Q = \Sigma$ is universal definite termination.

(7) The Turing [Turing 1950] & Floyd [Floyd 1967] proof method uses an invariant and a variant function into a well-founded set. The abstraction $\mathsf{post}(\supseteq, \subseteq) \circ \alpha_G(\mathsf{post}[\![\mathsf{S}]\!]_{\perp}) \triangleq \{\langle P,\ Q\rangle \in \wp(\Sigma) \times \wp(\Sigma) \mid \mathsf{post}[\![\mathsf{S}]\!]_{\perp}P \subseteq Q\}$ yields the theory of Apt and Plotkin [Apt and Plotkin 1986] logic in the assertional case (and that of Manna & Pnueli logic [Manna and Pnueli 1974] in the relational case). This claims follows from [Apt and Plotkin 1986] for an imperative language and [Cousot 2002] for arbitrary transition systems. The logic can be used to prove definite correctness or incorrectness.

I.3.14.3 Partial Possible Accessibility of All Final States From Some Initial State $Q \subseteq \mathsf{post}[\![\mathsf{S}]\!]_{\perp}P \Leftrightarrow Q \subseteq \mathsf{post}[\![\mathsf{S}]\!]P$, $P, Q \in \wp(\Sigma)$. This means that for any final state σ' in Q there exists at least one initial state σ in P and an execution from σ that will terminate in state σ'. Blocking states σ may be included in P. Moreover, this does not preclude executions from σ to make nondeterministic choices terminating normally with $\neg Q$ or do not terminate at all.

(10) By [de Vries and Koutavas 2011, Definition 1], $\mathsf{post}(\subseteq, \supseteq) \circ \alpha_G(\mathsf{post}[\![\mathsf{S}]\!])$ is the theory of De Vries and Koutavas reversed Hoare logic. This is also confirmed by the soundness and completeness

proofs in [de Vries and Koutavas 2011, section 6] based on a “weakest postcondition calculus” defined in [de Vries and Koutavas 2011, section 5] as “wpo(P, c), [is] the weakest postcondition given a precondition P and program c”. So “wpo” is nothing other than post and “$\langle P\rangle$ c $\langle Q\rangle$ is a valid triple if and only if $Q \Rightarrow$ wpo(P,c)”.

By [O’Hearn 2020, Fact 13], this is also incorrectness logic requiring any bug in Q to be possibly reachable in finitely many steps from P thus discarding infinite executions as possible errors.

The difference is in the examples handled where Q is “good” for De Vries and Koutavas and “bad” for O’Hearn.

I.3.14.4 Partial Possible Accessibility of Some Final State From All Initial States $P \subseteq \mathrm{pre}[\![\mathtt{S}]\!]Q$, $P, Q \in \wp(\Sigma)$. This prescribes that all initial states in P have at least one execution that does reach Q.

⑭ Dijkstra [Dijkstra 1982] shown the equivalence of $\mathrm{post}[\![\mathtt{S}]\!]P \subseteq Q$ (that is, Turing-Floyd-Naur-Hoare partial correctness and $P \subseteq \mathrm{pre}[\![\mathtt{S}]\!]Q$ (that is, Morris and Wegbreit subgoal induction, claiming “subgoal induction is indeed the next variation on an old theme”). By (12) this should have been $P \subseteq \widetilde{\mathrm{pre}}[\![\mathtt{S}]\!]Q$ in general, but Dijkstra considers total deterministic programs for which pre = $\widetilde{\mathrm{pre}}$. This is also the incorrectness part of the outcome logic [Zilberstein et al. 2023], the induction principle (i^{-1}) of [Cousot and Cousot 1982, p. 100], and (SIL) in [Ascari et al. 2023].

I.3.14.5 Possible Accessibility of Some Final State or Nontermination From All Initial States $P \subseteq \mathrm{pre}[\![\mathtt{S}]\!]_\perp Q$, $P \in \wp(\Sigma), Q \in \wp(\Sigma_\perp)$. For $Q = \Sigma$, this is *possible termination from all initial states* ⑰. For $Q = \{\perp\}$, this is *possible nontermination from all initial states.* Similarly, ⑪ is $P \subseteq \mathrm{pre}[\![\mathtt{S}]\!]Q$, named (NC) in [Ascari et al. 2023].

⑲ This logic will be formally developed by calculus in Sect. II.8.2.

We can also consider **disproofs of program properties** by the abstraction $\alpha^{\neg}$ (30) of the theory of a program logic.

I.3.14.6 Partial Possible Accessibility of Some Final States (or Nontermination) From Some Initial States $\mathrm{post}[\![\mathtt{S}]\!]P \cap Q \neq \varnothing$ for $P, Q \in \wp(\Sigma)$ (or $\mathrm{post}[\![\mathtt{S}]\!]_\perp P \cap Q \neq \varnothing$ for $Q \in \wp(\Sigma_\perp)$). This means that at least one execution from at least one initial state in P does terminate in a final state satisfying Q. Taking $Q = \Sigma$ is *possible termination from some initial states.*

㉓ Disproving a Hoare triple using the proof system would require to show that no proof does exist for this triple, a method no one ever consider. One can use incorrectness logic [O’Hearn 2020] or provide a counter-example (not supported by a logic). The *Hoare incorrectness logic* ㉓ can be used to prove that a Hoare specification is violated with a possible counter-example, since $\neg\big(\{P\}\mathtt{S}\{Q\}\big) = \neg\big(\mathrm{post}[\![S]\!]P \subseteq Q\big) = \mathrm{post}[\![S]\!]P \cap \neg Q \neq \varnothing$. It’s nothing but debugging in logic form.

This is weaker that the requirements of incorrectness logic, for which the principle of denial [O’Hearn 2020, Fig. 1] states that if $Q \subseteq \mathrm{post}[\![\mathtt{S}]\!]P \wedge \neg(Q \subseteq Q')$ then $Q \cap \neg Q' \neq \varnothing$ and therefore $\mathrm{post}[\![\mathtt{S}]\!]P \cap \neg Q' \neq \varnothing$ that is, $\neg\big(\{P\}\mathtt{S}\{Q'\}\big)$. However the converse is not true since the violation of $\{P\}\mathtt{S}\{Q\}$ only require one state of P definitely reaches one state not satisfying Q.

Other contrapositive logics or logics for disproving program properties are considered in the appendix Ⓐ.

I.3.15 The Combination of Logics

Program logics are generally composite that is, the result of combining elementary logics which are different abstractions of program executions e.g. [Bruni et al. 2023; Zilberstein et al. 2023].

I.3.15.1 The Conjunction/Disjunction of Logics. We have $\mathrm{wlp}(\mathtt{S}, Q) = \mathrm{pre}[\![\mathtt{S}]\!]Q \cap \widetilde{\mathrm{pre}}[\![\mathtt{S}]\!]Q$ while $\mathrm{wp}(\mathtt{S}, Q) = \mathrm{pre}[\![\mathtt{S}]\!]_\perp Q \cap \widetilde{\mathrm{pre}}[\![\mathtt{S}]\!]_\perp Q$ since blocking states must be prevented as well as nontermination for wp, see Fig. 4 in the appendix. The relevant abstractions of transformers τ_1, τ_2 are Ⓐ

$$\alpha^{\cap}\langle\tau_1,\ \tau_2\rangle r \triangleq \tau_1(r) \mathbin{\dot{\cap}} \tau_2(r) \quad \text{meet (or conjunction) where } \delta(x) = \langle x,\ x\rangle \text{ is duplication} \tag{43}$$

$$\langle(\wp(\mathcal{X}\times\mathcal{Y}) \to (\wp(\mathcal{X}) \to \wp(\mathcal{Y})))^2,\ \ddot{\supseteq}\rangle \xleftarrow[\alpha^{\cap}]{\delta} \langle\wp(\mathcal{X}\times\mathcal{Y}) \to (\wp(\mathcal{X}) \to \wp(\mathcal{Y})),\ \ddot{\supseteq}\rangle$$

$$\alpha^{\cup}\langle\tau_1,\ \tau_2\rangle r \triangleq \tau_1(r) \mathbin{\dot{\cup}} \tau_2(r) \quad \text{join (or disjunction)}$$

$$\langle(\wp(\mathcal{X}\times\mathcal{Y}) \to (\wp(\mathcal{X}) \to \wp(\mathcal{Y})))^2,\ \ddot{\subseteq}\rangle \xleftarrow[\alpha^{\cup}]{\delta} \langle\wp(\mathcal{X}\times\mathcal{Y}) \to (\wp(\mathcal{X}) \to \wp(\mathcal{Y})),\ \ddot{\subseteq}\rangle$$

I.3.15.2 The Product of Logics. One can imagine a Cartesian product $\{DT, PT, NT\}\mathsf{S}\{Q, R\}$ meaning that every execution of S starting with an initial state of DT will definitely terminate in a final state in Q, every execution of S starting with an initial state of PT will either terminate in a final state in R or not terminate, and every execution of S starting with an initial state of NT will never terminate. Q and R could further be decomposed into a product of good and bad states.

Similarly, [O'Hearn 2020, section 4] uses the notation $[p]C[ok : q][er : r]$ as a shorthand for $[p]C[ok : q]$ and $[p]C[er : r]$ resulting in a single deductive system instead of two independent ones. The definition of the relational semantics in (54) will use such a grouping to set apart `breaks`.

The relevant Cartesian abstraction $\alpha^{\times}$ merges two transformers into a single one. We assume that $\langle\mathcal{X} \to \mathcal{Y}_1,\ \sqsubseteq_1\rangle$ and $\langle\mathcal{X} \to \mathcal{Y}_2,\ \sqsubseteq_2\rangle$ are posets, $\tau_1 \in \mathcal{X} \to \mathcal{Y}_1$ and $\tau_2 \in \mathcal{X} \to \mathcal{Y}_2$. Ⓐ

$$\alpha^{\times}\langle\tau_1,\ \tau_2\rangle(P) \triangleq \langle\tau_1(P),\ \tau_2(P)\rangle \qquad \text{Cartesian product} \tag{44}$$

$$\gamma^{\times}(\bar{\tau}) \triangleq \langle\boldsymbol{\lambda} P \bullet \mathsf{let}\ \langle P_1,\ P_2\rangle = \bar{\tau}(P)\ \mathsf{in}\ P_1,\ \boldsymbol{\lambda} P \bullet \mathsf{let}\ \langle P_1,\ P_2\rangle = \bar{\tau}(P)\ \mathsf{in}\ P_2\rangle$$

with Galois connection $$\langle\mathcal{X} \to \mathcal{Y}_1 \times \mathcal{X} \to \mathcal{Y}_2,\ \dot{\sqsubseteq}_1.\dot{\sqsubseteq}_2\rangle \xleftarrow[\alpha^{\times}]{\gamma^{\times}} \langle\mathcal{X} \to (\mathcal{Y}_1 \times \mathcal{Y}_2),\ (\sqsubseteq_1 \overset{\bullet}{,} \sqsubseteq_2)\rangle$$

Example I.3.13. We mentioned the origin [Park 1979] of relational semantics that Park encodes by $\alpha^{\times}\langle\alpha^{\not\perp},\ \boldsymbol{\lambda} S \bullet \alpha^{\neg}((\dot{\alpha}^{-1}(\mathsf{post}))(S)\{\perp\})\rangle[\![\mathsf{S}]\!]_{\perp}$ i.e. the input-output relation $[\![\mathsf{S}]\!]$ computed by S and the definite termination domain of S which is the complement of possible nontermination. ■

Example I.3.14. Dijkstra's weakest precondition $\mathsf{wp}(\mathsf{S}, Q)$ [Dijkstra 1976] is $\boldsymbol{\lambda} Q \bullet \alpha^{\cap}(\mathsf{pre}[\![\mathsf{S}]\!]_{\perp},\ \widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]_{\perp})$, with $Q \in \wp(\Sigma)$, $\mathsf{pre} = \dot{\alpha}^{-1}(\mathsf{post})$, and $\widetilde{\mathsf{pre}} = \dot{\alpha}^{\sim}(\mathsf{pre})$. The weakest liberal condition $\mathsf{wlp}(\mathsf{S}, Q)$ is $\boldsymbol{\lambda} Q \bullet \alpha^{\cap}((\widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]_{\perp}) \circ \alpha^{\perp}, (\widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]_{\perp}) \circ \alpha^{\perp}) = \alpha^{\cap}(\mathsf{pre}[\![\mathsf{S}]\!], \widetilde{\mathsf{pre}}[\![\mathsf{S}]\!])$. ■

I.3.15.3 The Reduced Product of Logics. The components are usually not independent. For example one uses invariants of Hoare logic to prove termination, or definite termination implies possible termination. Another example is adversarial logic [Vanegue 2022] to describe the possible interaction between a program and an attacker. These are reductions (45) that have been studied in the context of program analysis [Cousot 2021, chapter 29] but also apply to any abstraction, including logics, e.g. [Bruni et al. 2023].

The functor $\alpha^{\circledast}$, inspired by the reduced product in abstract interpretation [Cousot and Cousot 1979b, section 10.1], is the Cartesian product where the information of one component is propagated, in abstract form, to the other. This is useful for combining program logics dealing with properties that are not independent.

Assume two abstractions of a (collecting) semantics in $\langle\mathcal{S},\ \sqsubseteq\rangle$ into different transformers $\langle\mathcal{S},\ \sqsubseteq\rangle \xleftarrow[\alpha_1]{\gamma_1} \langle\mathcal{X} \to \mathcal{Y}_1,\ \dot{\sqsubseteq}_1\rangle$ and $\langle\mathcal{S},\ \sqsubseteq\rangle \xleftarrow[\alpha_2]{\gamma_2} \langle\mathcal{X} \to \mathcal{Y}_1,\ \dot{\sqsubseteq}_2\rangle$. Assume that $\langle\mathcal{X} \to (\mathcal{Y}_1 \times \mathcal{Y}_2),\ (\sqsubseteq_1 \overset{\bullet}{,} \sqsubseteq_2),\ \sqcap\rangle$ is a complete lattice.

The reduced product combines two abstractions of the semantics S into transformers τ_1 and τ_2 into an abstraction of the semantics S into a single transformer with $\alpha^{\circledast,\sqcap} \triangleq \rho \circ \alpha^{\times}$ where the reduction operator is $\rho(\bar{\tau}) \triangleq \bigsqcap\{\bar{\tau}' \mid \mathsf{let}\ \langle\tau_1,\ \tau_2\rangle = \gamma^{\times}(\bar{\tau})\ \mathsf{and}\ \langle\tau_1',\ \tau_2'\rangle = \gamma^{\times}(\bar{\tau}')\ \mathsf{in}\ \gamma_1(\tau_1) \sqcap \gamma_2(\tau_2) \sqsubseteq \gamma_1(\tau_1') \wedge \gamma_1(\tau_1) \sqcap \gamma_2(\tau_2) \sqsubseteq \gamma_2(\tau_2')\}$. By [Cousot 2021, Theorem 36.24], we have the Galois connection

$$\langle\mathcal{X} \to \mathcal{Y}_1 \times \mathcal{X} \to \mathcal{Y}_2,\ \dot{\sqsubseteq}_1\ .\ \dot{\sqsubseteq}_2\rangle \xleftarrow[\rho \circ \alpha^{\times}]{\gamma^{\times}} \langle\mathcal{X} \to (\mathcal{Y}_1 \times \mathcal{Y}_2),\ (\sqsubseteq_1 \overset{\bullet}{,} \sqsubseteq_2)\rangle \tag{45}$$

Example I.3.15. Continuing example I.3.1, the reduced product of Hoare logic [Hoare 1978] (abstracting post) and subgoal induction logic [Morris Jr. and Wegbreit 1977] (in Dijkstra's version [Dijkstra 1982] abstracting pre) for the factorial with consequent specification $f = !\underline{n}$ is $\{n = \underline{n} \geqslant 0 \wedge f = 1\}$ $\texttt{fact}$ $\{\underline{n} \geqslant 0 \wedge f = !\underline{n}\}$. ■

I.3.16 Symbolic Inversion

Let us consider one more useful abstraction of transformers allowing for their inversion using symbolic execution. This reversal abstraction $\alpha^{\leftrightarrow}$ from [Cousot 1981, Theorem 10-13] allows to prove backward properties using a forward proof system by using auxiliary variables for initial values of variables (as in symbolic execution) and conversely (as an inverse symbolic execution starting with symbolic final values of variables). Given $\mathcal{D} \triangleq \wp(\mathcal{X} \times \mathcal{Y}) \to (\wp(\mathcal{X}) \to \wp(\mathcal{Y}))$, $\mathcal{D}' \triangleq \wp(\mathcal{X} \times \mathcal{Y}) \to (\wp(\mathcal{Y}) \to \wp(\mathcal{X}))$, and $\tau = \text{post}$, we have Ⓐ (and similarly for $\tau \in \{\text{pre}, \text{post}, \text{Pre}\}$)

$$\alpha^{\leftrightarrow}(\tau)(r)P \triangleq \{\sigma' \mid \exists \sigma \in P \,.\, \sigma \in \tau(r^{-1})\{\overline{\varsigma} \mid \overline{\varsigma} = \sigma'\}\} \qquad \langle \mathcal{D},\ \dddot{\subseteq} \rangle \underset{\alpha^{\leftrightarrow}}{\overset{\alpha^{\leftrightarrow}}{\leftrightarrows}} \langle \mathcal{D}',\ \dddot{\subseteq} \rangle \tag{46}$$

Example I.3.16. Consider the straight-line program `x = x+y; y = 2*x+y`. A forward symbolic execution $\text{post}[\![\texttt{S}]\!]\{\sigma \mid \sigma_{\mathsf{x}} = \underline{\sigma}_{\mathsf{x}} \wedge \sigma_{\mathsf{y}} = \underline{\sigma}_{\mathsf{y}}\}$ with Hoare logic for initial auxiliary variables $\underline{x}$, $\underline{y}$ is

$$\{x = \underline{x} \wedge y = \underline{y}\}\ \texttt{x = x + y;}\ \{x = \underline{x} + \underline{y} \wedge y = \underline{y}\}\ \texttt{y = 2*x+y;}\ \{x = \underline{x} + \underline{y} \wedge y = 2\underline{x} + 3\underline{y}\}$$

This information can be inferred automatically by forward static analyses using affine equalities [Karr 1976] or inequalities [Cousot and Halbwachs 1978]. This can be used to get a precondition $\text{pre}[\![\texttt{S}]\!]Q$ ensuring that a postcondition Q holds be defining $\text{pre}[\![\texttt{S}]\!]Q = \{\underline{\sigma} \mid \exists \overline{\sigma} \in Q \,.\, \overline{\sigma} \in \text{post}[\![\texttt{S}]\!]\{\sigma \mid \sigma_{\mathsf{x}} = \underline{\sigma}_{\mathsf{x}} \wedge \sigma_{\mathsf{y}} = \underline{\sigma}_{\mathsf{y}}\}\}$ which, in our example, is $\{\langle x,\ y\rangle \mid \exists \langle \overline{x},\ \overline{y}\rangle \in Q \wedge \overline{x} = x + y \wedge \overline{y} = 2x + 3y\}$ so that, e.g., for $Q = \{\langle \overline{x},\ \overline{y}\rangle \mid \overline{x} = \overline{y}\}$ stating that $x = y$ on exit, we get the precondition $\{\langle z,\ -z/3\rangle \mid z \in \mathbb{Z}\}$.

Inversely, using subgoal induction, a backward execution $\text{pre}[\![\texttt{S}]\!]\{\sigma \mid \sigma_{\mathsf{x}} = \overline{\sigma}_{\mathsf{x}} \wedge \sigma_{\mathsf{y}} = \overline{\sigma}_{\mathsf{y}}\}$ is

$$\{x = 3\overline{x} - \overline{y} \wedge y = \overline{y} - 2\overline{x}\}\ \ x = x + y;\ \ \{x = \overline{x} \wedge y = \underline{y} - 2\overline{x}\}\ \ y = 2 * x + y;\ \ \{x = \overline{x} \wedge y = \overline{y}\}$$

This information can be used to get a postcondition $\text{post}[\![\texttt{S}]\!]P$ hence holding for states reachable from the precondition P as $\{\sigma \mid \exists \underline{\sigma} \in P \,.\, \underline{\sigma} \in \text{pre}[\![\texttt{S}]\!]\{\sigma \mid \sigma_{\mathsf{x}} = \overline{\sigma}_{\mathsf{x}} \wedge \sigma_{\mathsf{y}} = \overline{\sigma}_{\mathsf{y}}\}\}$. For our example, we get $\{\langle x,\ y\rangle \mid \exists \langle \underline{x},\ \underline{y}\rangle \in P \,.\, \underline{x} = 3x - y \wedge \underline{y} = y - 2x\}$ which, e.g., for $P = \{\langle x,\ y\rangle \mid x = y\}$, yields $\{\langle x,\ y\rangle \mid 5x = 2y\}$. This calculation is mechanizable using the operations of the abstract domains for affine equalities [Karr 1976] or inequalities [Cousot and Halbwachs 1978]. ■

Part II: Design of the Proof Rules of Logics by Abstraction of Their Theory

Given the theory $\alpha([\![\texttt{S}]\!]_{\perp})$ of a logic defined by an abstraction α of the natural relational semantics $[\![\texttt{S}]\!]_{\perp}$, we now consider the problem of designing the proof/deductive system for that logic. The abstraction α can be decomposed into $\alpha_a \circ \alpha_t$ where α_t abstracts the natural relational semantics $[\![\texttt{S}]\!]_{\perp})$ into an exact transformer (isomorphically its antecedant-consequent graph) which is then over or under approximated by α_a.

We first express the natural relational semantics in structural fixpoint form in Sect. II.1. Then we use fixpoint abstraction of Sect. II.2 and structural induction to express the exact transformer $\alpha_t([\![\texttt{S}]\!]_{\perp})$ in structural fixpoint form. The approximation abstraction α_a is then handled using the fixpoint induction principles of Sect. II.3 to under or over approximate the transformer by $\alpha_a \circ \alpha_t([\![\texttt{S}]\!]_{\perp})$. [Aczel 1977] has shown that set theoretic fixpoints can be expressed as proof/deductive systems and conversely. We recall his method in Sect. II.5. This yields a method of designing proof system by calculus in Sect. II.5.3. This is applied to two new example logics. The first example in section II.8.1 is a forward transformational logic to express correct reachability of a postcondition (as in Hoare and Manna partial correctness logics), termination (as in Apt & Plokin and Manna &

Pnueli logics) as well as nontermination, all cases being expressible by a single formula of the logic (depending on initial values). The second example in section II.8.2 is a backward transformational logic to express correct accessibility of a postcondition or nontermination.

II.1 STRUCTURAL FIXPOINT NATURAL RELATIONAL SEMANTICS

We define the relational natural semantic $[\![\mathtt{S}]\!]_\perp \in \wp(\Sigma \times \Sigma_\perp)$ of statements $\mathtt{S}$ by structural induction on the program syntax and iteration defined as extremal fixpoints of increasing (monotone/isotone) functions on complete lattices [Tarski 1955].

The definition is in Milner/Tofte style [Milner and Tofte 1991], except that finite behaviors in $\wp(\Sigma\times\Sigma)$ are in inductive style with least fixpoints (lfp) and infinite behaviors in $\wp(\Sigma\times\{\perp\})$ are in co-inductive style with greatest fixpoints (gfp), as in [Cousot and Cousot 1992, 2009]. Milner/Tofte define both finite and infinite behaviors in co-inductive style [Leroy 2006; Milner and Tofte 1991], which looks more uniform. However, some fixpoint approximation techniques are more precise for least fixpoints than for greatest fixpoints [Cousot 2021, Chapter 18], which will be essential to prove completeness of proof methods[2].

Given the assignment $\sigma[\mathtt{x} \leftarrow v]$ of value $v \in \mathbb{V}$ to variable $\mathtt{x} \in \mathbb{X}$ in state $\sigma \in \Sigma \triangleq \mathbb{X} \to \mathbb{V}$ and the identity relation $\mathsf{id} \triangleq \{\langle\sigma,\ \sigma\rangle \mid \sigma \in \Sigma_\perp\}$, the basic statements have the following semantics. They all terminate and do not exit loops, but for `break`, that exits the closest outer loop (which existence must be checked syntactically) without changing the values of variables.

$$
\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
[\![\mathtt{x = A}]\!]^e &\triangleq& \{\langle\sigma,\ \sigma[\mathtt{x}\leftarrow\mathcal{A}[\![\mathtt{A}]\!]\sigma]\rangle \mid \sigma\in\Sigma\} & [\![\mathtt{x = A}]\!]^b &\triangleq& \varnothing & [\![\mathtt{x = A}]\!]^\perp &\triangleq& \varnothing\\
[\![\mathtt{x = [a,\ b]}]\!]^e &\triangleq& \{\langle\sigma,\ \sigma[\mathtt{x}\leftarrow i]\rangle \mid \sigma\in\Sigma\wedge & [\![\mathtt{x = [a,b]}]\!]^b &\triangleq& \varnothing & [\![\mathtt{x = [a,b]}]\!]^\perp &\triangleq& \varnothing\\
&& \qquad a-1<i<b+1\} &&&&&&\\
[\![\mathtt{break}]\!]^e &\triangleq& \varnothing & [\![\mathtt{break}]\!]^b &\triangleq& \mathsf{id} & [\![\mathtt{break}]\!]^\perp &\triangleq& \varnothing\\
[\![\mathtt{skip}]\!]^e &\triangleq& \mathsf{id} & [\![\mathtt{skip}]\!]^b &\triangleq& \varnothing & [\![\mathtt{skip}]\!]^\perp &\triangleq& \varnothing
\end{array}
\tag{47}
$$

For the conditional, we let $[\![\mathtt{B}]\!] \triangleq \{\langle\sigma,\ \sigma\rangle \mid \sigma \in \mathcal{B}[\![\mathtt{B}]\!]\}$ be the relational semantics of Boolean expressions. We define ($\fatsemi$ is the composition of relations, see Sect. A.1 in the appendix)

$$
\begin{array}{rcl@{\qquad}rcl}
[\![\mathtt{S_1;S_2}]\!]^e &\triangleq& [\![\mathtt{S_1}]\!]^e \fatsemi [\![\mathtt{S_2}]\!]^e & [\![\mathtt{if\ (B)\ S_1\ else\ S_2}]\!]^e &\triangleq& [\![\mathtt{B}]\!] \fatsemi [\![\mathtt{S_1}]\!]^e \cup [\![\neg\mathtt{B}]\!] \fatsemi [\![\mathtt{S_2}]\!]^e\\
[\![\mathtt{S_1;S_2}]\!]^b &\triangleq& [\![\mathtt{S_1}]\!]^b \cup ([\![\mathtt{S_1}]\!]^e \fatsemi [\![\mathtt{S_2}]\!]^b) & [\![\mathtt{if\ (B)\ S_1\ else\ S_2}]\!]^b &\triangleq& [\![\mathtt{B}]\!] \fatsemi [\![\mathtt{S_1}]\!]^b \cup [\![\neg\mathtt{B}]\!] \fatsemi [\![\mathtt{S_2}]\!]^b\\
[\![\mathtt{S_1;S_2}]\!]^\perp &\triangleq& [\![\mathtt{S_1}]\!]^\perp \cup ([\![\mathtt{S_1}]\!]^e \fatsemi [\![\mathtt{S_2}]\!]^\perp) & [\![\mathtt{if\ (B)\ S_1\ else\ S_2}]\!]^\perp &\triangleq& [\![\mathtt{B}]\!] \fatsemi [\![\mathtt{S_1}]\!]^\perp \cup [\![\neg\mathtt{B}]\!] \fatsemi [\![\mathtt{S_2}]\!]^\perp
\end{array}
\tag{48}
$$

For iteration, we define

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

$$F^\perp(X) \triangleq [\![\mathtt{B}]\!] \fatsemi [\![\mathtt{S}]\!]^e \fatsemi X, \quad X \in \wp(\Sigma\times\{\perp\}) \tag{50}$$

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

$$[\![\mathtt{while\ (B)\ S}]\!]^b \triangleq \varnothing \tag{52}$$

$$[\![\mathtt{while\ (B)\ S}]\!]^\perp \triangleq (\mathsf{lfp}^{\subseteq} F^e \fatsemi [\![\mathtt{B}]\!] \fatsemi [\![\mathtt{S}]\!]^\perp) \cup \mathsf{gfp}^{\subseteq} F^\perp \tag{53}$$

The transformers are defined on complete lattices, F^e on $\langle\wp(\Sigma\times\Sigma),\ \subseteq,\ \varnothing,\ \Sigma\times\Sigma,\ \cup,\ \cap\rangle$ and $F^\perp$ on $\langle\wp(\Sigma_\perp\times\{\perp\}),\ \subseteq,\ \varnothing,\ \vec{\infty},\ \cup,\ \cap\rangle$ with $\vec{\infty} \triangleq \Sigma\times\{\perp\}$ and are $\subseteq$-increasing, so do exist [Tarski 1955].

Moreover, the natural transformer F^e in (49) preserves arbitrary joins, so is continuous. By Scott-Kleene fixpoint theorem [Scott and Strachey 1971], its least fixpoint is the reflexive transitive closure $\mathsf{lfp}^{\subseteq} F^e = \bigcup_{n\geqslant 0}([\![\mathtt{B}]\!] \fatsemi [\![\mathtt{S}]\!]^e)^n = ([\![\mathtt{B}]\!] \fatsemi [\![\mathtt{S}]\!]^e)^*$. So $\mathsf{lfp}^{\subseteq} F^e$ is a relation between initial states before entering the loop and successive states at loop reentry after any number $n \geqslant 0$ of iterations. If,

[2] For example, Park induction Th. II.3.1 can be used to over approximate least fixpoints with an invariant only while approximating greatest fixpoints in the dual of Th. II.3.8 involves a variant function.

after n iterations, the test B ever becomes false then $[\![\mathtt{B}]\!] = \varnothing$ and so all later terms in the infinite disjunction are empty.

Then composing $\mathsf{lfp}^{\subseteq} F^e = ([\![\mathtt{B}]\!] \mathbin{⨟} [\![\mathtt{S}]\!]^e)^*$ with $[\![\neg\mathtt{B}]\!] \cup [\![\mathtt{B}]\!] \mathbin{⨟} [\![\mathtt{S}]\!]^b$ in (51) yields the relation between initial and final states in case of termination or in case of a break when excuting the loop body S. (52) states that a `break` exits the immediately enclosing loop, not any of the outer ones.

Composing $\mathsf{lfp}^{\subseteq} F^e = ([\![\mathtt{B}]\!] \mathbin{⨟} [\![\mathtt{S}]\!]^e)^*$ with $[\![\mathtt{B}]\!] \mathbin{⨟} [\![\mathtt{S}]\!]^\perp$ in (53) yields the possible cases of nontermination when the loop body S does not terminate after finitely many finite iterations in the loop.

Finally, the term $\mathsf{gfp}^{\subseteq} F^\perp$ in (53) represents infinitely many iterations of terminating body executions. Again if B becomes false after finitely many iterations then $[\![\mathtt{B}]\!] = \varnothing$ so that this infinite iteration term is $\varnothing$ (since $\varnothing$ is absorbant for ⨟). As shown by [Cousot 2002, Example 22], $F^\perp$ may not be co-continuous when considering unbounded nondeterminism so that transfinite decreasing fixpoint iterations from the supremum might be necessary [Cousot and Cousot 1979a]. The following lemma makes clear that $\mathsf{gfp}^{\subseteq} F^\perp$ characterizes (non)termination Ⓐ

Lemma II.1.1 (Termination). $\mathsf{gfp}^{\subseteq} F^\perp = \varnothing \Leftrightarrow \{\sigma \in \mathbb{N} \to \Sigma \mid \forall i \in \mathbb{N} \,.\, \langle \sigma_i,\ \sigma_{i+1}\rangle \in [\![\mathtt{B}]\!] \mathbin{⨟} [\![\mathtt{S}^e]\!]\} = \varnothing$.

Since $\perp \notin \Sigma$, $([\![\mathtt{S}]\!]^e \cup [\![\mathtt{S}]\!]^\perp) \cap \Sigma = [\![\mathtt{S}]\!]^e$ and $([\![\mathtt{S}]\!]^e \cup [\![\mathtt{S}]\!]^\perp) \cap \{\perp\} = [\![\mathtt{S}]\!]^\perp$, the semantics can be defined as

$$\begin{array}{rcllr} [\![\mathtt{S}]\!]_\perp & \triangleq & \langle [\![\mathtt{S}]\!]^e \cup [\![\mathtt{S}]\!]^\perp,\ [\![\mathtt{S}]\!]^b\rangle & \in \ \wp(\Sigma \times \Sigma_\perp) \times \wp(\Sigma \times \Sigma) & \text{natural semantics} \quad (54)\\ [\![\mathtt{S}]\!] & \triangleq & \langle [\![\mathtt{S}]\!]^e,\ [\![\mathtt{S}]\!]^b\rangle & \in \ \wp(\Sigma \times \Sigma) \times \wp(\Sigma \times \Sigma) & \text{angelic semantics} \end{array}$$

where $\langle \wp(\Sigma \times \Sigma_\perp),\ \sqsubseteq,\ \Sigma \times \{\perp\},\ \Sigma \times \Sigma,\ \sqcup,\ \sqcap\rangle$ is a complete lattice for the computational ordering $X \sqsubseteq Y \triangleq (X \cap (\Sigma \times \Sigma)) \subseteq (Y \cap (\Sigma \times \Sigma)) \wedge (X \cap (\Sigma \times \{\perp\})) \supseteq (Y \cap (\Sigma \times \{\perp\}))$. It follows that the definition of termination on normal exit or nontermination can be defined by a single transformer [Cousot 2002, Theorem 9] (but termination $[\![\mathtt{S}]\!]^e$ and break $[\![\mathtt{S}]\!]^b$ cannot be mixed without losing information).

This relational natural semantics can be extended to record a relation between the initial and current values of variables. This consists in considering the Galois connections $\langle \alpha_{\downarrow^2},\ \gamma_{\downarrow^2}\rangle$ for assertions and $\langle \dot{\alpha}_{\downarrow^2},\ \dot{\gamma}_{\downarrow^2}\rangle$ for relations in (24). This can be implemented using auxiliary variables without modification of the semantics Ⓐ.

Nondeterminism can be unbounded, as discussed in the appendix Ⓐ.

II.2 FIXPOINT ABSTRACTION

We recall classic fixpoint abstraction theorems [Cousot 2002], [Cousot 2021, Ch. 18] to abstract the fixpoint definition of the program relational semantics into a fixpoint definition of transformers (or their graph). Abstraction can also be applied to deductive systems Ⓐ.

Theorem II.2.1 (Fixpoint abstraction [Cousot and Cousot 1979b]). *If* $\langle C,\ \sqsubseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A,\ \preceq\rangle$ *is a Galois connection between complete lattices* $\langle C,\ \sqsubseteq\rangle$ *and* $\langle A,\ \preceq\rangle$, $f \in C \xrightarrow{i} C$ *and* $\bar{f} \in A \xrightarrow{i} A$ *are increasing and commuting, that is,* $\alpha \circ f = \bar{f} \circ \alpha$, *then* $\alpha(\mathsf{lfp}^{\sqsubseteq} f) = \mathsf{lfp}^{\preceq} \bar{f}$ *(while semi-commutation* $\alpha \circ f \preceq \bar{f} \circ \alpha$ *implies* $\alpha(\mathsf{lfp}^{\sqsubseteq} f) \preceq \mathsf{lfp}^{\preceq} \bar{f}$*).*

As a simple application, we will need the following corollary Ⓐ.

Corollary II.2.2 (Pointwise abstraction). *Let* $\langle L,\ \sqsubseteq,\ \top,\ \sqcup\rangle$ *and* $\langle L',\ \sqsubseteq',\ \top',\ \sqcup'\rangle$ *be complete lattices. Assume that* $F \in (L \to L') \xrightarrow{i} (L \to L')$ *is increasing and that for all* $Q \in L$, $\bar{F}_Q \in L' \xrightarrow{i} L'$ *is increasing. Assume* $\forall Q \in L \,.\, \forall f \in L \to L' \,.\, F(f)Q = \bar{F}_Q(f(Q))$. *Then* $\forall Q \in L \,.\, (\mathsf{lfp}^{\sqsubseteq'} F)Q = \mathsf{lfp}^{\sqsubseteq'} \bar{F}_Q$.

When the abstraction involves the negation abstraction $\alpha^{\neg}$, Park's classic fixpoint theorem [Park 1979, equation (4,1,2)] is useful (and generalizes to complete Boolean lattices).

Theorem II.2.3 (Complement dualization). *If X is a set and $f \in \wp(X) \xrightarrow{i} \wp(X)$ is $\subseteq$-increasing then* $\mathsf{lfp}^{\subseteq}\, \alpha^{\sim}(f) = \alpha^{\neg}(\mathsf{gfp}^{\subseteq} f)$.

II.3 FIXPOINT INDUCTION

Least or greatest fixpoint definitions of the graph of transformers provide strongest or antecedent-consequent (or weakest consequent-antecedent) pairs. Then we need to take into account consequence rules, that is, approximations discussed in Sect. I.3.4. In this section, and in addition to [Cousot 2019b] and [Cousot 2021, Ch. 24], we introduce fixpoint induction methods to handle such approximations $\mathsf{post}(\supseteq, \subseteq)$, $\mathsf{post}(\subseteq, \supseteq)$, etc. In this section II.3, $\perp$ is the infimum of a poset and possibly unrelated to nontermination.

II.3.1 Least Fixpoint Over Approximation

The classic least fixpoint (lfp) over approximation theorem (and order dually over approximation of greatest fixpoints (gfp)), called "fixpoint induction", is due to Park [Park 1969] and follows directly from Tarski's fixpoint theorem [Tarski 1955], $\mathsf{lfp}^{\sqsubseteq} f = \bigsqcap\{x \in L \mid f(x) \sqsubseteq x\}$.

Theorem II.3.1 (Least 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 $\mathsf{lfp}^{\sqsubseteq} f \sqsubseteq p$ if and only if $\exists i \in L \,.\, f(i) \sqsubseteq i \wedge i \sqsubseteq p$.*

Example II.3.2. An invariant of a conditional iteration `while(B) S` with precondition P must satisfy $\mathsf{lfp}^{\subseteq} \boldsymbol{\lambda} X \bullet P \cup \mathsf{post}[\![S]\!](B \cap X) \subseteq I$. The proof method provided by Park's Th. II.3.1 is $\exists J \,.\, P \subseteq J \wedge \mathsf{post}[\![S]\!](B \cap J) \subseteq J \wedge J \subseteq I$ which is Turing [Turing 1950]/Floyd [Floyd 1967] invariant proof method. ■

By order-duality, this is sound and complete greatest fixpoint under approximation $p \sqsubseteq \mathsf{gfp}^{\sqsubseteq} f$ proof method. i is called an invariant (a co-invariant for greatest fixpoints).

Example II.3.3. Continuing example II.3.2, by contraposition, the invariant must satisfy $\neg I \subseteq \neg\mathsf{lfp}^{\subseteq} \boldsymbol{\lambda} X \bullet P \cup \mathsf{post}[\![S]\!](B \cap X)$ that is $\neg I \subseteq \mathsf{gfp}^{\subseteq} \boldsymbol{\lambda} X \bullet \neg P \cap \widetilde{\mathsf{post}}[\![S]\!](\neg B \cup X)$ by Park's Th. II.2.3. The dual of Th. II.3.1 suggest the proof method $\exists J \,.\, J \subseteq \neg P \wedge J \subseteq \widetilde{\mathsf{post}}[\![S]\!](\neg B \cup J) \wedge I \subseteq \neg J$ which is methods $(\widetilde{\mathrm{i}^{-1}})$ and $(\widetilde{\mathrm{I}^{-1}})$ of [Cousot and Cousot 1982]. ■

II.3.2 Ordinals

We let $\langle \mathbb{O}, \in, \emptyset, \mathbb{O}, \cup, \cap\rangle$ be the von Neumann's ordinals [von Neumann 1923], writing the more intuitive $<$ for $\in$, 0 for $\emptyset$, $+1$ for the successor function, sometimes max for $\cup$, min for $\cap$, and ω for the first infinite limit ordinal. If necessary, a short refresher on ordinals is given in Sect. H of the appendix Ⓐ.

II.3.3 Over Approximation of the Abstraction of a Least Fixpoint

To solve the problem $\alpha(\mathsf{lfp}^{\sqsubseteq} F) \sqsubseteq P$ where α is a function on the domain of F, we can try to use fixpoint abstraction Th. II.2.1 to get $\alpha(\mathsf{lfp}^{\sqsubseteq} F) = \mathsf{lfp}^{\sqsubseteq} \bar{F}$ and then check $\mathsf{lfp}^{\sqsubseteq} \bar{F} \sqsubseteq P$ by fixpoint induction Th. II.3.1. But Th. II.2.1 requires α to preserves joins, which is not always the case (for the dual problem $\alpha = \mathsf{pre}$ in remark I.3.12 is a counter-example). If α does not preserves joins, we can nevertheless use the following theorem Ⓐ.

Theorem II.3.4 (Overapproximation of a least fixpoint image). *Let $\langle L, \sqsubseteq, \perp, \sqcup\rangle$ and $\langle \bar{L}, \bar{\sqsubseteq}, \bar{\top}, \bar{\sqcup}\rangle$ be complete lattices[3], $F \in L \xrightarrow{i} L$ and $\alpha \in L \xrightarrow{i} \bar{L}$ be increasing functions, and $P \in \bar{L}$.*

Then $\alpha(\mathsf{lfp}^{\sqsubseteq} F) \mathrel{\bar{\sqsubseteq}} P$ if and only if there exists $I \in \bar{L}$ such that (1) $\alpha(\perp) \mathrel{\bar{\sqsubseteq}} I$ (2) $\forall X \in L \,.\, \alpha(X) \mathrel{\bar{\sqsubseteq}} I \Rightarrow \alpha(F(X)) \mathrel{\bar{\sqsubseteq}} I$, (3) for any $\sqsubseteq$-increasing chain $\langle X^{\delta}, \delta \in \mathbb{O}\rangle$ of elements $X^{\delta} \sqsubseteq \mathsf{lfp}^{\sqsubseteq} F$, $\forall \beta < \lambda \,.\, \alpha(X^{\beta}) \mathrel{\bar{\sqsubseteq}} I$ implies $\alpha(\bigsqcup_{\beta<\lambda} X^{\beta}) \mathrel{\bar{\sqsubseteq}} I$, and (4) $I \mathrel{\bar{\sqsubseteq}} P$.

[3] or CPOs.

Let $\langle F^\delta,\ \delta \in \mathbb{O}\rangle$ be the increasing iterates of F from $\perp$ ultimately stationary at rank ϵ [Cousot and Cousot 1979a]. Then condition II.3.4.(2) in is only necessary for all $X = F^\delta$, $\delta \leqslant \epsilon$ while condition (3) is only necessary for $\langle X^\delta,\ \delta \leqslant \epsilon\rangle = \langle F^\delta,\ \delta \leqslant \epsilon\rangle$. These weaker conditions are assumed to prove completeness ("only if" in Th. II.3.4).

II.3.4 Fixpoint Under Approximation by Transfinite Iterates

For under approximation of least fixpoints (or order dually over approximation of greatest fixpoints), we can use the generalization [Cousot 2019b] of Scott-Kleene induction based on transfinite induction when continuity does not apply and follows directly from the constructive version of Tarski's fixpoint theorem [Cousot and Cousot 1979a].

Definition II.3.5 (Ultimately Over Approximating Transfinite Sequence). We say that "the transfinite sequence $\langle X^\delta,\ \delta \in \mathbb{O}\rangle$ of elements of poset $\langle L,\ \sqsubseteq\rangle$ for $f \in L \to L$ ultimately over approximates $P \in L$" if and only if $X^0 = \perp$, $X^{\delta+1} \sqsubseteq f(X^\delta)$ for successor ordinals, $\bigsqcup_{\delta<\lambda} X^\delta$ exists for limit ordinals λ such that $X^\lambda \sqsubseteq \bigsqcup_{\delta<\lambda} X^\delta$, and $\exists \delta \in \mathbb{O}\ .\ P \sqsubseteq X^\delta$.

The condition can equivalently be expressed as $\forall \delta \in \mathbb{O}\ .\ X^\delta \sqsubseteq f(\bigsqcup_{\beta<\delta} X^\beta + 1)$ which avoids to have to make the distinction between successor and limit ordinals Ⓐ.

Theorem II.3.6 (Fixpoint Under Approximation by Transfinite Iterates). *Let $f \in L \xrightarrow{i} L$ be an increasing function on a* cpo *$\langle L,\ \sqsubseteq,\ \perp,\ \sqcup\rangle$ (i.e. every increasing chain in L has a least upper bound in L, including $\perp = \sqcup\varnothing$). $P \in L$ is a fixpoint underapproximation, i.e. $P \sqsubseteq \mathsf{lfp}^{\sqsubseteq} f$, if and only if there exists an increasing transfinite sequence $\langle X^\delta,\ \delta \in \mathbb{O}\rangle$ for f ultimately over approximating P (Def. II.3.5).*

Notice that ordinals are an abstraction $\langle \mathfrak{Wf},\ \subseteq\rangle \underset{\rho}{\overset{\mathrm{id}}{\leftrightarrows}} \langle \mathbb{O},\ \leqslant\rangle$ of well-founded sets by their rank ρ, so that Th. II.3.6 could have assumed the existence of a well-founded set to replace the ordinals. The hypothesis that $\langle X^\delta,\ \delta \in \mathbb{O}\rangle$ is increasing is necessary in a cpo but not in a complete lattice, in which case this non-increasing sequence can be used to build an increasing one Ⓐ.

Lemma II.3.7. *Let $\langle X^\delta,\ \delta \in \mathbb{O}\rangle$ be a sequence in a complete lattice satisfying the hypotheses of Def. II.3.5, then there is an increasing one satisfying these same hypotheses.*

II.3.5 Fixpoint Under Approximation by Bounded Iterates

For iterations, under approximations such as $P \subseteq \mathsf{post}[\![\mathtt{S}]\!]_\perp Q$ (incorrectness logic), $P \subseteq \mathsf{pre}[\![\mathtt{S}]\!]_\perp \Sigma$ (possible termination), $P \subseteq \neg\mathsf{pre}[\![\mathtt{S}]\!]_\perp\{\perp\} = \widetilde{\mathsf{pre}}[\![\mathtt{S}]\!]_\perp \Sigma$ (definite termination), and $P \subseteq \mathsf{pre}[\![\mathtt{S}]\!]_\perp Q \cap \widetilde{\mathsf{pre}}[\![\mathtt{S}]\!]_\perp Q$ (weakest precondition, starting from any initial state of P, $\mathtt{S}_\perp$ "is certain to establish eventually the truth of" Q [Dijkstra 1976, page 17]) are fixpoint under approximations. Programmers almost never use Th. II.3.6 for proving termination using ordinals (or a well-founded set). They cannot use Hoare logic either since nontermination $\{P\}\mathtt{S}\{\mathsf{false}\}$ is provable by the logic but its negation $\neg(\{P\}\mathtt{S}\{\mathsf{false}\})$ is not in the logic. A first method for bounded iteration uses a loop counter incremented on each iteration and an invariant proving that the counter is bounded ("time clocks" in [Knuth 1997], [Luckham and Suzuki 1977; Sokolowski 1977]). This is sound but incomplete for unbounded nondeterminism. The most popular method uses well-founded sets, which can be generalized to fixpoints Ⓐ.

