8 Quantum Correctness Reasoning

This chapter develops the *correctness* side of quantum program logic, in the tradition of quantum Hoare logic (QHL) and its later variants. The core idea is the same as in classical Hoare logic: a derivation follows the program structure and produces a compositional proof that a program meets a specification. The quantum-specific aspects are (i) program states are (partial) density operators on a composite Hilbert space, (ii) assertions are operator-valued, and (iii) measurements introduce probabilistic branching that is internalized via subnormalized states and super-operators. Our semantic reference is the denotational semantics of Section 6, where each command denotes a super-operator on the fixed global space H.

Local assertions on subsystems. As in Sections 5–6, we fix a register set Reg and a global space $H \cong \bigotimes_{x\in\mathsf{Reg}} H_x$, and we write H_s for the subsystem space of $s \subseteq \mathsf{Reg}$. A quantum assertion is typically *local*: it is an operator $A \in L(H_s)$ that talks only about the degrees of freedom in s. To evaluate such an assertion on a *global* state $\rho \in \mathcal{D}_{\leq 1}(H)$, we lift it by cylindrical extension:

$$A^{(s)} := A \otimes \mathbf{1}_{H_{\bar{s}}} \in L(H),$$

and we read its *expected value* in state ρ as

$$\langle A \rangle_\rho \;:=\; \mathrm{tr}\big(A^{(s)}\,\rho\big).$$

This expectation-based reading is the core semantic bridge between operator-valued predicates and density operators: it is linear in ρ and compositional with respect to tensoring independent subsystems.

Quantum program logics in the literature differ mainly in what they allow as assertions; the main choices are:

Projections/subspaces. A projection $P = P^\dagger = P^2$ represents a closed subspace of H_s. For normalized ρ, $\mathrm{tr}(P^{(s)}\rho)$ is the probability that a projective measurement of P returns "true". Projection assertions connect cleanly to lattice-theoretic reasoning and support-based reasoning, but closure conditions can be cumbersome in mechanization.

Effects / quantum predicates. An *effect* is an operator $0 \sqsubseteq Q \sqsubseteq \mathbf{1}$ (Hermitian with spectrum in $[0,1]$). Then $\mathrm{tr}(Q^{(s)}\rho) \in [0, \mathrm{tr}(\rho)]$ is a quantitative degree of satisfaction (an expected truth value). This is the classical D'Hondt–Panangaden viewpoint used by many QHL systems.

General linear operators. CoqQ generalizes further and allows assertions to be arbitrary linear operators in $L(H_s)$, while the semantic validity of triples is still phrased using the trace pairing $\mathrm{tr}(A^{(s)}\rho)$. This has a pragmatic advantage: $L(H_s)$ is closed under linear combinations, so intermediate assertions do not require repeated side-condition proofs (e.g. that an expression remains an effect). For physical interpretation, however, one typically restricts attention to Hermitian operators, since $\mathrm{tr}(A^{(s)}\rho)$ is meaningful as an "observable expectation" only when A is Hermitian.

The appropriate entailment order between (Hermitian) assertions is the Löwner order: $A \sqsubseteq B$ iff $B - A \sqsupseteq 0$. It is compatible with the expectation semantics: if $A \sqsubseteq B$ and $\rho \sqsupseteq 0$, then $\mathrm{tr}(A^{(s)}\rho) \leq \mathrm{tr}(B^{(s)}\rho)$. This is the quantum analogue of semantic implication, and it underlies the rule of consequence in quantum Hoare-style systems.

Assertions attached to small subsystems let proofs follow the programmer's intent: most commands touch a limited footprint, and most specifications speak about a few registers (e.g.

"the output register x is in the Fourier basis"). CoqQ formalizes this principle via a *localization* property: every program has a local denotation on the tensor factor generated by its mentioned subsystems, and extends by identity elsewhere. This result is a key ingredient behind local/parallel reasoning rules such as invariance and framing.

8.1 CoqQ's program logic

CoqQ formalizes a Hoare-style program logic for qwhile, and proves it sound with respect to the denotational semantics $[\![C]\!]$ (Section 6). The logic is a mild adaptation of Ying's QHL family, but with (i) assertions allowed to be general linear operators, (ii) explicit support for local/parallel reasoning, and (iii) a new rule (R.Inner) to connect forward and backward reasoning.