Theorem II.3.8 (Least Fixpoint Under Approximation with a Variant Function). *We assume that (1) f is increasing on a* cpo *$\langle L,\ \sqsubseteq,\ \perp,\ \sqcup\rangle$; (2) that $P \in L$; (3) that there exists a sequence $\langle X^\delta,\ \delta \in \mathbb{O}\rangle$ of elements of L such that $X^0 = \perp$, $X^{\delta+1} \sqsubseteq f(X^\delta)$ for successor ordinals, and $X^\lambda \sqsubseteq \bigsqcup_{\beta<\lambda} X^\beta$ for limit ordinals λ; and (4) that there exists a well-founded set $\langle W,\ \preceq\rangle$ and a variant function $v \in \{X^\delta \mid \delta \in \mathbb{O}\} \to W$ such that for all $\beta < \delta$, we have $P \not\sqsubseteq X^\beta$ implies $v(X^\beta) \succ v(X^\delta)$.*

Hypotheses(1) to (4) imply that $\exists \delta < \omega\ .\ P \sqsubseteq X^\delta \sqsubseteq f^\delta \sqsubseteq \mathsf{lfp}^{\sqsubseteq} f$.

Because $\delta < \omega$ in Th. II.3.8, the proof method is sound but incomplete, as shown by the following counter example where the property holds but the proof method of Th. II.3.8, is inapplicable.

Example II.3.9. Consider the complete lattice $\langle \wp(\mathbb{Z}), \subseteq \rangle$. Define $f \in \wp(\mathbb{Z}) \to \wp(\mathbb{Z})$ by $f(X) = \{0\} \cup \{x \in \mathbb{Z} \mid x - 1 \in X\}$. The iterates are $f^0 = \emptyset$, $f^n = \{k \in \mathbb{N} \mid 0 \leqslant x < n\}$. The limit is $f^\omega = \bigcup_{n \in \mathbb{N}} f^n = \mathbb{N} = \mathsf{lfp}^{\subseteq} f$. Take $P = \mathbb{N}$ such that $P \subseteq \mathsf{lfp}^{\subseteq} f$. Then $\forall n \in \mathbb{N} \,.\, P \nsubseteq f^n$. It follows that Def. II.3.8 is infeasible since $\forall n \in \mathbb{N} \,.\, P \nsubseteq f^n$ implies for all $\beta < \delta$ that $\nu(X^\beta) \succ \nu(X^\delta)$. This infinite strictly decreasing chain is in contradiction with the well-foundness hypothesis. ∎

II.3.6 Void Intersection With Fixpoint Using Variant Functions

Turing and Floyd [Floyd 1967; Turing 1950] method for unbounded nondeterminism, uses *reductio ad absurdum*, proving that nontermination is impossible. This idea can also be generalized to fixpoints.

An atom of a poset $\langle L, \sqsubseteq \rangle$ is either a minimal element of L if L has no infimum or covers the infimum $\bot$ otherwise. So the set of atoms of a poset $\langle L, \sqsubseteq \rangle$ is $\mathsf{atoms}(L) \triangleq \{a \in L \mid \nexists x' \in L \,.\, x' \sqsubset a\}$ if L has no infimum and $\mathsf{atoms}(L) \triangleq \{a \in L \mid \nexists x' \in L \,.\, \bot \sqsubset x' \sqsubset a\}$ if $\bot$ is the infimum of L. The atoms of an element x of L are $\mathsf{atoms}(x) \triangleq \{a \in \mathsf{atoms}(L) \mid a \sqsubseteq x\}$. A poset is atomic if the atoms of any element x of L have a join which exists and is x, that is, $\forall x \in L \,.\, x = \bigsqcup \mathsf{atoms}(x)$. Co-atomicity is $\sqsubseteq$-order-dual. We have Ⓐ

Theorem II.3.10 (Void intersection with least fixpoint). *We assume that (1) $\langle L, \sqsubseteq, \bot, \top, \sqcap, \sqcup \rangle$ is an atomic complete lattice; (2) $f \in L \to L$ preserves non-empty joins; (3) there exists an invariant $I \in L$ of f (i.e. such that $f(I) \sqsubseteq I$); (4) that there exists a well-founded set $\langle W, \preceq \rangle$ and a variant function $\nu \in I \to W$ such that $\forall x \in \mathsf{atoms}(I) \,.\, (x \neq f(x)) \Rightarrow (\nu(x) \succ \nu(f(x)))$; (5) $Q \in L$; and (6) $\forall x \in \mathsf{atoms}(I) \,.\, (\nu(x) \nsucc \nu(f(x))) \Rightarrow (x \sqcap Q = \bot)$. Then, hypotheses (1) to (6) imply $\mathsf{lfp}^{\sqsubseteq} f \sqcap Q = \bot$.*

Th. II.3.10 is useful, in particular, to prove $\mathsf{lfp}^{\sqsubseteq} f = \bot$ for $Q = \top$. If $L = \wp(\Sigma_\bot)$ then $P \subseteq Q$ is $P \cap \neg Q = \emptyset$, another possible use of this theorem.

The proof method of Th. II.3.10 is incomplete, as shown by counter-example H.1 in the appendix. The completeness of Turing/Floyd variant function method is due to the additional property that the inverse of the transition relation of a terminating program is well-founded Ⓐ (see example H.2 in the appendix).

Theorem II.3.11 (Turing/Floyd). *Let $r \in \wp(\mathcal{X} \times \mathcal{X})$ be a relation on a set $\mathcal{X}$ and $P \in \wp(\mathcal{X})$. Then*

$$\begin{aligned} & \{x \in \mathcal{X} \mid \exists \sigma \in \mathbb{N} \to \mathcal{X} \,.\, \sigma_0 = x \in P \wedge \forall i \in \mathbb{N} \,.\, \langle \sigma_i, \sigma_{i+1} \rangle \in r\} = \emptyset \\ \Leftrightarrow\quad & \{x \in \mathcal{X} \mid x \in P \wedge \exists \langle W, \preceq \rangle \in \mathfrak{Wf} \,.\, \exists I \in \wp(\mathcal{X}) \,.\, P \cup \mathsf{post}(r) I \subseteq I \wedge \exists \nu \in I \to W \,.\, \\ & \qquad \forall y \in I \,.\, \forall y' \in \mathcal{X} . \langle y, y' \rangle \in r \Rightarrow \nu(y) \succ \nu(y')\} \end{aligned}$$

Notice that the soundness proof given in the appendix uses (the dual of) Th. II.3.8 which shows that it is a generalization for Turing/Floyd variant method. Notice that if the intersection of $\mathsf{gfp}^{\sqsubseteq} f$ with Q is empty ($\bot$ in the lattice) then so is the intersection of $\mathsf{lfp}^{\sqsubseteq} f$ with Q but not conversely, so in addition to theorem II.3.10, we also need the following Ⓐ

Theorem II.3.12 (Void intersection with greatest fixpoint). *We assume that (1) $\langle L, \sqsubseteq, \bot, \top, \sqcap, \sqcup \rangle$ is a coatomic complete lattice; (2) $f \in L \to L$ preserves non-empty meets; (3) there exists a coinvariant $I \in L$ of f (i.e. such that $I \sqsubseteq f(I)$); (4) that there exists a well-founded set $\langle W, \preceq \rangle$ and a variant function $\nu \in I \to W$ such that $\forall x \in \mathsf{coatoms}(I) \,.\, (x \neq f(x)) \Rightarrow (\nu(x) \succ \nu(f(x)))$; (5) $Q \in L$; and (6) $\forall x \in \mathsf{coatoms}(I) \,.\, (\nu(x) \nsucc \nu(f(x))) \Rightarrow (x \sqcap Q = \bot)$. Then, hypotheses (1) to (6) imply $\mathsf{gfp}^{\sqsubseteq} f \sqcap Q = \bot$.*

Notice that Th. II.3.12, as well as its proof in Sect. H of the appendix, are not the order dual of Th. II.3.10 since (6) have the same conclusion $x \sqcap Q = \bot$ and the dual of the conclusion $\mathsf{lfp}^{\sqsubseteq} f \sqcap Q = \bot$ would be $\mathsf{gfp}^{\sqsubseteq} f \sqcup Q = \top$.

II.3.7 Fixpoint Non Emptiness

Another result to handle greatest fixpoints, e.g. to prove definite nontermination, is the following theorem Ⓐ.

THEOREM II.3.13 (GREATEST FIXPOINT NON EMPTINESS). *Let* $f \in L \xrightarrow{i} L$ *be an increasing function of a complete lattice* $\langle L, \sqsubseteq, \bot, \top, \sqcup, \sqcap\rangle$ *and* $P \in L \setminus \{\bot\}$. *Then* $\mathsf{gfp}^{\sqsubseteq} f \sqcap P \neq \bot$ *if and only if* $\forall X \in L\ .\ (\mathsf{gfp}^{\sqsubseteq} f \sqsubseteq X \wedge f(X) \sqsubseteq X \wedge X \sqcap P \neq \bot) \Rightarrow (f(X) \sqcap P \neq \bot)$.

A fixpoint induction principle H.3 for $\alpha(\mathsf{lfp}^{\sqsubseteq} f) \sqcap P \neq \bot$ in (39.d) is given in the appendix. Ⓐ

II.4 DEDUCTIVE SYSTEMS OF PROGRAM LOGICS

Logics define the valid properties of a program as all provable facts by the formal proof system of the logic. These formal systems, introduced by Hilbert [Hilbert and Ackermann 1938, § 5], are "a system of axioms from which the remaining true sentences may be obtained by means of certain rules". Such a formal system is a finitely presented set of axioms c and rules $\frac{P}{c}$ where the axioms and conclusions c of the rules are terms with variables and the premisses P are formulas of a logic.

The semantics/interpretation of the logic maps logical terms to elements of a mathematical structure with universe $\mathcal{U}$. Logical formulas are interpreted as the subsets of $\mathcal{U}$ of elements satisfying the formulas. Therefore logical implication is subset inclusion $\subseteq$ in the complete Boolean lattice $\langle\wp(\mathcal{U}), \subseteq, \varnothing, \mathcal{U}, \cup, \cap, \neg\rangle$ where $\varnothing$ is false, $\mathcal{U}$ true, $\cup$ disjunction, $\cap$ conjunction, and $\neg$ negation. The semantics/interpretation of the formal rules is a deductive system $R = \left\{\frac{P_i}{c_i}\ \middle|\ i \in \Delta\right\}$ where $P_i \in \wp(\mathcal{U})$ is the finite premise and $c_i \in \mathcal{U}$ the conclusion of the rule. The axioms have $P_i = \varnothing$ (false) as premises. We have $R \in \wp(\wp(\mathcal{U}) \times \mathcal{U})$ where pairs $\langle P,\ c\rangle$ are conventionally written $\frac{P}{c}$.

Example II.4.1. The formal system $1 \in \mathcal{O}$ and inductive rule $\frac{n \in \mathcal{O}}{n+2 \in \mathcal{O}}$ (defining the odd naturals $\mathcal{O}$ on universe $\mathbb{N}$) has the interpretation $\{\frac{\varnothing}{1}\} \cup \{\frac{\{n\}}{n+2} \mid n \in \mathbb{N}\}$. For example if $2 \in \mathbb{N}$ is odd then 4 is odd. To prove that 2 is odd, the only way is to prove that 0 is odd which is not an axiom nor the conclusion of a rule, proving 2 not to be odd. ■

II.5 THE SEMANTICS OF DEDUCTIVE SYSTEMS

Aczel [Aczel 1977] has shown that there are two equivalent ways of defining the subset $\alpha^{\mathcal{I}}(R)$ of the universe $\mathcal{U}$ defined by a deductive system $R = \left\{\frac{P_i}{c_i}\ \middle|\ i \in \Delta\right\}$.

II.5.1 Proof-Theoretic Semantics of Deductive Systems

In the proof-theoretic approach, $\alpha^{\mathcal{I}}(R)$ is the set of provable elements where a formal proof is a finite sequence $t_1 \ldots\ t_n$ of terms (i.e. elements of the universe $\mathcal{U}$) such that any term is the conclusion of a rule which premise is implied by (i.e. included in $\subseteq$) the set of previous terms in the sequence (which have been already proved, starting with axioms). Therefore $\alpha^{\mathcal{I}}(R) = \{t_n \in \mathcal{U} \mid \exists t_1, \ldots, t_{n-1} \in \mathcal{U}\ .\ \forall k \in [1, n]\ .\ \exists \frac{P}{c} \in R\ .\ P \subseteq \{t_1, \ldots, t_{k-1}\} \wedge t_k = c\}$ (this requires P to be finite).

It follows that there is a Galois connection $\langle\wp(\wp(\mathcal{U}) \times \mathcal{U}),\ \subseteq\rangle \underset{\alpha^{\mathcal{I}}}{\overset{\gamma^{\mathcal{I}}}{\leftrightarrows}} \langle\wp(\mathcal{U}),\ \subseteq\rangle$ where $\alpha^{\mathcal{I}}$ is $\subseteq$-increasing (the more rules the larger is the defined set) and $\gamma^{\mathcal{I}}(X) = \{\frac{P}{c} \mid P \in \wp(\mathcal{U}) \wedge c \in X\}$ including axioms $\frac{\varnothing}{c}$ collecting all elements c of X. (As discussed thereafter, there are other, more natural and effective, possible deductive systems. Proof systems are not unique.)

II.5.2 Model-Theoretic Semantics of Deductive Systems

In the model-theoretic approach, the same $\alpha^{\mathcal{I}}(R)$ is defined as $\alpha^{\mathcal{I}}(R) = \mathsf{lfp}^{\subseteq} \alpha^F(R)$ where the consequence operator is $\alpha^F(R)X \triangleq \{c \mid \exists \frac{P}{c} \in R\ .\ P \subseteq X\}$. $\alpha^F(R)X$ is the set of consequences derivable from the hypotheses $X \in \wp(\mathcal{U})$ by one application of an axiom (with $P = \varnothing$) or a rule of the deductive system. $\alpha^F(R) \in \wp(\mathcal{U}) \xrightarrow{\sqcup} \wp(\mathcal{U})$ preserve nonempty joins and so is increasing. The least fixpoint $\mathsf{lfp}^{\subseteq}\,\alpha^F(R)$ of the consequence operator is well-defined [Tarski 1955] and is the set of all provable terms, that is, $\alpha^{\mathcal{I}}(R)$. For example II.4.1, $\mathcal{O} = \mathsf{lfp}^{\subseteq}\,\boldsymbol{\lambda} X\bullet\{1\}\cup\{n+2 \mid \{n\}\subseteq X\}$.

II.5.3 Equivalence of the Two Definitions of the Semantics of Deductive Systems

The definitions of a subset of the universe by a deductive system or by a fixpoint are equivalent [Aczel 1977]. We have recalled that a deductive system can be expressed in fixpoint form. Conversely, given any increasing operator F on $\langle\mathcal{U},\,\subseteq\rangle$, the terms provable by the deductive system $\gamma^F(F) = \left\{\frac{P}{c} \;\middle|\; P\in\wp(\mathcal{U})\wedge c\in F(P)\right\}$ (or $\left\{\frac{P}{c} \;\middle|\; P\in\wp(\mathcal{U})\wedge c\in F(P)\wedge\forall P'\in\wp(\mathcal{U})\,.\,c\in F(P')\Rightarrow P\subseteq P'\right\}$) are exactly its least fixpoint $\mathsf{lfp}^{\subseteq} F$. This yields a Galois connection between deductive systems and increasing consequence operators $\langle\wp(\wp(\mathcal{U})\times\mathcal{U}),\,\subseteq\rangle \underset{\alpha^F}{\overset{\gamma^F}{\leftrightarrows}} \langle\wp(\mathcal{U}) \xrightarrow{i} \wp(\mathcal{U}),\,\dot{\subseteq}\rangle$ where $\dot{\subseteq}$ is $\subseteq$, pointwise. Note that there is also a Galois connection between increasing operators and fixpoints $\langle\wp(\mathcal{U}) \xrightarrow{i} \wp(\mathcal{U}),\,\dot{\subseteq}\rangle \underset{\mathsf{lfp}^{\subseteq}}{\overset{\boldsymbol{\lambda} y\bullet\boldsymbol{\lambda} x\bullet(\!|x\subseteq y\,?\,y\,\mathrm{:}\,\mathcal{U}|\!)}{\leftrightarrows}} \langle\wp(\mathcal{U}),\,\subseteq\rangle$ such that $\alpha^{\mathcal{I}}$ is the composition of these two Galois connections.

The order dual of this result is defined by co-induction leading to greatest fixpoints $\langle\wp(\mathcal{U}) \xrightarrow{i} \wp(\mathcal{U}),\dot{\subseteq}\rangle \underset{\mathsf{gfp}^{\subseteq}}{\overset{\boldsymbol{\lambda} y\bullet\boldsymbol{\lambda} x\bullet(\!|x\supseteq y\,?\,y\,\mathrm{:}\,\varnothing|\!)}{\leftrightarrows}} \langle\wp(\mathcal{U}),\subseteq\rangle$, we get the coinductive interpretation of proof systems. It can also be biinductive, a mix of the two, taking the lfp of $\alpha^F(R)$ restricted to a subset of $\mathbb{V}\subseteq\mathcal{U}$ of the universe and gfp on $\alpha^F(R)$ restricted to the complement $\mathcal{U}\setminus\mathbb{V}$ [Cousot and Cousot 1992, 1995, 2009].

More generally, the results hold for any complete lattice $\langle L,\,\preceq\rangle$ thus generalizing the powerset case $\langle\wp(\mathcal{U}),\,\subseteq\rangle$ and its order dual [Cousot and Cousot 1995].

The take away is that, knowing the fixpoint semantics of the logic, there is a method for constructing the deductive system for that logic, which is both sound and complete, by construction. An example I.1 is given in the appendix showing how to construct the deductive natural relational semantics Sect. I.1.1 from its fixpoint definition of Sect. II.1 Ⓐ.

II.6 CALCULATIONAL DESIGN OF PROOF SYSTEMS

After defining the theory of a logic by abstraction $\alpha_a\circ\alpha_t(\llbracket \mathtt{S}\rrbracket_\perp)$ of the relational semantics $\llbracket \mathtt{S}\rrbracket_\perp$, we use the fixpoint abstraction theorems of Sect. II.2 to provide a fixpoint definition of $\alpha_t(\llbracket \mathtt{S}\rrbracket_\perp)$, which is most often a transformer or its graph. Then to handle α_a, which is an approximation abstraction like $\mathsf{post}(\subseteq,\supseteq)$ or $\mathsf{post}(\supseteq,\subseteq)$, we use the fixpoint induction theorems of Sect. II.3 to provide a set-theoretic of the theory of the logic which is then translated in a proof system by Aczel method of Sect. II.5.3.

Example II.6.1. Assume that $\alpha_t(\llbracket \mathtt{S}\rrbracket_\perp) = \mathsf{lfp}^{\subseteq} F_P$ and that we must derive the abstract theory $T = \alpha_a\circ\alpha_t(\llbracket \mathtt{S}\rrbracket_\perp) = \{\langle P,\,Q\rangle \mid \mathsf{lfp}^{\subseteq} F_P\subseteq Q\}$ (e.g. to handle the $\subseteq$ part in $\mathsf{post}(\subseteq,\supseteq) = \mathsf{post}(=,\supseteq)\circ\mathsf{post}(\subseteq,=)$ or $\mathsf{post}(\supseteq,\subseteq)$, the other part $\supseteq$ being dual). By Th. II.3.1, $T = \{\langle P,\,Q\rangle \mid \exists I\,.\,F_P(I)\subseteq I\wedge I\subseteq Q\}$. By Sect. II.5.2. this set T is defined by the axiom $\frac{F_P(I)\subseteq I,\ I\subseteq Q}{\langle P,\,Q\rangle\in T}$. ■

Remark II.6.2. *(On abstraction versus induction)* Hoare logic is the $\mathsf{post}(\subseteq,\supseteq)$ and it's reverse is the $\mathsf{post}(\supseteq,\subseteq)$ abstraction of the transformer graph $T = \{\langle P,\,\mathsf{post}\llbracket \mathtt{S}\rrbracket P\rangle \mid P\in\wp(\Sigma)\}$. Both proof systems can be designed, by the rules for T plus the consequence rules for $\mathsf{post}(\subseteq,\supseteq)$ and $\mathsf{post}(\supseteq,\subseteq)$. By (51), the theory T of the conditional iteration $\mathtt{W}$ without `breaks` would involve $T' = \{\langle P,\,\mathsf{post}(\mathsf{lfp}^{\subseteq} F^e)P\rangle \mid P\in\wp(\Sigma)\}$. The rule would be (9), using ordinals for unbounded nondeterminism. So to prove $\{P\}\,\mathtt{S}\,\{Q\}$, $P\neq\varnothing$, we would have to find a postcondition Q', prove that it is the strongest, and then use the consequence rule to prove that $Q'\subseteq Q$/ This is sound and complete but much too demanding. The fixpoint induction theorems of Sect. II.3 solve this problem by weakening the rules for iteration while preserving soundness and completeness. Contrary to fixpoint abstraction, fixpoint induction allows us to take the consequence rule into account in the design of proof rules for fixpoint semantics. So partial correctness need not be a consequence of total correctness and nontermination. ■

II.7 ON THE COMPARISON OF LOGICS

To compare logics, we first relate their theories, that is compare their expressivity, through their respective abstractions of the collecting semantics (as formalized by fixpoint abstraction in Sect. II.2). Different abstractions yield different logics, compared though their relation by Galois connections. The logics are equivalent when their theories are linked by a Galois isomorphism. An example is given in Sect. I.3.14.4 where Hoare logic and subgoal induction have the same theory but different proof method (as shown in figure 3).

The proof system of a logic is entirely determined by its theory (as proved in Sect. II.4), but up to an equivalence, since different induction principles may be used, as formalized in Sect. II.3, to exploit approximation so as to simplify induction. This is exemplified by Rem. II.6.2. Which induction principle is used is the second characteristic to compare logics.

II.8 APPLICATIONS

The development of Hoare incorrectness logic in Ex. I.3.11 is relegated to the appendix Ⓐ.

II.8.1 Application I: Calculational Design of a New Forward Logic for Termination with Correct Reachability of a Postcondition or Nontermination

Using $\perp$ to denote nontermination, we write $Q_\perp \triangleq Q \cup \{\perp\}$ and $Q_{\not\perp} \triangleq Q \setminus \{\perp\}$. The semantics and predicates/assertions are relational. They can establish a relation between initial and final values of a loop body to show that a variant function in a well founded set is decreasing (as in Turing/Floyd method formalized by Th. II.3.11). See example I.3.1.

The language includes a `break` out of the closest enclosing loop, so the specifications have the form $\{P\}\,\mathtt{S}\,\{ok : Q, br : T\}$ meaning that any execution of $\mathtt{S}$ started in a state of P will terminate in a state of $Q_{\not\perp}$, or not terminate if $\perp \in Q$, or break out of $\mathtt{S}$ to the closest enclosing loop in a state satisfying T. So $Q = \{\perp\}$ and $T = \varnothing$ would mean definite non termination (when $P \neq \varnothing$).

To design the logic, we first formally define the meaning of specifications as an abstraction of post. Then we proceed by structural induction on the syntax of the language. Using fixpoint over approximation Th. II.3.1, the iteration rule is ($\overline{\Sigma}$ is Σ extended to an auxiliary variable in $\overline{\mathbb{X}}$ for each variable in $\mathbb{X}$) Ⓐ

$$\frac{\begin{array}{c}\{\sigma \in \wp(\overline{\Sigma}) \mid \sigma_{\overline{\mathbb{X}}} = \sigma_{\mathbb{X}} \wedge \sigma_{\mathbb{X}} \in P\} \subseteq I \quad \{\mathcal{B}[\![\mathtt{B}]\!] \cap I_{\not\perp}\}\,\mathtt{S}\,\{ok : R, br : T\}\\ R_{\not\perp} \subseteq I \quad (\mathcal{B}[\![\neg\mathtt{B}]\!] \cap I) \subseteq Q \quad T \subseteq Q \quad R_\perp \subseteq Q\\ (\perp \notin Q) \Rightarrow (\exists\langle W, \preceq\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in I \to W \,.\, \forall\langle\underline{\sigma}, \sigma'\rangle \in I \,.\, \nu(\underline{\sigma}) \succ \nu(\sigma'))\end{array}}{\{P\}\,\mathtt{while\ (B)\ S}\,\{ok : Q, br : T\}} \tag{55}$$

Example II.8.1. For factorial `fact`, we choose the invariant $I = I_{\not\perp} \cup I_\perp$ with $I_{\not\perp} = \{n = \underline{n} \wedge f = 1\} \cup \{\underline{n} > n \geqslant 0 \wedge f = \prod_{i=n}^{\underline{n}} i\}$ with $\prod\varnothing = 1$ for termination and $I_\perp = \{n \leqslant \underline{n} < 0 \wedge f = \perp\}$ for nontermination, $\langle W, \preceq\rangle = \langle \mathbb{N}, \leqslant\rangle \in \mathfrak{Wf}$, $\nu(\underline{n}, n) = n$. We have $\mathcal{B}[\![\mathtt{B}]\!] = n \neq 0$ so that $\{\mathcal{B}[\![\mathtt{B}]\!] \cap I_{\not\perp}\}$ `f = f*n; n = n-1;` $\{ok : R, br : T\}$ is $\{(n = \underline{n} \neq 0 \wedge f = 1) \vee (n > \underline{n} > 0 \wedge f = \prod_{i=n}^{\underline{n}} i)\}$ `f = f*n; n = n-1;` $\{ok : R, br : \varnothing\}$ with $R = R_{\not\perp} = (n = \underline{n} - 1 \neq 0 \wedge f = \underline{n}) \vee (n > \underline{n} > 0 \wedge f = \prod_{i=n}^{\underline{n}} i) \subseteq I$, $R_\perp = \varnothing$ and $T = \varnothing$ by termination and absence of `break`. ■

II.8.2 Application II: Calculational Design of a New Program Logic for possible Accessibility of a Postcondition or Nontermination

As a second example, we design a logic $\{P\}\,\mathtt{S}\,\overleftarrow{\{}ok : Q, br : T\}$ for the language of Sect. II.1 with natural semantics (54). A quadruple $\{P\}\,\mathtt{S}\,\overleftarrow{\{}ok : Q, br : T\}$ means that for any state in P there exists at least one execution from that state that terminates in a state of Q, or T through a break, or does not terminate (contrary to incorrectness logic [O'Hearn 2020] requiring termination). For example if Q is bad, $\perp \notin Q$, and $T = \varnothing$ then from any state of P there must be a finite execution reaching a bad state in Q (unless all executions from that state in P do not terminate or P is empty), which corresponds to the incorrectness component of the outcome logic [Zilberstein et al. 2023] in case $\perp \notin Q$. $\{P\}\,\mathsf{S}\,\overleftarrow{\{}ok : \{\perp\}, br : \varnothing\}$ stipulates that any initial state in P can lead to at least one nonterminating execution (as opposed to the extended Hoare specification $\{P\}\,\mathsf{S}\,\{ok : \{\perp\}, br : \varnothing\}$ stating that no execution from P can terminate). Formally,

$$\{P\}\,\mathsf{S}\,\overleftarrow{\{}ok : Q, br : T\} \quad \triangleq \quad \langle P,\ Q,\ T\rangle \in \alpha_{\mathsf{pre}}(\llbracket \mathsf{S}\rrbracket_{\perp}) \tag{56}$$

where the abstraction $\alpha_{\mathsf{pre}}(\llbracket \mathsf{S}\rrbracket_{\perp})$ is

$$\{\langle P,\ Q,\ T\rangle \in \wp(\Sigma)\times\wp(\Sigma_{\perp})\times\wp(\Sigma) \mid (P \subseteq \mathsf{pre}(\llbracket \mathsf{S}\rrbracket^{e} \cup \llbracket \mathsf{S}\rrbracket^{\perp})Q) \vee (P \subseteq \mathsf{pre}(\llbracket \mathsf{S}\rrbracket^{b})T)\} \tag{57}$$

that is
$$\begin{aligned}\alpha_{\mathsf{pre}}(\llbracket \mathsf{S}\rrbracket_{\perp}) \quad = \quad & \{\langle P,\ Q,\ T\rangle \mid P \subseteq \mathsf{pre}(\llbracket \mathsf{S}\rrbracket^{e} \cup \llbracket \mathsf{S}\rrbracket^{\perp})Q\} && \text{(A)}\\ & \cup \{\langle P,\ Q,\ T\rangle \mid P \subseteq \mathsf{pre}(\llbracket \mathsf{S}\rrbracket^{b})T\} && \text{(B)} \qquad \text{(def. union } \cup\text{)}\end{aligned} \tag{58}$$

We proceed by structural induction on statements (details are found in the appendix Ⓐ). We consider the iteration `while(B) S` which is the most difficult case. Let us start with the easy case (B) of (58) for the iteration `while(B) S`.

$$\begin{aligned}& \{\langle P,\ Q,\ T\rangle \mid P \subseteq \mathsf{pre}(\llbracket \mathtt{while(B)\ S}\rrbracket^{b})T\}\\ = \quad & \{\langle P,\ Q,\ T\rangle \mid P \subseteq \mathsf{pre}(\varnothing)T\} = \{\langle P,\ Q,\ T\rangle \mid P \subseteq \varnothing\} && \text{(def. (52) of } \llbracket \mathtt{while(B)\ S}\rrbracket^{b} \text{ and (38) of pre)}\\ = \quad & \{\langle \varnothing,\ Q,\ T\rangle \mid Q \in \wp(\Sigma_{\perp}) \wedge T \in \wp(\Sigma)\} && \text{(def. inclusion} \subseteq \text{and empty set } \varnothing\text{)}\end{aligned}$$

Following Sect. II.5, (A) $\cup$ (B) can be defined by a deductive system with separate rules for (A) and (B). With notation (56), the deductive system for (B) of (58) for the iteration `while(B) S` is the axiom

$$\frac{\varnothing}{\{\varnothing\}\,\mathtt{while(B)\ S}\,\overleftarrow{\{}ok : Q, br : T\}} \tag{59}$$

meaning that if you never execute a program you can conclude anything on its executions. This is also valid in Hoare logic but is not given as an explicit axiom since it can be derived from other rules (by an extensive induction on all program statements).

We now have to consider case (A) of (58) for the iteration `while(B) S`. This is more difficult and requires three pages of calculation plus two pages of auxiliary propositions, too much for most readers. So we will sketch the main steps of this calculation for a global understanding and refer to Sect. J.4 of the appendix for all further details.

Starting from $\{\langle P,\ Q,\ T\rangle \mid P \subseteq \mathsf{pre}(\llbracket \mathtt{while(B)\ S}\rrbracket^{e} \cup \llbracket \mathtt{while(B)\ S}\rrbracket^{\perp})Q\}$ we get

$$\begin{aligned}\{\langle P^{e} \cup P^{\perp},\ Q,\ T\rangle \mid\ & P^{e} \subseteq (\mathsf{pre}(\mathsf{lfp}^{\subseteq} F^{e} \mathbin{⨾} \llbracket \neg\mathsf{B}\rrbracket)Q_{\not\perp} \cup \mathsf{pre}(\mathsf{lfp}^{\subseteq} F^{e} \mathbin{⨾} \llbracket \mathsf{B}\rrbracket \mathbin{⨾} \llbracket \mathsf{S}\rrbracket^{b})Q_{\not\perp}) \wedge\\ & P^{\perp} \subseteq (\mathsf{pre}(\mathsf{lfp}^{\subseteq} F^{e} \mathbin{⨾} \llbracket \mathsf{B}\rrbracket \mathbin{⨾} \llbracket \mathsf{S}\rrbracket^{\perp})\{\perp \mid \perp \in Q\} \cup \mathsf{pre}(\mathsf{gfp}^{\subseteq} F^{\perp})\{\perp \mid \perp \in Q\}))\}\end{aligned}$$

by case analysis and expanding definitions (51) and (53) of the semantics. For states in P^{e} there is an execution terminating in $Q_{\not\perp} \triangleq Q \setminus \{\perp\}$, possibly by a break. For states in $P^{\perp}_{b}$ there is an infinite execution consisting of zero or more finite iterations followed by a nonterminating execution of the loop body. Finally, for states in $P^{\perp}_{\ell}$ there is an execution consisting of infinitely many finite iterations. These cases are not exclusive and might be empty.

Grouping the cases $P = P^{e} \cup P^{\perp}_{b}$, we get (by case analysis and $\mathsf{pre}(r)\varnothing = \varnothing$)

$$\begin{aligned}\{\langle P \cup P^{\perp}_{\ell},\ Q,\ T\rangle \mid P \subseteq \mathsf{lfp}^{\subseteq} \boldsymbol{\lambda} X \bullet (\mathsf{pre}\llbracket \neg\mathsf{B}\rrbracket Q_{\not\perp}) \cup (\mathsf{pre}(\llbracket \mathsf{B}\rrbracket \mathbin{⨾} \llbracket \mathsf{S}\rrbracket^{b})Q_{\not\perp}) \cup (\mathsf{pre}(\llbracket \mathsf{B}\rrbracket \mathbin{⨾} \llbracket \mathsf{S}\rrbracket^{\perp})\{\perp \mid \perp \in Q\}) \cup\\ (\mathsf{pre}(\llbracket \mathsf{B}\rrbracket \mathbin{⨾} \llbracket \mathsf{S}\rrbracket^{e})X) \wedge (\!| \perp \in Q \mathrel{?} P^{\perp}_{\ell} \subseteq \mathsf{pre}(\mathsf{gfp}^{\subseteq} F^{\perp})\{\perp\} \mathrel{:} P^{\perp}_{\ell} = \varnothing |\!)\}\end{aligned}$$

We must now handle fixpoint under approximations using induction principles. Since the pre transformer preserves arbitrary joins, its least fixpoint iterations are stable at ω. So the hypotheses of Th. II.3.6 for $P \subseteq \mathsf{lfp}^{\subseteq} \boldsymbol{\lambda} X \bullet A \cup \mathsf{pre}(r)X$ become

$$\exists\langle I^n,\ n\in\mathbb{N}\rangle\ .\ I^0=\varnothing\wedge\forall n\in\mathbb{N}\ .\ I^n\subseteq I^{n+1}\subseteq A\cup\mathsf{pre}(r)I^n\wedge\exists\ell\in\mathbb{N}\ .\ P\subseteq I^\ell \tag{60}$$

Since pre does not preserve meets, we cannot use the dual of the fixpoint abstraction Th. II.2.1 to express $\mathsf{pre}(\mathsf{gfp}^{\subseteq}F^{\perp})\{\perp\mid\perp\in Q\}$ in fixpoint form and then use the dual of fixpoint induction Th. II.3.4. This is where the order dual Th. II.3.4 is useful to under approximate the image of the greatest fixpoint $\mathsf{gfp}^{\subseteq}F^{\perp}$ by $\boldsymbol{\lambda} r\bullet\mathsf{pre}(r)\{\perp\mid\perp\in Q\}$. The dual hypotheses of Th. II.3.4 are that there exists $J\in\wp(\Sigma_\perp)$ such that, after simplifications for that particular case, are

$$\exists J\in\wp(\Sigma_\perp)\ .\ \llparenthesis\,\perp\in Q\ ?\ \mathsf{pre}(\llbracket\mathsf{B}\rrbracket\mathbin{;}\llbracket\mathsf{S}\rrbracket^e)(J)\subseteq J\wedge P_\ell^\perp\subseteq J\ {:}\ \mathsf{true}\,\rrparenthesis \tag{61}$$

so that, by (60), (61), and $\mathsf{pre}(\llbracket\mathsf{B}\rrbracket)R=\mathcal{B}\llbracket\mathsf{B}\rrbracket\cap R$, we get
$\{\langle P\cup P_\ell^\perp,\ Q,\ T\rangle\mid\exists\langle I^n,\ n\in\mathbb{Q}\rangle\ .\ I^0=\varnothing\wedge\forall n\in\mathbb{N}\ .\ I^n\subseteq I^{n+1}\subseteq(\mathcal{B}\llbracket\neg\mathsf{B}\rrbracket\cap Q_{\not\perp})\cup(\mathcal{B}\llbracket\mathsf{B}\rrbracket\cap\mathsf{pre}(\llbracket\mathsf{S}\rrbracket^b)Q_{\not\perp})$
$\cup\,(\mathcal{B}\llbracket\mathsf{B}\rrbracket\cap\mathsf{pre}\llbracket\mathsf{S}\rrbracket^{\perp}\{\perp\mid\perp\in Q\})\cup(\llbracket\mathsf{B}\rrbracket\cap\mathsf{pre}\llbracket\mathsf{S}\rrbracket^e(I^n))\wedge\exists\ell\in\mathbb{N}\ .\ P\subseteq I^\ell\wedge\exists J\in\wp(\Sigma_\perp)\ .\ \llparenthesis\,\perp\in Q\ ?$
$\mathcal{B}\llbracket\mathsf{B}\rrbracket\cap\mathsf{pre}(\llbracket\mathsf{S}\rrbracket^e)(J)\subseteq J\wedge P_\ell^\perp\subseteq J\ {:}\ P_\ell^\perp=\varnothing\,\rrparenthesis\}$

Since we proceed by structural induction, we have, by definition (56), to make the under approximations of $\mathsf{pre}\llbracket\mathsf{S}\rrbracket$ for the loop body `S` to appear explicitly in the calculations, as follows

$\{\langle P\cup P_\ell^\perp,\ Q,\ T\rangle\mid\exists R^b\ .\ R^b\subseteq\mathsf{pre}(\llbracket\mathsf{S}\rrbracket^b)Q_{\not\perp}\wedge\exists R^\perp\ .\ \langle R^\perp,\ \{\perp\mid\perp\in Q\},\ \varnothing\rangle\in\alpha_{\mathsf{pre}}(\llbracket\mathsf{S}\rrbracket_\perp)\wedge\exists\langle I^n,$
$n\in\mathbb{Q}\rangle\ .\ I^0=\varnothing\wedge\forall n\in\mathbb{N}\ .\ \exists R_n^e\ .\ \langle R_n^e,\ I^n,\ \varnothing\rangle\in\alpha_{\mathsf{pre}}(\llbracket\mathsf{S}\rrbracket_\perp)\wedge I^n\subseteq I^{n+1}\subseteq(\mathcal{B}\llbracket\neg\mathsf{B}\rrbracket\cap Q_{\not\perp})\cup(\llbracket\mathsf{B}\rrbracket\cap R^b)\cup$
$(\llbracket\mathsf{B}\rrbracket\cap R^\perp)\cup(\llbracket\mathsf{B}\rrbracket\cap R_n^e)\wedge\exists\ell\in\mathbb{N}\ .\ P\subseteq I^\ell\wedge\exists J\in\wp(\Sigma_\perp)\ .\ \exists R_\ell^\perp\ .\ \langle R_\ell^\perp,\ J,\ \varnothing\rangle\in\alpha_{\mathsf{pre}}(\llbracket\mathsf{S}\rrbracket)\wedge\llparenthesis\,\perp\in Q\ ?$
$\mathcal{B}\llbracket\mathsf{B}\rrbracket\cap R_\ell^\perp\subseteq J\wedge P_\ell^\perp\subseteq J\ {:}\ P_\ell^\perp=\varnothing\,\rrparenthesis\}$

Following Sect. II.5 and using the notation (56), the theory $\alpha_{\mathsf{pre}}(\llbracket\texttt{while(B) S}\rrbracket_\perp)$ can be equivalently defined by the following deductive system of the logic.

$$\frac{\begin{array}{c}I^0=\varnothing\quad\{R^b\}\,\mathtt{S}\,\overleftarrow{\{}ok:\varnothing,br:Q_{\not\perp}\}\quad\{R^\perp\}\,\mathtt{S}\,\overleftarrow{\{}ok:\{\perp\mid\perp\in Q\},br:\varnothing\}\\ \forall n\in\mathbb{N}\ .\ \{R_n^e\}\,\llbracket\mathsf{S}\rrbracket\,\overleftarrow{\{}ok:I^n,br:\varnothing\}\\ I^n\subseteq I^{n+1}\subseteq(\mathcal{B}\llbracket\neg\mathsf{B}\rrbracket\cap Q_{\not\perp})\cup(\mathcal{B}\llbracket\mathsf{B}\rrbracket\cap R^b)\cup(\mathcal{B}\llbracket\mathsf{B}\rrbracket\cap R^\perp)\cup(\mathcal{B}\llbracket\mathsf{B}\rrbracket\cap R_n^e)\quad\exists\ell\in\mathbb{N}\ .\ P\subseteq I^\ell\\ \llparenthesis\,\perp\in Q\ ?\ \{R_\ell^\perp\}\,\llbracket\mathsf{S}\rrbracket\,\overleftarrow{\{}ok:J,br:\varnothing\}\wedge\mathcal{B}\llbracket\mathsf{B}\rrbracket\cap R_\ell^\perp\subseteq J\wedge P_\ell^\perp\subseteq J\ {:}\ P_\ell^\perp=\varnothing\,\rrparenthesis\end{array}}{\{P\cup P_\ell^\perp\}\,\texttt{while(B) S}\,\overleftarrow{\{}ok:Q,br:\varnothing\}} \tag{62}$$

Example II.8.2. Continuing Ex. I.3.1 and I.3.5, consider the factorial with postcondition contract `f` >0. An interval analysis produces an alarm $Q=Q_{\not\perp}=\mathtt{f}\leqslant 0$ where $\perp\notin Q$ so $Q_\perp=\varnothing$ and $P_\ell^\perp=\varnothing$. Take $R^\perp=R^b=\varnothing$ since the loop body terminates with no break. Let $I^k=n\leqslant k\wedge f\leqslant 0$ and $R_k^e=I^{k-1}$ so that $\{R_k^e\}\,\llbracket\texttt{f = f*n; n = n-1;}\rrbracket\,\overleftarrow{\{}ok:I^k,br:\varnothing\}$. Take $P=I^{|\underline{n}|}$. By (62), $\{P\}\,\texttt{fact}\,\overleftarrow{\{}ok:Q,br:\varnothing\}$. But P implies `f` $\leqslant 0$ in contradiction $\{\varnothing\}\,\texttt{f=1;}\,\overleftarrow{\{}ok:P,br:\varnothing\}$ with the initialization `f=1` proving the unreachable alarm to be false, which [O'Hearn 2020; Zilberstein et al. 2023] cannot do. ■

II.9 CONCLUSION

Related work was moved to the appendix Sect. K Ⓐ. We have shown that the theory of abstract interpretation can be used to design program transformational logics, including (non)termination, by defining their theory as an abstraction of the programming language fixpoint natural relational semantics and then their proof system (useful to support mechanization) by fixpoint induction and Aczel correspondence between set-theoretic fixpoint definitions and deductive systems [Aczel 1977]. The approach applies to all other abstractions of the collecting semantics into a relation, not necessarily into a logic. For future work, this same principled approach can be used to design hyper logics [Dardinier 2023], including dependency logics [Cousot 2019a], to include meta information in predicates with an instrumented semantics (e.g. [Vanegue 2022; Zhang and Kaminski 2022; Zilberstein et al. 2023]), and to extend temporal logics like [Pnueli 1979] to programming languages by structural induction and local invariants [Bubel et al. 2023].

DATA AVAILABILITY STATEMENT

The full version of this article is available in a single file on Zenodo with clickable hyper references to the appendix [Cousot 2024], https://doi.org/10.5281/zenodo.10439108.

ACKNOWLEDGMENTS

I thank the participants to the Dagstuhl Seminar on "Theoretical Advances and Emerging Applications in Abstract Interpretation" 09–14 July 2023 and Jeffery Wang for discussions. I thank the reviewers for appreciation and criticisms, corrections, and useful suggestions, Francesco Ranzato for improvement proposals, and Charles de Haro for numerous corrections.

REFERENCES

Peter Aczel. 1977. An Introduction to Inductive Definitions. In *Handbook of Mathematical Logic*, John Barwise (Ed.). North–Holland, Amsterdam, Chapter 7, 739–782.

Krzysztof R. Apt. 1981. Ten Years of Hoare's Logic: A Survey - Part I. *ACM Trans. Program. Lang. Syst.* 3, 4 (1981), 431–483. https://doi.org/10.1145/357146.357150

Krzysztof R. Apt. 1984. Ten Years of Hoare's Logic: A Survey Part II: Nondeterminism. *Theor. Comput. Sci.* 28 (1984), 83–109. https://doi.org/10.1016/0304-3975(83)90066-X

Krzysztof R. Apt and Ernst-Rüdiger Olderog. 2019. Fifty years of Hoare's logic. *Formal Aspects Comput.* 31, 6 (2019), 751–807. https://doi.org/10.1007/s00165-019-00501-3

Krzysztof R. Apt and Ernst-Rüdiger Olderog. 2021. Assessing the Success and Impact of Hoare's Logic. In *Theories of Programming*. ACM / Morgan & Claypool, 41–76. https://doi.org/10.1145/3477355.3477359

Krzysztof R. Apt and Gordon D. Plotkin. 1986. Countable Nondeterminism and Random Assignment. *J. ACM* 33, 4 (1986), 724–767. https://doi.org/10.1145/6490.6494

Ali Asadi, Krishnendu Chatterjee, Hongfei Fu, Amir Kafshdar Goharshady, and Mohammad Mahdavi. 2021. Polynomial reachability witnesses via Stellensätze. In *PLDI*. ACM, 772–787. https://doi.org/10.1145/3453483.3454076

Flavio Ascari, Roberto Bruni, and Roberta Gori. 2022. Limits and difficulties in the design of under-approximation abstract domains. In *FoSSaCS (Lecture Notes in Computer Science, Vol. 13242)*. Springer, 21–39. https://doi.org/10.1007/978-3-030-99253-8_2

Flavio Ascari, Roberto Bruni, Roberta Gori, and Francesco Logozzo. 2023. Sufficient Incorrectness Logic: SIL and Separation SIL. *CoRR* abs/2310.18156 (2023). https://doi.org/10.48550/arXiv.2310.18156

Arnon Avron, Furio Honsell, Ian A. Mason, and Robert Pollack. 1992. Using Typed Lambda Calculus to Implement Formal Systems on a Machine. *J. Autom. Reason.* 9, 3 (1992), 309–354. https://doi.org/10.1007/BF00245294

Thomas Ball, Orna Kupferman, and Greta Yorsh. 2005. Abstraction for Falsification. In *CAV (Lecture Notes in Computer Science, Vol. 3576)*. Springer, 67–81. https://doi.org/10.1007/11513988_8

Merrie Bergmann. 1977. Logic and Sortal Incorrectness. *The Review of Metaphysics* 31, 1 (September 1977), 61–79. https://www.jstor.org/stable/20127017

François Le Berre and Alexandre Tessier. 1996. Declarative Incorrectness Diagnosis in Constraint Logic Programming. In *APPIA-GULP-PRODE*. 379–390.

Andreas Blass and Yuri Gurevich. 2000. The Underlying Logic of Hoare Logic. *Bull. EATCS* 70 (2000), 82–111. https://doi.org/10.1142/4566

Roberto Bruni, Roberto Giacobazzi, Roberta Gori, and Francesco Ranzato. 2023. A Correctness and Incorrectness Program Logic. *J. ACM* 70, 2 (2023), 15:1–15:45. https://doi.org/10.1145/3582267

Richard Bubel, Dilian Gurov, Reiner Hähnle, and Marco Scaletta. 2023. Trace-based Deductive Verification. In *LPAR (EPiC Series in Computing, Vol. 94)*. EasyChair, 73–95. https://doi.org/10.29007/VDFD

Rod M. Burstall. 1969. Formal description of program structure and semantics in first order logic. In *Machine Intelligence 5*, Bernard Meltzer and Donald Michie (Eds.). Edinburgh University Press, 79–98. https://doi.org/10.2307/3612456

Michael R. Clarkson and Fred B. Schneider. 2010. Hyperproperties. *J. Comput. Secur.* 18, 6 (2010), 1157–1210. https://doi.org/10.3233/JCS-2009-0393

Stephen A. Cook. 1978. Soundness and Completeness of an Axiom System for Program Verification. *SIAM J. Comput.* 7, 1 (1978), 70–90. https://doi.org/10.1137/0207005

Stephen A. Cook. 1981. Corrigendum: Soundness and Completeness of an Axiom System for Program Verification. *SIAM J. Comput.* 10, 3 (1981), 612. https://doi.org/10.1137/0210045

Patrick Cousot. 1981. Semantic Foundations of Program Analysis. In *Program Flow Analysis: Theory and Applications*, S.S. Muchnick and N.D. Jones (Eds.). Prentice-Hall, Inc., Englewood Cliffs, New Jersey, Chapter 10, 303–342.

Patrick Cousot. 2002. Constructive Design of a Hierarchy of Semantics of a Transition System by Abstract Interpretation. *Theor. Comput. Sci.* 277, 1–2 (2002), 47–103. https://doi.org/10.1016/S0304-3975(00)00313-3

Patrick Cousot. 2019a. Abstract Semantic Dependency. In *SAS (Lecture Notes in Computer Science, Vol. 11822)*. Springer, 389–410. https://doi.org/10.1007/978-3-030-32304-2_19

Patrick Cousot. 2019b. On Fixpoint/Iteration/Variant Induction Principles for Proving Total Correctness of Programs with Denotational Semantics. In *LOPSTR (Lecture Notes in Computer Science, Vol. 12042)*. Springer, 3–18. https://doi.org/10.1007/978-3-030-45260-5_1

Patrick Cousot. 2021. *Principles of Abstract Interpretation* (1 ed.). MIT Press.

Patrick Cousot. 2024. Full version of "Calculational Design of [In]Correctness Transformational Program Logics by Abstract Interpretation", Proc. ACM Program. Lang. 8, POPL (2024), 7:1–10:33, https://doi.org/10.1145/3632849. *Zenodo* (Dec. 2024), 66 pages. https://doi.org/10.5281/zenodo.10439108

Patrick Cousot and Radhia Cousot. 1977. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In *POPL*. ACM, 238–252. https://doi.org/10.1145/512950.512973

Patrick Cousot and Radhia Cousot. 1979a. Constructive Versions of Tarski's Fixed Point Theorems. *Pacific J. of Math.* 82, 1 (1979), 43–57. https://doi.org/10.2140/pjm.1979.82.43

Patrick Cousot and Radhia Cousot. 1979b. Systematic Design of Program Analysis Frameworks. In *POPL*. ACM Press, 269–282. https://doi.org/10.1145/567752.567778

Patrick Cousot and Radhia Cousot. 1982. Induction principles for proving invariance properties of programs. In *Tools & Notions for Program Construction: an Advanced Course*, D. Néel (Ed.). Cambridge University Press, Cambridge, UK, 75–119.

Patrick Cousot and Radhia Cousot. 1992. Inductive Definitions, Semantics and Abstract Interpretation. In *POPL*. ACM Press, 83–94. https://doi.org/10.1145/143165.143184

Patrick Cousot and Radhia Cousot. 1995. Compositional and Inductive Semantic Definitions in Fixpoint, Equational, Constraint, Closure-condition, Rule-based and Game-Theoretic Form. In *CAV (Lecture Notes in Computer Science, Vol. 939)*. Springer, 293–308. https://doi.org/10.1007/3-540-60045-0_58

Patrick Cousot and Radhia Cousot. 2009. Bi-inductive structural semantics. *Inf. Comput.* 207, 2 (2009), 258–283. https://doi.org/10.1016/J.IC.2008.03.025

Patrick Cousot and Radhia Cousot. 2014. A Galois connection calculus for abstract interpretation. In *POPL*. ACM, 3–4. https://doi.org/10.1145/2535838.2537850

Patrick Cousot, Radhia Cousot, Manuel Fähndrich, and Francesco Logozzo. 2013. Automatic Inference of Necessary Preconditions. In *VMCAI (Lecture Notes in Computer Science, Vol. 7737)*. Springer, 128–148. https://doi.org/10.1007/978-3-642-35873-9_10

Patrick Cousot, Radhia Cousot, and Francesco Logozzo. 2011. Precondition Inference From Intermittent Assertions and Application to Contracts on Collections. In *VMCAI (Lecture Notes in Computer Science, Vol. 6538)*. Springer, 150–168. https://doi.org/10.1007/978-3-642-18275-4_12

Patrick Cousot, Radhia Cousot, Francesco Logozzo, and Michael Barnett. 2012. An abstract interpretation framework for refactoring with application to extract methods with contracts. In *OOPSLA*. ACM, 213–232. https://doi.org/10.1145/2384616.2384633

Patrick Cousot and Nicolas Halbwachs. 1978. Automatic Discovery of Linear Restraints Among Variables of a Program. In *POPL*. ACM Press, 84–96. https://doi.org/10.1145/512760.512770

Thibault Dardinier. 2023. Formalization of Hyper Hoare Logic: A Logic to (Dis-)Prove Program Hyperproperties. *Arch. Formal Proofs* 2023 (2023). https://www.isa-afp.org/entries/HyperHoareLogic.html

Edsko de Vries and Vasileios Koutavas. 2011. Reverse Hoare Logic. In *SEFM (Lecture Notes in Computer Science, Vol. 7041)*. Springer, 155–171. https://doi.org/10.1007/978-3-642-24690-6_12

Klaus Denecke, Marcel Erné, and Shelly L. Wismath. 2003. *Galois Connections and Applications*. Kluwer Academic Publishers. https://doi.org/10.1007/978-1-4020-1898-5

Edsger W. Dijkstra. 1975. Guarded Commands, Nondeterminacy and Formal Derivation of Programs. *Commun. ACM* 18, 8 (1975), 453–457. https://doi.org/10.1145/360933.360975

Edsger W. Dijkstra. 1976. *A Discipline of Programming*. Prentice-Hall.

Edsger W. Dijkstra. 1982. On subgoal induction. In *Selected Writings on Computing: A Personal Perspective*. Springer-Verlag New York, 223–224.

Edsger W. Dijkstra and Carel S. Scholten. 1990. *Predicate Calculus and Program Semantics*. Springer. https://doi.org/10.1007/978-1-4612-3228-5

James E. Donahue. 1976. *Complementary Definitions of Programming Language Semantics*. Lecture Notes in Computer Science, Vol. 42. Springer. https://doi.org/10.1007/BFB0025364

Vijay D'Silva and Caterina Urban. 2015. Conflict-Driven Conditional Termination. In *CAV (2) (Lecture Notes in Computer Science, Vol. 9207)*. Springer, 271–286. https://doi.org/10.1007/978-3-319-21668-3_16

Yuan Feng and Sanjiang Li. 2023. Abstract interpretation, Hoare logic, and incorrectness logic for quantum programs. *Inf. Comput.* 294 (2023), 105077. https://doi.org/10.1016/J.IC.2023.105077

Gérard Ferrand. 1993. The Notions of Symptom and Error in Declarative Diagnosis of Logic Programs. In *AADEBUG (Lecture Notes in Computer Science, Vol. 749)*. Springer, 40–57. https://doi.org/10.1007/BFb0019

Robert W. Floyd. 1967. Assigning Meaning to Programs. In *Proc. Symp. in Applied Math.*, J.T. Schwartz (Ed.). Vol. 19. Amer. Math. Soc., 19–32. https://doi.org/10.1007/978-94-011-1793-7_4

Alexey Gotsman, Josh Berdine, and Byron Cook. 2011. Precision and the Conjunction Rule in Concurrent Separation Logic. In *MFPS (Electronic Notes in Theoretical Computer Science, Vol. 276)*. Elsevier, 171–190. https://doi.org/10.1016/J.ENTCS.2011.09.021

George Grätzer. 1998. *General Lattice Theory* (2nd ed.). Birkhäuser.

David Harel. 1979. *First-Order Dynamic Logic*. Lecture Notes in Computer Science, Vol. 68. Springer. https://doi.org/10.1007/3-540-09237-4

Robert Harper, Furio Honsell, and Gordon D. Plotkin. 1993. A Framework for Defining Logics. *J. ACM* 40, 1 (1993), 143–184. https://doi.org/10.1145/138027.138060

David Hilbert and Wilhelm Ackermann. 1928, 1949, reprinted 1959. *Grundzüge der Theoretischen Logik* (6 ed.). Springer. Engl. Trans. "Principles of Mathematical Logic," Lewis M. Hammond, George G. Leckie, F. Steinhardt, AMS Chelsea, 1958, reprinted 2008.

David Hilbert and Wilhelm Ackermann. 1938. *Grundzüge der theoretischen Logik*. Springer-Verlag Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-41928-1

Charles Antony Richard Hoare. 1969. An Axiomatic Basis for Computer Programming. *Commun. ACM* 12, 10 (1969), 576–580. https://doi.org/10.1145/363235.363259

C. A. R. Hoare. 1978. Some Properties of Predicate Transformers. *J. ACM* 25, 3 (1978), 461–480. https://doi.org/10.1145/322077.322088

Michael Karr. 1976. Affine Relationships Among Variables of a Program. *Acta Inf.* 6 (1976), 133–151. https://doi.org/10.1007/BF00268497

Jinwoo Kim, Loris D'Antoni, and Thomas W. Reps. 2023. Unrealizability Logic. *Proc. ACM Program. Lang.* 7, POPL (2023), 659–688. https://doi.org/10.1145/3571216

Donald E. Knuth. 1997. *The Art of Computer Programming, Volume I: Fundamental Algorithms, 3rd Edition*. Addison-Wesley.

Quang Loc Le, Azalea Raad, Jules Villard, Josh Berdine, Derek Dreyer, and Peter W. O'Hearn. 2022. Finding real bugs in big programs with incorrectness logic. *Proc. ACM Program. Lang.* 6, OOPSLA1 (2022), 1–27. https://doi.org/10.1145/3527325

Xavier Leroy. 2006. Coinductive Big-Step Operational Semantics. In *ESOP (Lecture Notes in Computer Science, Vol. 3924)*. Springer, 54–68. https://doi.org/10.1007/11693024

John W. Lloyd. 1995. Debugging for a Declarative Programming Language. In *Machine Intelligence 15*. Oxford University Press, 341–359. https://doi.org/10.1093/oso/9780198538677.003.0019

David C. Luckham and Norihisa Suzuki. 1977. Proof of Termination within a Weak Logic of Programs. *Acta Informatica* 8 (1977), 21–36. https://doi.org/10.1007/BF00276182

Petar Maksimovic, Caroline Cronjäger, Andreas Lööw, Julian Sutherland, and Philippa Gardner. 2023. Exact Separation Logic: Towards Bridging the Gap Between Verification and Bug-Finding. In *ECOOP (LIPIcs, Vol. 263)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 19:1–19:27. https://doi.org/10.4230/LIPIcs.ECOOP.2023.19

Zohar Manna. 1971. Mathematical Theory of Partial Correctness. *J. Comput. Syst. Sci.* 5, 3 (1971), 239–253. https://doi.org/10.1016/S0022-0000(71)80035-1

Zohar Manna and Amir Pnueli. 1974. Axiomatic Approach to Total Correctness of Programs. *Acta Inf.* 3 (1974), 243–263. https://doi.org/10.1007/BF00288637

Marco Milanese and Francesco Ranzato. 2022. Local Completeness Logic on Kleene Algebra with Tests. In *SAS (Lecture Notes in Computer Science, Vol. 13790)*. Springer, 350–371. https://doi.org/10.1007/978-3-031-22308-2_16

Robin Milner and Mads Tofte. 1991. Co-Induction in Relational Semantics. *Theor. Comput. Sci.* 87, 1 (1991), 209–220. https://doi.org/10.1016/0304-3975(91)90033-X

Antoine Miné. 2014. Backward under-approximations in numeric abstract domains to automatically infer sufficient program conditions. *Sci. Comput. Program.* 93 (2014), 154–182. https://doi.org/10.1016/J.SCICO.2013.09.014

Bernhard Möller, Peter W. O'Hearn, and Tony Hoare. 2021. On Algebra of Program Correctness and Incorrectness. In *RAMiCS (Lecture Notes in Computer Science, Vol. 13027)*. Springer, 325–343. https://doi.org/10.1007/978-3-030-88701-8_20

James Donald Monk. 1969. *Introduction to Set Theory*. McGraw–Hill. http://euclid.colorado.edu/~monkd/monk11.pdf

F. Lockwood Morris and Cliff B. Jones. 1984. An Early Program Proof by Alan Turing. *IEEE Ann. Hist. Comput.* 6, 2 (1984), 139–143. https://doi.org/10.1109/MAHC.1984.10017

James H. Morris Jr. and Ben Wegbreit. 1977. Subgoal Induction. *Commun. ACM* 20, 4 (1977), 209–222. https://doi.org/10.1145/359461.359466

Toby Murray. 2020. An Under-Approximate Relational Logic. *Arch. Formal Proofs* 2020 (2020). https://www.isa-afp.org/entries/Relational-Incorrectness-Logic.html

Peter Naur. 1966. Proofs of Algorithms by General Snapshots. *BIT* 6 (1966), 310–316. https://doi.org/10.1007/BF01966091

Nico Naus, Freek Verbeek, Marc Schoolderman, and Binoy Ravindran. 2023. Low-Level Reachability Analysis Based on Formal Logic. In *TAP (Lecture Notes in Computer Science, Vol. 14066)*. Springer, 21–39. https://doi.org/10.1007/978-3-031-38828-6_2

Peter W. O'Hearn. 2020. Incorrectness logic. *Proc. ACM Program. Lang.* 4, POPL (2020), 10:1–10:32. https://doi.org/10.1145/3371078

David Michael Ritchie Park. 1969. Fixpoint Induction and Proofs of Program Properties. In *Machine Intelligence Volume 5*, Donald Mitchie and Bernard Meltzer (Eds.). Edinburgh Univ. Press, Chapter 3, 59–78.

David Michael Ritchie Park. 1979. On the Semantics of Fair Parallelism. In *Abstract Software Specifications (Lecture Notes in Computer Science, Vol. 86)*. Springer, 504–526. https://doi.org/10.1007/3-540-10007-5_47

Thomas Piecha and Peter Schroeder-Heister. 2019. General Proof Theory: Introduction. *Studia Logica* 107, 1 (2019), 1–5. https://doi.org/10.1007/s11225-018-9818-4

Gordon D. Plotkin. 1976. A Powerdomain Construction. *SIAM J. Comput.* 5, 3 (1976), 452–487. https://doi.org/10.1137/0205035

Gordon D. Plotkin. 1979. Dijkstra's Predicate Transformers & Smyth's Power Domaine. In *Abstract Software Specifications (Lecture Notes in Computer Science, Vol. 86)*. Springer, 527–553. https://doi.org/10.1007/3-540-10007-5_48

Gordon D. Plotkin. 2004a. The origins of structural operational semantics. *J. Log. Algebraic Methods Program.* 60-61 (2004), 3–15. https://doi.org/10.1016/J.JLAP.2004.03.009

Gordon D. Plotkin. 2004b. A structural approach to operational semantics. *J. Log. Algebraic Methods Program.* 60-61 (2004), 17–139.

Amir Pnueli. 1979. The Temporal Semantics of Concurrent Programs. In *Semantics of Concurrent Computation (Lecture Notes in Computer Science, Vol. 70)*. Springer, 1–20. https://doi.org/10.1007/BFB0022460

Christopher M. Poskitt and Detlef Plump. 2023. Monadic second-order incorrectness logic for GP 2. *J. Log. Algebraic Methods Program.* 130 (2023), 100825. https://doi.org/10.1016/J.JLAMP.2022.100825

Vaughan R. Pratt. 1976. Semantical Considerations on Floyd-Hoare Logic. In *FOCS*. IEEE Computer Society, 109–121. https://doi.org/10.1109/SFCS.1976.27

Azalea Raad, Josh Berdine, Hoang-Hai Dang, Derek Dreyer, Peter W. O'Hearn, and Jules Villard. 2020. Local Reasoning About the Presence of Bugs: Incorrectness Separation Logic. In *CAV (2) (Lecture Notes in Computer Science, Vol. 12225)*. Springer, 225–252. https://doi.org/10.1007/978-3-030-53291-8_14

Azalea Raad, Josh Berdine, Derek Dreyer, and Peter W. O'Hearn. 2022. Concurrent incorrectness separation logic. *Proc. ACM Program. Lang.* 6, POPL (2022), 1–29. https://doi.org/10.1145/3498695

Azalea Raad, Julien Vanegue, Josh Berdine, and Peter W. O'Hearn. 2023. A General Approach to Under-Approximate Reasoning About Concurrent Programs. In *CONCUR (LIPIcs, Vol. 279)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 25:1–25:17. https://doi.org/10.4230/LIPICS.CONCUR.2023.25

David A. Schmidt. 2007. Extracting Program Logics From Abstract Interpretations Defined by Logical Relations. In *MFPS (Electronic Notes in Theoretical Computer Science, Vol. 173)*. Elsevier, 339–356. https://doi.org/10.1016/J.ENTCS.2007.02.042

Dana S. Scott and Christopher Strachey. 1971. *Towards a Mathematical Semantics for Computer Languages*. Technical Report PRG-6. Oxford University Computer Laboratory. 49 pages. https://www.cs.ox.ac.uk/files/3228/PRG06.pdf

Ehud Y. Shapiro. 1982. Algorithmic Program Diagnosis. In *POPL*. ACM Press, 299–308. https://doi.org/10.1145/582153.582185

Michael B. Smyth. 1978. Power Domains. *J. Comput. Syst. Sci.* 16, 1 (1978), 23–36. https://doi.org/10.1016/0022-0000(78)90048-X

Stefan Sokolowski. 1977. Axioms for Total Correctness. *Acta Informatica* 9 (1977), 61–71. https://doi.org/10.1007/BF00263765

Vladimír Svoboda and Jaroslav Peregrin. 2016. Logically Incorrect Arguments. *Argumentation* 30 (2016), 263–287. https://doi.org/10.1007/s10503-015-9375-1

Alfred Tarski. 1933. The semantic conception of truth and the foundations of semantics. *Philosophy and Phenomenological Research* 4, 3 (1933), 341–376. https://doi.org/10.2307/2102968

Alfred Tarski. 1955. A Lattice Theoretical Fixpoint Theorem and Its Applications. *Pacific J. of Math.* 5 (1955), 285–310. https://doi.org/10.2140/pjm.1955.5.285

Alan Turing. 1949 [1950]. Checking a Large Routine. In *Report of a Conference on High Speed Automatic Calculating Machines*. University of Cambridge Mathematical Laboratory, Cambridge, England, 67–69. https://turingarchive.kings.cam.ac.uk/publications-lectures-and-talks-amtb/amt-b-8

Caterina Urban. 2013. The Abstract Domain of Segmented Ranking Functions. In *SAS (Lecture Notes in Computer Science, Vol. 7935)*. Springer, 43–62. https://doi.org/10.1007/978-3-642-38856-9_5

Caterina Urban. 2015. FuncTion: An Abstract Domain Functor for Termination - (Competition Contribution). In *TACAS (Lecture Notes in Computer Science, Vol. 9035)*. Springer, 464–466. https://doi.org/10.1007/978-3-662-46681-0_46

Caterina Urban, Arie Gurfinkel, and Temesghen Kahsai. 2016. Synthesizing Ranking Functions from Bits and Pieces. In *TACAS (Lecture Notes in Computer Science, Vol. 9636)*. Springer, 54–70. https://doi.org/10.1007/978-3-662-49674-9_4

Caterina Urban and Antoine Miné. 2014a. An Abstract Domain to Infer Ordinal-Valued Ranking Functions. In *ESOP (Lecture Notes in Computer Science, Vol. 8410)*. Springer, 412–431. https://doi.org/10.1007/978-3-642-54833-8_22

Caterina Urban and Antoine Miné. 2014b. A Decision Tree Abstract Domain for Proving Conditional Termination. In *SAS (Lecture Notes in Computer Science, Vol. 8723)*. Springer, 302–318. https://doi.org/10.1007/978-3-319-10936-7_19

Caterina Urban and Antoine Miné. 2015. Proving Guarantee and Recurrence Temporal Properties by Abstract Interpretation. In *VMCAI (Lecture Notes in Computer Science, Vol. 8931)*. Springer, 190–208. https://doi.org/10.1007/978-3-662-46081-8_11

Julien Vanegue. 2022. Adversarial Logic. In *SAS (Lecture Notes in Computer Science, Vol. 13790)*. Springer, 422–448. https://doi.org/10.1007/978-3-031-22308-2_19

John von Neumann. 1923. Zur Einführung der transfiniten Zahlen. *Acta Scientiarum Mathematicarum (Szeged)* 1, 4-4 (1923), 199–208. http://pub.acta.hu/acta/showCustomerArticle.action?id=4981&dataObjectType=article

Glynn Winskel. 1993. *The formal semantics of programming languages - an introduction.* MIT Press.

Peng Yan, Hanru Jiang, and Nengkun Yu. 2022. On incorrectness logic for Quantum programs. *Proc. ACM Program. Lang.* 6, OOPSLA1 (2022), 1–28. https://doi.org/10.1145/3527316

Cheng Zhang, Arthur Azevedo de Amorim, and Marco Gaboardi. 2022. On incorrectness logic and Kleene algebra with top and tests. *Proc. ACM Program. Lang.* 6, POPL (2022), 1–30. https://doi.org/10.1145/3498690

Linpeng Zhang and Benjamin Lucien Kaminski. 2022. Quantitative strongest post: a calculus for reasoning about the flow of quantitative information. *Proc. ACM Program. Lang.* 6, OOPSLA1 (2022), 1–29. https://doi.org/10.1145/3527331

Noam Zilberstein, Derek Dreyer, and Alexandra Silva. 2023. Outcome Logic: A Unifying Foundation for Correctness and Incorrectness Reasoning. *Proc. ACM Program. Lang.* 7, OOPSLA1 (2023), 522–550. https://doi.org/10.1145/3586045

Received 2023-07-11; accepted 2023-11-07

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

PATRICK COUSOT, Courant Institute of Mathematical Studies, New York University, USA

ACM Reference Format:
Patrick Cousot. 2024. Appendix of "Calculational Design of [In]Correctness Transformational Program Logics by Abstract Interpretation". *Proc. ACM Program. Lang.* 8, POPL, Article 7 (January 2024), 34 pages. https://doi.org/10.1145/3632849

A AUXILIARY MATERIAL FOR SECTION I.1 (RELATIONAL SEMANTICS)

A.1 Relations

The Cartesian product of sets $\mathcal{X}$ and $\mathcal{Y}$ is $\mathcal{X} \times \mathcal{Y} \triangleq \{\langle x, y\rangle \mid x \in \mathcal{X} \wedge y \in \mathcal{Y}\}$. The Cartesian power is $\mathcal{X}^n \triangleq \{\langle x_1, \ldots, x_n\rangle \mid \forall i \in [1,n] \text{ , } x_i \in \mathcal{X}\}$, $n \in \mathbb{N}$, with $\mathcal{X}^0 = \varnothing$ and $\mathcal{X}^1 = X$. The powerset $\wp(\mathcal{X})$ of a set $\mathcal{X}$ is the set of all subsets of $\mathcal{X}$. A relation r on sets $\mathcal{X}$ and $\mathcal{Y}$ is a set of pairs in their Cartesian product so $r \in \wp(\mathcal{X} \times \mathcal{Y})$. Its domain is $\mathsf{dom}(r) \triangleq \{x \mid \exists y \mathrel{.} \langle x, y\rangle \in r\}$ and its codomain is $\mathsf{codom}(r) \triangleq \{y \mid \exists x \mathrel{.} \langle x, y\rangle \in r\}$. The inverse of a relation r is $r^{-1} \triangleq \{\langle y, x\rangle \mid \langle x, y\rangle \in r\}$. The left composition of relations is $r_1 \mathbin{⨾} r2 \triangleq \{\langle x, z\rangle \mid \exists y \mathrel{.} \langle x, y\rangle \in r_1 \wedge \langle y, z\rangle \in r_2\}$. The composition of functions is $g \circ f \triangleq \boldsymbol{\lambda} x \bullet g(f(x))$ which, for their graphs, is the functional relation $\{\langle x, f(x)\rangle \mid x \in \mathsf{dom}(f)\} \mathbin{⨾} \{\langle y, g(y)\rangle \mid y \in \mathsf{dom}(g)\}$. A relation is functional if it is the graph of a total function. Therefore, the set of functional relations between $\mathcal{X}$ and $\mathcal{Y}$ is defined as the set of relations such that any element of their domain has a unique image in the codomain, that is, $\wp_{\mathsf{fun}}(\mathcal{X} \times \mathcal{Y}) \triangleq \{r \in \wp(\mathcal{X} \times \mathcal{Y}) \mid \forall x \in \mathcal{X} \mathrel{.} \exists y \in \mathcal{Y} \mathrel{.} \langle x, y\rangle \in r \wedge \forall y,z \in \mathcal{Y} \mathrel{.} \langle x, y\rangle \in r \wedge \langle x, z\rangle \in r \Rightarrow y = z\}$. We extend the definition of the left relation composition $⨾$ to nontermination $\perp$ by $r \mathbin{⨾} r' \triangleq \{\langle x, \perp\rangle \mid \langle x, \perp\rangle \in r\} \cup \{\langle x, y\rangle \mid (\exists z \in \mathcal{X} \setminus \{\perp\} \mathrel{.} \langle x, z\rangle \in r \wedge \langle z, y\rangle \in r'\}$. The conditional with several alternatives, à la C, is $(\!|\ldots \mathrel{?} \ldots \mathrel{[\!]} \ldots \mathrel{?} \ldots \mathrel{[\!]} \ldots \mathrel{:} \ldots|\!)$ where $[\!]$ is the optional "else if".

B AUXILIARY MATERIAL FOR SECTION I.2 (GALOIS CONNECTIONS)

Formally, a Galois connection $\langle C, \sqsubseteq\rangle \mathrel{\underset{\alpha}{\overset{\gamma}{\leftrightarrows}}} \langle A, \preceq\rangle$ is a pair $\langle \alpha, \gamma\rangle$ of functions between posets $\langle C, \sqsubseteq\rangle$ and $\langle A, \preceq\rangle$ satisfying $\forall x \in C \mathrel{.} \forall y \in A \mathrel{.} \alpha(x) \preceq y \Leftrightarrow x \sqsubseteq \gamma(y)$. We use a double headed arrow $\twoheadrightarrow$ to indicate surjection and $\twoheadleftarrow\!\!\twoheadrightarrow$ for bijections. We use classic properties of Galois connections for which proofs are found in [Cousot 2021, Ch. 11] and [Denecke et al. 2003]. In particular, the abstraction α preserves existing arbitrary joins (so in strict and increasing) and dually the concretization γ preserves existing arbitrary meets.

Proposition B.1. *The composition of Galois connections with Galois connections and isomorphisms is a Galois connection.*

Proof of (B.1). — Assume that $\langle \mathcal{X}, \sqsubseteq\rangle \mathrel{\underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}}} \langle \mathcal{Y}, \preceq\rangle$ and $\langle \mathcal{Y}, \preceq\rangle \mathrel{\underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}}} \langle \mathcal{Z}, \leq\rangle$. Then

$$\begin{array}{ll}
\quad \alpha_2 \circ \alpha_1(x) \leq y & \\
\Leftrightarrow \alpha_1(x) \preceq \gamma_2(y) & \text{(by } \langle \mathcal{Y}, \preceq\rangle \mathrel{\underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}}} \langle \mathcal{Z}, \leq\rangle\text{)} \\
\Leftrightarrow x \sqsubseteq \gamma_1 \circ \gamma_2(y) & \text{(by } \langle \mathcal{X}, \sqsubseteq\rangle \mathrel{\underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}}} \langle \mathcal{Y}, \preceq\rangle\text{)}
\end{array}$$

proving $\langle \mathcal{X}, \sqsubseteq\rangle \mathrel{\underset{\alpha_2 \circ \alpha_1}{\overset{\gamma_1 \circ \gamma_2}{\leftrightarrows}}} \langle \mathcal{Z}, \leq\rangle$.

— Assume that $\langle \mathcal{X}, =\rangle \mathrel{\underset{\alpha}{\overset{\alpha^{-1}}{\twoheadleftarrow\!\!\twoheadrightarrow}}} \langle \mathcal{Y}, =\rangle$ and $\langle \mathcal{Y}, \preceq\rangle \mathrel{\underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}}} \langle \mathcal{Z}, \leq\rangle$. Define the partial order $\langle \mathcal{X}, \sqsubseteq\rangle$ by $x \sqsubseteq y \Leftrightarrow \alpha(x) \preceq \alpha(y)$.

$$\begin{array}{ll}
\quad \alpha_2 \circ \alpha(x) \leq y & \\
\Leftrightarrow \alpha(x) \preceq \gamma_2(y) & \text{(by } \langle \mathcal{Y}, \preceq\rangle \mathrel{\underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}}} \langle \mathcal{Z}, \leq\rangle\text{)} \\
\Leftrightarrow \alpha(x) \preceq \alpha \circ \alpha^{-1} \circ \gamma_2(y) & \text{(by } \langle \mathcal{X}, =\rangle \mathrel{\underset{\alpha}{\overset{\alpha^{-1}}{\twoheadleftarrow\!\!\twoheadrightarrow}}} \langle \mathcal{Y}, =\rangle\text{)} \\
\Leftrightarrow x \sqsubseteq \alpha^{-1} \circ \gamma_2(y) & \text{(def. } \sqsubseteq\text{)}
\end{array}$$

proving $\langle \mathcal{X},\ \sqsubseteq\rangle \xleftrightarrow[\alpha_2 \circ \alpha]{\alpha^{-1} \circ \gamma_2} \langle \mathcal{Z},\ \leq\rangle$.

– The proof that $\langle \mathcal{X},\ \sqsubseteq\rangle \xleftrightarrow[\alpha_1]{\gamma_1} \langle \mathcal{Y},\ \leq\rangle$ and $\langle \mathcal{Y},\ =\rangle \xleftrightarrow[\alpha]{\alpha^{-1}} \langle \mathcal{Z},\ =\rangle$ implies $\langle \mathcal{X},\ \sqsubseteq\rangle \xleftrightarrow[\alpha \circ \alpha_1]{\gamma_1 \circ \alpha^{-1}} \langle \mathcal{Z},\ \leq\rangle$ where $x \leq y \Leftrightarrow \alpha^{-1}(x) \leq \alpha^{-1}(y)$ is similar. □

While a Galois connection $\langle C,\ \sqsubseteq\rangle \xleftrightarrow[\alpha]{\gamma} \langle A,\ \leq\rangle$ is appropriate for over approximation, a semidual Galois correspondence $\langle C,\ \sqsubseteq\rangle \xleftrightarrow[\alpha]{\gamma} \langle A,\ \geq\rangle$ (as originally defined by Évariste Galois for $\sqsubseteq \triangleq \leq \triangleq \subseteq$) is convenient for under approximation (as discussed e.g. by [Ascari et al. 2022]). The dual $\langle A,\ \geq\rangle \xleftrightarrow[\gamma]{\alpha} \langle C,\ \sqsupseteq\rangle$ may also be useful (e.g. to approximate greatest rather that least fixpoints).

By Prop. B.1, the composition of Galois connections (with corresponding partial orders) is a Galois connection. We will compose Galois connections to show that all known transformational logics are abstractions of the natural relation semantics.

C AUXILIARY MATERIAL FOR SECTION I.3 (THE DESIGN OF A NATURAL TRANSFORMATIONAL LOGIC THEORY BY COMPOSING ABSTRACTIONS OF THE NATURAL RELATIONAL SEMANTICS)

Proof of (7).

$$
\begin{array}{ll}
\quad \alpha_{\mathrm{G}}(f) = r & \wr\text{assuming } r \in \wp_{\mathsf{fun}}(X \times Y)\wr \\
\Leftrightarrow \{\langle x,\ f(x)\rangle \mid x \in X\} = r & \wr\text{def. } \alpha_{\mathrm{G}}(f)\wr \\
\Leftrightarrow \forall x \in X\ .\ f(x) = (y \textit{ such that } \langle x,\ y\rangle \in r) & \wr r \text{ is a functional relation}\wr \\
\Leftrightarrow f = \boldsymbol{\lambda} x \bullet (y \textit{ such that } \langle x,\ y\rangle \in r) & \wr\text{def. function equality}\wr \\
\Leftrightarrow f = \gamma_{\mathrm{G}}(r) & \wr\text{def. } \gamma_{\mathrm{G}}(r)\wr \quad \square
\end{array}
$$

Proof of (8).

$$
\begin{array}{ll}
\quad \alpha_{\mathrm{G}}(\mathsf{Post}(\alpha_C(\{\!|\mathsf{S}|\!\}_\perp))) & \\
= \alpha_{\mathrm{G}}(\mathsf{Post}([\![\mathsf{S}]\!]_\perp)) & \wr\text{from Sect. I.3.1}\wr \\
= \{\langle P,\ \mathsf{Post}([\![\mathsf{S}]\!]_\perp)P\rangle \mid P \in \wp(\Sigma)\} & \wr\text{def. } \alpha_{\mathrm{G}}\wr \\
= \{\langle P,\ \{\langle \sigma_0,\ \sigma'\rangle \mid \exists \sigma\ .\ \langle \sigma_0,\ \sigma\rangle \in P \wedge \langle \sigma,\ \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\}\rangle \mid P \in \wp(\Sigma)\} & \wr\text{def. (6) of Post}\wr \quad \square
\end{array}
$$

Example C.1. Continuing example I.3.3, instead of Floyd's verification condition $\{P\}\mathsf{S}\{\mathsf{post}[\![\mathsf{S}]\!]P\}$ [Floyd 1967], Hoare logic [Hoare 1969] uses $\{\widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]Q\}\mathsf{S}\{Q\}$ that is, $\{\{\sigma \mid \sigma[\mathsf{x} \leftarrow \mathcal{A}[\![\mathsf{A}]\!]\sigma \in Q\}\}\mathsf{S}\{Q\}$, in syntactic form $\{Q[\mathsf{A}/\mathsf{x}]\}\mathsf{S}\{Q\}$ where $Q[\mathsf{A}/\mathsf{x}]$ is Q where A is substituted for x. By (12), the two verification conditions for assignment are equivalent. ■

Proof of (13).

$$
\begin{array}{ll}
\quad \mathsf{post}(\sqsupseteq,\subseteq)(\alpha_{\mathrm{G}}(\mathsf{Post}([\![\mathsf{S}]\!]_\perp))) & \wr\text{def. (13) of consequence}\wr \\
= \mathsf{post}(\sqsupseteq,\subseteq)(\{\langle P,\ \mathsf{Post}([\![\mathsf{S}]\!]_\perp)P\rangle \mid P \in \wp(\Sigma \times \Sigma)\}) & \wr\text{def. } \alpha_{\mathrm{G}}\wr \\
= \mathsf{post}(\sqsupseteq,\subseteq)(\{\langle P,\ \{\langle \sigma_0,\ \sigma'\rangle \mid \exists \sigma\ .\ \langle \sigma_0,\ \sigma\rangle \in P \wedge \langle \sigma,\ \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\}\rangle \mid P \in \wp(\Sigma \times \Sigma)\}) & \wr\text{def. (6) of Post}\wr \\
= \{y \mid \exists x \in \{\langle P,\ \{\langle \sigma_0,\ \sigma'\rangle \mid \exists \sigma\ .\ \langle \sigma_0,\ \sigma\rangle \in P \wedge \langle \sigma,\ \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\}\rangle \mid P \in \wp(\Sigma \times \Sigma)\}\ .\ \langle x,\ y\rangle \in (\sqsupseteq,\subseteq)\} & \\
 & \wr\text{def. (10) of post}\wr \\
= \{\langle P',\ Q'\rangle \mid \exists \langle P,\ Q\rangle \in \{\langle P,\ \{\langle \sigma_0,\ \sigma'\rangle \mid \exists \sigma\ .\ \langle \sigma_0,\ \sigma\rangle \in P \wedge \langle \sigma,\ \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\}\rangle \mid P \in \wp(\Sigma \times \Sigma)\}\ .\ \langle\langle P,\ Q\rangle,\ \langle P', & \\
\quad Q'\rangle\rangle \in (\sqsupseteq,\subseteq)\} & \\
 & \wr y = \langle P',\ Q'\rangle \in \wp(\Sigma \times \Sigma) \times \wp(\Sigma \times \Sigma_\perp) \text{ and } x = \langle P,\ Q\rangle \in \wp(\Sigma \times \Sigma) \times \wp(\Sigma \times \Sigma_\perp)\ \wr
\end{array}
$$

$= \{\langle P', Q'\rangle \mid \exists\langle P, Q\rangle \in \{\langle P, \{\langle\sigma_0, \sigma'\rangle \mid \exists\sigma \,.\, \langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\}\rangle \mid P \in \wp(\Sigma\times\Sigma)\} \,.\, P \supseteq P' \wedge Q \subseteq Q'\}$ $\wr$def. $\supseteq, \subseteq\wr$

$= \{\langle P', Q'\rangle \mid \exists P \in \wp(\Sigma\times\Sigma) \,.\, P' \subseteq P \wedge \{\langle\sigma_0, \sigma'\rangle \mid \exists\sigma \,.\, \langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\} \subseteq Q'\}$ $\wr$def. $\in\wr$

$= \{\langle P', Q'\rangle \mid \exists P \in \wp(\Sigma\times\Sigma) \,.\, P' \subseteq P \wedge \forall\sigma_0,\sigma,\sigma' \,.\, (\langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp) \Rightarrow (\langle\sigma_0, \sigma'\rangle \in Q')\}$ $\wr$def. $\subseteq\wr$

$= \{\langle P', Q'\rangle \mid \forall\sigma_0,\sigma,\sigma' \,.\, (\langle\sigma_0, \sigma\rangle \in P' \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp) \Rightarrow (\langle\sigma_0, \sigma'\rangle \in Q')$

$\wr$($\subseteq$) If $\langle\sigma_0, \sigma\rangle \in P'$, then $\langle\sigma_0, \sigma\rangle \in P$ since $P' \subseteq P$ which implies $\langle\sigma_0, \sigma'\rangle \in Q'$ by hypothesis. Otherwise, $\langle\sigma_0, \sigma\rangle \notin P'$, and false $\Rightarrow (\langle\sigma_0, \sigma'\rangle \in Q')$.
($\supseteq$) Take $P = P'$ and reflexivity$\wr$

$= \{\langle P, Q\rangle \mid \forall\langle\sigma_0, \sigma'\rangle \,.\, (\exists\sigma \,.\, \langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp) \Rightarrow (\langle\sigma_0, \sigma'\rangle \in Q)\}$ $\wr$def. $\Rightarrow$, renaming$\wr$

$= \{\langle P, Q\rangle \mid \{\langle\sigma_0, \sigma'\rangle \mid \exists\sigma \,.\, \langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\} \subseteq Q\}$ $\wr$def. $\subseteq\wr$

$= \{\langle P, Q\rangle \mid \mathsf{Post}([\![\mathsf{S}]\!]_\perp)P \subseteq Q\}$ $\wr$def. (6) of Post$\wr$ □

Proof of (14).

$\mathsf{post}(\subseteq,\supseteq)(\alpha_{\mathrm{G}}(\mathsf{Post}([\![\mathsf{S}]\!]_\perp)))$ $\wr$def. (13) of consequence$\wr$

$= \mathsf{post}(\subseteq,\supseteq)(\{\langle P, \mathsf{Post}([\![\mathsf{S}]\!]_\perp)P\rangle \mid P \in \wp(\Sigma\times\Sigma)\})$ $\wr$def. $\alpha_{\mathrm{G}}\wr$

$= \mathsf{post}(\subseteq,\supseteq)(\{\langle P, \{\langle\sigma_0, \sigma'\rangle \mid \exists\sigma \,.\, \langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\}\rangle \mid P \in \wp(\Sigma\times\Sigma)\})$ $\wr$def. (6) of Post$\wr$

$= \{y \mid \exists x \in \{\langle P, \{\langle\sigma_0, \sigma'\rangle \mid \exists\sigma \,.\, \langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\}\rangle \mid P \in \wp(\Sigma\times\Sigma)\} \,.\, \langle x, y\rangle \in (\subseteq,\supseteq)\}$ $\wr$def. (10) of post$\wr$

$= \{\langle P', Q'\rangle \mid \exists\langle P, Q\rangle \in \{\langle P, \{\langle\sigma_0, \sigma'\rangle \mid \exists\sigma \,.\, \langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\}\rangle \mid P \in \wp(\Sigma\times\Sigma)\} \,.\, \langle\langle P, Q\rangle, \langle P', Q'\rangle\rangle \in (\subseteq,\supseteq)\}$
$\wr y = \langle P', Q'\rangle \in \wp(\Sigma\times\Sigma)\times\wp(\Sigma\times\Sigma_\perp)$ and $x = \langle P, Q\rangle \in \wp(\Sigma\times\Sigma)\times\wp(\Sigma\times\Sigma_\perp)\wr$

$= \{\langle P', Q'\rangle \mid \exists\langle P, Q\rangle \in \{\langle P, \{\langle\sigma_0, \sigma'\rangle \mid \exists\sigma \,.\, \langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\}\rangle \mid P \in \wp(\Sigma\times\Sigma)\} \,.\, P \subseteq P' \wedge Q \supseteq Q'\}$ $\wr$def. $\supseteq, \subseteq\wr$

$= \{\langle P', Q'\rangle \mid \exists P \in \wp(\Sigma\times\Sigma) \,.\, P \subseteq P' \wedge \{\langle\sigma_0, \sigma'\rangle \mid \exists\sigma \,.\, \langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\} \supseteq Q'\}$ $\wr$def. $\in\wr$

$= \{\langle P', Q'\rangle \mid \exists P \in \wp(\Sigma\times\Sigma) \,.\, P \subseteq P' \wedge \forall\langle\sigma_0, \sigma'\rangle \,.\, (\langle\sigma_0, \sigma'\rangle \in Q') \Rightarrow (\exists\sigma \,.\, \langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp)\}$ $\wr$def. $\subseteq\wr$

$= \{\langle P', Q'\rangle \mid \forall\langle\sigma_0, \sigma'\rangle \,.\, (\langle\sigma_0, \sigma'\rangle \in Q') \Rightarrow (\exists\sigma \,.\, \langle\sigma_0, \sigma\rangle \in P' \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp)\}$

$\wr$($\subseteq$) $\langle\sigma_0, \sigma\rangle \in P$ and $P \subseteq P'$ implies $\langle\sigma_0, \sigma\rangle \in P'$
($\supseteq$) Take $P' = P$ and reflexivity$\wr$

$= \{\langle P, Q\rangle \mid Q \subseteq \{\langle\sigma_0, \sigma'\rangle \mid \exists\sigma \,.\, \langle\sigma_0, \sigma\rangle \in P \wedge \langle\sigma, \sigma'\rangle \in [\![\mathsf{S}]\!]_\perp\}\}$ $\wr$def. $\subseteq$ and renaming$\wr$

$= \{\langle P, Q\rangle \mid Q \subseteq \mathsf{Post}([\![\mathsf{S}]\!]_\perp)P\}$ $\wr$def. (6) of Post$\wr$ □

Proof of (16).