Judgments and validity. A Hoare judgment has the form

$$\{\, A \,\}\ C\ \{\, B \,\},$$

where $A \in L(H_{s_1})$ and $B \in L(H_{s_2})$ are assertions on (possibly different) subsystems $s_1, s_2 \subseteq \mathsf{Reg}$, and C is a qwhile command.

We will distinguish *total correctness* and *partial correctness* validity by the following trace-expectation inequalities. Let $\rho \in \mathcal{D}_{\leq 1}(H)$ be any input partial state and let $[\![C]\!] : L(H) \to L(H)$ be the denotation.

Remark 8.1. We use two different turnstiles to separate *syntax* from *semantics*. The judgment

$$\vdash \{A\}\, C\, \{B\}$$

means that the Hoare triple is *derivable* in the proof system (there is a formal proof using the inference rules). In contrast,

$$\models \{A\}\, C\, \{B\}$$

means that the triple is *semantically valid* with respect to the denotational semantics $[\![C]\!]$ (Definitions 8.2 for total/partial correctness).

Some presentations sometimes write the rules directly with $\models$, because in a mechanized development each "rule" is proved as a lemma about the semantics (so it is already a sound semantic principle). Here we keep the traditional separation: rules are stated with $\vdash$, and soundness is expressed by the meta-theorem

$$\vdash_{p/t} \{A\}\, C\, \{B\} \implies \models_{p/t} \{A\}\, C\, \{B\}.$$

Definition 8.2. *Total correctness.* We write $\models_t \{A\}\, C\, \{B\}$ if for all $\rho \in \mathcal{D}_{\leq 1}(H)$,

$$\mathrm{tr}\big(A^{(s_1)}\,\rho\big) \;\leq\; \mathrm{tr}\big(B^{(s_2)}\,[\![C]\!](\rho)\big).$$

Partial correctness. We write $\models_p \{A\}\, C\, \{B\}$ if for all $\rho \in \mathcal{D}_{\leq 1}(H)$,

$$\mathrm{tr}\big(A^{(s_1)}\,\rho\big) \;\leq\; \mathrm{tr}\big(B^{(s_2)}\,[\![C]\!](\rho)\big) \;+\; \big(\mathrm{tr}(\rho) - \mathrm{tr}([\![C]\!](\rho))\big).$$

The extra term $\mathrm{tr}(\rho) - \mathrm{tr}([\![C]\!](\rho))$ is exactly the *missing probability mass* after running C in the subnormalized-state semantics. Recall that inputs are allowed to be partial states $\rho \in \mathcal{D}_{\leq 1}(H)$, where $\mathrm{tr}(\rho) \leq 1$ represents the total probability mass of being in the current configuration. A denotation $[\![C]\!]$ is trace-nonincreasing, so

$$\mathrm{tr}([\![C]\!](\rho)) \;\leq\; \mathrm{tr}(\rho).$$

The difference

$$\mathrm{tr}(\rho) - \mathrm{tr}([\![C]\!](\rho))$$

therefore measures how much probability mass is *lost* by executing C. In the intended operational reading, this lost mass corresponds to executions that do not produce a normal returned post-state. In particular, there are two standard sources of such loss:

Nontermination (divergence). For loops, some runs may iterate forever. In the limit-of-approximants semantics, the loop denotation collects only the mass of terminating runs. The remaining mass is precisely the probability of diverging, and it is reflected by the trace gap $\mathrm{tr}(\rho) - \mathrm{tr}(\llbracket C \rrbracket(\rho))$.

Aborting/blocked behavior. If a command is treated as *non-returning* in the super-operator model, it also contributes no returned post-state and therefore drops trace mass. In our development, abort is interpreted at the super-operator level as $\llbracket \mathsf{abort} \rrbracket(\rho) = 0$, so an execution that reaches abort loses all remaining mass from that point onward. More generally, any construct that is modelled as "no normal outcome" will be represented denotationally by mapping to 0, and its probability weight is counted into the trace gap.