$\alpha^2_{\not\perp}(R) \subseteq R'$ $\wr R \in \wp(\wp(\Sigma\times\Sigma)\times\wp(\Sigma\times\Sigma_\perp))$ and $R' \in \wp(\wp(\Sigma\times\Sigma)\times\wp(\Sigma\times\Sigma))$ by hypothesis$\wr$

$\Leftrightarrow \{\langle P, Q\rangle \mid \langle P, Q\rangle \in R \wedge Q \cap (\Sigma\times\{\perp\}) = \varnothing\} \subseteq R'$ $\wr$def. (15) of $\alpha^2_{\not\perp}\wr$

$\Leftrightarrow \forall P,Q \,.\, (\langle P, Q\rangle \in R \wedge Q \cap (\Sigma\times\{\perp\}) = \varnothing) \Rightarrow (\langle P, Q\rangle \in R')$ $\wr$def. subset $\subseteq\wr$

$\Leftrightarrow \forall P,Q \,.\, (\langle P, Q\rangle \in R) \Rightarrow ((Q \cap (\Sigma\times\{\perp\}) = \varnothing) \Rightarrow (\langle P, Q\rangle \in R'))$ $\wr$def. implication $\Rightarrow\wr$

$\Leftrightarrow R \subseteq \{\langle P, Q\rangle \mid (Q \cap (\Sigma\times\{\perp\}) = \varnothing) \Rightarrow (\langle P, Q\rangle \in R')\}$ $\wr$def. subset $\subseteq\wr$

$\Leftrightarrow R \subseteq R' \cup \{\langle P, Q\rangle \mid Q \cap (\Sigma\times\{\perp\}) \neq \varnothing\}$

$\wr$($\Rightarrow$) Assume $\langle P, Q\rangle \in R$. If $Q \subseteq \Sigma\times\Sigma$ then $Q \cap (\Sigma\times\{\perp\}) = \varnothing$ so $\langle P, Q\rangle \in R'$ by hypothesis. Otherwise $Q \cap (\Sigma\times\{\perp\}) \neq \varnothing$ so $(Q \cap (\Sigma\times\{\perp\}) = \varnothing) \Rightarrow (\langle P, Q\rangle \in R')$ is true, in which case $R \subseteq \{\langle P, Q\rangle \mid Q \cap (\Sigma\times\{\perp\}) \neq \varnothing\}$;
($\Leftarrow$) If $\langle P, Q\rangle \in R$ and $Q \cap (\Sigma\times\{\perp\}) = \varnothing$ then $\langle P, Q\rangle \in R'$ $\wr$

$\Leftrightarrow R \subseteq \gamma^2_{\not\perp}(R')$ $\wr$def. $\gamma^2_{\not\perp}\wr$ $\square$

Proof of (17).

$$\begin{aligned}
& \alpha^2_{\not\perp}(\mathsf{post}(\supseteq,\subseteq)(\alpha_{\mathrm{G}}(\mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)))) \\
= {} & \alpha^2_{\not\perp}(\{\langle P,\ Q\rangle \mid \mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)P \subseteq Q\}) && \wr\text{by (13)}\wr \\
= {} & \{\langle P,\ Q\rangle \mid \langle P,\ Q\rangle \in \{\langle P,\ Q\rangle \mid \mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)P \subseteq Q \wedge Q \cap (\Sigma \times \{\perp\}) = \varnothing\}\} && \wr\text{def. (15) of } \alpha^2_{\not\perp}\wr \\
= {} & \{\langle P,\ Q\rangle \mid \mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)P \subseteq Q \wedge Q \cap (\Sigma \times \{\perp\}) = \varnothing\} && \wr\text{def. } \in\wr \\
= {} & \{\langle P,\ Q \setminus (\Sigma \times \{\perp\})\rangle \mid \mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)P \subseteq Q \setminus (\Sigma \times \{\perp\})\} && \wr\text{since } Q \in \wp(\Sigma \times \Sigma \cup \{\perp\})\wr \quad \square
\end{aligned}$$

Proof of (19).

$$\begin{aligned}
& \alpha^2_{\perp}(R) \subseteq R' \\
\Leftrightarrow {} & \{\langle P,\ Q \cup (\Sigma \times \{\perp\})\rangle \mid \langle P,\ Q\rangle \in R\} \subseteq R' && \wr\text{def. (18) of } \alpha^2_{\perp}\wr \\
\Leftrightarrow {} & \forall \langle P,\ Q\rangle \in R \,.\, \langle P,\ Q \cup (\Sigma \times \{\perp\})\rangle \in R' && \wr\text{def. subset } \subseteq\wr \\
\Leftrightarrow {} & R \subseteq \{\langle P,\ Q\rangle \mid \langle P,\ Q \cup (\Sigma \times \{\perp\})\rangle \in R'\} && \wr\text{def. subset } \subseteq\wr \\
\Leftrightarrow {} & R \subseteq \gamma^2_{\perp}(R') && \wr\text{def. concretization } \gamma^2_{\perp}\wr \quad \square
\end{aligned}$$

Proof of (20).

$$\begin{aligned}
& \alpha^2_{\perp}(\mathsf{post}(\supseteq,\subseteq)(\alpha_{\mathrm{G}}(\mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)))) \\
= {} & \alpha^2_{\perp}(\{\langle P,\ Q\rangle \mid \mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)P \subseteq Q\}) && \wr\text{by (13)}\wr \\
= {} & \{\langle P,\ Q \cup (\Sigma \times \{\perp\})\rangle \mid \langle P,\ Q\rangle \in \{\langle P,\ Q\rangle \mid \mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)P \subseteq Q\}\} && \wr\text{def. (18) of } \alpha^2_{\perp}\wr \\
= {} & \{\langle P,\ Q \cup (\Sigma \times \{\perp\})\rangle \mid \mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)P \subseteq Q\} && \wr\text{def. } \in\wr \quad \square
\end{aligned}$$

Proof of (21).

$$\begin{aligned}
& \alpha^{2\,\prime}_{\perp}(R) \subseteq R' \\
\Leftrightarrow {} & \{\langle P,\ Q \cap (\Sigma \times \Sigma)\rangle \mid \langle P,\ Q\rangle \in R\} \subseteq R' && \wr\text{def. (22) of } \alpha^{2\,\prime}_{\perp}\wr \\
\Leftrightarrow {} & \forall \langle P,\ Q\rangle \in R \,.\, \langle P,\ Q \cap (\Sigma \times \Sigma)\rangle \in R' && \wr\text{def. } \subseteq\wr \\
\Leftrightarrow {} & R \subseteq \{\langle P,\ Q\rangle \mid \langle P,\ Q \cap (\Sigma \times \Sigma)\rangle \in R'\} && \wr\text{def. } \subseteq\wr \\
\Leftrightarrow {} & R \subseteq \gamma^{2\,\prime}_{\perp}(R') && \wr\text{def. (22) of } \gamma^{2\,\prime}_{\perp}\wr \quad \square
\end{aligned}$$

Proof of (22).

$$\begin{aligned}
& \alpha^{2\,\prime}_{\perp}(\mathsf{post}(\supseteq,\subseteq)(\alpha_{\mathrm{G}}(\mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)))) \\
= {} & \alpha^{2\,\prime}_{\perp}(\{\langle P,\ Q\rangle \mid \mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)P \subseteq Q\}) && \wr\text{by (13)}\wr \\
= {} & \{\langle P,\ Q \cap (\Sigma \times \Sigma)\rangle \mid \langle P,\ Q\rangle \in \{\langle P,\ Q\rangle \mid \mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)P \subseteq Q\}\} && \wr\text{def. (22) of } \alpha^{2\,\prime}_{\perp}\wr \\
= {} & \{\langle P,\ Q \cap (\Sigma \times \Sigma)\rangle \mid \mathsf{Post}(\llbracket \mathsf{S}\rrbracket_\perp)P \subseteq Q\} && \wr\text{def. } \in\wr \quad \square
\end{aligned}$$

Proof of (23).

$$\begin{aligned}
- \ & \alpha_{\downarrow^2}(P) \subseteq Q \\
\Leftrightarrow {} & \{\sigma \mid \exists \sigma_0 \,.\, \langle \sigma_0,\ \sigma\rangle \in P\} \subseteq Q && \wr\text{def. (24) of } \alpha_{\downarrow^2}\wr \\
\Leftrightarrow {} & \forall \sigma \,.\, (\exists \sigma_0 \,.\, \langle \sigma_0,\ \sigma\rangle \in P) \Rightarrow (\sigma \in Q) && \wr\text{def. } \subseteq\wr \\
\Leftrightarrow {} & \forall \sigma \,.\, \forall \sigma_0 \,.\, (\langle \sigma_0,\ \sigma\rangle \in P) \Rightarrow (\sigma \in Q) && \wr\text{def. } \Rightarrow\wr
\end{aligned}$$

$\Leftrightarrow \forall \sigma_0 \,.\, \forall \sigma \,.\, (\langle \sigma_0,\, \sigma\rangle \in P) \Rightarrow (\langle \sigma_0,\, \sigma\rangle \in \mathcal{X} \times Q)$ $\quad$ ≀def. $\forall$ and $\sigma_0 \in \mathcal{X}$≀

$\Leftrightarrow P \subseteq (\mathcal{X} \times Q)$ $\quad$ ≀def. $\subseteq$≀

$\Leftrightarrow P \subseteq \gamma_{\downarrow^2}(Q)$ $\quad$ ≀def. (24) of $\gamma_{\downarrow^2}$≀

— $\dot{\alpha}_{\downarrow^2}(R) \subseteq R'$

$\Leftrightarrow \{\langle \alpha_{\downarrow^2}(P),\, \alpha_{\downarrow^2}(Q)\rangle \mid \langle P,\, Q\rangle \in R\} \subseteq R'$ $\quad$ ≀def. $\dot{\alpha}_{\downarrow^2}$≀

$\Leftrightarrow \{\langle\{\sigma' \mid \exists \sigma \,.\, \langle \sigma,\, \sigma'\rangle \in p\},\, \{\sigma' \mid \exists \sigma \,.\, \langle \sigma,\, \sigma'\rangle \in Q\}\rangle \mid \langle P,\, Q\rangle \in R'\} \subseteq R'$ $\quad$ ≀def. $\alpha_{\downarrow^2}$≀

$\Leftrightarrow \forall \langle P,\, Q\rangle \in R \,.\, \langle\{\sigma' \mid \exists \sigma \,.\, \langle \sigma,\, \sigma'\rangle \in P\},\, \{\sigma' \mid \exists \sigma \,.\, \langle \sigma,\, \sigma'\rangle \in Q\}\rangle \in R'$ $\quad$ ≀def. $\subseteq$≀

$\Leftrightarrow R \subseteq \{\langle P,\, Q\rangle \mid \langle\{\sigma' \mid \exists \sigma \,.\, \langle \sigma,\, \sigma'\rangle \in P\} \in R',\, \{\sigma' \mid \exists \sigma \,.\, \langle \sigma,\, \sigma'\rangle \in Q\} \in R'\rangle$ $\quad$ ≀def. $\subseteq$≀

$\Leftrightarrow R \subseteq \{\langle P,\, Q\rangle \mid \exists \langle P',\, Q'\rangle \in' R \,.\, P' = \{\sigma' \mid \exists \sigma \,.\, \langle \sigma,\, \sigma'\rangle \in P\} \wedge Q' = \{\sigma' \mid \exists \sigma \,.\, \langle \sigma,\, \sigma'\rangle \in Q\}\}$ $\quad$ ≀def. $\in$≀

$\Leftrightarrow R \subseteq \{\langle \Sigma \times P',\, \Sigma \times Q'\rangle \mid \langle P',\, Q'\rangle \in R'\}$ $\quad$ ≀since $P = \Sigma \times P'$ by $\sigma \in \Sigma$, same for Q≀

$\Leftrightarrow R \subseteq \{\langle \gamma_{\downarrow^2}(P'),\, \gamma_{\downarrow^2}(Q')\rangle \mid \langle P',\, Q'\rangle \in R'\}$ $\quad$ ≀def. $\gamma_{\downarrow^2}$≀

$\Leftrightarrow R \subseteq \dot{\gamma}_{\downarrow^2}(R')$ $\quad$ ≀def. $\dot{\gamma}_{\downarrow^2}$≀ □

Proof of (28). We write $\subseteq/=$ to either stand for $\subseteq$ (in the Galois connection proof) or $=$ (in the isomorphism proof), everywhere in the proof.

— $\alpha^{-1}(r) \subseteq/= r'$

$\Leftrightarrow r^{-1} \subseteq/= r'$ $\quad$ ≀def. (27) of α^{-1}≀

$\Leftrightarrow (r^{-1})^{-1} \subseteq/= r'^{-1}$ $\quad$ ≀def. inverse $^{-1}$≀

$\Leftrightarrow r \subseteq/= r'^{-1}$ $\quad$ ≀def. inverse $^{-1}$≀

$\Leftrightarrow r \subseteq/= \alpha^{-1}(r')$ $\quad$ ≀def. (27) of α^{-1}≀

— $\dot{\alpha}^{-1}(f) \mathrel{\dot{\subseteq}/=} g$

$\Leftrightarrow \forall x \,.\, \dot{\alpha}^{-1}(f)(x) \subseteq/= g(x)$ $\quad$ ≀pointwise def. $\dot{\subseteq}$≀

$\Leftrightarrow \forall x \,.\, \alpha^{-1}(f(\alpha^{-1}(x))) \subseteq/= g(x)$ $\quad$ ≀def. (27) of $\dot{\alpha}^{-1}$≀

$\Leftrightarrow \forall y \,.\, \alpha^{-1}(f(\alpha^{-1}(\alpha^{-1}(y)))) \subseteq/= g(\alpha^{-1}(y))$

≀($\subseteq$) letting $x = \alpha^{-1}(y)$
($\supseteq$) $y = \alpha^{-1}(x)$, Galois isomorphism (28) for α^{-1}≀

$\Leftrightarrow \forall y \,.\, f(y) \subseteq/= \alpha^{-1}(g(\alpha^{-1}(y)))$ $\quad$ ≀Galois isomorphism (28) for α^{-1}≀

$\Leftrightarrow \forall x \,.\, f(x) \subseteq/= \dot{\alpha}^{-1}(g)(x)$ $\quad$ ≀pointwise def. (27) of $\dot{\alpha}^{-1}$≀

$\Leftrightarrow f \mathrel{\dot{\subseteq}/=} \dot{\alpha}^{-1}(g)$ $\quad$ ≀pointwise def. $\dot{\subseteq}$≀

— $\vec{\alpha}^{-1}(T) \mathrel{\ddot{\subseteq}/=} T'$

$\Leftrightarrow \dot{\alpha}^{-1} \circ T \circ \alpha^{-1} \mathrel{\ddot{\subseteq}/=} T'$ $\quad$ ≀def. (28) of $\vec{\alpha}^{-1}$≀

$\Leftrightarrow \forall r \,.\, \dot{\alpha}^{-1}(T(\alpha^{-1}(r))) \mathrel{\dot{\subseteq}/=} T'(r)$ $\quad$ ≀pointwise def. $\ddot{\subseteq}$ and def. function composition $\circ$≀

$\Leftrightarrow \forall r' \,.\, \dot{\alpha}^{-1}(T(\alpha^{-1}(\alpha^{-1}(r')))) \mathrel{\dot{\subseteq}/=} T'(\alpha^{-1}(r'))$ $\quad$ ≀letting $r = \alpha^{-1}(r')$≀

$\Leftrightarrow \forall r' \,.\, \dot{\alpha}^{-1}(T(r')) \mathrel{\dot{\subseteq}/=} T'(\alpha^{-1}(r'))$ $\quad$ ≀Galois isomorphism (28) for α^{-1}≀

$\Leftrightarrow \forall r' \,.\, \forall Q \,.\, \dot{\alpha}^{-1}(T(r'))Q \subseteq/= T'(\alpha^{-1}(r'))Q$ $\quad$ ≀pointwise def. $\dot{\subseteq}/=$≀

$\Leftrightarrow \forall r' \,.\, \forall Q \,.\, \alpha^{-1}(T(r')Q) \subseteq/= T'(\alpha^{-1}(r'))Q$ $\quad$ ≀pointwise def. (27) of $\dot{\alpha}^{-1}$≀

$\Leftrightarrow \forall r' \,.\, \forall Q \,.\, T(r')Q \subseteq/= \alpha^{-1}(T'(\alpha^{-1}(r'))Q)$ $\quad$ ≀Galois isomorphism (27) for α^{-1}≀

$\Leftrightarrow \forall r' \,.\, \forall Q \,.\, T(r')Q \subseteq/= \dot{\alpha}^{-1}(T'(\alpha^{-1}(r')))Q$ $\quad$ ≀def. (27) of $\dot{\alpha}^{-1}$≀

$\Leftrightarrow \forall r' \,.\, T(r') \mathrel{\dot{\subseteq}/=} \dot{\alpha}^{-1}(T'(\alpha^{-1}(r')))Q$ $\quad$ ≀pointwise def. $\dot{\subseteq}$≀

$\Leftrightarrow T \mathrel{\ddot{\subseteq}/=} \dot{\alpha}^{-1} \circ T' \circ \alpha^{-1}$ ⟅def. fonction composition $\circ$ and pointwise def. $\ddot{\subseteq}$⟆

$\Leftrightarrow T \mathrel{\ddot{\subseteq}/=} \vec{\alpha}^{-1}(T')$ ⟅def. (28) of $\vec{\alpha}^{-1}$⟆ □

Proof of (29).

$\vec{\alpha}^{-1}(\mathsf{Post})$

$= \dot{\alpha}^{-1} \circ \mathsf{Post} \circ \alpha^{-1}$ ⟅def. (28) of $\vec{\alpha}^{-1}$⟆

$= \boldsymbol{\lambda} r \bullet \dot{\alpha}^{-1}(\mathsf{Post}(\alpha^{-1}(r)))$ ⟅def. function composition $\circ$⟆

$= \boldsymbol{\lambda} r \bullet \dot{\alpha}^{-1}(\mathsf{Post}(r^{-1}))$ ⟅def. (27) of α^{-1}⟆

$= \boldsymbol{\lambda} r \bullet \dot{\alpha}^{-1}(\boldsymbol{\lambda} P \bullet \{\langle \sigma_0, \sigma' \rangle \mid \exists \sigma \,.\, \langle \sigma_0, \sigma \rangle \in P \wedge \langle \sigma, \sigma' \rangle \in r^{-1}\})$ ⟅def. (6) of Post⟆

$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \alpha^{-1}(\{\langle \sigma_0, \sigma' \rangle \mid \exists \sigma \,.\, \langle \sigma, \sigma_0 \rangle \in \alpha^{-1}(P) \wedge \langle \sigma, \sigma' \rangle \in r^{-1}\})$ ⟅pointwise def. of $\dot{\alpha}^{-1}$, def. α^{-1}⟆

$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \alpha^{-1}(\{\langle \sigma_0, \sigma' \rangle \mid \exists \sigma \,.\, \langle \sigma, \sigma_0 \rangle \in \alpha^{-1}(P) \wedge \langle \sigma', \sigma \rangle \in r)\})$ ⟅def. inverse r^{-1}⟆

$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \alpha^{-1}(\{\langle \sigma_0, \sigma' \rangle \mid \exists \sigma \,.\, \langle \sigma', \sigma \rangle \in r \wedge \langle \sigma, \sigma_0 \rangle \in \alpha^{-1}(P))\})$ ⟅$\wedge$ commutative⟆

$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \alpha^{-1}(\{\langle \sigma_f, \sigma' \rangle \mid \exists \sigma \,.\, \langle \sigma', \sigma \rangle \in r \wedge \langle \sigma, \sigma_f \rangle \in \alpha^{-1}(P))\})$ ⟅renaming σ_0 into σ_f⟆

$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \alpha^{-1}(\{\langle \sigma_f, \sigma \rangle \mid \exists \sigma' \,.\, \langle \sigma, \sigma' \rangle \in r \wedge \langle \sigma', \sigma_f \rangle \in \alpha^{-1}(P))\})$

⟅renaming σ into σ' and vice-versa⟆

$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \{\langle \sigma, \sigma_f \rangle \mid \exists \sigma' \,.\, \langle \sigma, \sigma' \rangle \in r \wedge \langle \sigma', \sigma_f \rangle \in Q)\}$ ⟅def. (27) of α^{-1}⟆

$= \mathsf{Pre}$ ⟅def. (29) of Pre⟆ □

Proof of (32).

— $\alpha^{\vec{\varnothing}}(\dot{\bigcup\limits_{i \in \Delta}} \tau_i)$

$= \{\langle P, Q \rangle \mid (\dot{\bigcup\limits_{i \in \Delta}} \tau_i)(P) \subseteq \neg Q\}$

⟅def. (31) of $\alpha^{\vec{\varnothing}}(\tau) = \{\langle P, Q \rangle \mid Q \cap \tau(P) = \varnothing\} = \{\langle P, Q \rangle \mid \tau(P) \subseteq \neg Q\}$⟆

$= \{\langle P, Q \rangle \mid (\bigcup\limits_{i \in \Delta} \tau_i(P)) \subseteq \neg Q\}$ ⟅pointwise def. $\dot{\bigcup}$⟆

$= \{\langle P, Q \rangle \mid \forall i \in \Delta \,.\, \tau_i(P) \subseteq \neg Q\}$ ⟅pointwise def. $\bigcup$ and $\subseteq$⟆

$= \bigcap\limits_{i \in \Delta} \{\langle P, Q \rangle \mid \tau_i(P) \subseteq \neg Q\}$ ⟅def. $\bigcap$⟆

$= \bigcap\limits_{i \in \Delta} \alpha^{\vec{\varnothing}}(\tau_i)$ ⟅def. (31) of $\alpha^{\vec{\varnothing}}$⟆

— The other $\alpha^{\overleftarrow{\varnothing}}$, $\alpha^{\overleftarrow{\varnothing}}$, and $\alpha^{\overleftarrow{\otimes}}$ follow from proposition B.1 by composition of the Galois connection (32) for $\alpha^{\vec{\varnothing}}$ and those (30) for $\alpha^{\neg}$ and (26) for α^{-1}. □

Proof of (33).

— $\alpha^{\sim}(f) \mathrel{\dot{\subseteq}/=} g$

$\Leftrightarrow \forall X \in \wp(\mathcal{X}) \,.\, \alpha^{\sim}(f)X \subseteq/= g(X)$ ⟅pointwise def. $\dot{\subseteq}$ or $=$⟆

$\Leftrightarrow \forall X \in \wp(\mathcal{X}) \,.\, \neg \circ f \circ \neg(X) \subseteq/= g(X)$ ⟅def. $\alpha^{\sim}$⟆

$\Leftrightarrow \forall X \in \wp(\mathcal{X}) \,.\, f \circ \neg(X) \supseteq/= \neg g(X)$ ⟅$A \subseteq B$ iff $\neg B \subseteq \neg A$ i.e. $\neg A \supseteq \neg B$ and $A = B \Leftrightarrow \neg A = \neg B$⟆

$\Leftrightarrow \forall Y \in \wp(\mathcal{X}) \,.\, f(Y) \supseteq/= \neg g(\neg Y)$ ⟅letting $X = \neg Y$ and $\neg \circ \neg$ is the identity⟆

$\Leftrightarrow f \mathrel{\dot{\supseteq}/=} \neg \circ g \circ \neg$ ⟅pointwise def. $\dot{\supseteq}$ and function composition $\circ$⟆

$\Leftrightarrow f \mathrel{\dot{\supseteq}/=} \alpha^{\sim}(g)$ ⟅def. $\alpha^{\sim}$⟆ □

PROOF OF (34).

$$
\begin{aligned}
-\ & \widetilde{\mathsf{Post}} \\
=\ & \dot{\alpha}^{\sim}(\mathsf{Post}) && \text{(def. (34) of } \widetilde{\mathsf{Post}}\text{)} \\
=\ & \boldsymbol{\lambda} r \bullet \alpha^{\sim}(\mathsf{Post}(r)) && \text{(pointwise def. } \dot{\alpha}^{\sim}\text{)} \\
=\ & \boldsymbol{\lambda} r \bullet \neg \circ \mathsf{Post}(r) \circ \neg && \text{(def. } \alpha^{\sim}\text{)} \\
=\ & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \neg(\mathsf{Post}(r)(\neg P)) && \text{(def. function composition } \circ\text{)} \\
=\ & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \neg\{\langle\sigma_0,\ \sigma'\rangle \mid \exists\sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in (\neg P) \wedge \langle\sigma,\ \sigma'\rangle \in r\} && \text{(def. (6) of Post)} \\
=\ & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \{\langle\sigma_0,\ \sigma'\rangle \mid \forall\sigma\ .\ \langle\sigma_0,\ \sigma\rangle \notin (\neg P) \vee \langle\sigma,\ \sigma'\rangle \notin r\} && \text{(def.negation } \neg\text{)} \\
=\ & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \{\langle\sigma_0,\ \sigma'\rangle \mid \forall\sigma\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma_0,\ \sigma\rangle \in P\} && \text{(def. implication } \Rightarrow \text{ and } \neg\text{)}
\end{aligned}
$$

$$
\begin{aligned}
-\ & \widetilde{\mathsf{Pre}} \\
=\ & \dot{\alpha}^{\sim}(\mathsf{Pre}) && \text{(def. (34) of } \widetilde{\mathsf{Pre}}\text{)} \\
=\ & \boldsymbol{\lambda} r \bullet \alpha^{\sim}(\mathsf{Pre}(r)) && \text{(pointwise def. } \dot{\alpha}^{\sim}\text{)} \\
=\ & \boldsymbol{\lambda} r \bullet \neg \circ \mathsf{Pre}(r) \circ \neg && \text{(def. } \alpha^{\sim}\text{)} \\
=\ & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \neg(\mathsf{Pre}(r)(\neg Q)) && \text{(def. function composition } \circ\text{)} \\
=\ & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \neg\{\langle\sigma,\ \sigma_f\rangle \mid \exists\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \wedge \langle\sigma',\ \sigma_f\rangle \in (\neg Q))\} && \text{(def. (29) of Pre)} \\
=\ & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \{\langle\sigma,\ \sigma_f\rangle \mid \forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \notin r \vee \langle\sigma',\ \sigma_f\rangle \in Q\} && \text{(def. negation } \neg\text{)} \\
=\ & \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \{\langle\sigma,\ \sigma_f\rangle \mid \forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma',\ \sigma_f\rangle \in Q\} && \text{(def. implication } \Rightarrow\text{)} \quad \square
\end{aligned}
$$

PROOF OF (35).

$$
\begin{aligned}
-\ & \mathsf{Post}(r)P \subseteq Q && \text{(}P, Q \in \wp(\mathcal{X} \times \mathcal{X})\text{)} \\
\Leftrightarrow\ & \{\langle\sigma_0,\ \sigma'\rangle \mid \exists\sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in P \wedge \langle\sigma,\ \sigma'\rangle \in r\} \subseteq Q && \text{(def. (6) of Post)} \\
\Leftrightarrow\ & \forall\langle\sigma_0,\ \sigma'\rangle\ .\ (\exists\sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in P \wedge \langle\sigma,\ \sigma'\rangle \in r) \Rightarrow \langle\sigma_0,\ \sigma'\rangle \in Q && \text{(def. } \subseteq\text{)} \\
\Leftrightarrow\ & \forall\sigma_0, \sigma', \sigma\ .\ (\langle\sigma_0,\ \sigma\rangle \in P \wedge \langle\sigma,\ \sigma'\rangle \in r) \Rightarrow \langle\sigma_0,\ \sigma'\rangle \in Q && \text{(def. } \Rightarrow \text{ and } \exists/\forall\text{)} \\
\Leftrightarrow\ & \forall\sigma_0, \sigma\ .\ (\langle\sigma_0,\ \sigma\rangle \in P \Rightarrow (\forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma_0,\ \sigma'\rangle \in Q) && \text{(def. } \Rightarrow \text{ and } \wedge\text{)} \\
\Leftrightarrow\ & P \subseteq \{\langle\sigma_0,\ \sigma\rangle \mid \forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma_0,\ \sigma'\rangle \in Q\} && \text{(def. } \subseteq\text{)} \\
\Leftrightarrow\ & P \subseteq \{\langle\sigma_f,\ \sigma\rangle \mid \forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma_f,\ \sigma'\rangle \in Q\} && \text{(renaming)} \\
\Leftrightarrow\ & P \subseteq \alpha^{-1}(\{\langle\sigma,\ \sigma_f\rangle \mid \forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma',\ \sigma_f\rangle \in \alpha^{-1}(Q)\}) && \text{(def. (27) of } \alpha^{-1}\text{)} \\
\Leftrightarrow\ & P \subseteq \alpha^{-1} \circ (\boldsymbol{\lambda} Q \bullet \{\langle\sigma,\ \sigma_f\rangle \mid \forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma',\ \sigma_f\rangle \in Q\}) \circ \alpha^{-1}(Q) \\
& && \text{(def. function application and composition } \circ\text{)} \\
\Leftrightarrow\ & P \subseteq \dot{\alpha}^{-1}(\boldsymbol{\lambda} Q \bullet \{\langle\sigma,\ \sigma_f\rangle \mid \forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma',\ \sigma_f\rangle \in Q\})Q && \text{(def. (27) of } \dot{\alpha}^{-1}\text{)} \\
\Leftrightarrow\ & P \subseteq \dot{\alpha}^{-1}(\widetilde{\mathsf{Pre}}(r))Q && \text{(def. (34) of } \widetilde{\mathsf{Pre}}\text{)}
\end{aligned}
$$

$$
\begin{aligned}
-\ & \mathsf{Pre}(r)Q \subseteq P \\
\Leftrightarrow\ & \{\langle\sigma,\ \sigma_f\rangle \mid \exists\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \wedge \langle\sigma',\ \sigma_f\rangle \in Q)\} \subseteq P && \text{(def. (29) of Pre)} \\
\Leftrightarrow\ & \forall\sigma, \sigma_f\ .\ (\exists\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \wedge \langle\sigma',\ \sigma_f\rangle \in Q)\}) \Rightarrow \langle\sigma,\ \sigma_f\rangle \in P && \text{(def. } \subseteq\text{)} \\
\Leftrightarrow\ & \forall\sigma, \sigma_f, \sigma'\ .\ (\langle\sigma,\ \sigma'\rangle \in r \wedge \langle\sigma',\ \sigma_f\rangle \in Q)\}) \Rightarrow \langle\sigma,\ \sigma_f\rangle \in P && \text{(def. } \Rightarrow \text{ and } \forall/\exists\text{)} \\
\Leftrightarrow\ & \forall\sigma, \sigma_f, \sigma'\ .\ (\langle\sigma',\ \sigma_f\rangle \in Q) \Rightarrow (\langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma,\ \sigma_f\rangle \in P) && \text{(def. } \Rightarrow \text{ and } \wedge\text{)} \\
\Leftrightarrow\ & \forall\sigma_f, \sigma'\ .\ (\langle\sigma',\ \sigma_f\rangle \in Q) \Rightarrow (\forall\sigma\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma,\ \sigma_f\rangle \in P) && \text{(def. } \Rightarrow \text{ and } \forall\text{)} \\
\Leftrightarrow\ & Q \subseteq \{\langle\sigma',\ \sigma_f\rangle \mid \forall\sigma\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma,\ \sigma_f\rangle \in P\} && \text{(def. } \subseteq\text{)}
\end{aligned}
$$

$\Leftrightarrow Q \subseteq \alpha^{-1}(\{\langle\sigma_f,\ \sigma'\rangle \mid \forall\sigma\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma_f,\ \sigma\rangle \in \alpha^{-1}(P)\})$ $\wr$def. (27) of $\alpha^{-1}\wr$
$\Leftrightarrow Q \subseteq \alpha^{-1}(\{\langle\sigma_0,\ \sigma'\rangle \mid \forall\sigma\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma_0,\ \sigma\rangle \in \alpha^{-1}(P)\})$ $\wr$renaming$\wr$
$\Leftrightarrow Q \subseteq \alpha^{-1}(\widetilde{\mathsf{Post}}(r)(\alpha^{-1}(P)))$ $\wr$def. (34) of $\widetilde{\mathsf{Post}}\wr$
$\Leftrightarrow Q \subseteq \alpha^{-1} \circ \widetilde{\mathsf{Post}}(r) \circ \alpha^{-1}(P)$ $\wr$def. function composition $\circ\wr$
$\Leftrightarrow Q \subseteq \dot{\alpha}^{-1}(\widetilde{\mathsf{Post}}(r))P$ $\wr$def. (27) of $\dot{\alpha}^{-1}\wr$ $\square$

Proof of (37). This is an application of [Cousot 2021, Theorem 11.78] and [Cousot 2021, Exercise 11.21]. We provide the proof for the appendix to be self-contained.

– $\alpha_2^A(\Theta) \,\dot{\subseteq}\, \theta$ $\wr$where $\Theta \in \wp(\mathcal{Z}\times\mathcal{X}) \xrightarrow{i} \wp(\mathcal{Z}\times\mathcal{Y})$ and $\theta \in \wp(\mathcal{X}) \xrightarrow{i} \wp(\mathcal{Y})\wr$
$\Leftrightarrow \alpha_2^A(\Theta)P \subseteq \theta(P)$ $\wr$pointwise def. $\dot{\subseteq}\wr$
$\Leftrightarrow \alpha_{\downarrow^2} \circ \Theta \circ \gamma_{\downarrow^2}(P) \subseteq \theta(P)$ $\wr$def. (24) of $\alpha_2^A\wr$
$\Leftrightarrow \Theta \circ \gamma_{\downarrow^2}(P) \subseteq \gamma_{\downarrow^2} \circ \theta(P)$ $\wr$Galois connection (23)$\wr$
$\Leftrightarrow \Theta(Q) \subseteq \gamma_{\downarrow^2} \circ \theta \circ \alpha_{\downarrow^2}(Q)$
$\wr(\Rightarrow)$ choosing $P = \alpha_{\downarrow^2}(Q)$, $\gamma_{\downarrow^2} \circ \alpha_{\downarrow^2}$ is extensive (by the Galois connection (23) for $\alpha_{\downarrow^2}$), and Θ is increasing by hypothesis;
$(\Leftarrow)$ choosing $Q = \gamma_{\downarrow^2}(P)$, $\alpha_{\downarrow^2} \circ \gamma_{\downarrow^2}$ is reductive (by the Galois connection (23) for $\gamma_{\downarrow^2}$), and θ is increasing by hypothesis$\wr$
$\Leftrightarrow \Theta \,\dot{\subseteq}\, \gamma_{\downarrow^2} \circ \theta \circ \alpha_{\downarrow^2}$ $\wr$pointwise def. $\dot{\subseteq}\wr$
$\Leftrightarrow \Theta \,\dot{\subseteq}\, \gamma_2^A(\theta)$ $\wr$def. $\gamma_2^A\wr$

– $\dot{\alpha}_2^A(\Theta) \,\ddot{\subseteq}\, \bar{\Theta}$ $\wr$where $\Theta \in \wp(\mathcal{X}\times\mathcal{Y}) \to \wp(\mathcal{Z}\times\mathcal{X}) \xrightarrow{i} \wp(\mathcal{Z}\times\mathcal{Y})\wr$
$\Leftrightarrow \dot{\alpha}_2^A(\Theta)(r) \,\dot{\subseteq}\, \bar{\Theta}(r)$ $\wr$pointwise def. $\ddot{\subseteq}\wr$
$\Leftrightarrow \alpha_2^A(\Theta(r)) \,\dot{\subseteq}\, \bar{\Theta}(r)$ $\wr$def. (36) of $\dot{\alpha}_2^A\wr$
$\Leftrightarrow \Theta(r) \,\dot{\subseteq}\, \gamma_2^A(\bar{\Theta}(r))$ $\wr$Galois connection (37) for $\alpha_2^A\wr$
$\Leftrightarrow \Theta(r) \,\dot{\subseteq}\, \dot{\gamma}_2^A(\bar{\Theta})(r)$ $\wr$def. $\dot{\gamma}_2^A \triangleq \boldsymbol{\lambda} r \bullet \gamma_2^A(r)\wr$
$\Leftrightarrow \Theta \,\ddot{\subseteq}\, \dot{\gamma}_2^A(\bar{\Theta})$ $\wr$pointwise def. $\ddot{\subseteq}\wr$ $\square$

Proof of (38).

– $\dot{\alpha}_2^A(\mathsf{post})$
$= \boldsymbol{\lambda} r \bullet \alpha_2^A(\mathsf{post}(r))$ $\wr$def. (36) of $\dot{\alpha}_2^A\wr$
$= \boldsymbol{\lambda} r \bullet \alpha_{\downarrow^2} \circ \mathsf{post}(r) \circ \gamma_{\downarrow^2}$ $\wr$def. (36) of $\alpha_2^A\wr$
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \alpha_{\downarrow^2}(\mathsf{post}(r)(\gamma_{\downarrow^2}(P)))$ $\wr$def. function composition $\circ\wr$
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \alpha_{\downarrow^2}(\{\langle\sigma_0,\ \sigma'\rangle \mid \exists\sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in \gamma_{\downarrow^2}(P) \wedge \langle\sigma,\ \sigma'\rangle \in r\})$ $\wr$def. (10) of post$\wr$
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \{\sigma' \mid \exists\sigma_0\ .\ \langle\sigma_0,\ \sigma'\rangle \in \{\langle\sigma_0,\ \sigma'\rangle \mid \exists\sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in \Sigma\times P \wedge \langle\sigma,\ \sigma'\rangle \in r\}\}$
$\wr$def. (24) of $\alpha_{\downarrow^2}$ and $\gamma_{\downarrow^2}\wr$
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \{\sigma' \mid \exists\sigma_0\ .\ \exists\sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in \Sigma\times P \wedge \langle\sigma,\ \sigma'\rangle \in r\}$ $\wr$def. $\in\wr$
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \{\sigma' \mid \exists\sigma\ .\ \sigma \in P \wedge \langle\sigma,\ \sigma'\rangle \in r\}$ $\wr$taking $\sigma_0 \in \Sigma\wr$
$=$ post $\wr$def. (10) of post$\wr$ $\square$

– $\dot{\alpha}_2^A(\widetilde{\mathsf{Post}})$
$= \boldsymbol{\lambda} r \bullet \alpha_2^A(\widetilde{\mathsf{Post}}(r))$ $\wr$def. (36) of $\dot{\alpha}_2^A\wr$

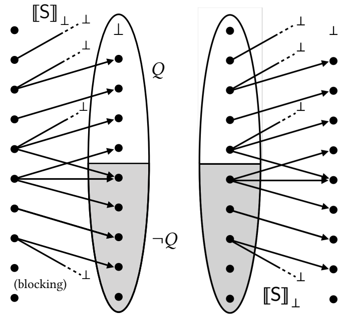
$= \boldsymbol{\lambda} r \bullet \alpha_{\downarrow^2} \circ \widetilde{\mathsf{Post}}(r) \circ \gamma_{\downarrow^2}$ ⟅def. (36) of α_2^A⟆
$= \boldsymbol{\lambda} r \bullet \alpha_{\downarrow^2} \circ \boldsymbol{\lambda} P \bullet \{\langle\sigma_0,\ \sigma'\rangle \mid \forall\sigma\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma_0,\ \sigma\rangle \in P\} \circ \gamma_{\downarrow^2}$ ⟅def. (34) of $\widetilde{\mathsf{Post}}$⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \alpha_{\downarrow^2}(\{\langle\sigma_0,\ \sigma'\rangle \mid \forall\sigma\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma_0,\ \sigma\rangle \in \gamma_{\downarrow^2}(P)\})$ ⟅def. function composition $\circ$⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \{\sigma' \mid \forall\sigma\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma_0,\ \sigma\rangle \in \mathcal{X} \times P\}$ ⟅def. (24) of $\alpha_{\downarrow^2}$ and $\gamma_{\downarrow^2}$⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \{\sigma' \mid \forall\sigma\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \sigma \in P\}$ ⟅def. pairs and $\in$, Q.E.D. for case $\widetilde{\mathsf{post}}$ of (38)⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \neg\{\sigma' \mid \exists\sigma\ .\ \langle\sigma,\ \sigma'\rangle \in r \wedge \sigma \notin P\}$ ⟅def. $\neg$⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \neg\mathsf{post}(r)(\neg P)$ ⟅def. (10) of post ⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \neg \circ \mathsf{post}(r) \circ \neg(P)$ ⟅def. function composition $\circ$ ⟆
$= \boldsymbol{\lambda} r \bullet \alpha^{\sim}(\mathsf{post}(r))$ ⟅def. $\alpha^{\sim}(f) \triangleq \neg \circ f \circ \neg$ in (33) and $\boldsymbol{\lambda}$-calculus⟆
$= \boldsymbol{\lambda} r \bullet \dot{\alpha}^{\sim}(\mathsf{post})(r)$ ⟅pointwise def. $\dot{\alpha}^{\sim}(\mathsf{post}) \triangleq \boldsymbol{\lambda} r \bullet \alpha^{\sim}(\mathsf{post}(r))$ in (33)⟆
$= \dot{\alpha}^{\sim} \circ \mathsf{post}$ ⟅def. function composition $\circ$ and $\boldsymbol{\lambda}$-calculus, Q.E.D. for case $\widetilde{\mathsf{post}}$ of (38)⟆

– $\dot{\alpha}_1^A(\widetilde{\mathsf{Pre}})$
$= \boldsymbol{\lambda} r \bullet \alpha_2^A(\widetilde{\mathsf{Pre}}(r))$ ⟅def. (36) of $\dot{\alpha}_1^A$⟆
$= \boldsymbol{\lambda} r \bullet \alpha_{\downarrow^1} \circ \widetilde{\mathsf{Pre}}(r) \circ \gamma_{\downarrow^1}$ ⟅def. (36) of α_1^A⟆
$= \boldsymbol{\lambda} r \bullet \alpha_{\downarrow^1} \circ \boldsymbol{\lambda} Q \bullet \{\langle\sigma,\ \sigma_f\rangle \mid \forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma',\ \sigma_f\rangle \in Q\} \circ \gamma_{\downarrow^1}$ ⟅def. (34) of $\widetilde{\mathsf{Pre}}$⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \alpha_{\downarrow^1}(\{\langle\sigma,\ \sigma_f\rangle \mid \forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \langle\sigma',\ \sigma_f\rangle \in \gamma_{\downarrow^1}(Q)\})$ ⟅def. function composition $\circ$⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \{\sigma \mid \forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \sigma' \in Q\}$ ⟅def. (26) of $\alpha_{\downarrow^1}$ and $\gamma_{\downarrow^1}$ Q.E.D. for case $\widetilde{\mathsf{pre}}$ of (38)⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \neg\{\sigma \mid \exists\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \wedge \sigma' \in \neg(Q)\}$ ⟅def. $\neg$⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \neg\mathsf{pre}(r)(\neg P)$ ⟅def. (38) of pre ⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \neg \circ \mathsf{pre}(r) \circ \neg(P)$ ⟅def. function composition $\circ$ ⟆
$= \boldsymbol{\lambda} r \bullet \alpha^{\sim}(\mathsf{pre}(r))$ ⟅def. $\alpha^{\sim}(f) \triangleq \neg \circ f \circ \neg$ in (33) and $\boldsymbol{\lambda}$-calculus⟆
$= \boldsymbol{\lambda} r \bullet \dot{\alpha}^{\sim}(\mathsf{pre})(r)$ ⟅pointwise def. $\dot{\alpha}^{\sim}(\mathsf{pre}) \triangleq \boldsymbol{\lambda} r \bullet \alpha^{\sim}(\mathsf{pre}(r))$ in (33)⟆
$= \dot{\alpha}^{\sim} \circ \mathsf{pre}$ ⟅def. function composition $\circ$ and $\boldsymbol{\lambda}$-calculus, Q.E.D. for case pre of (38)⟆

– $\dot{\alpha}_1^A(\mathsf{Pre})$
$= \boldsymbol{\lambda} r \bullet \alpha_2^A(\widetilde{\mathsf{Pre}}(r))$ ⟅def. (36) of $\dot{\alpha}_1^A$⟆
$= \boldsymbol{\lambda} r \bullet \alpha_{\downarrow^1} \circ \mathsf{Pre}(r) \circ \gamma_{\downarrow^1}$ ⟅def. (36) of α_1^A⟆
$= \boldsymbol{\lambda} r \bullet \alpha_{\downarrow^1} \circ \boldsymbol{\lambda} Q \bullet \{\langle\sigma,\ \sigma_f\rangle \mid \exists\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \wedge \langle\sigma',\ \sigma_f\rangle \in Q\} \circ \gamma_{\downarrow^1}$ ⟅def. (34) of Pre⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \alpha_{\downarrow^1}(\{\langle\sigma,\ \sigma_f\rangle \mid \exists\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \wedge \langle\sigma',\ \sigma_f\rangle \in \gamma_{\downarrow^1}(Q)\})$ ⟅def. function composition $\circ$⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \{\sigma \mid \exists\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \wedge \sigma' \in Q\}$ ⟅def. (26) of $\alpha_{\downarrow^1}$ and $\gamma_{\downarrow^1}$ Q.E.D. for case pre of (38)⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} Q \bullet \neg\{\sigma \mid \forall\sigma'\ .\ \langle\sigma,\ \sigma'\rangle \in r \Rightarrow \sigma' \in \neg(Q)\}$ ⟅def. $\neg$⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \neg\widetilde{\mathsf{pre}}(r)(\neg P)$ ⟅def. (38) of $\widetilde{\mathsf{pre}}$ ⟆
$= \boldsymbol{\lambda} r \bullet \boldsymbol{\lambda} P \bullet \neg \circ \widetilde{\mathsf{pre}}(r) \circ \neg(P)$ ⟅def. function composition $\circ$ ⟆
$= \boldsymbol{\lambda} r \bullet \alpha^{\sim}(\widetilde{\mathsf{pre}}(r))$ ⟅def. $\alpha^{\sim}(f) \triangleq \neg \circ f \circ \neg$ in (33) and $\boldsymbol{\lambda}$-calculus⟆
$= \boldsymbol{\lambda} r \bullet \dot{\alpha}^{\sim}(\widetilde{\mathsf{pre}})(r)$ ⟅pointwise def. $\dot{\alpha}^{\sim}(\widetilde{\mathsf{pre}}) \triangleq \boldsymbol{\lambda} r \bullet \alpha^{\sim}(\widetilde{\mathsf{pre}}(r))$ in (33)⟆
$= \dot{\alpha}^{\sim} \circ \widetilde{\mathsf{pre}}$ ⟅def. function composition $\circ$ and $\boldsymbol{\lambda}$-calculus, Q.E.D. for case $\widetilde{\mathsf{pre}}$ of (38)⟆ □

In Fig. 4, the points • represent states, the arrows between two states or a state and $\perp$ represent a pair in $[\![\mathsf{S}]\!]_\perp$. For the angelic semantics $[\![\mathsf{S}]\!]$, the states and arrows marked $\perp$ should be ignored. The consequent states on the left are partitioned into Q and $\neg Q$. In the column for each transformer τ in the left table, tags

$\mathrm{wp}(\mathsf{S}, Q_{\not\perp})$	$\mathrm{wlp}(\mathsf{S}, Q_{\not\perp})$	$\widetilde{\mathrm{pre}}[\![\mathsf{S}]\!]Q$	$\widetilde{\mathrm{pre}}[\![\mathsf{S}]\!]_\perp Q$	$\mathrm{pre}[\![\mathsf{S}]\!]Q$	$\mathrm{pre}[\![\mathsf{S}]\!]_\perp Q$
		×	×		
		×			×
	×	×	×	×	×
×	×	×	×	×	×
				×	×
				×	×
					×
		×	×		
		×	×		



$\mathrm{post}[\![\mathsf{S}]\!]_\perp Q$	$\mathrm{post}[\![\mathsf{S}]\!]Q$	$\widetilde{\mathrm{post}}[\![\mathsf{S}]\!]_\perp Q$	$\widetilde{\mathrm{post}}[\![\mathsf{S}]\!]Q$	$\mathrm{slp}(\mathsf{S}, Q)$	$\mathrm{sp}(\mathsf{S}, Q)$
×			×		
×	×	×	×	×	
×	×	×	×	×	×
×	×	×	×	×	
×	×			×	

Fig. 4. Illustration of property transformers

× indicate that the antecedent state on the same line belongs to $\tau(Q)$. Similarly, the antecedent states on the left are partitioned into Q and $\neg Q$. In the column for each transformer τ in the right table, consequent states belonging to $\tau(Q)$ are tagged × on the same line.

PROOF OF (39.D). Let us show that $\mathrm{post}(R)P \cap Q \neq \varnothing \Leftrightarrow \exists \sigma \in P \,.\, \exists \sigma' \notin Q \,.\, \langle \sigma,\ \sigma' \rangle \in R \Leftrightarrow$

$$
\begin{aligned}
& \mathrm{post}(R)P \cap Q \neq \varnothing \\
\Leftrightarrow\ & \{\sigma' \mid \exists \sigma \in P \,.\, \langle \sigma,\ \sigma' \rangle \in R\} \cap Q \neq \varnothing && \text{(def. post)} \\
\Leftrightarrow\ & \exists \sigma' \,.\, \sigma' \in \{\sigma' \mid \exists \sigma \in P \,.\, \langle \sigma,\ \sigma' \rangle \in R\} \wedge \sigma' \in Q && \text{(def. } \cap \text{ and } \varnothing\text{)} \\
\Leftrightarrow\ & \exists \sigma' \,.\, \exists \sigma \in P \,.\, \langle \sigma,\ \sigma' \rangle \in R \wedge \sigma' \in Q && \text{(def. } \in\text{)} \\
\Leftrightarrow\ & \exists \sigma \in P \,.\, \exists \sigma' \in Q \,.\, \langle \sigma,\ \sigma' \rangle \in R && \text{(commutativity)} \\
\Leftrightarrow\ & \exists \sigma \,.\, \sigma \in P \wedge \sigma \in \{\sigma \mid \exists \sigma' \in Q \,.\, \langle \sigma,\ \sigma' \rangle \in R\} && \text{(def. } \in\text{)} \\
\Leftrightarrow\ & P \cap \{\sigma \mid \exists \sigma' \in Q \,.\, \langle \sigma,\ \sigma' \rangle \in R\} \neq \varnothing && \text{(def. } \cap \text{ and } \varnothing\text{)} \\
\Leftrightarrow\ & P \cap \mathrm{pre}(R)Q \neq \varnothing && \text{(LEM. 6)} \quad \square
\end{aligned}
$$

PROOF OF REMARK I.3.12. Define $t_i = \{\langle n,\ n+i \rangle \mid n \in \mathbb{N}\}$. If $\langle n,\ m \rangle \in \bigcap_{i\in\mathbb{N}} t_i$ then it belongs to all t_i so $\forall i \in \mathbb{N} \,.\, m = n + i$, a contradiction. So $\bigcap_{i\in\mathbb{N}} t_i = \varnothing$ hence $\mathrm{pre}(\bigcap_{i\in\mathbb{N}} t_i)\mathbb{Z} = \varnothing$. On the other hand, $\mathrm{pre}(t_i)\mathbb{Z} = \{n \mid \exists m \in \mathbb{Z} \,.\, \langle n,\ m \rangle \in t_i\} = \{n \mid \exists m \in \mathbb{Z} \,.\, m = n + i\} = \mathbb{Z}$. Obviously $\mathrm{pre}(\bigcap_{i\in\mathbb{N}} t_i)\mathbb{Z} = \varnothing \neq \mathbb{Z} = \bigcap_{i\in\mathbb{N}} \mathrm{pre}(t_i)\mathbb{Z}$.

Take $t = \{\langle 1,\ m \rangle \mid m \in \mathbb{N}\}$ and $X_i = \{n \in \mathbb{N} \mid n \geqslant i\}$. If $m \in \bigcap_{i\in\mathbb{N}} X_i$ then for all $i \in \mathbb{N}$, $m \in X_i$, in contradiction with $m \notin X_{m+1}$ proving $\bigcap_{i\in\mathbb{N}} X_i = \varnothing$ so that $\mathrm{pre}(t)(\bigcap_{i\in\mathbb{N}} X_i) = \varnothing$. Now $\mathrm{pre}(t)X_i = \{n \mid \exists m \in X_i \,.\, \langle n,\ m \rangle \in t_i\}$ $= \{1\}$ since no X_i is empty. It follows that $\mathrm{pre}(t)(\bigcap_{i\in\mathbb{N}} X_i) = \varnothing \neq \{1\} = \bigcap_{i\in\mathbb{N}} \mathrm{pre}(t)X_i$. □

PROOF OF (42).

$$
\begin{aligned}
-\ & \alpha_{\not\perp}(P) \subseteq Q \\
\Leftrightarrow\ & P \setminus \{\perp\} \subseteq Q && \text{(def. (40) of } \alpha_{\not\perp}\text{)} \\
\Leftrightarrow\ & P \subseteq Q \cup \{\perp\} && \text{(def. } \subseteq\text{)} \\
\Leftrightarrow\ & P \subseteq \gamma_{\not\perp}(Q) && \text{(by defining } \gamma_{\not\perp}(Q) \triangleq Q \cup \{\perp\}\text{)} \\
-\ & \vec{\alpha}_{\not\perp}(\theta) \mathrel{\dot\subseteq} \bar\theta \\
\Leftrightarrow\ & \forall P \in \mathcal{X} \,.\, \vec{\alpha}_{\not\perp}(\theta)P \subseteq \bar\theta(P) && \text{(pointwise def. } \dot\subseteq\text{)}
\end{aligned}
$$

$\Leftrightarrow \forall P \in \mathcal{X} \,.\, \alpha_{\not{\perp}}(\theta(P)) \subseteq \bar{\theta}(P)$ ⟅def. (40) of $\overrightarrow{\alpha}_{\not{\perp}}$⟆

$\Leftrightarrow \forall P \in \mathcal{X} \,.\, \theta(P) \subseteq \gamma_{\not{\perp}}(\bar{\theta}(P))$ ⟅Galois connection $\langle \alpha_{\not{\perp}},\ \gamma_{\not{\perp}} \rangle$ (42)⟆

$\Leftrightarrow \theta \mathrel{\dot{\subseteq}} \boldsymbol{\lambda} P \bullet \gamma_{\not{\perp}}(\bar{\theta}(P))$ ⟅pointwise def. $\subseteq$⟆

$\Leftrightarrow \theta \mathrel{\dot{\subseteq}} \overrightarrow{\gamma}_{\not{\perp}}(\bar{\theta})$ ⟅by defining $\overrightarrow{\gamma}_{\not{\perp}}(\bar{\theta}) \triangleq \boldsymbol{\lambda} P \bullet \gamma_{\not{\perp}}(\bar{\theta}(P))$⟆

— The compositions $\overleftarrow{\alpha}_{\not{\perp}}$ and $\overleftarrow{\gamma}_{\not{\perp}}$ of increasing functions is increasing, Moreover,

$\overleftarrow{\alpha}_{\not{\perp}}(\theta) \mathrel{\dot{\subseteq}} \bar{\theta}$

$\Leftrightarrow \forall Q \in \wp(\Sigma_{\perp}) \,.\, \overleftarrow{\alpha}_{\not{\perp}}(\theta)Q \subseteq \bar{\theta}(Q)$ ⟅pointwise def. $\dot{\subseteq}$⟆

$\Leftrightarrow \forall Q \in \wp(\Sigma_{\perp}) \,.\, \theta(\gamma_{\not{\perp}}(Q)) \subseteq \bar{\theta}(Q)$ ⟅def. (42) of $\overleftarrow{\alpha}_{\not{\perp}}$⟆

$\Rightarrow \forall P \in \wp(\Sigma) \,.\, \theta(\gamma_{\not{\perp}}(\alpha_{\not{\perp}}(P))) \subseteq \bar{\theta}(\alpha_{\not{\perp}}(P))$ ⟅for $Q = \alpha_{\not{\perp}}(P)$⟆

$\Rightarrow \forall P \in \wp(\Sigma) \,.\, \theta(P) \subseteq \bar{\theta}(\alpha_{\not{\perp}}(P))$

⟅since $\gamma_{\not{\perp}} \circ \alpha_{\not{\perp}}$ is extensive by Galois connection $\langle \alpha_{\not{\perp}},\ \gamma_{\not{\perp}} \rangle$ (42) and θ is increasing by hypothesis⟆

$\Leftrightarrow \theta \mathrel{\dot{\subseteq}} \bar{\theta}(\alpha_{\not{\perp}})$ ⟅pointwise def. $\dot{\subseteq}$⟆

Conversely,

$\forall P \in \wp(\Sigma) \,.\, \theta(P) \subseteq \bar{\theta}(\alpha_{\not{\perp}}(P))$

$\Rightarrow \forall Q \in \wp(\Sigma_{\perp}) \,.\, \theta(\gamma_{\not{\perp}}(Q)) \subseteq \bar{\theta}(\alpha_{\not{\perp}}(\gamma_{\not{\perp}}(Q)))$ ⟅for $P = \gamma_{\not{\perp}}(Q)$⟆

$\Rightarrow \forall Q \in \wp(\Sigma_{\perp}) \,.\, \theta(\gamma_{\not{\perp}}(Q)) \subseteq \bar{\theta}(Q)$

⟅since $\alpha_{\not{\perp}} \circ \gamma_{\not{\perp}}$ is reductive by Galois connection $\langle \alpha_{\not{\perp}},\ \gamma_{\not{\perp}} \rangle$ (42) and $\bar{\theta}$ is increasing by hypothesis⟆ □

D THE SUBHIERARCHY OF ASSERTIONAL LOGICS, CONTINUED

Continuing Sect. I.3.14 on “The subhierarchy of assertional logics”, we consider other **contrapositive** logics or properties **disproving program properties**.

D.1 Partial Definite Inaccessibility of All Final States From All Non-Initial States $\mathsf{pre}[\![\mathsf{S}]\!]Q \subseteq P \Leftrightarrow Q \subseteq \widetilde{\mathsf{post}}[\![\mathsf{S}]\!]P \Leftrightarrow \neg P \subseteq \widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]\neg Q \Leftrightarrow \mathsf{pre}[\![\mathsf{S}]\!]_{\perp}Q \subseteq P \Leftrightarrow Q \subseteq \widetilde{\mathsf{post}}[\![\mathsf{S}]\!]_{\perp}P \Leftrightarrow \neg P \subseteq \widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]_{\perp}\neg Q$, $P, Q \in \wp(\Sigma)$

This specifies the fact that any executions from initial states not in P can never reach a state in Q. This provides a necessary precondition for partial possible accessibility of some final states from some initial states but does not prevent nontermination. This formalizes by Galois connections the relation inductive principles $(\bar{\mathrm{I}})$, $(\widetilde{\bar{\mathrm{I}}})$, $(\overline{\mathrm{I}^{-1}})$, $(\widetilde{\overline{\mathrm{I}^{-1}}})$ and the assertional ones $(\bar{\mathrm{i}})$, $(\widetilde{\bar{\mathrm{i}}})$, $(\overline{\mathrm{i}^{-1}})$, $(\widetilde{\overline{\mathrm{i}^{-1}}})$ of [Cousot and Cousot 1982].

Notice that complement duality can be used to perform under (respectively over) approximation by over (respectively under) approximations of the complement.

D.2 Partial Definite Inaccessibility of Some Final State From All Non-Initial States $\widetilde{\mathsf{post}}[\![\mathsf{S}]\!]P \cap Q \neq \varnothing \Leftrightarrow \widetilde{\mathsf{post}}[\![\mathsf{S}]\!]_{\perp}P \cap Q \neq \varnothing$, $P, Q \in \wp(\Sigma)$

This expresses that there exists at least one state σ' in Q, which is not accessible by any execution from an initial state not in P.

D.3 Partial Definite Accessibility of Some Final State From Some Initial State $\mathsf{pre}[\![\mathsf{S}]\!]Q \cap P \neq \varnothing \Leftrightarrow \mathsf{pre}[\![\mathsf{S}]\!]_{\perp}Q \cap P \neq \varnothing$, $P, Q \in \wp(\Sigma)$

This indicates that there is at least one initial state in P from which there is at least one execution that does terminate in state Q. We have $\neg(\{P\}\mathsf{S}\{Q\}) \Leftrightarrow \neg(P \subseteq \widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]Q) \Leftrightarrow (P \cap \neg(\widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]Q)) \neq \varnothing \Leftrightarrow P \cap \mathsf{pre}[\![\mathsf{S}]\!]\neg Q \neq \varnothing$ which is another incorrectness logic [Cousot 2021, Ch. 50].

D.4 Partial Possible Accessibility of Some Non-Final State From All Non-Initial States $\widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]Q \subseteq P \Leftrightarrow \widetilde{\mathsf{pre}}[\![\mathsf{S}]\!]_{\perp}Q \subseteq P \Leftrightarrow \neg P \subseteq \mathsf{pre}[\![\mathsf{S}]\!]\neg Q \Leftrightarrow \neg P \subseteq \mathsf{pre}[\![\mathsf{S}]\!]_{\perp}\neg Q$, $P, Q \in \wp(\Sigma)$

The meaning is that executions from initial states not in P have at least one execution not reaching a state in Q. This provides a necessary precondition for universal possible accessibility [Cousot et al. 2013].

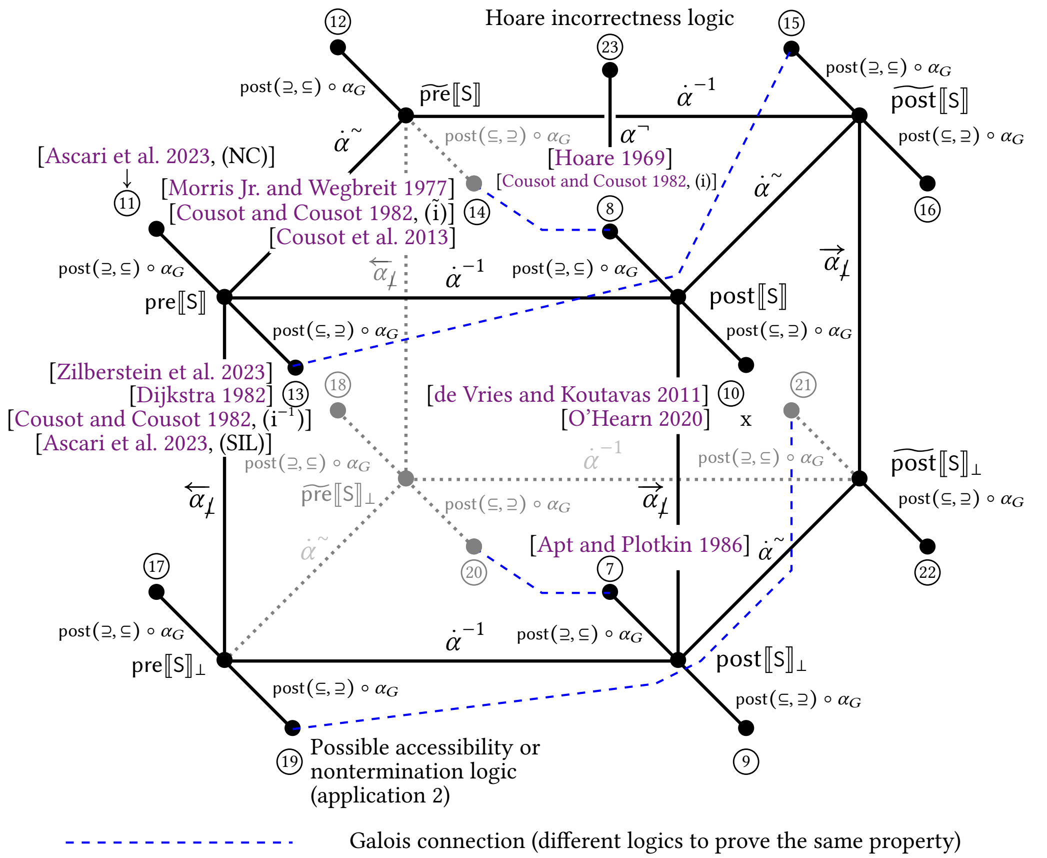


Fig. 3. Hierarchical taxonomy of transformational assertional logics

D.5 Partial Possible Accessibility of All Non-Final States From Some Non-Initial State

$\widetilde{\mathrm{post}}[\![\mathsf{S}]\!]P \subseteq Q \Leftrightarrow \neg Q \subseteq \mathrm{post}[\![\mathsf{S}]\!]\neg P \Leftrightarrow \neg Q \subseteq \mathrm{post}[\![\mathsf{S}]\!]_\bot\neg P \Leftrightarrow \neg Q \subseteq \neg\widetilde{\mathrm{post}}[\![\mathsf{S}]\!]_\bot P \Leftrightarrow \widetilde{\mathrm{post}}[\![\mathsf{S}]\!]_\bot P \subseteq Q$

The signification is that for any state σ' not in Q there exists at least one initial state σ not in P and an execution from σ that will terminate in state σ'. Letting $P' = \neg P$ and $Q' = \neg Q$, this is partial possible accessibility of all final states from some initial state $Q' \subseteq \mathrm{post}[\![\mathsf{S}]\!]P'$ from Sect. I.3.14.3. This shows that the under approximation $Q \subseteq \mathrm{post}[\![\mathsf{S}]\!]P$ is equivalent to an over approximation $\widetilde{\mathrm{post}}[\![\mathsf{S}]\!]\neg P \subseteq \neg Q$ of the complement, that is, a proof by contradiction.

D.6 Total Definite Accessibility of Some Final State From Some Initial State $\widetilde{\mathrm{pre}}[\![\mathsf{S}]\!]Q \cap P \neq \emptyset \Leftrightarrow \widetilde{\mathrm{pre}}[\![\mathsf{S}]\!]_\bot Q \cap P \neq \emptyset$, $P, Q \in \wp(\Sigma)$

This states that there is at least one initial state in P from which all executions do terminate in Q.

Proof of (43).

– $\alpha^{\cap}\langle\tau_1,\ \tau_2\rangle \ddot{\supseteq} \bar{\tau}$

$\Leftrightarrow \alpha^{\cap}\langle\tau_1,\ \tau_2\rangle(r) \dot{\supseteq} \bar{\tau}(r)$ ⟅pointwise def. $\ddot{\subseteq}$⟆

$\Leftrightarrow \tau_1(r) \,\dot{\cap}\, \tau_2(r) \dot{\supseteq} \bar{\tau}(r)$ ⟅def. $\alpha^{\cap}$⟆

$\Leftrightarrow \tau_1(r) \dot{\supseteq} \bar{\tau}(r) \wedge \tau_2(r) \dot{\supseteq} \bar{\tau}(r)$ ⟅def. $\dot{\cap}$⟆

$\Leftrightarrow \langle\tau_1(r),\ \tau_2(r)\rangle \ddot{\supseteq} \langle\bar{\tau}(r),\ \bar{\tau}(r)\rangle$ ⟅componentwise def. $\ddot{\supseteq}$ for pairs⟆

$\Leftrightarrow \langle\tau_1(r),\ \tau_2(r)\rangle \ddot{\supseteq} \delta(\bar{\tau}(r))$ ⟅def. δ⟆

– Similarly $\alpha^{\cup}\langle\tau_1,\ \tau_2\rangle \ddot{\sqsubseteq} \bar{\tau} \Leftrightarrow \langle\tau_1(r),\ \tau_2(r)\rangle \ddot{\sqsubseteq} \delta(\bar{\tau}(r))$ by $\subseteq$-order duality. □

Proof of (44).

$\alpha^{\times}\langle\tau_1,\ \tau_2\rangle\ (\sqsubseteq_1 \dot{,} \sqsubseteq_2)\ \bar{\tau}$

$\Leftrightarrow \forall P\ .\ \alpha^{\times}\langle\tau_1,\ \tau_2\rangle(P)\ (\sqsubseteq_1, \sqsubseteq_2)\ \bar{\tau}(P)$ ⟨pointwise def. $(\sqsubseteq_1 \dot{,} \sqsubseteq_2)$⟩

$\Leftrightarrow \forall P\ .\ \langle\tau_1(P),\ \tau_2(P)\rangle\ (\sqsubseteq_1, \sqsubseteq_2)\ \bar{\tau}(P)$ ⟨def. $\alpha^{\times}$⟩

$\Leftrightarrow \forall P\ .\ \langle\tau_1(P),\ \tau_2(P)\rangle\ (\sqsubseteq_1, \sqsubseteq_2)\ \ \mathsf{let}\ \langle P_1,\ P_2\rangle = \bar{\tau}(P)\ \mathsf{in}\ \langle P_1,\ P_2\rangle$ ⟨$\bar{\tau}(P) \in \mathcal{Y}_1 \times \mathcal{Y}_2$⟩

$\Leftrightarrow \forall P\ .\ \ \mathsf{let}\ \langle P_1,\ P_2\rangle = \bar{\tau}(P)\ \mathsf{in}\ \tau_1(P) \sqsubseteq_1 P_1 \wedge \alpha_2(P) \sqsubseteq_2 P_2$ ⟨componentwise def. of $(\sqsubseteq_1, \sqsubseteq_2)$⟩

$\Leftrightarrow \tau_1\ \dot{\sqsubseteq}_1\ \boldsymbol{\lambda} P \bullet\ \mathsf{let}\ \langle P_1,\ P_2\rangle = \bar{\tau}(P)\ \mathsf{in}\ P_1 \wedge \tau_2\ \dot{\sqsubseteq}_2\ \boldsymbol{\lambda} P \bullet\ \mathsf{let}\ \langle P_1,\ P_2\rangle = \bar{\tau}(P)\ \mathsf{in}\ P_2$
⟨pointwise def. $\dot{\sqsubseteq}_1$ and $\dot{\sqsubseteq}_2$⟩

$\Leftrightarrow\ \mathsf{let}\ \bar{\tau}_1 = (\boldsymbol{\lambda} P \bullet\ \mathsf{let}\ \langle P_1,\ P_2\rangle = \bar{\tau}(P)\ \mathsf{in}\ P_1)\ \mathsf{and}\ \bar{\tau}_2 = (\boldsymbol{\lambda} P \bullet\ \mathsf{let}\ \langle P_1,\ P_2\rangle = \bar{\tau}(P)\ \mathsf{in}\ P_2)\ \mathsf{in}\ \tau_1\ \dot{\sqsubseteq}_1\ \bar{\tau}_1 \wedge \tau_2\ \dot{\sqsubseteq}_2\ \bar{\tau}_2$
⟨def. let⟩

$\Leftrightarrow \mathsf{let}\ \langle\bar{\tau}_1,\ \bar{\tau}_2\rangle = \gamma^{\times}(\bar{\tau})\ \mathsf{in}\ \tau_1\ \dot{\sqsubseteq}_1\ \bar{\tau}_1 \wedge \tau_2\ \dot{\sqsubseteq}_2\ \bar{\tau}_2$ ⟨def. $\gamma^{\times}$⟩

$\Leftrightarrow \mathsf{let}\ \langle\bar{\tau}_1,\ \bar{\tau}_2\rangle = \gamma^{\times}(\bar{\tau})\ \mathsf{in}\ \langle\tau_1,\ \tau_2\rangle\ \dot{\sqsubseteq}_1.\dot{\sqsubseteq}_2\ \langle\bar{\tau}_1,\ \bar{\tau}_2\rangle$ ⟨componentwise def. $\dot{\sqsubseteq}_1.\dot{\sqsubseteq}_2$⟩

$\Leftrightarrow \langle\tau_1,\ \tau_2\rangle\ \dot{\sqsubseteq}_1.\dot{\sqsubseteq}_2\ \gamma^{\times}(\bar{\tau})$ ⟨def. let⟩ □

Proof of (46). We use $\{\varsigma \mid \varsigma = \sigma\}$ rather than $\{\sigma\}$ to make bindings more clear.

$\mathsf{post}(r)P$

$= \{\sigma' \mid \exists \sigma \in P\ .\ \langle\sigma,\ \sigma'\rangle \in r\}$ ⟨def. post⟩

$= \{\sigma' \mid \exists \sigma \in P\ .\ \exists \overline{\sigma}\ .\ \overline{\sigma} = \sigma' \wedge \langle\sigma,\ \overline{\sigma}\rangle \in r\}$ ⟨def. =⟩

$= \{\sigma' \mid \exists \sigma \in P\ .\ \sigma \in \{\sigma \mid \exists \overline{\sigma}\ .\ \overline{\sigma} = \sigma' \wedge \langle\sigma,\ \overline{\sigma}\rangle \in r\}\}$ ⟨def. ∈⟩

$= \{\sigma' \mid \exists \sigma \in P\ .\ \sigma \in \{\sigma \mid \exists \overline{\sigma}\ .\ \overline{\sigma} \in \{\overline{\varsigma} \mid \overline{\varsigma} = \sigma'\} \wedge \langle\sigma,\ \overline{\sigma}\rangle \in r\}\}$ ⟨def. ∈⟩

$= \{\sigma' \mid \exists \sigma \in P\ .\ \sigma \in \{\sigma \mid \exists \overline{\sigma}\ .\ \langle\sigma,\ \overline{\sigma}\rangle \in r \wedge \overline{\sigma} \in \{\overline{\varsigma} \mid \overline{\varsigma} = \sigma'\}\}\}$ ⟨commutativity ∧⟩

$= \{\sigma' \mid \exists \sigma \in P\ .\ \sigma \in \mathsf{pre}(r)\{\overline{\varsigma} \mid \overline{\varsigma} = \sigma'\}\}$ ⟨def. pre⟩

$= \{\sigma' \mid \exists \sigma \in P\ .\ \sigma \in \mathsf{post}(r^{-1})\{\overline{\varsigma} \mid \overline{\varsigma} = \sigma'\}\}$ ⟨def. post⟩

$= \alpha^{\leftrightarrow}(\mathsf{post})(r)P$ ⟨def. $\alpha^{\leftrightarrow}$⟩

The proof for pre is similar with $\mathcal{D} = (\wp(\mathcal{X} \times \mathcal{Y}) \to (\wp(\mathcal{Y}) \to \wp(\mathcal{X}))) \to (\wp(\mathcal{X} \times \mathcal{Y}) \to (\wp(\mathcal{Y}) \to \wp(\mathcal{X})))$. For Post, we have

$\mathsf{Post}(r)P$

$= \{\langle\sigma_0,\ \sigma'\rangle \mid \exists \sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in P \wedge \langle\sigma,\ \sigma'\rangle \in r\}$ ⟨def. post⟩

$= \{\langle\sigma_0,\ \sigma'\rangle \mid \exists \sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in P \wedge \langle\sigma',\ \sigma\rangle \in r^{-1}\}$ ⟨def. inverse r^{-1}⟩

$= \{\langle\sigma_0,\ \sigma'\rangle \mid \exists \sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in P \wedge \langle\sigma_0,\ \sigma'\rangle \in \{\langle\varsigma_0,\ \varsigma'\rangle \mid \exists \langle\varsigma_0,\ \varsigma\rangle \in \{\langle\overline{\varsigma}_0,\ \overline{\varsigma}\rangle \mid \overline{\varsigma}_0 = \sigma_0 \wedge \overline{\varsigma} = \sigma\}\ .\ \langle\varsigma',\ \varsigma\rangle \in r^{-1}\}\}$ ⟨def. membership ∈⟩

$= \{\langle\sigma_0,\ \sigma'\rangle \mid \exists \sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in P \wedge \langle\sigma_0,\ \sigma'\rangle \in \mathsf{Post}(r^{-1})\{\langle\overline{\varsigma}_0,\ \overline{\varsigma}\rangle \mid \overline{\varsigma}_0 = \sigma_0 \wedge \overline{\varsigma} = \sigma\}\}$ ⟨def. Post⟩

$= \alpha^{\leftrightarrow}(\mathsf{Post})(r)P$ ⟨def. $\alpha^{\leftrightarrow}$⟩

where in this case, we have the definition

$$\alpha^{\leftrightarrow}(\tau)(r)P \quad \triangleq \quad \{\langle\sigma_0,\ \sigma'\rangle \mid \exists \sigma\ .\ \langle\sigma_0,\ \sigma\rangle \in P \wedge \langle\sigma_0,\ \sigma'\rangle \in \tau(r^{-1})(\{\langle\overline{\varsigma}_0,\ \overline{\varsigma}\rangle \mid \overline{\varsigma}_0 = \sigma_0 \wedge \overline{\varsigma} = \sigma\})\}$$

The case of Pre is similar with $\mathcal{D} = (\wp(\mathcal{X} \times \mathcal{Y}) \to (\wp(\mathcal{Y} \times \mathcal{Z}) \to \wp(\mathcal{X} \times \mathcal{Z}))) \to (\wp(\mathcal{X} \times \mathcal{Y}) \to (\wp(\mathcal{Y} \times \mathcal{Z}) \to \wp(\mathcal{X} \times \mathcal{Z})))$. □

E AUXILIARY MATERIAL FOR SECTION II.1 (STRUCTURAL FIXPOINT NATURAL RELATIONAL SEMANTICS)

PROOF OF LEMMA II.1.1. Let $\langle X^n,\ n \in \mathbb{N}\rangle$ be the iterates of $F^\perp(X) \triangleq [\![\mathsf{B}]\!] \fatsemi [\![\mathsf{S}]\!]^e \fatsemi X$ in (50) on the complete lattice $\langle \wp(\Sigma \times \{\perp\}),\ \subseteq,\ \emptyset,\ \Sigma \times \{\perp\},\ \cup,\ \cap\rangle$ from the supremum $X^0 = \Sigma \times \{\perp\}$.

Assume that $X^n = ([\![\mathsf{B}]\!] \fatsemi [\![\mathsf{S}]\!]^e)^n \fatsemi (\Sigma \times \{\perp\})$ by induction hypothesis.

Then by def. (50) of $F^\perp$, we have $X^{n+1} = F^\perp(X^n) = [\![\mathsf{B}]\!]\fatsemi[\![\mathsf{S}]\!]^e\fatsemi([\![\mathsf{B}]\!]\fatsemi[\![\mathsf{S}]\!]^e)^n\fatsemi(\Sigma\times\{\perp\}) = ([\![\mathsf{B}]\!]\fatsemi[\![\mathsf{S}]\!]^e)^{n+1}\fatsemi\Sigma\times\{\perp\}$.

Since $F^\perp$ preserves joins, by [Cousot and Cousot 1979a], the limit is $\mathsf{gfp}^{\subseteq} F^\perp = X^\omega = \bigcap_{n\in\mathbb{N}} X^n = \bigcap_{n\in\mathbb{N}}([\![\mathsf{B}]\!] \fatsemi [\![\mathsf{S}^e]\!])^n \fatsemi \Sigma \times \{\perp\}$.

For X^ω not to be empty, none of the iterates $([\![\mathsf{B}]\!] \fatsemi [\![\mathsf{S}^e]\!])^n \fatsemi \Sigma \times \{\perp\}$ must be empty, meaning that $\forall n \in \mathbb{N}\ .\ ([\![\mathsf{B}]\!] \fatsemi [\![\mathsf{S}^e]\!])^n \neq \emptyset$. By definition of the power of a relation, this implies that $\exists \sigma \in \mathbb{N} \to \Sigma\ .\ \forall i \in \mathbb{N}\ .\ \langle \sigma_i,\ \sigma_{i+1}\rangle \in [\![\mathsf{B}]\!] \fatsemi [\![\mathsf{S}^e]\!]$. The result follows by contraposition. □

E.1 Auxiliary Variables

The relational natural semantics $[\![\mathsf{S}]\!]_\perp \in \wp(\Sigma\times\Sigma_\perp)$ of Sect. II.1 can be extended to record the values of variables of entry of statements S. The states $\sigma \in \Sigma \triangleq \mathbb{X} \to \mathcal{V}$ are extended with fresh auxiliary variables $\overline{\mathbb{X}} \triangleq \{\bar{x} \mid x \in \mathbb{X}\}$ to $\sigma \in \overline{\Sigma} \triangleq \overline{\mathbb{X}} \cup \mathbb{X} \to \mathcal{V}$. We define the projection on auxiliary variables as $\sigma_{\overline{\mathbb{X}}} \in \overline{\mathbb{X}} \to \mathcal{V}$ such that $\forall \bar{x} \in \overline{\mathbb{X}}\ .\ \sigma_{\overline{\mathbb{X}}}(\bar{x}) = \sigma(\bar{x})$ while the projection on program variables is $\sigma_{\mathbb{X}} \in \mathbb{X} \to \mathcal{V}$ such that $\forall x \in \mathbb{X}\ .\ \sigma_{\mathbb{X}}(x) = \sigma(x)$. A predicate $P \in \wp(\Sigma)$ is extended to $P \in \wp(\overline{\Sigma}) = \{\sigma \mid \sigma_{\mathbb{X}} \in P\}$. For initialization of fresh auxiliary variables to the values of the program variables, we define $\mathbb{I}_{\overline{\mathbb{X}}}(P) \triangleq \{\sigma \in \overline{\Sigma} \mid \sigma_{\overline{\mathbb{X}}} = \sigma_{\mathbb{X}} \wedge \sigma_{\mathbb{X}} \in P\}$. The relations $r \in \wp(\Sigma \times \Sigma_\perp)$ involved in the relational natural semantics of Sect. II.1 are extended to $r \in \wp(\overline{\Sigma} \times \overline{\Sigma}_\perp) = \{\langle \sigma,\ \sigma'\rangle \mid \sigma'_{\overline{\mathbb{X}}} = \sigma_{\overline{\mathbb{X}}} \wedge \langle \sigma_{\mathbb{X}},\ \sigma'_{\mathbb{X}}\rangle \in r\}$ since the values of initial values are kept unchanged by program execution. We leave implicit the fact that batches of fresh auxiliary variables can be successively added to program variables, for example on entry of each loop body in imbricated loops.

E.2 Bounded Versus Unbounded Nondeterminism

Unbounded nondeterminism was strongly rejected by Dijkstra in [Dijkstra 1976, chapter 9] as unimplementable. Park was rather critical about this restriction [Park 1969] since time is unbounded. Later Dijkstra changed his mind [?], [Dijkstra and Scholten 1990, pages 174–180 of chapter 9] using arbitrary well-founded sets as by Turing [Turing 1950] and Floyd [Floyd 1967]. This is the approach we use, using ordinals [Monk 1969, Ch. 2] to formalize well-founded sets used in termination proofs.

F AUXILIARY MATERIAL FOR SECTION II.2 (FIXPOINT ABSTRACTION)

PROOF OF COROLLARY II.2.2. Given any $Q \in L$, define $\alpha_Q(f) \triangleq f(Q)$ and $\gamma_Q(y) \triangleq \boldsymbol{\lambda} x \bullet (\!| x = Q \mathrel{?} y \mathrel{:} \top |\!)$. We have $\langle L \to L',\ \dot{\sqsubseteq}'\rangle \underset{\alpha_Q}{\overset{\gamma_Q}{\leftrightarrows}} \langle L',\ \sqsubseteq'\rangle$ as follows

$$
\begin{array}{ll}
\quad \alpha_Q(f) \sqsubseteq' y & \\
\Leftrightarrow f(Q) \sqsubseteq' y & \text{(def. } \alpha_Q\text{)} \\
\Leftrightarrow f \mathrel{\dot{\sqsubseteq}'} \boldsymbol{\lambda} x \bullet (\!| x = Q \mathrel{?} y \mathrel{:} \top' |\!) & \text{(pointwise def. } \dot{\sqsubseteq}' \text{ and } \top' \text{ is supremum for } \sqsubseteq'\text{)} \\
\Leftrightarrow f \mathrel{\dot{\sqsubseteq}'} \gamma_Q(y) & \text{(def. } \gamma_Q(y)\text{)}
\end{array}
$$

We have the commutation

$$
\begin{array}{lll}
& \alpha_Q(F(f)) & \\
= & F(f)(Q) & \text{(def. } \alpha_Q\text{)} \\
= & \bar{F}_Q(f(Q)) & \text{(hypothesis)} \\
= & \bar{F}_Q(\alpha_Q(f)) & \text{(def. } \alpha_Q\text{)}
\end{array}
$$

By the fixpoint abstraction Th. II.2.1 and definition of α_Q, we conclude that $\mathsf{lfp}^{\sqsubseteq'} \bar{F}_Q = \alpha_Q(\mathsf{lfp}^{\dot{\sqsubseteq}'} F) = (\mathsf{lfp}^{\dot{\sqsubseteq}'} F)Q$. □

G ABSTRACTIONS OF DEDUCTIVE SYSTEMS

In order to abstract a deductive system R, we can consider its model semantics $\mathsf{lfp}^{\subseteq} F_R$ as in Sect. II.5.2, abstract this fixpoint into $\mathsf{lfp}^{\subseteq} \bar{F}_R$ by abstracting its transformer F_R into an abstract one $\bar{F}_R$ using Th. II.2.1, and then this abstract fixpoint back to an abstract deductive system, as explained in Sect. II.5.3. A more direct way is as follows

Theorem G.1 (Deductive system abstraction). *Let $S \in \wp(X)$ be the set defined by the deductive system R. Let $\langle \wp(X), \subseteq \rangle \xleftrightarrow[\alpha]{\gamma} \langle \wp(Y), \subseteq \rangle$. Let $\bar{S} \in \wp(Y)$ be defined by the abstract deductive system $\bar{R} \triangleq \big\{ \frac{\alpha(P)}{\bar{c}} \bigm| \frac{P}{c} \in R \wedge \bar{c} \in \alpha(\{c\}) \big\}$. Then $\alpha(S) \subseteq \bar{S}$. For a Galois isomorphism, $\alpha(S) = \bar{S}$.*

Proof of Th. G.1. Consider the consequence operators F_R and $F_{\bar{R}}$ of the two deductive systems R and $\bar{R}$. We have

$$
\begin{array}{lll}
& \alpha(F_R(X)) & \\
= & \alpha(\big\{c \bigm| \frac{P}{c} \in R \wedge P \subseteq X\big\}) & \text{\textit{def. consequence operator } } F_R \\
= & \alpha(\bigcup\big\{\{c\} \bigm| \frac{P}{c} \in R \wedge P \subseteq X\big\}) & \text{def. } \bigcup \\
= & \bigcup\big\{\alpha(\{c\}) \bigm| \frac{P}{c} \in R \wedge P \subseteq X\big\}) & \text{in a Galois connection } \alpha \text{ preserves existing joins} \\
= & \big\{\bar{c} \bigm| \bar{c} \in \alpha(\{c\}) \wedge \frac{P}{c} \in R \wedge P \subseteq X\big\} & \text{def. } \bigcup \\
\subseteq/= & \big\{\bar{c} \bigm| \bar{c} \in \alpha(\{c\}) \wedge \frac{P}{c} \in R \wedge \alpha(P) \subseteq \alpha(X)\big\} & \\
\end{array}
$$

(for $\subseteq$ in case of a Galois connection, α is increasing
for = in case of a Galois isomorphism such that, conversely, $\alpha(X) \subseteq \alpha(Y)$ implies $X \subseteq Y$)

$$
= \quad F_{\bar{R}}(\alpha(X)) \qquad \text{def. consequence operator } F_{\bar{R}}
$$

We conclude, by semi-commutativity and fixpoint abstraction Th. II.2.1 that $S = \mathsf{lfp}^{\subseteq} F_R \subseteq \mathsf{lfp}^{\subseteq} F_{\bar{R}} = \bar{S}$ with equality in case of a Galois isomorphism implying commutativity $\alpha \circ F_R = F_{\bar{R}} \circ \alpha$. □

H AUXILIARY MATERIAL FOR SECTION II.3 (FIXPOINT INDUCTION)

We let $\langle \mathbb{O}, \in, \varnothing, \mathbb{O}, \cup, \cap \rangle$ be the von Neumann's ordinals [von Neumann 1923] (see [Monk 1969] for an introduction). Von Neumann's ordinals are $0 = \varnothing$, $1 = \{\varnothing\}$, $2 = \{\varnothing, \{\varnothing\}\} = \{0, 1\}$, $3 = \{0, 1, 2\}$, ..., $\delta + 1 = \{\delta\} \cup \delta$ for successor ordinals, and $\lambda = \bigcup_{\beta \in \delta} \beta$ for limit ordinals ordered by $\in$. We let ω be the first infinite limit ordinal. For better intuition, we will denote the strict order $\in$ by $<$, 0 for $\varnothing$, $+1$ for the successor function, sometimes max for $\cup$, min for $\cap$. For this total order $\langle \mathbb{O}, \leqslant \rangle$ where $\leqslant$ is $<$ or equality, $\cup$ is the least upper bound and $\cap$ is the greatest lower bound.

We let $\mathfrak{Wf}$ be the class of all well-founded sets $\langle W, \leqslant \rangle$, that is, a set W equipped with a binary relation $\leqslant$, with no infinite strictly decreasing chain for $\leqslant$. $\leqslant$ is often a partial order, but this is not necessary. In that case $\langle W, \leqslant \rangle$ is called a well-ordered set. Any well-founded sets $\langle W, \leqslant \rangle \in \mathfrak{Wf}$ can be mapped to a unique ordinal. Define the ranking function $\rho(w) = \bigcup_{w' \in W \wedge w' < w} \rho(w')$. Notice that for minimal elements m of W (i.e. $\forall w' \in W : w' \nless m$ and there must be some by wellfoundedness), $\rho(m) = \bigcup \varnothing = \varnothing$ that is, 0. For any subset $X \in \wp(W)$ define $\rho(X) = \bigcup_{w \in X} \rho(w)$. The order of the well-founded set $\langle W, \leqslant \rangle$, is $\rho(W)$. The elements w of W are mapped to an ordinal $\rho(w) \in \rho(W)$, The crucial property, that we will use, is that $w < w' \Rightarrow \rho(w) \in \rho(w')$ i.e. $\rho(w) < \rho(w')$. So ordinals are a well founded set (indeed a well founded total ordering) and any well founded set can be abstracted into an ordinal, that is $\langle \mathfrak{Wf}, \subseteq \rangle \xleftrightarrow[\rho]{\mathsf{id}} \langle \mathbb{O}, \leqslant \rangle$ where id is the identity function and all well founded set of the same rank are abstracted to the same ordinal, while larger well-founded sets can only be mapped to larger ordinals.

Proof of Th. II.3.4. Let $\langle F^{\delta}, \delta \in \mathbb{O} \rangle$ be the increasing iterates of F from $\perp$ ultimately stationary at rank ϵ[Cousot and Cousot 1979a].

Soundness ($\Leftarrow$). We have $\alpha(F^0) = \alpha(\perp) \mathrel{\bar{\sqsubseteq}} I$ by II.3.4.(1). Assume $\alpha(F^{\delta}) \mathrel{\bar{\sqsubseteq}} I$ by induction hypothesis. Then $\alpha(F^{\delta+1}) = \alpha(F(F^{\delta})) \mathrel{\bar{\sqsubseteq}} I$ by II.3.4.(2). If λ is limit ordinal and $\forall \beta < \lambda \,.\, \alpha(F^{\beta}) \mathrel{\bar{\sqsubseteq}} I$. Then $\alpha(F^{\lambda}) =$

$\alpha(\bigsqcup_{\beta<\lambda} F^{\beta}) \mathrel{\dot{\sqsubseteq}} I$ by II.3.4.(3). By transfinite induction $\forall \delta \in \mathbb{O} \mathrel{.} \alpha(F^{\delta}) \mathrel{\dot{\sqsubseteq}} I$. So $\alpha(\mathsf{lfp}^{\sqsubseteq} F) = \alpha(F^{\epsilon}) \mathrel{\dot{\sqsubseteq}} I \mathrel{\dot{\sqsubseteq}} P$ by II.3.4.(4).

Completeness ($\Rightarrow$), for the weaker condition II.3.4.(2) and II.3.4.(3). Assume that $\alpha(\mathsf{lfp}^{\sqsubseteq} F) \mathrel{\dot{\sqsubseteq}} P$ and define $I = \alpha(\mathsf{lfp}^{\sqsubseteq} F)$, $\langle X^{\delta}, \delta \in \mathbb{O}\rangle = \langle F^{\delta}, \delta \in \mathbb{O}\rangle$. We have II.3.4.(1) $\alpha(\bot) \mathrel{\dot{\sqsubseteq}} \alpha(\mathsf{lfp}^{\sqsubseteq} F) = I$ since $\bot$ is the infimum and α is increasing. For II.3.4.(2) with $X = F^{\delta}$ we have $F^{\delta} \sqsubseteq \mathsf{lfp}^{\sqsubseteq} F$ so $\alpha(F(X)) = \alpha(F(F^{\delta})) = \alpha(F^{\delta+1}) \mathrel{\dot{\sqsubseteq}} \alpha(\mathsf{lfp}^{\sqsubseteq} F) = I$. For II.3.4.(3) with $\langle X^{\delta}, \delta \in \mathbb{O}\rangle = \langle F^{\delta}, \delta \in \mathbb{O}\rangle$ we have $\bigsqcup_{\beta<\lambda} X^{\beta} \sqsubseteq \mathsf{lfp}^{\sqsubseteq} F$ so $\alpha(\bigsqcup_{\beta<\lambda} X^{\beta}) \mathrel{\dot{\sqsubseteq}} \alpha(\mathsf{lfp}^{\sqsubseteq} F) = I$. Finally II.3.4.(4), by hypothesis $I = \alpha(\mathsf{lfp}^{\sqsubseteq} F) \mathrel{\dot{\sqsubseteq}} P$. □

Proof of Th. II.3.6. Since f in increasing on L, the iterates $I^0 = \bot$, $I^{\delta+1} = f(I^{\delta})$, and $I^{\lambda} = \bigsqcup_{\beta<\lambda} I^{\beta}$ are an increasing chain in L so, its cardinality being bounded by that of L, the sequence must be ultimately stationary at rank ϵ, to a fixpoint I^{δ} for all $\delta \geqslant \epsilon$, which, f being increasing, is the least one [Cousot and Cousot 1979a]. By transfinite induction, $\langle X^{\delta}, \delta \in \mathbb{O}\rangle$ is pointwise bounded by $\langle I^{\delta}, \delta \in \mathbb{O}\rangle$ i.e. $\forall \delta \in \mathbb{O} \mathrel{.} X^{\delta} \sqsubseteq I^{\delta}$. Therefore, $\langle X^{\delta}, \delta \in \mathbb{O}\rangle$ being increasing, we have $\exists \delta \in \mathbb{O} \mathrel{.} P \sqsubseteq X^{\delta} \sqsubseteq I^{\delta} \sqsubseteq I^{\max(\delta,\epsilon)} = \mathsf{lfp}^{\sqsubseteq} f$. Conversely, choosing $\langle X^{\delta}, \delta \in \mathbb{O}\rangle = \langle I^{\delta}, \delta \in \mathbb{O}\rangle$ will do since $P \sqsubseteq \mathsf{lfp}^{f} = I^{\epsilon}$. □

Proof of lemma II.3.7. Let $\langle X^{\delta}, \delta \in \mathbb{O}\rangle$ be a sequence in a complete lattice $\langle L, \sqsubseteq, \bot, \sqcup\rangle$ satisfying the hypotheses of Def. II.3.5. Define $Y^0 = \bot$, $Y^{\delta+1} = f(Y^{\delta})$, and $Y^{\lambda} = \bigsqcup_{\beta<\lambda} Y^{\beta}$ which is well defined in the complete lattice L and, by transfinite induction, increasing since for the basis $X^0 = \bot \sqsubseteq X^1$ and for the induction f is increasing [Cousot and Cousot 1979a]. To prove that this sequence $\langle Y^{\delta}, \delta \in \mathbb{O}\rangle$ is an upper bound of $\langle X^{\delta}, \delta \in \mathbb{O}\rangle$, observe, for the basis, that $X^0 = \bot \sqsubseteq \bot = Y^0$ by reflexivity and definition of the iterates. Assume $X^{\delta} \sqsubseteq Y^{\delta}$ by induction hypothesis. Then $X^{\delta+1} \sqsubseteq f(X^{\delta}) \sqsubseteq f(Y^{\delta}) = Y^{\delta+1}$ since f is increasing and by definition of the iterates. Moreover, $\forall \beta < \lambda \mathrel{.} X^{\beta} \sqsubseteq Y^{\beta}$ implies $\forall \beta < \lambda \mathrel{.} X^{\beta} \sqsubseteq \bigsqcup_{\beta<\lambda} Y^{\beta}$ implies $X^{\lambda} \sqsubseteq \bigsqcup_{\beta<\lambda} X^{\beta} \sqsubseteq \bigsqcup_{\beta<\lambda} Y^{\beta} = Y^{\lambda}$ by definition of the least upper bound and the iterates. By transfinite induction, $\forall \delta \in \mathbb{O} \mathrel{.} X^{\delta} \sqsubseteq Y^{\delta}$. By hypothesis $\exists \delta \in \mathbb{O} \mathrel{.} P \sqsubseteq X^{\delta}$ so $\exists \delta' \in \mathbb{O} \mathrel{.} P \sqsubseteq Y^{\delta'}$ with $\delta' = \delta$, $X^{\delta} \sqsubseteq Y^{\delta}$, and transitivity. In conclusion, $\langle Y^{\delta}, \delta \in \mathbb{O}\rangle$ is well-defined, increasing, and satisfies Def. II.3.5. □

Proof of Th. II.3.8. Observe that, by hypothesis II.3.8.(1), the transfinite iterates $f^0 = \bot$, $f^{\delta+1} = f(f^{\delta})$, and $f^{\lambda} = \bigsqcup_{\beta<\lambda} f^{\beta}$ of f, starting at $\bot$, are increasing and converging at rank $\epsilon \in \mathbb{O}$ to $\mathsf{lfp}^{\sqsubseteq} f$ [Cousot and Cousot 1979a]. Consider $\langle X^{\delta}, \delta \leqslant \epsilon\rangle$ as defined by hypothesis II.3.8.(3). Since f is increasing, it follows, by transfinite induction, that $\forall \delta \leqslant \epsilon \mathrel{.} X^{\delta} \sqsubseteq f^{\delta}$. If $\exists \delta \in \mathbb{O} \mathrel{.} P \sqsubseteq X^{\delta}$ then we are done since $P \sqsubseteq X^{\delta} \sqsubseteq f^{\delta} \sqsubseteq \mathsf{lfp}^{\sqsubseteq} f$. Otherwise, $\forall \delta \in \mathbb{O} \mathrel{.} P \not\sqsubseteq X^{\delta}$ and so, by hypothesis II.3.8.(4), the chain $\langle \nu(X^{\delta}), \delta \in \mathbb{O}\rangle$ is strictly $\prec$-decreasing, in contradiction with the hypothesis that $\langle W, \preceq\rangle$ is well-founded, proving $\exists \delta \in \mathbb{O} \mathrel{.} P \sqsubseteq X^{\delta} \sqsubseteq f^{\delta} \sqsubseteq \mathsf{lfp}^{\sqsubseteq} f$. Notice that, by well-foundedness, $\langle \nu(X^{\delta}), \delta \in \mathbb{O}\rangle$ reaches a minimal element at some rank less than ω so $\delta < \omega$. □

Proof of Th. II.3.10. By refth:Intersection-Lfp.(2), f preserves non-empty joins implies that the iterates $f^0(\bot) \triangleq \bot$ and $f^{n+1}(\bot) \triangleq f(f^n(\bot))$ of f from the infimum $\bot$ on the L form an increasing chain which limit, by refth:Intersection-Lfp.(1) exists in the complete lattice L, and is $\mathsf{lfp}^{\sqsubseteq} f = f^{\omega}(\bot) \triangleq \bigsqcup_{n\in\mathbb{N}} f^n(\bot)$ [Cousot and Cousot 1979a]. Moreover, by refth:Intersection-Lfp.(3), $f(I) \sqsubseteq I$ implies, by recurrence, that $\forall n \in \mathbb{N} \cup \{\omega\} \mathrel{.} f^n(\bot) \sqsubseteq I$ since by refth:Intersection-Lfp.(2) f preserves non-empty joins so is increasing.