This is why the partial-correctness validity condition adds the trace-gap term: partial correctness does not require that all probability mass reaches a normal final state. Instead, it allows the possibility that some mass is lost (due to divergence or aborting behavior), and only insists that *whenever* the program does return normally, the postcondition is satisfied in the quantitative sense encoded by $\mathrm{tr}(B^{(s_2)}\llbracket C \rrbracket(\rho))$. Concretely, the inequality

$$\mathrm{tr}\big(A^{(s_1)}\rho\big) \;\le\; \mathrm{tr}\big(B^{(s_2)}\llbracket C \rrbracket(\rho)\big) + \big(\mathrm{tr}(\rho) - \mathrm{tr}(\llbracket C \rrbracket(\rho))\big)$$

can be read as: "the amount of A-mass initially present is bounded by the amount of B-mass in the normal output, *plus* the amount of mass that fails to produce a normal output." When C is terminating with probability 1 on ρ (i.e. $\mathrm{tr}(\llbracket C \rrbracket(\rho)) = \mathrm{tr}(\rho)$), the trace-gap vanishes and partial correctness collapses to total correctness for that input.

Saturated judgments. CoqQ also uses a *saturated* variant (superscript s) in which the inequality in the validity definition is strengthened to equality. For instance, $\models_t^s \{A\}C\{B\}$ means $\mathrm{tr}(A^{(s_1)}\rho) = \mathrm{tr}(B^{(s_2)}\llbracket C \rrbracket(\rho))$ for all ρ. Saturated total correctness is especially useful when preconditions are scalars (probabilities) and postconditions are effects that represent measurement events, as in typical textbook statements of algorithm success probabilities.

Readable state-to-state judgments. A practical feature of the paper is syntactic sugar for state-based assertions expressed in labelled Dirac notation. Informally, the judgment

$$\models_{pt}^{st} \; \{\, |\psi\rangle\langle\psi| \,\}\; C \;\{\, |\phi\rangle\langle\phi| \,\}$$

can be read as "C transforms the input state $|\psi\rangle$ into the output state $|\phi\rangle$" (under appropriate normalization side conditions).

Concrete syntax The abstract qwhile syntax is convenient for meta-theory: commands act directly on named subsystems $s \subseteq \mathsf{Reg}$ via init ρ_s, apply U_s, and measurements M_s. However, when formalizing textbook algorithms, it is more convenient to write programs in terms of *typed quantum variables* (register names) and use a small set of high-level commands, notably initialization into computational basis states and computational-basis measurement as guards.

Types and their Hilbert spaces. The paper assigns each quantum variable a *type* T together with a Hilbert space H_T. The intended use is that T can represent qubits, qudits, classical finite domains, and products.
-If T is a finite set, then H_T is the Hilbert space with the elements of T as the computational basis:

$$H_T \;:=\; \mathrm{span}\{|t\rangle \;:\; t \in T\}, \qquad \langle t|t'\rangle = \delta_{t,t'}.$$

-If $T = T_1 \times T_2$ is a Cartesian product, then $H_T \cong H_{T_1} \otimes H_{T_2}$, with basis $|t_1, t_2\rangle \equiv |t_1\rangle \otimes |t_2\rangle$.

This closure under products is essential: it lets one package several registers into a single "compound" variable and then project back to components when needed.

Definition 8.3. A quantum variable x of type T consists of:

a symbolic name x;

a *location* (subsystem) $\mathrm{set}(x) \subseteq \mathsf{Reg}$;

a fixed unitary identification (encoding) between the abstract type space and the concrete subsystem space,

$$\iota_x : \; H_T \xrightarrow{\cong} H_{\mathrm{set}(x)}.$$

We write the image of a vector $|\psi\rangle \in H_T$ and an operator $A \in L(H_T)$ at location x as

$$|\psi\rangle_x \;:=\; \iota_x\,|\psi\rangle \in H_{\mathrm{set}(x)}, \qquad A[x] \;:=\; \iota_x\,A\,\iota_x^\dagger \in L(H_{\mathrm{set}(x)}).$$

Thus $A[x]$ is simply "the operator A acting on the subsystem $\mathrm{set}(x)$, expressed in the concrete Hilbert space of that subsystem".