If $f(\bot) = \bot$ then obviously $\mathsf{lfp}^{\sqsubseteq} f = \bot = \bot \sqcap Q$. So in the following, we assume that $\bot \neq f(\bot)$ that is, $\bot \sqsubset f(\bot)$ by definition of the infimum $\bot$.

In an atomic complete lattice, all elements have a non-empty set of atoms but $\bot$. Since $\bot \neq f(\bot)$, we have $\mathsf{atoms}(f(\bot)) \neq \emptyset$.

We have $\bot = f^0(\bot) \sqsubset f^1(\bot) = \bigsqcup\{f^0(x) \mid x \in \mathsf{atoms}(f(\bot))\}$ by definition of the identity f^0. Assume that, for $n \geqslant 1$, $f^n(\bot) = \bigsqcup\{f^{n-1}(x) \mid x \in \mathsf{atoms}(f(\bot))\}$ by recurrence hypothesis. By refth:Intersection-Lfp.(2), f

preserves non-empty joins so $f^{n+1}(\bot) = f(f^n(\bot)) = f(\bigsqcup\{f^{n-1}(x) \mid x \in \text{atoms}(f(\bot))\}) = \bigsqcup\{f(f^{n-1}(x)) \mid x \in \text{atoms}(f(\bot))\} = \bigsqcup\{f^n(x) \mid x \in \text{atoms}(f(\bot))\}$. By recurrence, $\forall n \geqslant 1 \,.\, f^n(\bot) = \bigsqcup\{f^{n-1}(x) \mid x \in \text{atoms}(f(\bot))\}$.

By refth:Intersection-Lfp.(1) and definition of an atom $x \in \text{atoms}(f(\bot))$, we have $\bot \sqsubset x \sqsubseteq f(\bot)$ that is $f^0(\bot) \sqsubset f^0(x) \sqsubseteq f(\bot)$. Since, by refth:Intersection-Lfp.(2), f preserves non-empty joins it is increasing and therefore $\forall n \in \mathbb{N} \,.\, f^n(\bot) \sqsubseteq f^n(x) \sqsubseteq f^{n+1}(\bot)$ by recurrence. By definition of a least upper bound, it follows that $\bigsqcup_{n\in\mathbb{N}} f^n(\bot) \sqsubseteq \bigsqcup_{n\in\mathbb{N}} f^n(x) \sqsubseteq \bigsqcup_{n\in\mathbb{N}} f^{n+1}(\bot) = \bigsqcup_{n\in\mathbb{N}} f^n(\bot)$ since $\bot \sqsubseteq f(\bot) = f^1(\bot)$. This implies that $\forall x \in \text{atoms}(f(\bot)) \,.\, \bigsqcup_{n\in\mathbb{N}} f^n(x) = \text{lfp}^{\sqsubseteq} f$.

Because $\forall n \geqslant 1 \,.\, f^n(x) = f(f^{n-1}(x)) \sqsubseteq I$, refth:Intersection-Lfp.(4) implies that the chain $\langle \nu(f^n(x)),\ n \geqslant 1\rangle$ in the well-founded set $\langle W,\ \preceq\rangle$ cannot infinitely decrease so is ultimately stationary at some $n = \ell_x$. By contraposition of refth:Intersection-Lfp.(4), $\forall y \sqsubseteq I \,.\, (\nu(y) \not\succ \nu(f(y))) \implies (y = f(y)) \sqsubseteq I$ so that $f^{\ell_x}(x) = f^m(x) \sqsubseteq I$ for all $m \geqslant \ell_x$. By refth:Intersection-Lfp.(5) and refth:Intersection-Lfp.(6), it follows that $\forall m \geqslant \ell_x \,.\, f^m(x) \sqcap Q = \bot$

It follows, by definition of the least upper bound, that $\text{lfp}^{\sqsubseteq} f = \bigsqcup_{n\in\mathbb{N}} f^n = \bigsqcup_{n\geqslant 1} \bigsqcup\{f^{n-1}(x) \mid x \in \text{atoms}(f(\bot))\} = \bigsqcup\{\bigsqcup_{n\geqslant 1} f^{n-1}(x) \mid x \in \text{atoms}(f(\bot))\} = \bigsqcup\{f^{\ell_x}(x) \mid x \in \text{atoms}(f(\bot))\}$. We have shown that $\forall x \in \text{atoms}(f(\bot)) \,.\, f^{\ell_x}(x) \sqcap Q = \bot$ so $\text{lfp}^{\sqsubseteq} f \sqcap Q = \bigsqcup\{f^{\ell_x}(x) \mid x \in \text{atoms}(f(\bot))\} \sqcap Q = \bot$ in the atomic complete lattice L. □

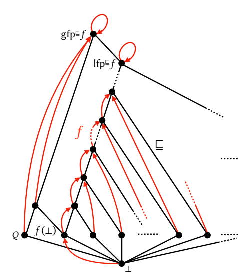


Fig. 5. Counter-example to the completeness of Th. II.3.10

Example H.1 (Counter-example to the completeness of Th. II.3.10). The iterates $\langle f^n(\bot),\ n \in \mathbb{N}\rangle$ of f from $\bot$ are the same as those for the atoms of $f(\bot)$ which are nothing but $f(\bot)$ itself and are infinitely strictly increasing, so the hypothesis (4) of Th. II.3.10 cannot be satisfied since it would imply convergence in finitely many steps. ■

Example H.2 (Transfinite variant function). The construction of the variant function ν for the program `x=[0,∞]; while (x>0) {x=x-1; y=[0,∞]; while (y>0) y=y-1;}` with unbounded nondeterminism is illustrated on Fig. 6. ■

Proof of Th. II.3.11. For soundness ($\supseteq$), we look for $P \cap \neg\text{pre}(r)\{\bot\} = P \cap \widetilde{\text{pre}}(r)\Sigma$ and apply the order dual of Th. II.3.8 observing that $I \subseteq P \cap \widetilde{\text{pre}}(r)I$ is equivalent to $P \cup \text{post}(r)I \subseteq I$.

For completeness ($\supseteq$), we let $r^* = \text{lfp}^{\subseteq} \boldsymbol{\lambda} X \bullet \text{id} \cup X \circ r$ and choose $W = I = \text{post}(r^*)P$, $\nu(x) = x$, and $x \succ y \triangleq (x \in W \wedge \langle x,\ y\rangle \in r)$. We must prove the following three conditions (we write $a \xrightarrow{r} b$ for $\langle a,\ b\rangle \in r$).

(1) $\langle W,\ \preceq\rangle \in \mathfrak{Wf}$.

By reductio ad absurdum, assume that there exists an infinite sequence strictly decreasing sequence $x_0 \succ x_1 \succ \ldots$ of elements of $W = I$. By definition of W, $x_0 \in \text{post}(r^*)P$ so that there exists a finite sequence $y_1 \in P$, $y_1 \xrightarrow{r} y_2 \xrightarrow{r} \ldots \xrightarrow{r} y_{n-1} \xrightarrow{r} y_n$ with $y_i \in W = I$, $i \in [1, n]$, $n \geqslant 0$ (0 if $x_0 \in P$) and $y_n = x_0$. By definition of $x \succ y$ implying $x \xrightarrow{r} y$, there exists an infinite sequence $y_1 \in P$, $y_1 \xrightarrow{r} y_2 \xrightarrow{r} \ldots \xrightarrow{r} y_{n-1} \xrightarrow{r} x_0 \xrightarrow{r} x_1 \xrightarrow{r} \ldots$ in contradiction with $\neg(\exists \sigma \in \mathbb{N} \to \mathcal{X} \,.\, \sigma_0 = x \in P \wedge \forall i \in \mathbb{N} \,.\, \langle \sigma_i,\ \sigma_{i+1}\rangle \in r)$.

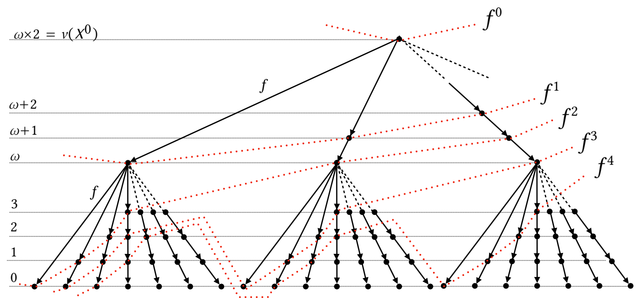


Fig. 6. Construction of the variant function ν

(2) By the fixpoint abstraction Th. II.2.1, the abstraction of $r^* = \mathsf{lfp}^{\subseteq} \boldsymbol{\lambda} X \bullet \mathsf{id} \cup X \circ r$ by $\boldsymbol{\lambda} X \bullet \mathsf{post}(X)P$ is $W = I = \mathsf{post}(r^*)P = \mathsf{lfp}^{\subseteq} \boldsymbol{\lambda} X \bullet \mathsf{P} \cup \mathsf{post}(r)X$ so that $I = \mathsf{P} \cup \mathsf{post}(r)I$ hence $P \cup \mathsf{post}(r)I \subseteq I$ by reflexivity.

(3) if $y \in W = I = \mathsf{post}(r^*)P$ and $y' \in \mathcal{X}$ is such that $\langle y, y'\rangle \in r$ then $y' \in \mathsf{post}(r^*)P = I = W$. Then $(y \in W \wedge \langle y, y'\rangle \in r)$ implies $y \succ y'$ by definition of $\succ$ and therefore $\nu(y) \succ \nu(y')$ by definition of ν. $\square$

Proof of Th. II.3.12. By (2), f preserves non-empty meets implies that the iterates $f^0(\top) \triangleq \top$ and $f^{n+1}(\bot) \triangleq f(f^n(\bot))$ of f from the supremum $\top$ on the L form an decreasing chain which limit, by (1) exists in the complete lattice L, and is $\mathsf{gfp}^{\sqsubseteq} f = f^{\omega}(\bot) \triangleq \bigsqcap_{n\in\mathbb{N}} f^n(\top)$ [Cousot and Cousot 1979a]. Moreover, by (3), $I \sqsubseteq f(I)$ implies, by recurrence, that $\forall n \in \mathbb{N} \cup \{\omega\} \;.\; I \sqsubseteq f^n(\top)$ since by (2) f preserves non-empty meets so is increasing.

If $f(\top) = \top$ then obviously $\mathsf{gfp}^{\sqsubseteq} f = \top$. By (6) $\forall \top \sqsupseteq I$ and $(\nu(\top) \succ \nu(f(\top)))$ implies $(\top \sqcap Q = \bot)$. So in the following, we assume that $\top \neq f(\top)$ that is, $\top \sqsupset f(\top)$ by definition of the supremum $\top$.

In a coatomic complete lattice, all elements have a non-empty set of coatoms but $\top$. Since $\top \neq f(\top)$, we have $\mathsf{coatoms}(f(\top)) \neq \varnothing$.

We have $\top = f^0(\top) \sqsupset f^1(\top) = \bigsqcap\{f^0(x) \mid x \in \mathsf{coatoms}(f(\top))\}$ by definition of the identity f^0. Assume that, for $n \geqslant 1$, $f^n(\top) = \bigsqcap\{f^{n-1}(x) \mid x \in \mathsf{coatoms}(f(\top))\}$ by recurrence hypothesis. By (2), f preserves non-empty meets so $f^{n+1}(\top) = f(f^n(\top)) = f(\bigsqcap\{f^{n-1}(x) \mid x \in \mathsf{coatoms}(f(\top))\}) = \bigsqcap\{f(f^{n-1}(x)) \mid x \in \mathsf{coatoms}(f(\top))\} = \bigsqcap\{f^n(x) \mid x \in \mathsf{coatoms}(f(\top))\}$. By recurrence, $\forall n \geqslant 1 \;.\; f^n(\top) = \bigsqcup\{f^{n-1}(x) \mid x \in \mathsf{coatoms}(f(\top))\}$.

By (1) and definition of a coatom $x \in \mathsf{coatoms}(f(\top))$, we have $\top \sqsupset x \sqsupset f(\top)$ that is $f^0(\top) \sqsupset f^0(x) \sqsupset f(\top)$. Since, by (2), f preserves non-empty meets it is increasing and therefore $\forall n \in \mathbb{N} \;.\; f^n(\top) \sqsupseteq f^n(x) \sqsupseteq f^{n+1}(\top)$ by recurrence. By definition of a greatest lower bound, it follows that $\bigsqcap_{n\in\mathbb{N}} f^n(\jmath) \sqsupseteq \bigsqcap_{n\in\mathbb{N}} f^n(x) \sqsubseteq \bigsqcap_{n\in\mathbb{N}} f^{n+1}(\top) = \bigsqcap_{n\in\mathbb{N}} f^n(\top)$ since $\jmath \sqsupseteq f(\jmath) = f^1(\jmath)$. This implies that $\forall x \in \mathsf{coatoms}(f(\jmath)) \;.\; \bigsqcap_{n\in\mathbb{N}} f^n(x) = \mathsf{gfp}^{\sqsubseteq} f$.

Because $\forall n \geqslant 1 \;.\; f^n(x) = f(f^{n-1}(x)) \sqsupseteq I$, (4) implies that the chain $\langle \nu(f^n(x)),\ n \geqslant 1\rangle$ in the well-founded set $\langle W, \preceq\rangle$ cannot infinitely decrease so is ultimately stationary at some $n = \ell_x$. By contraposition of (4), $\forall y \sqsupseteq I \;.\; (\nu(y) \not\succ \nu(f(y))) \implies (y = f(y)) \sqsupseteq I$ so that $f^{\ell_x}(x) = f^m(x) \sqsupseteq I$ for all $m \geqslant \ell_x$. By (5) and (6), it follows that $\forall m \geqslant \ell_x \;.\; f^m(x) \sqcap Q = \bot$

It follows, by definition of the greatest lower bound, that $\mathsf{gfp}^{\sqsubseteq} f = \bigsqcap_{n\in\mathbb{N}} f^n = \bigsqcap_{n\geqslant 1} \bigsqcap\{f^{n-1}(x) \mid x \in \mathsf{atoms}(f(\top))\} = \bigsqcap\{\bigsqcap_{n\geqslant 1} f^{n-1}(x) \mid x \in \mathsf{atoms}(f(\top))\} = \bigsqcap\{f^{\ell_x}(x) \mid x \in \mathsf{coatoms}(f)(\top)\}$. We have shown that $\forall x \in \mathsf{coatoms}(f(\top)) \;.\; f^{\ell_x}(x) \sqcap Q = \bot$ so $\mathsf{lfp}^{\sqsubseteq} f \sqcap Q = \bigsqcap\{f^{\ell_x}(x) \mid x \in \mathsf{coatoms}(f(\top))\} \sqcap Q = \bot$ in the coatomic complete lattice L. $\square$

Proof of Th. II.3.13. The transfinite iterates $X^0 = \top$, $X^{\delta+1} = f(X^{\delta})$, and $X^{\lambda} = \bigsqcap_{\beta<\lambda} X^{\beta}$ of f from $\top$ are decreasing, ultimately stationary, with limit $\mathsf{gfp}^{\sqsubseteq} f$ [Cousot and Cousot 1979a], so $\forall \delta \in \mathbb{O} \;.\; \mathsf{gfp}^{\sqsubseteq} f \sqsubseteq X^{\delta}$.

For soundness, we have $X^0 \sqcap P \neq \bot$ since $P \neq \bot$. If $X^\delta \sqcap P \neq \bot$ then $f(X^\delta) = X^{\delta+1} \sqsubseteq X^\delta$ implies $f(X^\delta) \sqcap P \neq \bot$ by hypothesis, that is, $X^{\delta+1} \sqcap P \neq \bot$. If, by induction hypothesis, $X^\beta \sqcap P \neq \bot$ for all $\beta < \lambda$ then $X^\lambda \sqcap P = (\bigsqcap_{\beta<\lambda} X^\beta) \sqcap P = \bigsqcap_{\beta<\lambda}(X^\beta \sqcap P) \neq \bot$ since $\langle X^\beta,\ \beta < \lambda\rangle$ is decreasing and so is $\langle X^\beta \sqcap P,\ \beta < \lambda\rangle$. By transfinite induction, we conclude that $\forall \delta \in \mathbb{O}\ .\ X^\delta \sqcap P \neq \bot$ and so $\mathsf{gfp}^{\sqsubseteq} f \sqcap P \neq \bot$.

For completeness, $\mathsf{gfp}^{\sqsubseteq} f \sqcap P \neq \bot$ then $P \neq \bot$. If $\mathsf{gfp}^{\sqsubseteq} f \sqsubseteq X \wedge f(X) \sqsubseteq X \wedge X \sqcap P \neq \bot$ then $\mathsf{gfp}^{\sqsubseteq} f = f(\mathsf{gfp}^{\sqsubseteq}) \sqsubseteq f(X)$ since f is increasing so $\mathsf{gfp}^{\sqsubseteq} f \sqcap P \neq \bot$ implies $f(X) \sqcap P \neq \bot$. ◻

Theorem H.3 (Non empty intersection with abstraction of least fixpoint). *Assume that (1) $\langle L, \sqsubseteq, \bot, \top, \sqcap, \sqcup\rangle$ is an atomic complete lattice; (2) $f \in L \to L$ preserves nonempty joins $\sqcup$; (3) $\langle L, \sqsubseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \bar{L}, \preceq, \curlywedge\rangle$; (4) $\bar{Q} \in \bar{L} \setminus \{0\}$ where $0 \triangleq \alpha(\bot)$; (5) There exists an inductive invariant $I \in L$ of f (i.e. $f(I) \sqsubseteq I$); (6) $\langle W, \leqslant\rangle$ is a well-founded set and $\nu \in \mathsf{atoms}(I) \to W$ is a (variant) function; (7) There exists a sequence $\langle a_i \in \mathsf{atoms}(I), i \in [1,\infty]\rangle$ that (7.a) $a_1 \in f(\bot)$, (7.b) $\forall i \in [1,\infty]\ .\ a_{i+1} \in \mathsf{atoms}(f(a_i))$, (7.c) $\forall i \in [1,\infty]\ .\ (a_i \neq a_{i+1}) \Rightarrow (\nu(a_i) > \nu(a_{i+1})$, (7.d) $\forall i \in [1,\infty]\ .\ (\nu(a_i) \not> \nu(a_{i+1})) \Rightarrow \alpha(a_i) \curlywedge \bar{Q} \neq 0$; Then, hypotheses (1) to (7) imply $\alpha(\mathsf{lfp}^{\sqsubseteq} f) \curlywedge \bar{Q} \neq 0$. Conversely (1) to (4) and $\mathsf{lfp}^{\sqsubseteq} f \sqcap \gamma(\bar{Q}) \neq \bot$ imply (5) to (7).*

Notice that if $L = \wp(\Sigma)$ then $\mathsf{atoms}(L) = \{\{x\} \mid x \in L\}$ so that $I \in \wp(\Sigma)$ and ν can be chosen in $I \to W$ instead of $\{\{x\} \mid x \in I\} \to W$.

Proof of Th. 6. By (1) and (2), $\mathsf{lfp}^{\sqsubseteq} f = \bigsqcup_{n\in\mathbb{N}} f^n(\bot)$ where the iterates of f from $x \in L$ are $f^0(x) = x$ and $f^{n+1}(x) = f(f^n(x))$ [Cousot and Cousot 1979a]. By (5), $f(I) \sqsubseteq I$ so that $\mathsf{lfp}^{\sqsubseteq} f \sqsubseteq I$ by Tarski's fixpoint theorem [Tarski 1955]. Consider $\langle a_i \in \mathsf{atoms}(I),\ i \in [1,\infty]\rangle$. By (7.a), $a_1 \in f(\bot)$ so $a_1 \in \mathsf{atoms}(f^1(\bot))$. Assume $a_n \in \mathsf{atoms}(f^n(\bot))$ so that $a_n \sqsubseteq f^n(\bot)$. By (2), f is increasing so $f(a_n) \sqsubseteq f(f^n(\bot)) = f^{n+1}(\bot)$. By (7.b), $a_{n+1} \in \mathsf{atoms}(f(a_n)) \subseteq \mathsf{atoms}(f^{n+1}(\bot))$. By recurrence $\forall n \in \mathbb{N}\ .\ a_n \in \mathsf{atoms}(f^n(\bot))$. This implies $a_n \sqsubseteq f^n(\bot) \sqsubseteq \bigsqcup_{n\in\mathbb{N}} f^n(\bot) = \mathsf{lfp}^{\sqsubseteq} f \subseteq I$ so that $a_n \in \mathsf{atoms}(I)$ proving that $\nu(a_n)$ is well-defined for all $n \in \mathbb{N}$. By (6), the sequence $\langle \nu(a_n),\ n \in \mathbb{N}\rangle$ cannot be strictly $>$-decreasing. So there is some $\ell \in \mathbb{N}$ such that $\nu(a_\ell) = \nu(a_{\ell+1})$. By (7.d), this implies that $\alpha(a_\ell) \curlywedge \bar{Q} \neq 0$. By (3), $\langle \alpha(L), \preceq, 0, 1, \curlywedge, \curlyvee\rangle$ is a complete lattice. We have $\alpha(\mathsf{lfp}^{\sqsubseteq} f) = \alpha(\bigsqcup_{n\in\mathbb{N}} f^n(\bot)) = \bigcurlyvee_{n\in\mathbb{N}} \alpha(f^n(\bot)) \succeq \alpha(f^\ell(\bot)) \succeq \alpha(a_\ell)$ so that $\alpha(a_\ell) \curlywedge \bar{Q} \neq 0$ implies $\alpha(\mathsf{lfp}^{\sqsubseteq} f) \curlywedge \bar{Q} \neq 0$.

Conversely assume (1) to (4) and $\mathsf{lfp}^{\sqsubseteq} f \sqcap \gamma(\bar{Q}) \neq \bot$. Let x_0 be an atom common to $\mathsf{lfp}^{\sqsubseteq} f$ and $\gamma(\bar{Q})$ so that $x_0 \in \mathsf{atoms}(f^n(\bot))$ for some $n > 0$. Assume we have constructed $x_0, \ldots, x_{n-i}$, $0 < i \leqslant n$ such that $\forall k \in [0,i]\ .\ x_k \in \mathsf{atoms}(f^{n-k})$ which elements are two by two different. There are two cases.

(1) If $x_{n-i} \in f(\bot)$ then define the finite sequence $a_1 = x_{n-i}$, $a_2 = x_{n-i+1}$, ..., $a_{n-i+1} = x_0$. Define $I = \mathsf{lfp}^{\sqsubseteq} f$, $\langle W, \leqslant\rangle = \langle [1, n-i+1], \leqslant\rangle$, $\nu(x) = (\!| x = a_i \mathrel{?} i \mathrel{¿} 1 |\!)$, which is well-founded since the elements of a_1, ..., a_{n-i+1} are two by two different. Then (5) to (7) are satisfied, Q.E.D.
(2) Otherwise $x_{n-i} \notin f(\bot)$ and $x_{n-i} \in \mathsf{atoms}(f^{n-i}) = \mathsf{atoms}(f(f^{n-i-1}))$. Pick x_{n-i-1} as an atom of f^{n-i-1} different from x_0, ..., x_{n-i}. Notice that if there no such x_{n-i-1}, we are in the previous case (1). This extends the sequence by one element, and we must terminate ultimately at $f^1(\bot)$ for which case (1) concludes. ◻

I AUXILIARY MATERIAL FOR SECTION II.5 (THE SEMANTICS OF DEDUCTIVE SYSTEMS)

Example I.1 (Design of the deductive natural relational semantics). The rule-based deductive natural relational semantics of Sect. I.1.1 is derived from its fixpoint definition of Sect. II.1, by structural induction. The base cases in (47) are understood as constant fixpoints $S = \mathsf{lfp}^{\sqsubseteq} \boldsymbol{\lambda} X \bullet S$ so that Sect. II.5.2 yields axioms. For the assignment, we get $\sigma \vdash \mathtt{x} = \mathtt{A} \overset{e}{\Rightarrow} \sigma[\mathtt{x} \leftarrow \mathcal{A}[\![\mathtt{A}]\!]\sigma]$. Since there are no rules for $\overset{b}{\Rightarrow}$ and $\overset{\infty}{\Rightarrow}$, $[\![\mathtt{x} = \mathtt{A}]\!]^b$ and $[\![\mathtt{x} = \mathtt{A}]\!]^\infty$ are empty.

For the induction cases, consider for example $[\![\mathsf{S}_1 ; \mathsf{S}_2]\!]^e \triangleq [\![\mathsf{S}_1]\!]^e \mathbin{⨾} [\![\mathsf{S}_2]\!]^e$ in (48). By structural induction hypothesis and definition of $⨾$, we get $\dfrac{\sigma \vdash \mathsf{S}_1 \overset{e}{\Rightarrow} \sigma',\ \sigma' \vdash \mathsf{S}_2 \overset{e}{\Rightarrow} \sigma''}{\sigma \vdash \mathsf{S}_1 ; \mathsf{S}_2 \overset{e}{\Rightarrow} \sigma''}$ where the comma means conjunction. We are in the constant fixpoint case, so the rule is actually an axiom for $\mathsf{S}_1 ; \mathsf{S}_2$ and, more rigorously, the premiss should be a side condition.

For iteration $\mathtt{W} = \mathtt{while\ (B)\ S}$, $\llbracket \mathtt{W} \rrbracket^{e}$ in (51) involves $\mathsf{lfp}^{\subseteq} F^{e}$. In (2), we write $\sigma \vdash \mathtt{W} \stackrel{i}{\Rightarrow} \sigma'$ for $\langle \sigma, \sigma' \rangle \in \mathsf{lfp}^{\subseteq} F^{e}$. By $F^{e}(X) \triangleq \mathsf{id} \cup (\llbracket \mathtt{B} \rrbracket \fatsemi \llbracket \mathtt{S} \rrbracket^{e} \fatsemi (X \setminus \Sigma \times \{\bot\}))$ in (49), we decompose the union into two rules (i.e. $\frac{x \in X}{x \in X \cup Y}$ and $\frac{y \in Y}{y \in X \cup Y}$). This decomposition yields the axiom $\sigma \vdash \mathtt{W} \stackrel{i}{\Rightarrow} \sigma$ for id and $\frac{\mathcal{B}\llbracket \mathtt{B} \rrbracket \sigma, \quad \sigma \vdash \mathtt{S} \stackrel{e}{\Rightarrow} \sigma', \quad \sigma' \vdash \mathtt{W} \stackrel{i}{\Rightarrow} \sigma''}{\sigma \vdash \mathtt{W} \stackrel{i}{\Rightarrow} \sigma''}$ for $\llbracket \mathtt{B} \rrbracket \fatsemi \llbracket \mathtt{S} \rrbracket^{e} \fatsemi (X \setminus \Sigma \times \{\bot\})$. Then $\llbracket \mathtt{W} \rrbracket^{e} \triangleq \mathsf{lfp}^{\subseteq} F^{e} \fatsemi (\llbracket \neg\mathtt{B} \rrbracket \cup \llbracket \mathtt{B} \rrbracket \fatsemi \llbracket \mathtt{S} \rrbracket^{b})$ in (51) is handled like the sequential composition and union case to get (3).

For the case $\mathsf{gfp}^{\subseteq} F^{\bot}$ of non termination, we use the dual interpretation $\alpha^{\mathcal{I}}(R) = \mathsf{gfp}^{\subseteq} \alpha^{F}(R)$ of rules R so that $F^{\bot}(X) \triangleq \llbracket \mathtt{B} \rrbracket \fatsemi \llbracket \mathtt{S} \rrbracket^{e} \fatsemi X$ yields the coinductive rule $\frac{\mathcal{B}\llbracket \mathtt{B} \rrbracket \sigma, \quad \sigma \vdash \mathtt{S} \stackrel{e}{\Rightarrow} \sigma', \quad \sigma' \vdash \mathtt{W} \stackrel{\infty}{\Rightarrow}}{\sigma \vdash \mathtt{W} \stackrel{\infty}{\Rightarrow}}\infty$ of (4). The other nontermination rule $\frac{\sigma \vdash \mathtt{W} \stackrel{i}{\Rightarrow} \sigma', \quad \mathcal{B}\llbracket \mathtt{B} \rrbracket \sigma', \quad \sigma' \vdash \mathtt{S} \stackrel{\infty}{\Rightarrow}}{\sigma \vdash \mathtt{W} \stackrel{\infty}{\Rightarrow}}$ follows, by structural induction, from the term $\mathsf{lfp}^{\subseteq} F^{e} \fatsemi \llbracket \mathtt{B} \rrbracket \fatsemi \llbracket \mathtt{S} \rrbracket^{\bot}$ of the union. ■

J AUXILIARY MATERIAL FOR SECTION II.8 (APPLICATIONS)

J.1 Application 0: Calculational Design of Hoare Incorrectness Logic

We design by calculus the Hoare incorrectness logic of Ex. I.3.11 which theory is $\mathcal{T}_{\overline{\mathrm{HL}}}(\mathtt{S}) = \mathsf{post}(\subseteq, \supseteq) \circ \alpha^{\neg} \circ \mathcal{T}_{\mathrm{HL}}(\mathtt{S})$. The proof is by structural induction. We consider the case of the conditional iteration $\mathtt{W} = \mathtt{while\ (B)\ S}$ (without `break`, to simplify). All other cases are similar and simpler.

J.1.1 Strongest Postcondition Over Approximation. We start by characterizing the theory of classic Hoare logic.

Lemma J.1 (Strongest postcondition). $\mathcal{T}(\mathtt{S}) = \alpha_{G} \circ \mathsf{post}\llbracket \mathtt{S} \rrbracket = \{\langle P,\ \mathsf{post}\llbracket \mathtt{S} \rrbracket P \rangle \mid P \in \wp(\Sigma)\}$.

Proof of lem. J.1.

$$
\begin{array}{ll}
 & \mathcal{T}(\mathtt{S}) \\
= & \alpha_{G} \circ \mathsf{post} \circ \alpha_{\not\bot} \circ \alpha_{C}(\{\llbracket \mathtt{S} \rrbracket_{\bot}\}) \hfill \wr \text{def. } \mathcal{T} \int \\
= & \alpha_{G} \circ \mathsf{post} \circ \alpha_{\not\bot}(\llbracket \mathtt{S} \rrbracket_{\bot}) \hfill \wr \text{def. } \alpha_{C} \int \\
= & \alpha_{G} \circ \mathsf{post}(\llbracket \mathtt{S} \rrbracket_{\bot} \cap (\Sigma \times \Sigma)) \hfill \wr \text{def. } \alpha_{\not\bot} \int \\
= & \alpha_{G} \circ \mathsf{post}\llbracket \mathtt{S} \rrbracket \hfill \wr \text{def. (1) of the angelic semantics } \llbracket \mathtt{S} \rrbracket \int \\
= & \{\langle P,\ \mathsf{post}\llbracket \mathtt{S} \rrbracket P \rangle \mid P \in \wp(\Sigma)\} \hfill \wr \text{def. } \alpha_{G} \int \quad \square
\end{array}
$$

Lemma J.2 (Strongest postcondition over approximation).

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

Proof of Lem. J.2.

$$
\begin{array}{ll}
 & \mathsf{post}(\supseteq.\subseteq) \circ \mathcal{T}(\mathtt{S}) \\
= & \mathsf{post}(\supseteq.\subseteq)(\mathcal{T}(\mathtt{S})) \hfill \wr \text{def. function composition } \circ \int \\
= & \mathsf{post}(\supseteq.\subseteq)(\{\langle P,\ \mathsf{post}\llbracket \mathtt{S} \rrbracket P \rangle \mid P \in \wp(\Sigma)\}) \hfill \wr \text{lem. J.1} \int \\
= & \{\langle P',\ Q' \rangle \mid \exists \langle P,\ Q \rangle \in \{\langle P,\ \mathsf{post}\llbracket \mathtt{S} \rrbracket P \rangle \mid P \in \wp(\Sigma)\} \mathrel{.} \langle \langle P,\ Q \rangle,\ \langle P',\ Q' \rangle \rangle \in \supseteq.\subseteq\} \hfill \wr \text{def. (10) of post} \int \\
= & \{\langle P',\ Q' \rangle \mid \exists P \mathrel{.} \langle \langle P,\ \mathsf{post}\llbracket \mathtt{S} \rrbracket P \rangle,\ \langle P',\ Q' \rangle \rangle \in \supseteq.\subseteq\} \hfill \wr \text{def. } \in \int \\
= & \{\langle P',\ Q' \rangle \mid \exists P \mathrel{.} \langle P,\ \mathsf{post}\llbracket \mathtt{S} \rrbracket P \rangle \supseteq.\subseteq \langle P',\ Q' \rangle\} \hfill \wr \text{def. } \in \int \\
= & \{\langle P',\ Q' \rangle \mid \exists P \mathrel{.} P \supseteq P' \wedge \mathsf{post}\llbracket \mathtt{S} \rrbracket P \subseteq Q'\} \hfill \wr \text{def. } \supseteq.\subseteq \int \\
= & \{\langle P',\ Q' \rangle \mid \exists P \mathrel{.} P' \subseteq P \wedge \mathsf{post}\llbracket \mathtt{S} \rrbracket P \subseteq Q'\} \hfill \wr \text{def. } \supseteq \int \\
= & \{\langle P',\ Q' \rangle \mid \mathsf{post}\llbracket \mathtt{S} \rrbracket P' \subseteq Q'\}
\end{array}
$$

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

$= \{\langle P', Q'\rangle \mid \exists P \,.\, P' = P \wedge \mathsf{post}[\![\mathsf{S}]\!]P \subseteq Q'\}$ ⟨def. $=$⟩

$= \{\langle P', Q'\rangle \mid \exists P \,.\, \langle P, \mathsf{post}[\![\mathsf{S}]\!]P\rangle =,\subseteq \langle P', Q'\rangle\}$ ⟨def. $=,\subseteq$⟩

$= \{\langle P', Q'\rangle \mid \exists P \,.\, \langle\langle P, \mathsf{post}[\![\mathsf{S}]\!]P\rangle, \langle P', Q'\rangle\rangle \in =,\subseteq\}$ ⟨def. $\in$⟩

$= \{\langle P', Q'\rangle \mid \exists\langle P, Q\rangle \in \{\langle P, \mathsf{post}[\![\mathsf{S}]\!]P\rangle \mid P \in \wp(\Sigma)\} \,.\, \langle\langle P, Q\rangle, \langle P', Q'\rangle\rangle \in =,\subseteq\}$ ⟨def. $\in$⟩

$= \{\langle P', Q'\rangle \mid \exists\langle P, Q\rangle \in \mathcal{T}(\mathsf{S}) \,.\, \langle\langle P, Q\rangle, \langle P', Q'\rangle\rangle \in =,\subseteq\}$ ⟨lem. J.1⟩

$= \mathsf{post}(=,\subseteq)(\mathcal{T}(\mathsf{S}))$ ⟨def. (10) of post⟩

$= \mathsf{post}(=,\subseteq) \circ \mathcal{T}(\mathsf{S})$ ⟨def. function composition $\circ$⟩ □

J.1.2 *Calculational Design of Hoare Incorrectness Logic Theory.*

Theorem J.3 (Equivalent definitions of $\overline{\mathrm{HL}}$ theories).

$$\mathcal{T}_{\overline{HL}}(\mathtt{W}) \quad \triangleq \quad \mathsf{post}(\subseteq,\supseteq) \circ \alpha^{\neg} \circ \mathcal{T}_{HL}(\mathtt{W}) \quad = \quad \alpha^{\neg} \circ \mathcal{T}_{HL}(\mathtt{W}) \qquad \mathtt{W} = \mathtt{while\ (B)\ S}$$

Observe that Th. J.3 shows that $\mathsf{post}(\subseteq,\supseteq)$ can be dispensed with. This implies that the consequence rule is useless for Hoare incorrectness logic.

Proof of Th. J.3.

$\mathcal{T}_{\overline{\mathrm{HL}}}(\mathtt{W}) \quad = \quad \mathsf{post}(\subseteq,\supseteq) \circ \alpha^{\neg} \circ \mathcal{T}_{\mathrm{HL}}(\mathtt{W})$ ⟨def. $\mathcal{T}_{\overline{\mathrm{HL}}}$⟩

$= \mathsf{post}((\subseteq,\supseteq)(\neg\{\langle P, Q\rangle \mid \mathsf{post}[\![\mathtt{W}]\!]P \subseteq Q\})$

⟨lem. J.2 and def. (30) of $\alpha^{\neg}$⟩

$= \mathsf{post}(\subseteq,\supseteq)(\{\langle P, Q\rangle \mid \neg(\mathsf{post}[\![\mathtt{W}]\!]P \subseteq Q)\})$ ⟨def. $\neg$⟩

$= \mathsf{post}(\subseteq,\supseteq)(\{\langle P, Q\rangle \mid \mathsf{post}[\![\mathtt{W}]\!]P \cap \neg Q \neq \varnothing\})$ ⟨def. $\subseteq$ and $\neg$⟩

$= \{\langle P', Q'\rangle \mid \exists\langle P, Q\rangle \in \{\langle P, Q\rangle \mid \mathsf{post}[\![\mathtt{W}]\!]P \cap \neg Q \neq \varnothing\} \,.\, \langle P, Q\rangle \subseteq,\supseteq \langle P', Q'\rangle\}$ ⟨def. post⟩

$= \{\langle P', Q'\rangle \mid \exists\langle P, Q\rangle \,.\, \mathsf{post}[\![\mathtt{W}]\!]P \cap \neg Q \neq \varnothing \wedge \langle P, Q\rangle \subseteq,\supseteq \langle P', Q'\rangle\}$ ⟨def. $\in$⟩

$= \{\langle P', Q'\rangle \mid \exists\langle P, Q\rangle \,.\, \mathsf{post}[\![\mathtt{W}]\!]P \cap \neg Q \neq \varnothing \wedge P \subseteq P' \wedge Q \supseteq Q'\}$ ⟨component wise def. of $\subseteq,\supseteq$⟩

$= \{\langle P', Q'\rangle \mid \exists Q \,.\, \mathsf{post}[\![\mathtt{W}]\!]P' \cap \neg Q \neq \varnothing \wedge Q \supseteq Q'\}$

⟨($\subseteq$) if $P \subseteq P'$ then $\mathsf{post}[\![\mathtt{W}]\!]P \subseteq \mathsf{post}[\![\mathtt{W}]\!]P'$ by (12) so that $\mathsf{post}[\![\mathtt{W}]\!]P \cap \neg Q \neq \varnothing$ implies $\mathsf{post}[\![\mathtt{W}]\!]P' \cap \neg Q \neq \varnothing$;

($\supseteq$) conversely, if $\exists Q \,.\, \mathsf{post}[\![\mathtt{W}]\!]P'$, then $\exists P \,.\, \mathsf{post}[\![\mathtt{W}]\!]P \cap \neg Q \neq \varnothing \wedge P \subseteq P'$ by choosing $P = P'$. ⟩

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

⟨($\subseteq$) if $Q \supseteq Q'$ then $\neg Q' \supseteq \neg Q$ so $\mathsf{post}[\![\mathtt{W}]\!]P' \cap \neg Q \neq \varnothing$ implies $\mathsf{post}[\![\mathtt{W}]\!]P' \cap \neg Q' \neq \varnothing$;

($\supseteq$) conversely $\mathsf{post}[\![\mathtt{W}]\!]P' \cap \neg Q' \neq \varnothing$ implies $\exists Q \,.\, \mathsf{post}[\![\mathtt{W}]\!]P' \cap \neg Q \neq \varnothing \wedge Q \supseteq Q'$ by choosing $Q = Q'$. ⟩

$= \{\langle P, Q\rangle \mid \neg(\mathsf{post}[\![\mathtt{W}]\!]P \subseteq Q)\}$ ⟨def. $\subseteq$ and $\neg$⟩

$= \alpha^{\neg} \circ \mathcal{T}_{\mathrm{HL}}(\mathtt{W})$ ⟨def. $\alpha^{\neg}$ and $\mathcal{T}_{\mathrm{HL}}$ for Hoare logic⟩ □

Theorem J.4 (Theory of $\overline{\mathrm{HL}}$).

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

Proof of Th. J.4.

$\mathcal{T}_{\overline{\mathrm{HL}}}(\mathtt{W})$

$= \{\langle P, Q\rangle \mid \mathsf{post}[\![\neg\mathtt{B}]\!](\mathsf{lfp}^{\subseteq} \bar{\bar{F}}^{e}_{P}) \cap \neg Q \neq \varnothing\}$ ⟨lem. J.1, where $\bar{\bar{F}}^{e}_{P}(X) \triangleq P \cup \mathsf{post}([\![\mathtt{B}]\!] \,⨾\, [\![\mathtt{S}]\!]^{e})X$ ⟩

$= \{\langle P, Q\rangle \mid \mathsf{lfp}^{\subseteq} \bar{\bar{F}}^{e}_{P} \cap \mathsf{pre}[\![\neg\mathtt{B}]\!](\neg Q) \neq \varnothing\}$ ⟨(39.d)⟩

$= \{\langle P, Q\rangle \mid \exists I \in \wp(\Sigma) \,.\, \bar{\bar{F}}^{e}_{P}(I) \subseteq I \wedge \exists\langle W, \leqslant\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in I \to W \,.\, \exists\langle\sigma_i \in I,\ i \in [1, \infty]\rangle \,.\, \sigma_1 \in \bar{\bar{F}}^{e}_{P}(\varnothing) \wedge \forall i \in [1, \infty] \,.\, \sigma_{i+1} \in \bar{\bar{F}}^{e}_{P}(\{\sigma_i\}) \wedge \forall i \in [1, \infty] \,.\, (\sigma_i \neq \sigma_{i+1}) \Rightarrow (\nu(\sigma_i) > \nu(\sigma_{i+1}) \wedge \forall i \in [1, \infty] \,.\, (\nu(\sigma_i) \not> \nu(\sigma_{i+1}) \Rightarrow \{\sigma_i\} \cap \mathsf{pre}[\![\neg\mathtt{B}]\!](\neg Q) \neq 0\}$ ⟨Th. H.3⟩

$= \{\langle P,\ Q\rangle \mid \exists I \in \wp(\Sigma)\ .\ P \subseteq I \wedge \mathsf{post}(\llbracket \mathsf{B}\rrbracket \mathbin{;} \llbracket \mathsf{S}\rrbracket^{e})I \subseteq I \wedge \exists\langle W,\ \leqslant\rangle \in \mathfrak{Wf}\ .\ \exists \nu \in I \to W\ .\ \exists\langle \sigma_i \in I,\ i \in [1,\infty]\rangle\ .\ \sigma_1 \in P \wedge \forall i \in [1,\infty]\ .\ (\sigma_{i+1} \in P \vee \{\sigma_{i+1}\} \subseteq \mathsf{post}(\llbracket \mathsf{B}\rrbracket \mathbin{;} \llbracket \mathsf{S}\rrbracket^{e})\{\sigma_i\}) \wedge \forall i \in [1,\infty]\ .\ (\sigma_i \neq \sigma_{i+1}) \Rightarrow (\nu(\sigma_i) > \nu(\sigma_{i+1}) \wedge \forall i \in [1,\infty]\ .\ (\nu(\sigma_i) \not> \nu(\sigma_{i+1}) \Rightarrow \sigma_i \in \mathsf{pre}\llbracket\neg\mathsf{B}\rrbracket(\neg Q))\}$

$\wr$def. $\bar{\bar{F}}^{e}_{P}(X) \triangleq P \cup \mathsf{post}(\llbracket \mathsf{B}\rrbracket \mathbin{;} \llbracket \mathsf{S}\rrbracket^{e})X$, $\subseteq$, and post, which is $\varnothing$-strict$\wr$

$= \{\langle P,\ Q\rangle \mid \exists I \in \wp(\Sigma)\ .\ P \subseteq I \wedge \mathsf{post}(\llbracket \mathsf{B}\rrbracket \mathbin{;} \llbracket \mathsf{S}\rrbracket^{e})I \subseteq I \wedge \exists\langle W,\ \leqslant\rangle \in \mathfrak{Wf}\ .\ \exists \nu \in I \to W\ .\ \exists\langle \sigma_i \in I,\ i \in [1,\infty]\rangle\ .\ \sigma_1 \in P \wedge \forall i \in [1,\infty]\ .\ \{\sigma_{i+1}\} \subseteq \mathsf{post}(\llbracket \mathsf{B}\rrbracket \mathbin{;} \llbracket \mathsf{S}\rrbracket^{e})\{\sigma_i\} \wedge \forall i \in [1,\infty]\ .\ (\sigma_i \neq \sigma_{i+1}) \Rightarrow (\nu(\sigma_i) > \nu(\sigma_{i+1}) \wedge \forall i \in [1,\infty]\ .\ (\nu(\sigma_i) \not> \nu(\sigma_{i+1}) \Rightarrow \sigma_i \in \mathsf{pre}\llbracket\neg\mathsf{B}\rrbracket(\neg Q))\}$

$\wr$since if $\sigma_{i+1} \in P$, we can equivalently consider the sequence $\langle \sigma_j \in I,\ j \in [i+1,\infty]\rangle\wr$

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

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

$= \{\langle P,\ Q\rangle \mid \exists I \in \wp(\Sigma)\ .\ P \subseteq I \wedge \mathsf{post}(\llbracket \mathsf{B}\rrbracket \mathbin{;} \llbracket \mathsf{S}\rrbracket^{e})I \subseteq I \wedge \exists n \geqslant 1\ .\ \exists\langle \sigma_i \in I,\ i \in [1,n]\rangle\ .\ \sigma_1 \in P \wedge \forall i \in [1,n[\ .\ \{\sigma_{i+1}\} \subseteq \mathsf{post}(\llbracket \mathsf{B}\rrbracket \mathbin{;} \llbracket \mathsf{S}\rrbracket^{e})\{\sigma_i\} \wedge \sigma_n \notin \mathcal{B}\llbracket\mathsf{B}\rrbracket \wedge \sigma_n \notin Q\}$ $\wr$def. pre$\wr$

$= \{\langle P, Q\rangle \mid \exists n \geqslant 1\ .\ \exists\langle \sigma_i \in I, i \in [1,n]\rangle\ .\ \sigma_1 \in P \wedge \forall i \in [1,n[\ .\ \{\sigma_{i+1}\} \subseteq \mathsf{post}(\llbracket \mathsf{B}\rrbracket \mathbin{;} \llbracket \mathsf{S}\rrbracket^{e})\{\sigma_i\} \wedge \sigma_n \notin \mathcal{B}\llbracket\mathsf{B}\rrbracket \wedge \sigma_n \notin Q\}$ $\wr I$ is not used and can always be chosen to be $\Sigma\wr$

$= \{\langle P,\ Q\rangle \mid \exists n \geqslant 1\ .\ \exists\langle \sigma_i \in I,\ i \in [1,n]\rangle\ .\ \sigma_1 \in P \wedge \forall i \in [1,n[\ .\ \mathsf{post}(\llbracket \mathsf{B}\rrbracket \mathbin{;} \llbracket \mathsf{S}\rrbracket^{e})\{\sigma_i\} \cap \{\sigma_{i+1}\} \neq \varnothing \wedge \sigma_n \notin \mathcal{B}\llbracket\mathsf{B}\rrbracket \wedge \sigma_n \notin Q\}$ $\wr$since $x \in X \Leftrightarrow X \cap \{x\} \neq \varnothing\wr$

$= \{\langle P,\ Q\rangle \mid \exists n \geqslant 1\ .\ \exists\langle \sigma_i \in I,\ i \in [1,n]\rangle\ .\ \sigma_1 \in P \wedge \forall i \in [1,n[\ .\ \mathsf{post}(\llbracket \mathsf{B}\rrbracket \mathbin{;} \llbracket \mathsf{S}\rrbracket^{e})\{\sigma_i\} \cap \neg(\neg\{\sigma_{i+1}\}) \neq \varnothing \wedge \sigma_n \notin \mathcal{B}\llbracket\mathsf{B}\rrbracket \wedge \sigma_n \notin Q\}$ $\wr$def. $\neg X = \Sigma \setminus X\wr$

$= \{\langle P,\ Q\rangle \mid \exists n \geqslant 1\ .\ \exists\langle \sigma_i \in I,\ i \in [1,n]\rangle\ .\ \sigma_1 \in P \wedge \forall i \in [1,n[\ .\ \neg(\mathsf{post}(\llbracket \mathsf{B}\rrbracket \mathbin{;} \llbracket \mathsf{S}\rrbracket^{e})\{\sigma_i\} \subseteq (\neg\{\sigma_{i+1}\})) \wedge \sigma_n \notin \mathcal{B}\llbracket\mathsf{B}\rrbracket \wedge \sigma_n \notin Q\}$ $\wr\neg(X \subseteq Y) \Leftrightarrow (X \cap \neg Y \neq \varnothing\wr$

$= \{\langle P, Q\rangle \mid \exists n \geqslant 1\ .\ \exists\langle \sigma_i \in I, i \in [1,n]\rangle\ .\ \sigma_1 \in P \wedge \forall i \in [1,n[\ .\ \neg(\mathsf{post}(\llbracket \mathsf{S}\rrbracket^{e})(\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \{\sigma_i\}) \subseteq (\neg\{\sigma_{i+1}\})) \wedge \sigma_n \notin \mathcal{B}\llbracket\mathsf{B}\rrbracket \wedge \sigma_n \notin Q\}$ $\wr$def. post, $\llbracket \mathsf{B}\rrbracket$, and $\mathbin{;}\wr$

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

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

J.1.3 *Calculational Design of $\overline{\mathrm{HL}}$ Proof Rules.*

Theorem 7 ($\overline{\mathrm{HL}}$ rules for conditional iteration).

$$\frac{\exists\langle \sigma_i \in I,\ i \in [1,n]\rangle\ .\ \sigma_1 \in P \wedge \forall i \in [1,n[\ .\ (\!|\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \{\sigma_i\}|\!)\ \mathsf{S}\ (\!|\neg\{\sigma_{i+1}\}|\!) \wedge \sigma_n \notin \mathcal{B}\llbracket\mathsf{B}\rrbracket \wedge \sigma_n \notin Q}{(\!|P|\!)\ \texttt{while (B) S}\ (\!|Q|\!)} \tag{63}$$

Proof of (63). We write $(\!|P|\!)\ \mathsf{S}\ (\!|Q|\!) \triangleq \langle P,\ Q\rangle \in \overline{\mathrm{HL}}(\mathsf{S})$;

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

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

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

Traditionally, the side condition is written as a premiss, to get (63). □

This is nothing but debugging formalized as a logic since $\langle \sigma_i \in I,\ i \in [1,n]\rangle$ is a finite iteration in the loop starting with P true and finishing with Q false, which is obviously a counter example to Hoare triple

$\{P\}$ `while (B) S` $\{Q\}$. Notice that recursively $(\!|\mathcal{B}[\![\mathsf{B}]\!] \cap \{\sigma_i\}|\!)\, \mathsf{S}\, (\!|\{\sigma_{i+1}\}|\!)$ enforces the execution of the loop body S to start in state σ_i and terminate in state σ_{i+1}.

J.2 The post image transformer with breaks

To handle breaks we extend the definition of post$[\![\mathsf{S}]\!]$ componentwise.

$$\text{post}^{\times}([\![\mathsf{S}]\!]_{\perp}) \quad \triangleq \quad \langle \text{post}([\![\mathsf{S}]\!]^{e} \cup [\![\mathsf{S}]\!]^{\perp}),\ \text{post}[\![\mathsf{S}]\!]^{b}\rangle \tag{64}$$

The semantic of extended Hoare triples is

$$\{P\}\,\mathsf{S}\,\{ok : Q, br : T\} \quad \triangleq \quad \langle P,\ Q,\ T\rangle \in \alpha_{\text{HL}}(\text{post}^{\times}([\![\mathsf{S}]\!]_{\perp})) \tag{65}$$

where

$$\begin{aligned}
\alpha_{\text{HL}}(\text{post}^{\times}([\![\mathsf{S}]\!]_{\perp})) \quad &\triangleq \quad \{\langle P,\ Q,\ T\rangle \mid \text{post}^{\times}[\![\mathsf{S}]\!]_{\perp} P \mathrel{\dot{\subseteq}} \langle Q,\ T\rangle\} && (66)\\
&= \quad \{\langle P,\ Q,\ T\rangle \mid \text{post}([\![\mathsf{S}]\!]^{e} \cup [\![\mathsf{S}]\!]^{\perp})P \subseteq Q \wedge \text{post}[\![\mathsf{S}]\!]^{b} P \subseteq T\} && \text{(by def. of post}^{\times}\text{)}
\end{aligned}$$

Note that this may involve several batches of never modified auxiliary variables.

J.3 Auxiliary Propositions

We will use the following auxiliary lemmas.

- join preservation $\quad \text{post}(\bigcup_{i\in\Delta} \tau_i)Q \;=\; \bigcup_{i\in\Delta} \text{post}(\tau_i)Q \quad$ and $\quad \text{post}(\tau) \bigcup_{i\in\Delta} Q_i \;=\; \bigcup_{i\in\Delta} \text{post}(\tau)Q_i$ (67)

PROOF OF (67). By the Galois connections $\langle \wp(X \times Y),\ \subseteq\rangle \underset{\text{post}}{\overset{\text{post}^{-1}}{\leftrightarrows}} \langle \wp(X) \xrightarrow{\sqcup} \wp(Y),\ \dot{\subseteq}\rangle$ and $\langle \wp(X),\ \subseteq\rangle \underset{\text{post}(\tau)}{\overset{\widetilde{\text{pre}}(\tau)}{\leftrightarrows}} \langle \wp(Y),\ \subseteq\rangle$ where the lower adjoint preserves arbitrary joins. □

- composition $\quad \text{post}(r_1 \mathbin{⨾} r_2)P \;=\; \text{post}(r_2) \circ \text{post}(r_1)P$ (68)

PROOF OF (68).

$$\begin{aligned}
&\text{post}(r_1 \mathbin{⨾} r_2)P\\
=\;& \{\sigma' \in Y \mid \exists \sigma \in P \,.\, \langle \sigma,\ \sigma'\rangle \in r_1 \mathbin{⨾} r_2\} && \text{(def. (10) of post)}\\
=\;& \{\sigma' \in Y \mid \exists \sigma \in P \,.\, \exists \sigma'' \,.\, \langle \sigma,\ \sigma''\rangle \in r_1 \wedge \langle \sigma'',\ \sigma'\rangle \in r_2\} && \text{(def. relation composition ⨾)}\\
=\;& \{\sigma' \in Y \mid \exists \sigma'' \,.\, \langle \sigma'',\ \sigma'\rangle \in r_2 \wedge \exists \sigma \in P \,.\, \langle \sigma,\ \sigma''\rangle \in r_1\} && \text{(associativity of disjunction)}\\
=\;& \{\sigma' \in Y \mid \exists \sigma'' \,.\, \langle \sigma'',\ \sigma'\rangle \in r_2 \wedge \sigma'' \in \{\sigma'' \mid \exists \sigma \in P \,.\, \langle \sigma,\ \sigma''\rangle \in r_1\}\} && \text{(def. } \in\text{)}\\
=\;& \{\sigma' \in Y \mid \exists \sigma'' \,.\, \langle \sigma'',\ \sigma'\rangle \in r_2 \wedge \sigma'' \in \text{post}(r_1)P\} && \text{(def. (10) of post)}\\
=\;& \text{post}(r_2)(\text{post}(r_1)P) && \text{(def. (10) of post)}\\
=\;& \text{post}(r_2) \circ \text{post}(r_1)P && \text{(def. function composition } \circ\text{)}
\end{aligned}$$

□

- Boolean postcondition $\quad \text{post}([\![\mathsf{B}]\!])P = \quad P \cap \mathcal{B}[\![\mathsf{B}]\!]$ (69)

PROOF OF (69).

$$\begin{aligned}
&\text{post}([\![\mathsf{B}]\!])Q\\
=\;& \{\sigma' \mid \exists \sigma \in P \,.\, \langle \sigma,\ \sigma'\rangle \in [\![\mathsf{B}]\!]\} && \text{(def. (10) of post)}\\
=\;& \{\sigma' \mid \exists \sigma \in P \,.\, \sigma = \sigma' \in \mathcal{B}[\![\mathsf{B}]\!]\} && \text{(def. } [\![\mathsf{B}]\!] \triangleq \{\langle \sigma,\ \sigma\rangle \mid \sigma \in \mathcal{B}[\![\mathsf{B}]\!]\}\text{)}\\
=\;& \{\sigma \mid \sigma \in P \wedge \sigma \in \mathcal{B}[\![\mathsf{B}]\!]\} && \text{(def. equality)}\\
=\;& P \cap \mathcal{B}[\![\mathsf{B}]\!] && \text{(def. intersection } \cap\text{)}
\end{aligned}$$

□

- commutativity
$$\begin{array}{rcll} \mathsf{lfp}^{\subseteq} F^{e} &=& \mathsf{lfp}^{\subseteq} F'^{e} & (70)\\ F^{e} &\triangleq& \boldsymbol{\lambda} X \bullet \mathsf{id} \cup (\llbracket \mathsf{B} \rrbracket \mathbin{⨾} \llbracket \mathsf{S} \rrbracket^{e} \mathbin{⨾} X), \quad X \in \wp(\Sigma \times \Sigma) & (49)\\ F'^{e} &\triangleq& \boldsymbol{\lambda} X \bullet \mathsf{id} \cup (X \mathbin{⨾} \llbracket \mathsf{B} \rrbracket \mathbin{⨾} \llbracket \mathsf{S} \rrbracket^{e}) & \end{array}$$

PROOF OF (70). Let $\langle X^n, n \in \mathbb{N}\rangle$ and $\langle X'^n, n \in \mathbb{N}\rangle$ be the respective iterates of F^e and F'^e (which preserve joins) starting from the infimum $\varnothing$. We have $X^0 = X'^0$ so $X^1 = X'^1 = \mathsf{id} = (\llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e})^0$. Assume $X^n = X'^n = \bigcup_{k<n}(\llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e})^k$, $n > 0$ by induction hypothesis. Then

$$\begin{array}{ll}
& X^{n+1}\\
= & F^{e}(X^{n}) \hfill \text{(def. iterates)}\\
= & \mathsf{id} \cup (\llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e} ⨾ X^{n}) \hfill \text{(def. (49) of } F^{e} \text{ and } X \cap \Sigma \times \{\bot\} = \varnothing\text{)}\\
= & \mathsf{id} \cup (\llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e} ⨾ \bigcup_{k<n} (\llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e})^{k}) \hfill \text{(induction hypothesis)}\\
= & (\llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e})^{0} \cup \bigcup_{k<n} (\llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e})^{k+1}) \hfill \text{(def. power and relation composition ⨾)}\\
= & \bigcup_{k<n+1} (\llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e})^{k} \hfill \text{(which is the induction hypotehsis)}\\
= & \mathsf{id} \cup (\bigcup_{k<n} (\llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e})^{k} ⨾ \llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e}) \hfill \text{(def. power and relation composition ⨾)}\\
= & \mathsf{id} \cup (X'^{n} ⨾ \llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e}) \hfill \text{(induction hypothesis)}\\
= & F'^{e}(X'^{n}) \hfill \text{(def. (70) of } F'^{e}\text{)}\\
= & X'^{n+1} \hfill \text{(def. iterates)}
\end{array}$$

It follows that $\mathsf{lfp}^{\subseteq} F^{e} = \bigcup_{k\in\mathbb{X}} X^{k} = \bigcup_{k\in\mathbb{X}} X'^{k} = \mathsf{lfp}^{\subseteq} F'^{e}$. □

- commutation
$$\begin{array}{rcll} \mathsf{post}(F'^{e}(X))P &=& \bar{F}'^{e}_{P}(\mathsf{post}(X)P) & (71)\\ \bar{F}'^{e}_{P}(X) &\triangleq& P \cup \mathsf{post}(\llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e})X, \quad X \in \wp(\Sigma) \to \wp(\Sigma) & \end{array}$$

PROOF OF (71).

$$\begin{array}{ll}
& \mathsf{post}(F'^{e}(X))P\\
= & \mathsf{post}(\mathsf{id} \cup (X ⨾ \llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e}))P \hfill \text{(def. (70) of } F'^{e}\text{)}\\
= & \mathsf{post}(\mathsf{id})P \cup \mathsf{post}(X ⨾ \llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e})P \hfill \text{(join preservation (67))}\\
= & P \cup \mathsf{post}(X ⨾ \llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e})P \hfill \text{(def. (10) of post and def. identity relation id)}\\
= & P \cup \mathsf{post}(\llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{e})(\mathsf{post}(X)P) \hfill \text{(composition (68))}\\
= & \bar{F}'^{e}_{P}(\mathsf{post}(X)P) \hfill \text{(def. (71) of } \bar{F}'^{e}\text{)} \quad □
\end{array}$$

J.4 Design of the Extended Hoare Logic

The deductive rules of Hoare logic are derived by structural induction on the program syntax and abstraction of the semantics for each statement S.

– Iteration `while(B) S`

$$\begin{array}{ll}
& \alpha_{\mathrm{HL}}(\mathsf{post}^{\times}(\llbracket \mathtt{while(B)\ S} \rrbracket_{\bot}))\\
= & \{\langle P, Q, T\rangle \mid \mathsf{post}(\llbracket \mathtt{while(B)\ S} \rrbracket^{e} \cup \llbracket \mathtt{while(B)\ S} \rrbracket^{\bot})P \subseteq Q \wedge \mathsf{post}\llbracket \mathtt{while(B)\ S} \rrbracket^{b} P \subseteq T\}\\
& \quad \text{(def. } \alpha_{\mathrm{HL}} \circ \mathsf{post}^{\times}\text{)}\\
= & \{\langle P, Q, T\rangle \mid \mathsf{post}((\mathsf{lfp}^{\subseteq} F^{e} ⨾ (\llbracket \neg\mathsf{B} \rrbracket \cup \llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{b})) \cup (\mathsf{lfp}^{\subseteq} F^{e} ⨾ \llbracket \mathsf{B} \rrbracket ⨾ \llbracket \mathsf{S} \rrbracket^{\bot}) \cup \mathsf{gfp}^{\subseteq} F^{\bot})P \subseteq Q \wedge \mathsf{post}(\varnothing)P \subseteq T\}\\
& \quad \text{(by definition (51), (53), and (52) of the relational semantics of } \llbracket \mathtt{while(B)\ S} \rrbracket\text{)}
\end{array}$$

$= \{\langle P, Q, T\rangle \mid \mathsf{post}((\mathsf{lfp}^{\subseteq} F^e \mathbin{;} (\llbracket\neg\mathsf{B}\rrbracket \cup \llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^b)) \cup (\mathsf{lfp}^{\subseteq} F^e \mathbin{;} \llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^{\perp}) \cup \mathsf{gfp}^{\subseteq} F^{\perp})P \subseteq Q\}$

$\wr \mathsf{post}(\varnothing)P = \varnothing \int$

$= \{\langle P, Q, T\rangle \mid \mathsf{post}(\mathsf{lfp}^{\subseteq} F^e \mathbin{;} (\llbracket\neg\mathsf{B}\rrbracket \cup \llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^b)P \subseteq Q \wedge \mathsf{post}(\mathsf{lfp}^{\subseteq} F^e \mathbin{;} \llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^{\perp}))P \subseteq Q \wedge \mathsf{post}(\mathsf{gfp}^{\subseteq} F^{\perp})P \subseteq Q\}$

$\wr$post preserves arbitrary joins (67), and $(A \cup B) \subseteq q \Leftrightarrow A \subseteq q \wedge B \subseteq q \int$

$= \{\langle P, Q, T\rangle \mid \mathsf{post}(\llbracket\neg\mathsf{B}\rrbracket \cup \llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^b)(\mathsf{post}(\mathsf{lfp}^{\subseteq} F^e)P) \subseteq Q \wedge \mathsf{post}(\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^{\perp})(\mathsf{post}(\mathsf{lfp}^{\subseteq} F^e)P) \subseteq Q \wedge \mathsf{post}(\mathsf{gfp}^{\subseteq} F^{\perp})P \subseteq Q\}$

$\wr$by composition (68) $\int$

$= \{\langle P, Q, T\rangle \mid \exists I \in \wp(\Sigma_{\perp}) \,.\, \mathsf{post}(\mathsf{lfp}^{\subseteq} F^e)P \subseteq I \wedge \mathsf{post}(\llbracket\neg\mathsf{B}\rrbracket \cup (\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^b))I \subseteq Q \wedge \mathsf{post}(\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^{\perp})I \subseteq Q \wedge \mathsf{post}(\mathsf{gfp}^{\subseteq} F^{\perp})P \subseteq Q\}$

$\wr$introducing an over approximation $I \in \wp(\Sigma_{\perp})$.
($\subseteq$) post(r) preserves joins so is increasing and transitivity;
($\supseteq$) take $I = \mathsf{post}(\mathsf{lfp}^{\subseteq} F^e)P \int$

$= \{\langle P, Q, T\rangle \mid \exists I \in \wp(\Sigma_{\perp}) \,.\, \mathsf{post}(\mathsf{lfp}^{\subseteq} F'^e)P \subseteq I \wedge \mathsf{post}(\llbracket\neg\mathsf{B}\rrbracket \cup (\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^b))I \subseteq Q \wedge \mathsf{post}(\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^{\perp})I \subseteq Q \wedge \mathsf{post}(\mathsf{gfp}^{\subseteq} F^{\perp})P \subseteq Q\}$

$\wr$by $\mathsf{lfp}^{\subseteq} F^e = \mathsf{lfp}^{\subseteq} F'^e$ at (70)$\int$

$= \{\langle P, Q, T\rangle \mid \exists I \in \wp(\Sigma_{\perp}) \,.\, \mathsf{lfp}^{\subseteq} \bar{F}'^e_P \subseteq I \wedge \mathsf{post}(\llbracket\neg\mathsf{B}\rrbracket \cup (\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^b))I \subseteq Q \wedge \mathsf{post}(\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^{\perp})I \subseteq Q \wedge (\!| \perp \in Q \;?\; \mathsf{true} \;\text{:}\; \mathsf{gfp}^{\subseteq} F^{\perp} \cap (P \times \{\perp\}) = \varnothing |\!)\}$

$\wr$– by Galois connection(12), commutativity (71), and Th. II.2.1 for $\alpha(X) = \mathsf{post}(X)P$
– By def. (50) of $F^{\perp}(X) \triangleq \llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^e \mathbin{;} X$ where $X \in \wp(\Sigma \times \{\perp\})$, we have $\mathsf{gfp}^{\subseteq} F^{\perp} \in \wp(\Sigma \times \{\perp\})$ and so by def. (10) of post, we have $\mathsf{post}(r)P = \{\perp \mid \exists\sigma \,.\, \sigma \in P \wedge \langle\sigma, \perp\rangle \in \mathsf{gfp}^{\subseteq} F^{\perp}\} = \{\perp \mid \mathsf{gfp}^{\subseteq} F^{\perp} \cap (P \times \{\perp\}) \neq \varnothing\}$. If follows that $\mathsf{post}(\mathsf{gfp}^{\subseteq} F^{\perp})P \subseteq Q$ when $\perp \in Q$ and otherwise $\mathsf{gfp}^{\subseteq} F^{\perp} \cap (P \times \{\perp\}) = \varnothing \int$

$= \{\langle P, Q, T\rangle \mid \exists I \in \wp(\Sigma_{\perp}) \,.\, \mathsf{lfp}^{\subseteq} \boldsymbol{\lambda} X \bullet P \cup \mathsf{post}(\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^e)X \subseteq I \wedge \mathsf{post}(\llbracket\neg\mathsf{B}\rrbracket \cup (\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^b))I \subseteq Q \wedge \mathsf{post}(\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^{\perp})I \subseteq Q \wedge (\perp \notin Q) \Rightarrow (\mathsf{gfp}^{\subseteq} F^{\perp} \cap (P \times \{\perp\}) = \varnothing)\}$

$\wr$– def. (71) of $\bar{F}'^e_P$
– def. conditional $(\!| \ldots \;?\; \ldots \;\text{:}\; \ldots |\!)$ and implication $\Rightarrow \int$

$= \{\langle P, Q, T\rangle \mid \exists I \in \wp(\Sigma_{\perp}) \,.\, P \subseteq I \wedge \mathsf{post}(\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^e)I \subseteq I \wedge \mathsf{post}(\llbracket\neg\mathsf{B}\rrbracket \cup (\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^b))I \subseteq Q \wedge \mathsf{post}(\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^{\perp})I \subseteq Q \wedge (\perp \notin Q) \Rightarrow \{\sigma \in \mathbb{N} \to \Sigma \mid \sigma_0 \in P \wedge \forall i \in \mathbb{N} \,.\, \langle\sigma_i, \sigma_{i+1}\rangle \in \llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}^e\rrbracket\} = \varnothing\}$

$\wr$– induction for fixpoint over approximation Th. II.3.1 noting that the invariant I need not be strengthened since it can be always be chosen to be strong enough
– by lemma II.1.1$\int$

$= \{\langle P, Q, T\rangle \mid \exists I \in \wp(\Sigma_{\perp}) \,.\, P \subseteq I \wedge \mathsf{post}(\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^e)I \subseteq I \wedge \mathsf{post}\llbracket\neg\mathsf{B}\rrbracket I \subseteq Q \wedge \mathsf{post}(\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^b)I \subseteq Q \wedge \mathsf{post}(\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}\rrbracket^{\perp})I \subseteq Q \wedge (\perp \notin Q) \Rightarrow \{\sigma \in \mathbb{N} \to \Sigma \mid \sigma_0 \in P \wedge \forall i \in \mathbb{N} \,.\, \langle\sigma_i, \sigma_{i+1}\rangle \in \llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}^e\rrbracket\} = \varnothing\}$

$\wr$post preserves joins and union subsumption law, $(A \cup B) \subseteq q \Leftrightarrow A \subseteq q \wedge B \subseteq q \int$

$= \{\langle P, Q, T\rangle \mid \exists I \in \wp(\Sigma_{\perp}) \,.\, P \subseteq I \wedge \mathsf{post}\llbracket\mathsf{B}\rrbracket(\mathsf{post}\llbracket\mathsf{S}\rrbracket^e I) \subseteq I \wedge \mathsf{post}\llbracket\neg\mathsf{B}\rrbracket I \subseteq Q \wedge \mathsf{post}\llbracket\mathsf{B}\rrbracket(\mathsf{post}\llbracket\mathsf{S}\rrbracket^b I) \subseteq Q \wedge \mathsf{post}\llbracket\mathsf{B}\rrbracket(\mathsf{post}\llbracket\mathsf{S}\rrbracket^{\perp} I) \subseteq Q \wedge (\perp \notin Q) \Rightarrow (\exists\langle W, \preceq\rangle \in \mathfrak{Wf} \,.\, \exists J \in \wp(\Sigma) \,.\, \exists \nu \in J \to W \,.\, P \cup \mathsf{post}(\llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}^e\rrbracket)J \subseteq J \wedge \forall \underline{\sigma} \in J \,.\, \forall \sigma' \in \Sigma \,.\, \langle\underline{\sigma}, \sigma'\rangle \in \llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}^e\rrbracket \Rightarrow \nu(\underline{\sigma}) \succ \nu(\sigma'))\}$

$\wr$– composition (68)
– by Turing/Floyd Th. II.3.11$\int$

$= \{\langle P, Q, T\rangle \mid \exists I \in \wp(\Sigma_{\perp}) \,.\, P \subseteq I \wedge \mathsf{post}\llbracket\mathsf{S}\rrbracket^e(\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap I) \subseteq I \wedge (\mathcal{B}\llbracket\neg\mathsf{B}\rrbracket \cap I) \subseteq Q \wedge \mathsf{post}\llbracket\mathsf{S}\rrbracket^b(\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap I) \subseteq Q \wedge \mathsf{post}\llbracket\mathsf{S}\rrbracket^{\perp}(\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap I) \subseteq Q \wedge (\perp \notin Q) \Rightarrow (\exists\langle W, \preceq\rangle \in \mathfrak{Wf} \,.\, \exists J \in \wp(\Sigma) \,.\, \exists \nu \in J \to W \,.\, P \subseteq J \wedge (\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{post}(\llbracket\mathsf{S}^e\rrbracket)J \subseteq J \wedge \forall \underline{\sigma} \in J \,.\, \forall \sigma' \in \Sigma \,.\, \langle\underline{\sigma}, \sigma'\rangle \in \llbracket\mathsf{B}\rrbracket \mathbin{;} \llbracket\mathsf{S}^e\rrbracket \Rightarrow \nu(\underline{\sigma}) \succ \nu(\sigma'))\}$

$\wr$– Boolean composition (69)
– by composition (68), Boolean postcondition (69), and union subsumption law, $(A \cup B) \subseteq q \Leftrightarrow A \subseteq q \wedge B \subseteq q \int$

$= \{\langle P, Q, T\rangle \mid \exists I' \,.\, P \subseteq I' \wedge \text{post}[\![\mathsf{S}]\!]^{e}(\mathcal{B}[\![\mathsf{B}]\!] \cap I') \subseteq I' \wedge (\mathcal{B}[\![\neg\mathsf{B}]\!] \cap I') \subseteq Q \wedge \text{post}[\![\mathsf{S}]\!]^{b}(\mathcal{B}[\![\mathsf{B}]\!] \cap I') \subseteq Q \wedge \text{post}[\![\mathsf{S}]\!]^{\perp}(\mathcal{B}[\![\mathsf{B}]\!] \cap I') \subseteq Q \wedge (\perp \notin Q) \Rightarrow (\exists\langle W, \preceq\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in I' \to W \,.\, \forall \underline{\sigma} \in I' \,.\, \forall \sigma' \in \Sigma \,.\, \langle \underline{\sigma}, \sigma'\rangle \in [\![\mathsf{B}]\!] \,\mathring{,}\, [\![\mathsf{S}^{e}]\!] \Rightarrow \nu(\underline{\sigma}) \succ \nu(\sigma'))\}$

(⊇) take $I = J = I'$

(⊇) take I' to be the greatest lower bound of I and J in the complete lattice of post fixpoints of $\boldsymbol{\lambda} X \bullet P \cup \text{post}[\![\mathsf{S}]\!]^{e}(\mathcal{B}[\![\mathsf{B}]\!] \cap X)$, [Cousot and Cousot 1979a, Corollary 4.2]; ∩ and post are increasing in both of their parameters

$= \{\langle P, Q, T\rangle \mid \exists I \in \wp(\Sigma_{\perp}) \,.\, P \subseteq I \wedge \text{post}[\![\mathsf{S}]\!]^{e}(\mathcal{B}[\![\mathsf{B}]\!] \cap I) \subseteq I \wedge (\mathcal{B}[\![\neg\mathsf{B}]\!] \cap I) \subseteq Q \wedge \text{post}[\![\mathsf{S}]\!]^{b}(\mathcal{B}[\![\mathsf{B}]\!] \cap I) \subseteq Q \wedge \text{post}[\![\mathsf{S}]\!]^{\perp}(\mathcal{B}[\![\mathsf{B}]\!] \cap I) \subseteq Q \wedge (\perp \notin Q) \Rightarrow (\exists\langle W, \preceq\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in I \to W \,.\, \forall \underline{\sigma} \in I \,.\, \forall \sigma' \in \text{post}[\![\mathsf{S}^{e}]\!](\mathcal{B}[\![\mathsf{B}]\!] \cap \{\varsigma \mid \varsigma = \underline{\sigma}\}) \,.\, \nu(\underline{\sigma}) \succ \nu(\sigma'))\}$

by composition (68) and Boolean postcondition (69), writing $\{\underline{\sigma}\} = \{\varsigma \mid \varsigma = \underline{\sigma}\}$ to make the binding of $\underline{\sigma}$ explicit

$= \{\langle P, Q, T\rangle \mid \exists \bar{I} \in \wp(\overline{\Sigma}_{\perp}) \,.\, \mathbb{1}_{\overline{\mathbb{X}}}(P) \subseteq \bar{I} \wedge \text{post}[\![\mathsf{S}]\!]^{e}(\mathcal{B}[\![\mathsf{B}]\!] \cap \bar{I}) \subseteq \bar{I} \wedge (\mathcal{B}[\![\neg\mathsf{B}]\!] \cap \bar{I})_{\mathbb{X}} \subseteq Q \wedge \text{post}[\![\mathsf{S}]\!]^{b}(\mathcal{B}[\![\mathsf{B}]\!] \cap \bar{I})_{\mathbb{X}} \subseteq Q \wedge \text{post}[\![\mathsf{S}]\!]^{\perp}(\mathcal{B}[\![\mathsf{B}]\!] \cap \bar{I})_{\mathbb{X}} \subseteq Q \wedge (\perp \notin Q) \Rightarrow (\exists\langle W, \preceq\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in \bar{I} \to W \,.\, \forall \langle \underline{\sigma}, \sigma'\rangle \in \text{post}[\![\mathsf{S}^{e}]\!](\mathcal{B}[\![\mathsf{B}]\!] \cap \bar{I}) \,.\, \nu(\underline{\sigma}) \succ \nu(\sigma'))\}$

by introducing auxiliary variables, with $I = \bar{I}_{\mathbb{X}}$

$= \{\langle P, Q, T\rangle \mid \exists \bar{I} \in \wp(\overline{\Sigma}_{\perp}) \,.\, \mathbb{1}_{\overline{\mathbb{X}}}(P) \subseteq \bar{I} \wedge \text{post}[\![\mathsf{S}]\!]^{e}(\mathcal{B}[\![\mathsf{B}]\!] \cap \bar{I}) \subseteq \bar{I} \wedge (\mathcal{B}[\![\neg\mathsf{B}]\!] \cap \bar{I})_{\mathbb{X}} \subseteq Q \wedge \text{post}[\![\mathsf{S}]\!]^{b}(\mathcal{B}[\![\mathsf{B}]\!] \cap \bar{I})_{\mathbb{X}} \subseteq Q \wedge \text{post}[\![\mathsf{S}]\!]^{\perp}(\mathcal{B}[\![\mathsf{B}]\!] \cap \bar{I})_{\mathbb{X}} \subseteq Q \wedge (\perp \notin Q) \Rightarrow (\exists\langle W, \preceq\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in \bar{I} \to W \,.\, \forall \langle \underline{\sigma}, \sigma'\rangle \in \bar{I} \,.\, \nu(\underline{\sigma}) \succ \nu(\sigma'))\}$

(⊆) by $x \in A \wedge (A \subseteq B) \Rightarrow (x \in B)$

(⊇) by taking the greatest lower bound of $(\bar{I})$ and $\text{post}[\![\mathsf{S}]\!]^{e}(\mathcal{B}[\![\mathsf{B}]\!] \cap \bar{I})$ in the complete lattice of post fixpoints of $\boldsymbol{\lambda} X \bullet \text{post}[\![\mathsf{S}]\!]^{e}(\mathcal{B}[\![\mathsf{B}]\!] \cap P \cap X)$, [Cousot and Cousot 1979a, Corollary 4.2] and increasingness

$= \{\langle P, Q, T\rangle \mid \exists I \in \wp(\overline{\Sigma}_{\perp}) \,.\, \mathbb{1}_{\overline{\mathbb{X}}}(P) \subseteq \bar{I} \wedge \exists R^{e} \,.\, \text{post}[\![\mathsf{S}]\!]^{e}(\mathcal{B}[\![\mathsf{B}]\!] \cap I) \subseteq R^{e} \wedge R^{e} \subseteq I \wedge (\mathcal{B}[\![\neg\mathsf{B}]\!] \cap I)_{\mathbb{X}} \subseteq Q \wedge \exists R^{b} \,.\, \text{post}[\![\mathsf{S}]\!]^{b}(\mathcal{B}[\![\mathsf{B}]\!] \cap I) \subseteq R^{b} \wedge R^{b}_{\mathbb{X}} \subseteq Q \wedge \exists R^{\perp} \,.\, \text{post}[\![\mathsf{S}]\!]^{\perp}(\mathcal{B}[\![\mathsf{B}]\!] \cap I) \subseteq R^{\perp} \wedge R^{\perp}_{\mathbb{X}} \subseteq Q \wedge (\perp \notin Q) \Rightarrow (\exists\langle W, \preceq\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in I \to W \,.\, \forall \langle \underline{\sigma}, \sigma'\rangle \in I \,.\, \nu(\underline{\sigma}) \succ \nu(\sigma'))\}$

choosing $I = \bar{I}$ and introducing possible approximations by $X \subseteq Y \Leftrightarrow \exists Z \,.\, X \subseteq Z \wedge Z \subseteq Y$

$= \{\langle P, Q, T\rangle \mid \exists I \in \wp(\overline{\Sigma}_{\perp}) \,.\, \mathbb{1}_{\overline{\mathbb{X}}}(P) \subseteq \bar{I} \wedge \exists R, R^{b} \,.\, \langle \mathcal{B}[\![\mathsf{B}]\!] \cap I_{\not\perp}, R, R^{b}\rangle \in \alpha_{\text{HL}}(\text{post}^{\times}[\![\mathsf{S}]\!]_{\perp}) \wedge R_{\not\perp} \subseteq I \wedge (\mathcal{B}[\![\neg\mathsf{B}]\!] \cap I)_{\mathbb{X}} \subseteq Q \wedge R^{b}_{\mathbb{X}} \subseteq Q \wedge R_{\perp\mathbb{X}} \subseteq Q \wedge (\perp \notin Q) \Rightarrow (\exists\langle W, \preceq\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in I \to W \,.\, \forall \langle \underline{\sigma}, \sigma'\rangle \in I \,.\, \nu(\underline{\sigma}) \succ \nu(\sigma'))\}$

def. (64) of $\text{post}^{\times}$ and (66) of α_{HL}

$= \{\langle P, Q, T\rangle \mid \exists I \in \wp(\overline{\Sigma}_{\perp}) \,.\, \{\sigma \in \wp(\overline{\Sigma}) \mid \sigma_{\overline{\mathbb{X}}} = \sigma_{\mathbb{X}} \wedge \sigma_{\mathbb{X}} \in P\} \subseteq I \wedge \exists R, R^{b} \,.\, \{\mathcal{B}[\![\mathsf{B}]\!] \cap I_{\not\perp}\}\, \mathsf{S}\, \{ok : R, br : R^{b}\} \wedge R_{\not\perp} \subseteq I \wedge (\mathcal{B}[\![\neg\mathsf{B}]\!] \cap I)_{\mathbb{X}} \subseteq Q \wedge R^{b}_{\mathbb{X}} \subseteq Q \wedge R_{\perp\mathbb{X}} \subseteq Q \wedge (\perp \notin Q) \Rightarrow (\exists\langle W, \preceq\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in I \to W \,.\, \forall \langle \underline{\sigma}, \sigma'\rangle \in I \,.\, \nu(\underline{\sigma}) \succ \nu(\sigma'))\}$

def. $\mathbb{1}_{\overline{\mathbb{X}}}(P)$ and (65) of the semantics of Hoare triples

The computation clearly shows that Q can be chosen in $\wp(\overline{\Sigma} \times \overline{\Sigma}_{\perp})$ to relate initial to final states or ⊥, in which case we get

$\{\langle P, Q, T\rangle \mid \exists I \in \wp(\overline{\Sigma}_{\perp}) \,.\, \{\sigma \in \wp(\overline{\Sigma}) \mid \sigma_{\overline{\mathbb{X}}} = \sigma_{\mathbb{X}} \wedge \sigma_{\mathbb{X}} \in P\} \subseteq I \wedge \exists R, T \,.\, \{\mathcal{B}[\![\mathsf{B}]\!] \cap I_{\not\perp}\}\, \mathsf{S}\, \{ok : R, br : T\} \wedge R_{\not\perp} \subseteq I \wedge (\mathcal{B}[\![\neg\mathsf{B}]\!] \cap I) \subseteq Q \wedge T \subseteq Q \wedge R_{\perp} \subseteq Q \wedge (\perp \notin Q) \Rightarrow (\exists\langle W, \preceq\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in I \to W \,.\, \forall \langle \underline{\sigma}, \sigma'\rangle \in I \,.\, \nu(\underline{\sigma}) \succ \nu(\sigma'))\}$

$Q \in \wp(\overline{\Sigma} \times \overline{\Sigma}_{\perp})$

Following Sect. II.5, this set $\{\langle P, Q, T\rangle \mid \langle P, Q, T\rangle \in \alpha_{\text{HL}}(\text{post}^{\times}([\![\texttt{while(B) S}]\!]_{\perp}))\} = \{\langle P, Q, T\rangle \mid \{P\}\, \mathsf{S}\, \{ok : Q, br : T\}$ by (65) can be equivalently defined by the following deductive system.

$$\frac{\begin{array}{c}\{\sigma \in \wp(\overline{\Sigma}) \mid \sigma_{\overline{\mathbb{X}}} = \sigma_{\mathbb{X}} \wedge \sigma_{\mathbb{X}} \in P\} \subseteq I \\ \{\mathcal{B}[\![\mathsf{B}]\!] \cap I_{\not\perp}\}\ \mathsf{S}\ \{ok : R, br : T\} \\ R_{\not\perp} \subseteq I \quad (\mathcal{B}[\![\neg\mathsf{B}]\!] \cap I_{\not\perp}) \subseteq Q \quad T \subseteq Q \quad R_{\perp} \subseteq Q \\ (\perp \notin Q) \Rightarrow (\exists \langle W, \preceq\rangle \in \mathfrak{Wf} \,.\, \exists v \in I \to W \,.\, \forall \langle \underline{\sigma}, \sigma'\rangle \in I \,.\, v(\underline{\sigma}) \succ v(\sigma'))\end{array}}{\{P\}\ \texttt{while(B) S}\ \{ok : Q, br : T\}} \tag{72}$$

J.5 Auxiliary Propositions

We will use the following auxiliary lemmas.

- join preservation $\quad \mathsf{pre}(\bigcup_{i\in\Delta} \tau_i)Q = \bigcup_{i\in\Delta} \mathsf{pre}(\tau_i)Q$ and $\mathsf{pre}(\tau)\bigcup_{i\in\Delta} Q_i = \bigcup_{i\in\Delta} \mathsf{pre}(\tau)Q_i$ (73)

PROOF OF (73). By the Galois connections $\langle \wp(X\times Y), \subseteq\rangle \xrightleftharpoons[\mathsf{pre}]{\mathsf{pre}^{-1}} \langle \wp(X) \xrightarrow{\sqcup} \wp(Y), \dot{\subseteq}\rangle$ and $\langle \wp(X), \subseteq\rangle \xrightleftharpoons[\mathsf{pre}(\tau)]{\widetilde{\mathsf{post}}(\tau)} \langle \wp(Y), \subseteq\rangle$ where the lower adjoint preserves arbitrary joins. ◻

- composition $\quad \mathsf{pre}(r_1 \mathbin{⨾} r_2)Q = \mathsf{pre}(r_1) \circ \mathsf{pre}(r_2)Q$ (74)

PROOF OF (74).

$$\begin{array}{ll}
& \mathsf{pre}(r_1 \mathbin{⨾} r_2)Q \\
= & \{\sigma \in X \mid \exists \sigma' \in Q \,.\, \langle \sigma, \sigma'\rangle \in r_1 \mathbin{⨾} r_2\} \qquad \text{(def. (38) of pre)} \\
= & \{\sigma \in X \mid \exists \sigma' \in Q \,.\, \exists \sigma'' \,.\, \langle \sigma, \sigma''\rangle \in r_1 \wedge \langle \sigma'', \sigma'\rangle \in r_2\} \qquad \text{(def. relation composition ⨾)} \\
= & \{\sigma \in X \mid \exists \sigma'' \,.\, \langle \sigma, \sigma''\rangle \in r_1 \wedge \exists \sigma' \in Q \,.\, \langle \sigma'', \sigma'\rangle \in r_2\} \qquad \text{(associativity of disjunction)} \\
= & \{\sigma \in X \mid \exists \sigma'' \,.\, \langle \sigma, \sigma''\rangle \in r_1 \wedge \sigma'' \in \{\sigma'' \mid \exists \sigma' \in Q \,.\, \langle \sigma'', \sigma'\rangle \in r_2\}\} \qquad \text{(def. ∈)} \\
= & \{\sigma \in X \mid \exists \sigma'' \,.\, \langle \sigma, \sigma''\rangle \in r_1 \wedge \sigma'' \in \mathsf{pre}(r_2)Q\} \qquad \text{(def. (38) of pre)} \\
= & \mathsf{pre}(r_1)(\mathsf{pre}(r_2)Q) \qquad \text{(def. (38) of pre)} \\
= & \mathsf{pre}(r_1) \circ \mathsf{pre}(r_2)Q \qquad \text{(def. function composition ∘)}
\end{array}$$

◻

- partitioning $\quad \mathsf{pre}(r)Q = \mathsf{pre}(r \cap (\Sigma\times\Sigma))Q_{\not\perp} \cup \mathsf{pre}(r \cap (\Sigma\times\{\perp\}))\{\perp \mid \perp \in Q\}$ (75)

PROOF OF (75).

$$\begin{array}{ll}
& \mathsf{pre}(r)Q \\
= & \{\sigma \mid \exists \sigma' \,.\, \langle \sigma, \sigma'\rangle \in r \wedge \sigma' \in Q\} \qquad \text{(def. (38) of pre)} \\
= & \{\sigma \mid \exists \sigma' \,.\, \langle \sigma, \sigma'\rangle \in (r \cap (\Sigma\times\Sigma) \cup r \cap (\Sigma\times\{\perp\})) \wedge \sigma' \in Q\} \\
& \quad \text{(}r \in \wp(\Sigma\times\Sigma_{\perp}), \Sigma_{\perp} = \Sigma \cup \{\perp\} \text{ and } \perp \notin \Sigma\text{)} \\
= & \{\sigma \mid \exists \sigma' \,.\, (\langle \sigma, \sigma'\rangle \in r \cap (\Sigma\times\Sigma) \vee \langle \sigma, \sigma'\rangle \in r \cap (\Sigma\times\{\perp\})) \wedge \sigma' \in Q\} \qquad \text{(def. ∪)} \\
= & \{\sigma \mid \exists \sigma' \,.\, (\langle \sigma, \sigma'\rangle \in r \cap (\Sigma\times\Sigma) \wedge \sigma' \in Q) \vee (\langle \sigma, \sigma'\rangle \in r \cap (\Sigma\times\{\perp\}) \wedge \sigma' \in Q)\} \qquad \text{(distributivity)} \\
= & \{\sigma \mid (\exists \sigma' \,.\, \langle \sigma, \sigma'\rangle \in r \cap (\Sigma\times\Sigma) \wedge \sigma' \in Q) \vee (\exists \sigma' \,.\, \langle \sigma, \sigma'\rangle \in r \cap (\Sigma\times\{\perp\}) \wedge \sigma' \in Q)\} \qquad \text{(associativity)} \\
= & \{\sigma \mid \exists \sigma' \,.\, \langle \sigma, \sigma'\rangle \in r \cap (\Sigma\times\Sigma) \wedge \sigma' \in Q\} \cup \{\sigma \mid \exists \sigma' \,.\, \langle \sigma, \sigma'\rangle \in r \cap (\Sigma\times\{\perp\}) \wedge \sigma' \in Q\} \qquad \text{(def. ∪)} \\
= & \{\sigma \mid \exists \sigma' \,.\, \langle \sigma, \sigma'\rangle \in r \cap (\Sigma\times\Sigma) \wedge \sigma' \in Q_{\not\perp}\} \cup \{\sigma \mid \exists \sigma' \,.\, \langle \sigma, \sigma'\rangle \in r \cap (\Sigma\times\{\perp\}) \wedge \sigma' \in \{\perp \mid \perp \in Q\}\}
\end{array}$$

(for the first term, $\sigma' \in \Sigma$ and $\sigma' \in Q$ so $\sigma' \in Q \cap \Sigma \triangleq Q_{\not\perp}$, while, for the second term, $\sigma' \in \{\perp\}$ and $\sigma' \in Q$ so $\sigma' \in Q \cap \{\perp\} = \{\perp \mid \perp \in Q\}$)

$= \quad \mathsf{pre}(r \cap (\Sigma \times \Sigma))Q_{\not\perp} \cup \mathsf{pre}(r \cap (\Sigma \times \{\perp\}))\{\perp \mid \perp \in Q\}$ ⟅def. (38) of pre⟆ □

- join covering $\quad P \subseteq A \cup B \quad \Leftrightarrow \quad \exists P_A, P_B \,.\, P = P_A \cup P_B \wedge P_A \subseteq A \wedge P_B \subseteq B$ (76)

PROOF OF (76).