Composing variables. If $x_1 : T_1$ and $x_2 : T_2$ are quantum variables with disjoint locations $\mathrm{set}(x_1) \cap \mathrm{set}(x_2) = \emptyset$, then we treat the pair $x \;\equiv\; [x_1, x_2]$ as a quantum variable of type $T_1 \times T_2$ with location $\mathrm{set}(x) = \mathrm{set}(x_1) \cup \mathrm{set}(x_2)$, and with injections compatible with tensor products:

$$|\psi_1, \psi_2\rangle_{[x_1,x_2]} \;=\; |\psi_1\rangle_{x_1} \otimes |\psi_2\rangle_{x_2}\,, \qquad (A_1 \otimes A_2)[[x_1, x_2]] \;=\; A_1[x_1] \otimes A_2[x_2].$$

This lets one define and manipulate "structured" registers while retaining locality at the level of subsystems.

For a command C, we write $\mathrm{set}(C)$ for the union of all subsystems mentioned in C (i.e. the program's footprint). This notion is used pervasively in proof-rule side conditions such as invariance and framing.

Concrete command syntax Under the concrete syntax, the core constructs are written using typed variables and computational basis measurement as guards. A representative fragment is:

$$\begin{aligned} C \;::=\;& \mathsf{skip} \;\mid\; C_1; C_2 \;\mid\; x := |t\rangle \;\mid\; x := U[x] \\ &\mid\; \mathsf{if}\ \mathsf{meas}[x] = t \to C_t\ \mathsf{fi} \;\mid\; \mathsf{while}\ \mathsf{meas}[x] = b\ \mathsf{do}\ C\ \mathsf{od} \\ &\mid\; \mathsf{for}\ i\ \mathsf{do}\ C_i \quad \text{(parallel/independent family, used in the logic)} \end{aligned}$$

where t ranges over the (finite) type of x, U is a unitary on H_T, and $\mathsf{meas}[x]$ denotes *computational basis* measurement on the variable x.

Computational-basis measurement. If $x : T$ with T finite, then $\mathsf{meas}[x]$ is the projective measurement $\{\,|t\rangle\langle t|\,\}_{t\in T}$ on H_T. At location x, its outcome operators on the concrete subsystem are

$$M_t[x] \;:=\; \big(|t\rangle\langle t|\big)[x] \;\in\; L\big(H_{\mathrm{set}(x)}\big), \qquad t \in T.$$

When interpreting these on the *global* space H, we further apply cylindrical extension:

$$M_t^{(\mathrm{set}(x))} \;=\; M_t[x] \otimes \mathbf{1}_{H_{\overline{\mathrm{set}(x)}}}.$$

Thus, the concrete guard $\mathsf{meas}[x] = t$ is a special case of the abstract measurement $M_s = \{(t, M_t)\}$ used, where $s = \mathrm{set}(x)$ and $M_t = M_t[x]$.

Each concrete construct is an instance of the abstract subsystem-based qwhile constructs from (and therefore inherits their operational and denotational semantics). Concretely:

Initialization. The concrete command $x := |t\rangle$ prepares the basis state at x and forgets any previous contents of $\mathrm{set}(x)$. In the abstract syntax, this is

$$x := |t\rangle \quad \rightsquigarrow \quad \mathsf{init}\ \rho_{\mathrm{set}(x)} \qquad \text{with} \qquad \rho_{\mathrm{set}(x)} := |t\rangle_x\langle t|\,.$$

Its denotation is $\rho \mapsto \rho_{\mathrm{set}(x)} \otimes \mathrm{tr}_{\mathrm{set}(x)}(\rho)$.

Unitary application. The concrete command $x := U[x]$ applies the unitary U on the type space H_T, transported to the location $\mathrm{set}(x)$ as $U[x] = \iota_x U \iota_x^\dagger$:

$$x := U[x] \quad \rightsquigarrow \quad \mathsf{apply}\ U_{\mathrm{set}(x)} \qquad \text{with} \qquad U_{\mathrm{set}(x)} := U[x].$$

Globally this acts by conjugation with $(U[x])^{(\mathrm{set}(x))}$.

Conditionals. The concrete conditional

$$\mathsf{if}\ \mathsf{meas}[x] = t \to C_t\ \mathsf{fi}$$

is the abstract conditional

$$\mathsf{if}\ (\square t.\ M_{\mathrm{set}(x)} = t \to C_t)\ \mathsf{fi}$$

with Kraus/projector operators $M_t[x] = (|t\rangle\langle t|)[x]$ (and their cylindrical extensions on H).