(⇒) take $P_A = P \cap A$ and $P_B = P \cap B$ so that $P_A \subseteq P \cap A \subseteq A$ and $P_B \subseteq P \cap B \subseteq B$;

(⇐) we have $P = P_A \cup P_B \subseteq A \cup B$. □

- Boolean precondition $\quad \mathsf{pre}(\llbracket \mathsf{B} \rrbracket)Q = \quad \mathcal{B}\llbracket \mathsf{B} \rrbracket \cap Q$ (77)

PROOF OF (77).

$$
\begin{aligned}
& \mathsf{pre}(\llbracket \mathsf{B} \rrbracket)Q \\
= \quad & \{\sigma \mid \exists \sigma' \in Q \,.\, \langle \sigma,\ \sigma' \rangle \in \llbracket \mathsf{B} \rrbracket\} && \text{⟅def. (38) of pre⟆} \\
= \quad & \{\sigma \mid \exists \sigma' \in Q \,.\, \sigma = \sigma' \in \mathcal{B}\llbracket \mathsf{B} \rrbracket\} && \text{⟅def. } \llbracket \mathsf{B} \rrbracket \triangleq \{\langle \sigma,\ \sigma \rangle \mid \sigma \in \mathcal{B}\llbracket \mathsf{B} \rrbracket\}\text{⟆} \\
= \quad & Q \cap \mathcal{B}\llbracket \mathsf{B} \rrbracket && \text{⟅def. intersection } \cap\text{⟆} \quad \square
\end{aligned}
$$

- F^e commutation $\quad F^e$ and $F^e_{\mathsf{pre}} \triangleq \boldsymbol{\lambda} X \bullet \boldsymbol{\lambda} Q \bullet Q \cup (\mathcal{B}\llbracket \mathsf{B} \rrbracket \cap \mathsf{pre}\llbracket \mathsf{S} \rrbracket^e(X(Q)))$ commute for $\alpha(X) = \boldsymbol{\lambda} Q \bullet \mathsf{pre}(X)Q$ (78)

PROOF OF (78).

$$
\begin{aligned}
& \alpha(F^e(X)) \\
= \quad & \mathsf{pre}(F^e(X)) && \text{⟅def. } \alpha\text{⟆} \\
= \quad & \mathsf{pre}(\mathsf{id} \cup (\llbracket \mathsf{B} \rrbracket \fatsemi \llbracket \mathsf{S} \rrbracket^e \fatsemi X)) && \text{⟅def. (49) of } F^e\text{⟆} \\
= \quad & \mathsf{pre}(\mathsf{id}) \mathbin{\dot{\cup}} \mathsf{pre}(\llbracket \mathsf{B} \rrbracket \fatsemi \llbracket \mathsf{S} \rrbracket^e \fatsemi X)) && \text{⟅join preservation (73)⟆} \\
= \quad & \boldsymbol{\lambda} Q \bullet \mathsf{pre}(\mathsf{id})Q \mathbin{\dot{\cup}} \mathsf{pre}(\llbracket \mathsf{B} \rrbracket \fatsemi \llbracket \mathsf{S} \rrbracket^e \fatsemi X)Q && \text{⟅def. } \boldsymbol{\lambda} \text{ notation⟆} \\
= \quad & \boldsymbol{\lambda} Q \bullet Q \cup \mathsf{pre}\llbracket \mathsf{B} \rrbracket(\mathsf{pre}\llbracket \mathsf{S} \rrbracket^e(\mathsf{pre}(X)Q)) && \text{⟅def. (38) of pre and composition (74)⟆} \\
= \quad & \boldsymbol{\lambda} Q \bullet Q \cup (\mathcal{B}\llbracket \mathsf{B} \rrbracket \cap \mathsf{pre}\llbracket \mathsf{S} \rrbracket^e(\mathsf{pre}(X)Q)) && \text{⟅Boolean precondition (77)⟆} \\
= \quad & \boldsymbol{\lambda} Q \bullet Q \cup (\mathcal{B}\llbracket \mathsf{B} \rrbracket \cap \mathsf{pre}\llbracket \mathsf{S} \rrbracket^e(\alpha(X)Q)) && \text{⟅def. } \alpha = \mathsf{pre}(X)\text{⟆} \\
= \quad & F^e_{\mathsf{pre}}(\alpha(X)) && \text{⟅def. (78) of } F^e_{\mathsf{pre}}\text{⟆} \quad \square
\end{aligned}
$$

- Observe that $F^e_{\mathsf{pre}}(X)Q = \bar{F}^e_Q(X(Q))$ by defining $\bar{F}^e_Q \triangleq \boldsymbol{\lambda} X \bullet Q \cup (\mathcal{B}\llbracket \mathsf{B} \rrbracket \cap \mathsf{pre}\llbracket \mathsf{S} \rrbracket^e(X))$ (79)

- $F^\perp$ commutation $\quad F^\perp$ and $F^\perp_{\mathsf{pre}} \triangleq \boldsymbol{\lambda} X \bullet \boldsymbol{\lambda} Q \bullet \mathcal{B}\llbracket \mathsf{B} \rrbracket \cap (\mathsf{pre}\llbracket \mathsf{S} \rrbracket^e(X(Q))$ commute for $\alpha(X) = \boldsymbol{\lambda} Q \bullet \mathsf{pre}(X)Q$ (80)

PROOF OF (80).

$$
\begin{aligned}
& \alpha(F^\perp(X)) \\
= \quad & \mathsf{pre}(F^\perp(X)) && \text{⟅def. } \alpha\text{⟆} \\
= \quad & \mathsf{pre}(\llbracket \mathsf{B} \rrbracket \fatsemi \llbracket \mathsf{S} \rrbracket^e \fatsemi X) && \text{⟅def. (50) of } F^\perp\text{⟆} \\
= \quad & \mathsf{pre}\llbracket \mathsf{B} \rrbracket \circ \mathsf{pre}\llbracket \mathsf{S} \rrbracket^e \circ \mathsf{pre}(X) && \text{⟅composition (74)⟆} \\
= \quad & \boldsymbol{\lambda} Q \bullet \mathsf{pre}\llbracket \mathsf{B} \rrbracket \circ \mathsf{pre}\llbracket \mathsf{S} \rrbracket^e \circ \mathsf{pre}(X)(Q) && \text{⟅def. } \boldsymbol{\lambda} \text{ notation⟆} \\
= \quad & \boldsymbol{\lambda} Q \bullet \mathsf{pre}\llbracket \mathsf{B} \rrbracket(\mathsf{pre}\llbracket \mathsf{S} \rrbracket^e(\mathsf{pre}(X)Q) && \text{⟅def. function composition } \circ\text{⟆}
\end{aligned}
$$

$$= \quad \boldsymbol{\lambda} Q \bullet \mathcal{B}[\![\mathsf{B}]\!] \cap (\mathsf{pre}[\![\mathsf{S}]\!]^e(\mathsf{pre}(X)Q)) \qquad \text{\{composition (74)\}}$$

$$= \quad \boldsymbol{\lambda} Q \bullet \mathcal{B}[\![\mathsf{B}]\!] \cap (\mathsf{pre}[\![\mathsf{S}]\!]^e(\alpha(X)Q)) \qquad \text{\{def. } \alpha = \mathsf{pre}(X)\text{\}}$$

$$= \quad F^{\perp}_{\mathsf{pre}}(\alpha(X)) \qquad \text{\{def. (80) of } F^{\perp}_{\mathsf{pre}}\text{\}} \qquad \square$$

- Let us now calculate $\alpha_{\mathsf{pre}}([\![\mathsf{S}]\!]_{\perp})$. We consider the case of the iteration `while(B) S` (which covers the conditional and sequential composition). Let us start with the easier case (B).

$$\{\langle P,\ Q,\ T\rangle \mid P \subseteq \mathsf{pre}([\![\mathtt{while(B)\ \ S}]\!]^b)T\}$$

$$= \{\langle P,\ Q,\ T\rangle \mid P \subseteq \mathsf{pre}(\varnothing)T\} \qquad \text{\{def. (52) of the natural relational semantics\}}$$

$$= \{\langle P,\ Q,\ T\rangle \mid P \subseteq \varnothing\} \qquad \text{\{pre preserves arbitrary joins (73), hence } \varnothing\text{\}}$$

$$= \{\langle \varnothing,\ Q,\ T\rangle \mid Q \in \wp(\Sigma_{\perp}) \wedge T \in \wp(\Sigma)\} \qquad \text{\{def. } \subseteq\text{\}}$$

which we can interpret as "if you believe that a loop breaks out of an outer loop, which is impossible, then you can state anything on the behavior of this outer loop".

J.6 Design of the Deductive System for Tthe iteration `while(B) S` in Case (A) of (58)

$$\{\langle P,\ Q,\ T\rangle \mid P \subseteq \mathsf{pre}([\![\mathtt{while(B)\ \ S}]\!]^e \cup [\![\mathtt{while(B)\ \ S}]\!]^{\perp})Q\}$$

$$= \{\langle P,\ Q,\ T\rangle \mid P \subseteq (\mathsf{pre}([\![\mathtt{while(B)\ \ S}]\!]^e)Q \cup \mathsf{pre}([\![\mathtt{while(B)\ \ S}]\!]^{\perp})Q)\} \qquad \text{\{join preservation (73)\}}$$

$$= \{\langle P,\ Q,\ T\rangle \mid P \subseteq (\mathsf{pre}([\![\mathtt{while(B)\ \ S}]\!]^e)Q_{\not\perp} \cup \mathsf{pre}([\![\mathtt{while(B)\ \ S}]\!]^{\perp})\{\perp \mid \perp \in Q\})\}$$

$$\text{\{partitioning (75) with } [\![\mathtt{while(B)\ \ S}]\!]^e \in \wp(\Sigma\times\Sigma) \text{ and } [\![\mathtt{while(B)\ \ S}]\!]^{\perp} \in \wp(\Sigma\times\{\perp\})\text{\}}$$

$$= \{\langle P^e \cup P^{\perp},\ Q,\ T\rangle \mid P^e \subseteq \mathsf{pre}([\![\mathtt{while(B)\ \ S}]\!]^e)Q_{\not\perp} \wedge P^{\perp} \subseteq \mathsf{pre}([\![\mathtt{while(B)\ \ S}]\!]^{\perp})\{\perp \mid \perp \in Q\})\}$$

{By join covering (76) of P by the initial states P^e from which executions may terminate in $Q_{\not\perp}$ and those $P^{\perp}$with possible nonterminating executions.}

$$= \{\langle P^e \cup P^{\perp},\ Q,\ T\rangle \mid P^e \subseteq \mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e \mathbin{\fatsemi} [\![\neg\mathsf{B}]\!] \cup [\![\mathsf{B}]\!] \mathbin{\fatsemi} [\![\mathsf{S}]\!]^b)Q_{\not\perp} \wedge P^{\perp} \subseteq \mathsf{pre}((\mathsf{lfp}^{\subseteq} F^e \mathbin{\fatsemi} [\![\mathsf{B}]\!] \mathbin{\fatsemi} [\![\mathsf{S}]\!]^{\perp}) \cup \mathsf{gfp}^{\subseteq} F^{\perp})\{\perp \mid \perp \in Q\})\}$$

$$\text{\{by definitions (51) and (53)\}}$$

$$= \{\langle P^e \cup P^{\perp},\ Q,\ T\rangle \mid P^e \subseteq (\mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e \mathbin{\fatsemi} [\![\neg\mathsf{B}]\!])Q_{\not\perp} \cup \mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e \mathbin{\fatsemi} [\![\mathsf{B}]\!] \mathbin{\fatsemi} [\![\mathsf{S}]\!]^b)Q_{\not\perp}) \wedge P^{\perp} \subseteq (\mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e \mathbin{\fatsemi} [\![\mathsf{B}]\!] \mathbin{\fatsemi} [\![\mathsf{S}]\!]^{\perp})\{\perp \mid \perp \in Q\} \cup \mathsf{pre}(\mathsf{gfp}^{\subseteq} F^{\perp})\{\perp \mid \perp \in Q\}))\} \qquad \text{\{by join preservation (73)\}}$$

$$= \{\langle P^{ok} \cup P^{br} \cup P^{\perp}_b \cup P^{\perp}_{\ell},\ Q,\ T\rangle \mid P^{ok} \subseteq \mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e \mathbin{\fatsemi} [\![\neg\mathsf{B}]\!])Q_{\not\perp} \wedge P^{br} \subseteq \mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e \mathbin{\fatsemi} [\![\mathsf{B}]\!] \mathbin{\fatsemi} [\![\mathsf{S}]\!]^b)Q_{\not\perp}) \wedge P^{\perp}_b \subseteq (\mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e \mathbin{\fatsemi} [\![\mathsf{B}]\!] \mathbin{\fatsemi} [\![\mathsf{S}]\!]^{\perp})\{\perp \mid \perp \in Q\}) \wedge P^{\perp}_{\ell} \subseteq (\mathsf{pre}(\mathsf{gfp}^{\subseteq} F^{\perp})\{\perp \mid \perp \in Q\})\}$$

{By join covering (76) of P^e into the initial states P^{ok} for which the loop terminates normally and P^{br} for which the loop terminates through a break and join convering of $P^{\perp}$ into the initial states $P^{\perp}_b$ from which the loop body ultimately does not terminate and $P^{\perp}_{\ell}$ from which there are infinitely many terminating executions of the loop body}

$$= \{\langle P^{ok} \cup P^{br} \cup P^{\perp}_b \cup P^{\perp}_{\ell},\ Q,\ T\rangle \mid P^{ok} \subseteq \mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e)(\mathsf{pre}[\![\neg\mathsf{B}]\!]Q_{\not\perp}) \wedge P^{br} \subseteq \mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e)(\mathsf{pre}([\![\mathsf{B}]\!] \mathbin{\fatsemi} [\![\mathsf{S}]\!]^b)Q_{\not\perp}) \wedge P^{\perp}_b \subseteq \mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e)(\mathsf{pre}([\![\mathsf{B}]\!] \mathbin{\fatsemi} \mathsf{S}^{\perp})\{\perp \mid \perp \in Q\}) \wedge P^{\perp}_{\ell} \subseteq (\mathsf{pre}(\mathsf{gfp}^{\subseteq} F^{\perp})\{\perp \mid \perp \in Q\})\}$$

$$\text{\{composition (74)\}}$$

$$= \{\langle P \cup P^{\perp}_{\ell}, Q, T\rangle \mid P \subseteq (\mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e)(\mathsf{pre}[\![\neg\mathsf{B}]\!]Q_{\not\perp}) \cup \mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e)(\mathsf{pre}([\![\mathsf{B}]\!] \mathbin{\fatsemi} [\![\mathsf{S}]\!]^b)Q_{\not\perp}) \cup \mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e)(\mathsf{pre}([\![\mathsf{B}]\!] \mathbin{\fatsemi} [\![\mathsf{S}]\!]^{\perp})\{\perp \mid \perp \in Q\})) \wedge P^{\perp}_{\ell} \subseteq (\mathsf{pre}(\mathsf{gfp}^{\subseteq} F^{\perp})\{\perp \mid \perp \in Q\})\}$$

$$\text{\{join covering (76) with } P = P^{ok} \cup P^{br} \cup P^{\perp}_b\text{\}}$$

$$= \{\langle P \cup P^{\perp}_{\ell}, Q, T\rangle \mid P \subseteq \mathsf{pre}(\mathsf{lfp}^{\subseteq} F^e)((\mathsf{pre}[\![\neg\mathsf{B}]\!]Q_{\not\perp}) \cup (\mathsf{pre}([\![\mathsf{B}]\!] \mathbin{\fatsemi} [\![\mathsf{S}]\!]^b)Q_{\not\perp}) \cup (\mathsf{pre}([\![\mathsf{B}]\!] \mathbin{\fatsemi} [\![\mathsf{S}]\!]^{\perp})\{\perp \mid \perp \in Q\})) \wedge P^{\perp}_{\ell} \subseteq (\mathsf{pre}(\mathsf{gfp}^{\subseteq} F^{\perp})\{\perp \mid \perp \in Q\})\} \qquad \text{\{join preservation (73)\}}$$

$= \{\langle P \cup P^{\perp}_{\ell}, Q, T\rangle \mid P \subseteq (\mathsf{lfp}^{\dot\subseteq} F^{e}_{\mathsf{pre}})((\mathsf{pre}[\![\neg\mathsf{B}]\!]Q_{\not\perp}) \cup (\mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{b})Q_{\not\perp}) \cup (\mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{\perp})\{\perp \mid \perp \in \cup(\mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{\perp})\{\perp \mid \perp \in Q\})) \wedge P^{\perp}_{\ell} \subseteq (\mathsf{pre}(\mathsf{gfp}^{\subseteq} F^{\perp})\{\perp \mid \perp \in Q\})\}$

≀by the Galois connection $\langle \mathsf{pre}, \mathsf{pre}^{-1}\rangle$ of (12) F^{e} and F^{e}_{pre} are increasing, and, by (78), they commute so that $\mathsf{pre}(\mathsf{lfp}^{\subseteq} F^{e}) = \mathsf{lfp}^{\dot\subseteq} F^{e}_{\mathsf{pre}}$ by the fixpoint abstraction Th. II.2.1≀

$= \{\langle P \cup P^{\perp}_{\ell}, Q, T\rangle \mid P \subseteq \mathsf{lfp}^{\subseteq} \bar{F}^{e}_{((\mathsf{pre}[\![\neg\mathsf{B}]\!]Q_{\not\perp}) \cup (\mathsf{pre}([\![\mathsf{B}]\!] ⨾ [\![\mathsf{S}]\!]^{b})Q_{\not\perp}) \cup (\mathsf{pre}([\![\mathsf{B}]\!] ⨾ [\![\mathsf{S}]\!]^{\perp})\{\perp \mid \perp \in Q\}))} \wedge P^{\perp}_{\ell} \subseteq (\mathsf{pre}(\mathsf{gfp}^{\subseteq} F^{\perp})\{\perp \mid \perp \in Q\})\}$

≀by $F^{e}_{\mathsf{pre}}(X)Q = \bar{F}^{e}_{Q}(X(Q))$ at (79) and the pointwise abstraction corollary II.2.2≀

$= \{\langle P \cup P^{\perp}_{\ell}, Q, T\rangle \mid P \subseteq \mathsf{lfp}^{\subseteq} \boldsymbol{\lambda} X \bullet (\mathsf{pre}[\![\neg\mathsf{B}]\!]Q_{\not\perp}) \cup (\mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{b})Q_{\not\perp}) \cup (\mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{\perp})\{\perp \mid \perp \in Q\}) \cup (\mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{e})X) \wedge P^{\perp}_{\ell} \subseteq (\mathsf{pre}(\mathsf{gfp}^{\subseteq} F^{\perp})\{\perp \mid \perp \in Q\})\}$ ≀def. (79) of $\bar{F}^{e}_{Q}$≀

$= \{\langle P \cup P^{\perp}_{\ell}, Q, T\rangle \mid P \subseteq \mathsf{lfp}^{\subseteq} \boldsymbol{\lambda} X \bullet (\mathsf{pre}[\![\neg\mathsf{B}]\!]Q_{\not\perp}) \cup (\mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{b})Q_{\not\perp}) \cup (\mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{\perp})\{\perp \mid \perp \in Q\}) \cup (\mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{e})X) \wedge (\!|\, \perp \in Q \;?\; P^{\perp}_{\ell} \subseteq (\mathsf{pre}(\mathsf{gfp}^{\subseteq} F^{\perp})\{\perp\} \;:\; P^{\perp}_{\ell} = \varnothing \,|\!))\}$ ≀since $\mathsf{pre}(r)\varnothing = \varnothing$≀

— Since we proceed by structural induction on statements, $\mathsf{pre}[\![\mathsf{S}]\!]^{\perp}(\{\perp \mid \perp \in Q\})$ is already determined so can be considered to be a constant the same way that $\mathcal{B}[\![\mathsf{B}]\!]$ or $\mathcal{B}[\![\neg\mathsf{B}]\!]$ are. We now have to choose induction principles to under approximate the fixpoints.

— Since pre preserves arbitrary joins, its least fixpoint iterations are stable at ω. So the hypotheses of Th. II.3.6 for $P \subseteq \mathsf{lfp}^{\subseteq} \boldsymbol{\lambda} X \bullet A \cup \mathsf{pre}(r)X$ become

$$\exists \langle I^{n}, n \in \mathbb{N}\rangle \,.\, I^{0} = \varnothing \wedge \forall n \in \mathbb{N} \,.\, I^{n} \subseteq I^{n+1} \subseteq A \cup \mathsf{pre}(r)I^{n} \wedge \exists \ell \in \mathbb{N} \,.\, P \subseteq I^{\ell} \tag{60}$$

since, $\langle I^{n}, n \in \mathbb{N}\rangle$ being an increasing chain for $\subseteq$, we have $P \subseteq \bigcup_{n\in\mathbb{N}} I^{n}$ if and only if $\exists \ell \in \mathbb{N} \,.\, P \subseteq I^{\ell}$.

— The hypotheses of the dual of Th. II.3.4 to under approximate the image of the greatest fixpoint $\mathsf{gfp}^{\subseteq} F^{\perp}$ by $\boldsymbol{\lambda} r \bullet \mathsf{pre}(r)\{\perp\}$ are that there exists $J \in \wp(\Sigma_{\perp})$ such that

(1) $\mathsf{pre}(\varnothing)\{\perp\} \subseteq J$ which trivially holds since $\mathsf{pre}(\varnothing)\{\perp\} = \varnothing$.

(2) $\forall X \in \wp(\Sigma \times \Sigma_{\perp}) \,.\, \mathsf{pre}(X)\{\perp\} \subseteq J \Rightarrow \mathsf{pre}(F^{\perp}(X))\{\perp\} \subseteq J$

$\Leftrightarrow (\!|\, \perp \in Q \;?\; \forall X \,.\, \mathsf{pre}(X)\{\perp\} \subseteq J \Rightarrow \mathsf{pre}(F^{\perp}(X))\{\perp\} \subseteq J \;:\; \mathsf{true} \,|\!)$ ≀since $\mathsf{pre}(X)\varnothing = \varnothing$≀

$\Leftrightarrow (\!|\, \perp \in Q \;?\; \forall X \,.\, \mathsf{pre}(X)\{\perp\} \subseteq J \Rightarrow \mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{e} \,⨾\, X)\{\perp\} \subseteq J \;:\; \mathsf{true} \,|\!)$ ≀def. (50) of $F^{\perp}$≀

$\Leftrightarrow (\!|\, \perp \in Q \;?\; \forall X \,.\, \mathsf{pre}(X)\{\perp\} \subseteq J \Rightarrow \mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{e})(\mathsf{pre}(X)\{\perp\}) \subseteq J \;:\; \mathsf{true} \,|\!)$

≀composition (74)≀

$\Leftrightarrow (\!|\, \perp \in Q \;?\; \mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{e})(J) \subseteq J \;:\; \mathsf{true} \,|\!)$

≀($\Rightarrow$) take $X = \{\langle \sigma, \perp\rangle \mid \sigma \in J\}$ so that $\mathsf{pre}(X)\{\perp\} = J$;
($\Leftarrow$) for all $X \in \wp(\Sigma \times \Sigma_{\perp})$, if $\mathsf{pre}(X)\{\perp\} \subseteq J$ by hypothesis then $\mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{e})(\mathsf{pre}(X))\{\perp\} \subseteq \mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{e})(J)$ since $\mathsf{pre}(r)$ preserves joins so is increasing, and therefore the hypothesis $\mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{e})(J) \subseteq J$ implies $\mathsf{pre}([\![\mathsf{B}]\!] \,⨾\, [\![\mathsf{S}]\!]^{e})(\mathsf{pre}(X)\{\perp\}) \subseteq J$ by transitivity.≀

(3) for any $\sqsubseteq$-increasing chain $\langle X^{\delta}, \delta \in \mathbb{O}\rangle$ of elements $X^{\delta} \sqsubseteq \mathsf{lfp}^{\sqsubseteq} F^{\perp}$, $\forall \beta < \lambda \,.\, \mathsf{pre}(X^{\beta})\{\perp\} \subseteq J$ implies $\mathsf{pre}(\bigcup_{\beta<\lambda} X^{\beta})\{\perp\} \subseteq J$. The case $\perp \notin Q$ being trivially true, let us assume that $\mathsf{pre}(X^{\beta})\{\perp\} \subseteq J$ that is, $\{\sigma \mid \langle \sigma, \perp\rangle \in X^{\beta}\} \subseteq J$ for all $\beta < \lambda$, λ limit ordinal. Then we have

$$\mathsf{pre}(\bigcup_{\beta<\lambda} X^{\beta})\{\perp\} \subseteq J$$

$\Leftrightarrow \{\sigma \mid \langle \sigma, \perp\rangle \in \bigcup_{\beta<\lambda} X^{\beta}\} \subseteq J$ ≀def. (38) of pre≀

$\Leftrightarrow \bigcup_{\beta<\lambda} \{\sigma \mid \langle \sigma, \perp\rangle \in X^{\beta}\} \subseteq J$ ≀def. union $\cup$≀

$\Leftrightarrow \forall \beta < \lambda \,.\, \{\sigma \mid \langle\sigma,\, \bot\rangle \in X^\beta\} \subseteq J$ ≀def. inclusion ⊆≀

$\Leftrightarrow$ true ≀by hypothesis≀

(4) $J \subseteq P$.

So the four hypotheses of the dual of Th. II.3.4 for proving that $P_\ell^\bot \subseteq (\mathsf{pre}(\mathsf{gfp}^\subseteq F^\bot)\{\bot\}$ boil down to

$$\exists J \in \wp(\Sigma_\bot) \,.\, \mathsf{pre}(\llbracket \mathsf{B}\rrbracket \,\mathbin{⨾}\, \llbracket \mathsf{S}\rrbracket^e)(J) \subseteq J \wedge P_\ell^\bot \subseteq J \tag{61}$$

so we can go on with our formal calculation. We left it at

$\Leftrightarrow \{\langle P \cup P_\ell^\bot,\, Q,\, T\rangle \mid P \subseteq \mathsf{lfp}^\subseteq \boldsymbol{\lambda} X \bullet (\mathsf{pre}\llbracket\neg\mathsf{B}\rrbracket Q_{\not\bot}) \cup (\mathsf{pre}(\llbracket\mathsf{B}\rrbracket \mathbin{⨾} \llbracket\mathsf{S}\rrbracket^b) Q_{\not\bot}) \cup (\mathsf{pre}(\llbracket\mathsf{B}\rrbracket \mathbin{⨾} \llbracket\mathsf{S}\rrbracket^\bot)\{\bot \mid \bot \in Q\}) \cup (\mathsf{pre}(\llbracket\mathsf{B}\rrbracket \mathbin{⨾} \llbracket\mathsf{S}\rrbracket^e)X) \wedge (\!| \bot \in Q \;?\; P_\ell^\bot \subseteq (\mathsf{pre}(\mathsf{gfp}^\subseteq F^\bot)\{\bot\} \;:\; P_\ell^\bot = \varnothing |\!)\}$

$\Leftrightarrow \{\langle P \cup P_\ell^\bot, Q, T\rangle \mid \exists\langle I^n, n \in \mathbb{Q}\rangle \,.\, I^0 = \varnothing \wedge \forall n \in \mathbb{N} \,.\, I^n \subseteq I^{n+1} \subseteq (\mathsf{pre}\llbracket\neg\mathsf{B}\rrbracket Q_{\not\bot}) \cup (\mathsf{pre}(\llbracket\mathsf{B}\rrbracket \mathbin{⨾} \llbracket\mathsf{S}\rrbracket^b) Q_{\not\bot}) \cup (\mathsf{pre}(\llbracket\mathsf{B}\rrbracket \mathbin{⨾} \llbracket\mathsf{S}\rrbracket^\bot)\{\bot \mid \bot \in Q\}) \cup (\mathsf{pre}(\llbracket\mathsf{B}\rrbracket \mathbin{⨾} \llbracket\mathsf{S}\rrbracket^e) I^n) \wedge \exists \ell \in \mathbb{N} \,.\, P \subseteq I^\ell \wedge \exists J \in \wp(\Sigma_\bot) \,.\, (\!| \bot \in Q \;?\; \mathsf{pre}(\llbracket\mathsf{B}\rrbracket \mathbin{⨾} \llbracket\mathsf{S}\rrbracket^e)(J) \subseteq J \wedge P_\ell^\bot \subseteq J \;:\; P_\ell^\bot = \varnothing |\!)\}$ ≀by (60) and (61)≀

$\Leftrightarrow \{\langle P \cup P_\ell^\bot, Q, T\rangle \mid \exists\langle I^n, n \in \mathbb{Q}\rangle \,.\, I^0 = \varnothing \wedge \forall n \in \mathbb{N} \,.\, I^n \subseteq I^{n+1} \subseteq (\mathcal{B}\llbracket\neg\mathsf{B}\rrbracket \cap Q_{\not\bot}) \cup (\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^b) Q_{\not\bot}) \cup (\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}\llbracket\mathsf{S}\rrbracket^\bot\{\bot \mid \bot \in Q\}) \cup (\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}\llbracket\mathsf{S}\rrbracket^e(I^n)) \wedge \exists \ell \in \mathbb{N} \,.\, P \subseteq I^\ell \wedge \exists J \in \wp(\Sigma_\bot) \,.\, (\!| \bot \in Q \;?\; \mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^e)(J) \subseteq J \wedge P_\ell^\bot \subseteq J \;:\; P_\ell^\bot = \varnothing |\!)\}$ ≀Boolean precondition (77)≀

$\Leftrightarrow \{\langle P \cup P_\ell^\bot, Q, T\rangle \mid \exists\langle I^n, n \in \mathbb{Q}\rangle \,.\, I^0 = \varnothing \wedge \forall n \in \mathbb{N} \,.\, I^n \subseteq I^{n+1} \subseteq (\mathcal{B}\llbracket\neg\mathsf{B}\rrbracket \cap Q_{\not\bot}) \cup (\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^b) Q_{\not\bot}) \cup (\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}\llbracket\mathsf{S}\rrbracket^\bot\{\bot \mid \bot \in Q\}) \cup (\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}\llbracket\mathsf{S}\rrbracket^e(I^n)) \wedge \exists \ell \in \mathbb{N} \,.\, P \subseteq I^\ell \wedge \exists J \in \wp(\Sigma) \,.\, (\!| \bot \in Q \;?\; \mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^e)(J) \subseteq J \wedge P_\ell^\bot \subseteq J \;:\; P_\ell^\bot = \varnothing |\!)\}$

≀since $\mathsf{pre}(\llbracket\mathsf{S}\rrbracket^e)(J) = \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^e)(J \setminus \{\bot\})$ by (51) and (49)≀

$\Leftrightarrow \{\langle P \cup P_\ell^\bot,\, Q,\, T\rangle \mid \exists R^b \,.\, R^b \subseteq \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^b) Q_{\not\bot} \wedge \exists\langle I^n,\, n \in \mathbb{Q}\rangle \,.\, I^0 = \varnothing \wedge \forall n \in \mathbb{N} \,.\, I^n \subseteq I^{n+1} \subseteq (\mathcal{B}\llbracket\neg\mathsf{B}\rrbracket \cap Q_{\not\bot}) \cup (\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap R^b) \cup (\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}\llbracket\mathsf{S}\rrbracket^\bot\{\bot \mid \bot \in Q\}) \cup (\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}\llbracket\mathsf{S}\rrbracket^e(I^n)) \wedge \exists \ell \in \mathbb{N} \,.\, P \subseteq I^\ell \wedge \exists J \in \wp(\Sigma) \,.\, (\!| \bot \in Q \;?\; \mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^e)(J) \subseteq J \wedge P_\ell^\bot \subseteq J \;:\; P_\ell^\bot = \varnothing |\!)\}$

≀by under approximation of $\mathsf{pre}(\llbracket\mathsf{S}\rrbracket^b) Q_{\not\bot}$ by R^b

($\Rightarrow$) Take $R^b = \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^b) Q_{\not\bot}$;

($\Leftarrow$) since $R^b \subseteq \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^b) Q_{\not\bot}$ and $X \subseteq R^b$ implies $X \subseteq \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^b) Q_{\not\bot}$≀

$\Leftrightarrow \{\langle P \cup P_\ell^\bot,\, Q,\, T\rangle \mid \exists R^b \,.\, R^b \subseteq \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^b) Q_{\not\bot} \wedge \exists R^\bot \,.\, R^\bot \subseteq \mathsf{pre}\llbracket\mathsf{S}\rrbracket^\bot\{\bot \mid \bot \in Q\} \wedge \exists\langle I^n,\, n \in \mathbb{Q}\rangle \,.\, I^0 = \varnothing \wedge \forall n \in \mathbb{N} \,.\, I^n \subseteq I^{n+1} \subseteq (\mathcal{B}\llbracket\neg\mathsf{B}\rrbracket \cap Q_{\not\bot}) \cup (\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap R^b) \cup (\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap R^\bot) \cup (\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}\llbracket\mathsf{S}\rrbracket^e(I^n)) \wedge \exists \ell \in \mathbb{N} \,.\, P \subseteq I^\ell \wedge \exists J \in \wp(\Sigma_\bot) \,.\, (\!| \bot \in Q \;?\; \mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^e)(J) \subseteq J \wedge P_\ell^\bot \subseteq J \;:\; P_\ell^\bot = \varnothing |\!)\}$

≀by under approximation of $\mathsf{pre}\llbracket\mathsf{S}\rrbracket^\bot\{\bot \mid \bot \in Q\}$ by $R^\bot$

($\Rightarrow$) Take $R^\bot = \mathsf{pre}\llbracket\mathsf{S}\rrbracket^\bot\{\bot \mid \bot \in Q\}$;

($\Leftarrow$) since $R^\bot \subseteq \mathsf{pre}\llbracket\mathsf{S}\rrbracket^\bot\{\bot \mid \bot \in Q\}$ and $X \subseteq R^\bot$ implies $X \subseteq \mathsf{pre}\llbracket\mathsf{S}\rrbracket^\bot\{\bot \mid \bot \in Q\}$≀

$\Leftrightarrow \{\langle P \cup P_\ell^\bot,\, Q,\, T\rangle \mid \exists R^b \,.\, R^b \subseteq \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^b) Q_{\not\bot} \wedge \exists R^\bot \,.\, R^\bot \subseteq \mathsf{pre}\llbracket\mathsf{S}\rrbracket_\bot\{\bot \mid \bot \in Q\} \wedge \exists\langle I^n,\, n \in \mathbb{Q}\rangle \,.\, I^0 = \varnothing \wedge \forall n \in \mathbb{N} \,.\, I^n \subseteq I^{n+1} \subseteq (\mathcal{B}\llbracket\neg\mathsf{B}\rrbracket \cap Q_{\not\bot}) \cup (\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap R^b) \cup (\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap R^\bot) \cup ((\mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}\llbracket\mathsf{S}\rrbracket^e(I^n)) \wedge \exists \ell \in \mathbb{N} \,.\, P \subseteq I^\ell \wedge \exists J \in \wp(\Sigma_\bot) \,.\, (\!| \bot \in Q \;?\; \mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^e)(J) \subseteq J \wedge P_\ell^\bot \subseteq J \;:\; P_\ell^\bot = \varnothing |\!)\}$

≀def. (54) of the relational semantics and (38) of pre so that $\mathsf{pre}\llbracket\mathsf{S}\rrbracket^\bot\varnothing = \mathsf{pre}\llbracket\mathsf{S}\rrbracket_\bot\varnothing$ and $\mathsf{pre}\llbracket\mathsf{S}\rrbracket^\bot\{\bot\} = \mathsf{pre}\llbracket\mathsf{S}\rrbracket_\bot\{\bot\}$ ≀

$\Leftrightarrow \{\langle P \cup P_\ell^\bot,\, Q,\, T\rangle \mid \exists R^b \,.\, R^b \subseteq \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^b) Q_{\not\bot} \wedge \exists R^\bot \,.\, \langle R^\bot,\, \{\bot \mid \bot \in Q\},\, \varnothing\rangle \in \alpha_{\mathsf{pre}}(\llbracket\mathsf{S}\rrbracket_\bot) \wedge \exists\langle I^n,\, n \in \mathbb{Q}\rangle \,.\, I^0 = \varnothing \wedge \forall n \in \mathbb{N} \,.\, I^n \subseteq I^{n+1} \subseteq (\mathcal{B}\llbracket\neg\mathsf{B}\rrbracket \cap Q_{\not\bot}) \cup (\llbracket\mathsf{B}\rrbracket \cap R^b) \cup (\llbracket\mathsf{B}\rrbracket \cap R^\bot) \cup (\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}\llbracket\mathsf{S}\rrbracket^e(I^n)) \wedge \exists \ell \in \mathbb{N} \,.\, P \subseteq I^\ell \wedge \exists J \in \wp(\Sigma_\bot) \,.\, (\!| \bot \in Q \;?\; \mathcal{B}\llbracket\mathsf{B}\rrbracket \cap \mathsf{pre}(\llbracket\mathsf{S}\rrbracket^e)(J) \subseteq J \wedge P_\ell^\bot \subseteq J \;:\; P_\ell^\bot = \varnothing |\!)\}$

≀def. (57) of α_{pre}≀

$\Leftrightarrow \{\langle P \cup P_\ell^\perp,\ Q,\ T\rangle \mid \exists R^b \,.\, R^b \subseteq \mathsf{pre}(\llbracket \mathsf{S}\rrbracket^b)Q_{\not\perp} \wedge \exists R^\perp \,.\, \langle R^\perp,\ \{\perp \mid \perp \in Q\},\ \varnothing\rangle \in \alpha_{\mathsf{pre}}(\llbracket \mathsf{S}\rrbracket_\perp) \wedge \exists\langle I^n,\ n \in \mathbb{Q}\rangle \,.\, I^0 = \varnothing \wedge \forall n \in \mathbb{N} \,.\, \exists R_n^e \,.\, \langle R_n^e,\ I^n,\ \varnothing\rangle \in \alpha_{\mathsf{pre}}(\llbracket \mathsf{S}\rrbracket_\perp) \wedge I^n \subseteq I^{n+1} \subseteq (\mathcal{B}\llbracket \neg\mathsf{B}\rrbracket \cap Q_{\not\perp}) \cup (\llbracket \mathsf{B}\rrbracket \cap R^b) \cup (\llbracket \mathsf{B}\rrbracket \cap R^\perp) \cup (\llbracket \mathsf{B}\rrbracket \cap R_n^e) \wedge \exists \ell \in \mathbb{N} \,.\, P \subseteq I^\ell \wedge \exists J \in \wp(\Sigma_\perp) \,.\, \llparenthesis \perp \in Q \mathrel{?} \mathcal{B}\llbracket \mathsf{B}\rrbracket \cap \mathsf{pre}(\llbracket \mathsf{S}\rrbracket^e)(J) \subseteq J \wedge P_\ell^\perp \subseteq J \fatsemi P_\ell^\perp = \varnothing \rrparenthesis\}$

$\wr$by under approximation R_n^e of $\mathsf{pre}\llbracket \mathsf{S}\rrbracket^e(I^n)$$\int$

$\Leftrightarrow \{\langle P \cup P_\ell^\perp,\ Q,\ T\rangle \mid \exists R^b \,.\, R^b \subseteq \mathsf{pre}(\llbracket \mathsf{S}\rrbracket^b)Q_{\not\perp} \wedge \exists R^\perp \,.\, \langle R^\perp,\ \{\perp \mid \perp \in Q\},\ \varnothing\rangle \in \alpha_{\mathsf{pre}}(\llbracket \mathsf{S}\rrbracket_\perp) \wedge \exists\langle I^n,\ n \in \mathbb{Q}\rangle \,.\, I^0 = \varnothing \wedge \forall n \in \mathbb{N} \,.\, \exists R_n^e \,.\, \langle R_n^e, I^n, \varnothing\rangle \in \alpha_{\mathsf{pre}}(\llbracket \mathsf{S}\rrbracket_\perp) \wedge I^n \subseteq I^{n+1} \subseteq (\mathcal{B}\llbracket \neg\mathsf{B}\rrbracket \cap Q_{\not\perp}) \cup (\llbracket \mathsf{B}\rrbracket \cap R^b) \cup (\llbracket \mathsf{B}\rrbracket \cap R^\perp) \cup (\llbracket \mathsf{B}\rrbracket \cap R_n^e) \wedge \exists \ell \in \mathbb{N} \,.\, P \subseteq I^\ell \wedge \exists J \in \wp(\Sigma_\perp) \,.\, \exists R_\ell^\perp \,.\, \langle R_\ell^\perp,\ J,\ \varnothing\rangle \in \alpha_{\mathsf{pre}}(\llbracket \mathsf{S}\rrbracket) \wedge \llparenthesis \perp \in Q \mathrel{?} \mathcal{B}\llbracket \mathsf{B}\rrbracket \cap R_\ell^\perp \subseteq J \wedge P_\ell^\perp \subseteq J \fatsemi P_\ell^\perp = \varnothing \rrparenthesis\}$

$\wr$by under approximation $R_\ell^\perp$ of $\mathsf{pre}(\llbracket \mathsf{S}\rrbracket^e)(J)$$\int$

Following Sect. II.5, the theory $\alpha_{\mathsf{pre}}(\llbracket \texttt{while(B) S}\rrbracket_\perp)$ can be equivalently defined by the following deductive system.

$$\frac{\begin{array}{c} I^0 = \varnothing \\ \{R^b\}\ \mathsf{S}\ \overleftarrow{\{}ok : \varnothing, br : Q_{\not\perp}\} \\ \{R^\perp\}\ \mathsf{S}\ \overleftarrow{\{}ok : \{\perp \mid \perp \in Q\}, br : \varnothing\} \\ \forall n \in \mathbb{N} \,.\, \{R_n^e\}\ \llbracket \mathsf{S}\rrbracket\ \overleftarrow{\{}ok : I^n, br : \varnothing\} \wedge \\ I^n \subseteq I^{n+1} \subseteq (\mathcal{B}\llbracket \neg\mathsf{B}\rrbracket \cap Q_{\not\perp}) \cup (\mathcal{B}\llbracket \mathsf{B}\rrbracket \cap R^b) \cup (\mathcal{B}\llbracket \mathsf{B}\rrbracket \cap R^\perp) \cup (\mathcal{B}\llbracket \mathsf{B}\rrbracket \cap R_n^e) \quad \exists \ell \in \mathbb{N} \,.\, P \subseteq I^\ell \\ \llparenthesis \perp \in Q \mathrel{?} \{R_\ell^\perp\}\llbracket \mathsf{S}\rrbracket\ \overleftarrow{\{}ok : J, br : \varnothing\} \wedge \mathcal{B}\llbracket \mathsf{B}\rrbracket \cap R_\ell^\perp \subseteq J \wedge P_\ell^\perp \subseteq J \fatsemi P_\ell^\perp = \varnothing \rrparenthesis \end{array}}{\{P \cup P_\ell^\perp\}\ \texttt{while(B) S}\ \overleftarrow{\{}ok : Q, br : \varnothing\}} \tag{62}$$

K RELATED WORK

The paper provides ample citations in the text and mainly owes to the extensive work on Hoare logic [Apt 1981, 1984; Apt and Olderog 2019, 2021; Hoare 1969], abstract interpretation [Cousot 2021] (for the design of semantics by abstraction [Cousot 2002] and the specification of program properties by Galois connections [Cousot and Cousot 2014]), fixpoint induction [Cousot 2019b; Park 1969] to handle invariants and termination, a previous study of proof methods [Cousot and Cousot 1982] (now expressed with Galois connections and fixpoint induction), and the nonconformist idea of Derek Dreyer, Ralf Jung, and Peter O'Hearn [O'Hearn 2020] originating the interest in incorrectness e.g. [Ascari et al. 2022, 2023; Bruni et al. 2023; de Vries and Koutavas 2011; Feng and Li 2023; Le et al. 2022; Maksimovic et al. 2023; Möller et al. 2021; Naus et al. 2023; Poskitt and Plump 2023; Raad et al. 2020, 2022, 2023; Vanegue 2022; Yan et al. 2022; Zhang and Kaminski 2022; Zilberstein et al. 2023].

Incorrectness has also be studied in the context of logic [Ferrand 1993; Lloyd 1995; Shapiro 1982] and constraint programming [Berre and Tessier 1996] as well as mathematical logic. [Bergmann 1977] discusses [in]correctness in the presence of undefined. The definition of incorrectness requires a referent [Svoboda and Peregrin 2016], which for programming languages is their semantics.

For simplicity, we have considered antecedent and consequent to be sets of states. Using logics instead is a further abstraction with no best abstraction (e.g. non-compact infinite disjunction in first-order logic). This abstraction introduces incompleteness inherited by transformational logics which by themselves are complete (under expressivity conditions of the underlying logic [Blass and Gurevich 2000; Cook 1978, 1981], which amounts to consider the interpretation of logic formulas as sets of states).

The abstraction $\mathsf{post}(\supseteq, \subseteq)$ comes from the specification of reverse Hoare logic by $Q \subseteq \mathsf{post}\llbracket \mathsf{S}\rrbracket P$ in [de Vries and Koutavas 2011] and then a.o. [O'Hearn 2020; Poskitt and Plump 2023; Zhang and Kaminski 2022]. [Cousot and Cousot 1982; Pratt 1976; Yan et al. 2022; Zhang and Kaminski 2022] also consider complement duality (12) [Maksimovic et al. 2023] incorporate reasoning about non-terminating specifications. [Dardinier 2023, Section 5] expresses different (hyper) logics in a common framework. [Schmidt 2007] derives a logic from an abstraction. In the proof theoretic approach, [Harper et al. 1993] designs deductive systems by encoding in the common Edinburgh Logical Framework (LF), e.g. Hoare logic [Avron et al. 1992, Sect. 6.1].