While loops. The concrete loop

$$\mathsf{while}\ \mathsf{meas}[x] = b\ \mathsf{do}\ C\ \mathsf{od}$$

is the abstract while-loop driven by the two-outcome measurement $M'_{\mathrm{set}(x)} = \{M_{\neg b}[x], M_b[x]\}$ on subsystem $\mathrm{set}(x)$, where $M_b[x] = (|b\rangle\langle b|)[x]$ and $M_{\neg b}[x] = (|\neg b\rangle\langle \neg b|)[x]$.

For loops (independent families). The construct $\mathsf{for}\ i\ \mathsf{do}\ C_i$ is used in the logic as a convenient way to express a family of commands acting on pairwise disjoint footprints. Under the disjointness condition

$$\forall i \neq j,\ \mathrm{set}(C_i) \cap \mathrm{set}(C_j) = \emptyset,$$

their denotations commute and can be composed in any order; correctness rules therefore allow one to reason about them componentwise with tensor-product assertions (cf. rule R.PC.P).

Proof rules.

$$\frac{}{\vdash_p \{A\}\ \mathsf{skip}\ \{A\}}\ (\textsc{Ax.Sk}) \qquad \frac{}{\vdash_p \{A\}\ x := U[x]\ \{\, U[x]\ A\ U[x]^\dagger \,\}}\ (\textsc{Ax.UTF})$$

$$\frac{s \cap \mathrm{set}(x) = \emptyset}{\vdash_p \{A_s\}\ x := |t\rangle\ \{\, A_s \otimes |t\rangle_x\langle t| \,\}}\ (\textsc{Ax.InF})$$

$$\frac{\vdash_p \{A\}\ C_1\ \{B\} \quad \vdash_p \{B\}\ C_2\ \{C\}}{\vdash_p \{A\}\ C_1; C_2\ \{C\}}\ (\mathrm{R.SC})$$

$$\frac{\forall t : T,\ \vdash_p \{A_t\}\ C_t\ \{B\}}{\vdash_p \left\{\ \sum_{t:T} |t\rangle_x\langle t|\ A_t\ |t\rangle_x\langle t|\ \right\}\ \mathsf{if}\ \mathsf{meas}[x] = t \to C_t\ \mathsf{fi}\ \{B\}}\ (\mathrm{R.IF})$$

$$\frac{R := |b\rangle_x\langle b|\ A\ |b\rangle_x\langle b| + |\neg b\rangle_x\langle \neg b|\ B\ |\neg b\rangle_x\langle \neg b| \quad \vdash_p \{A\}\ C\ \{R\} \quad A \sqsubseteq \mathbf{1} \quad B \sqsubseteq \mathbf{1}}{\vdash_p \{R\}\ \mathsf{while}\ \mathsf{meas}[x] = b\ \mathsf{do}\ C\ \mathsf{od}\ \{B\}}\ (\mathrm{R.LP.P})$$

$$\frac{\forall i \in I.\ \vdash_p \{A_{i,s_{A_i}}\}\ C_i\ \{B_{i,s_{B_i}}\}\ \wedge\ (0 \sqsubseteq A_{i,s_{A_i}} \sqsubseteq \mathbf{1})\ \wedge\ (0 \sqsubseteq B_{i,s_{B_i}} \sqsubseteq \mathbf{1})\ \wedge\ \forall i \neq j.\ \mathcal{F}_i \cap \mathcal{F}_j = \emptyset}{\vdash_p \left\{\ \bigotimes_{i\in I} A_{i,s_{A_i}}\ \right\}\ \mathsf{for}\ i\ \mathsf{do}\ C_i\ \left\{\ \bigotimes_{i\in I} B_{i,s_{B_i}}\ \right\}}\ (\text{R.PC.P})$$

$$\text{where} \qquad \mathcal{F}_i := \mathrm{set}(C_i) \cup s_{A_i} \cup s_{B_i}.$$

$$\frac{\forall i \neq j,\ \mathrm{set}(x_i) \cap \mathrm{set}(x_j) = \emptyset}{\vdash_p \{A\}\ \mathsf{for}\ i\ \mathsf{do}\ x_i := U_i[x_i]\ \left\{\ (\bigotimes_{i\in I} U_i[x_i])\ A\ (\bigotimes_{i\in I} U_i[x_i])^\dagger\ \right\}}\ (\text{Ax.UTFP})$$

$$\frac{\forall i \neq j,\ \mathrm{set}(x_i) \cap \mathrm{set}(x_j) = \emptyset}{\vdash_p \{1\}\ \mathsf{for}\ i\ \mathsf{do}\ x_i := |t_i\rangle\ \left\{\ \bigotimes_{i\in I} |t_i\rangle_{x_i}\langle t_i|\ \right\}}\ (\text{Ax.InFP})$$

$$\frac{A \sqsubseteq A' \quad \vdash_p \{A'\}\ C\ \{B'\} \quad B' \sqsubseteq B}{\vdash_p \{A\}\ C\ \{B\}}\ (\text{R.Or})$$

$$\frac{\forall i \in I,\ \vdash_p \{A_i\}\ C\ \{B_i\} \quad \forall i \in I,\ 0 \le \lambda_i \quad \sum_{i\in I} \lambda_i \le 1}{\vdash_p \left\{\ \sum_{i\in I} \lambda_i A_i\ \right\}\ C\ \left\{\ \sum_{i\in I} \lambda_i B_i\ \right\}}\ (\text{R.CC.P})$$

$$\frac{A_s \sqsubseteq \mathbf{1} \quad s \cap \mathrm{set}(C) = \emptyset}{\vdash_p \{A_s\}\ C\ \{A_s\}}\ (\text{Ax.Inv})$$

$$\frac{\vdash_p \{A_{s_A} \otimes \mathbf{1}_s\}\ C\ \{B\}}{\vdash_p \{A_{s_A}\}\ C\ \{B\} \quad s_A \cap s = \emptyset}\ (\text{R.EI})$$

$$\frac{\vdash_p \{A_{s_A}\}\ C\ \{B_{s_B}\} \quad 0 \sqsubseteq R_s \sqsubseteq \mathbf{1} \quad (\mathrm{set}(C) \cup s_A \cup s_B) \cap s = \emptyset}{\vdash_p \{A_{s_A} \otimes R_s\}\ C\ \{B_{s_B} \otimes R_s\}}\ (\text{Frame.P})$$

$$\frac{\vdash^s_t \{1\}\ C\ \{|v\rangle_{s_2}\langle v|\} \quad \| |v\rangle_{s_2} \| \le 1 \quad s_1 \subseteq s_2}{\vdash^s_t \left\{\ \left\|\mathbf{1}_{s_1}(\langle u|_{s_1} |v\rangle_{s_2})\right\|^2\ \right\}\ C\ \left\{\ |u\rangle_{s_1}\langle u|\ \right\}}\ (\text{R.Inner})$$

$$\frac{}{\vdash_p \{\ |\psi\rangle\langle\psi|\ \}\ x := U[x]\ \{\ |U[x]\psi\rangle\langle U[x]\psi|\ \}}\ (\text{Ax.UTF'})$$

$$\frac{s \cap \mathrm{set}(x) = \emptyset}{\vdash_p \{\ |\psi\rangle\langle\psi|\ \}\ x := |t\rangle\ \{\ (|\psi\rangle \otimes |t\rangle_x)(\langle\psi| \otimes \langle t|_x)\ \}}\ (\text{Ax.InF'})$$

8.2 Backward and forward reasoning

In classical Hoare logic, backward reasoning is guided by weakest preconditions. The quantum analogue is a *predicate-transformer* view induced by the denotational semantics. Given a post-assertion B (typically an effect), one can define a weakest (liberal) precondition $\mathrm{wlp}(C, B)$ such that

$$\models_t \{A\}\, C\, \{B\} \quad \Longleftrightarrow \quad A \ \sqsubseteq\ \mathrm{wlp}(C, B)$$

(and similarly for partial correctness with an appropriate liberal variant). At the semantic level, this transformer is naturally expressed via the Heisenberg dual $[\![C]\!]^*$ characterized by

$$\mathrm{tr}\big(B^{(s_2)}\, [\![C]\!](\rho)\big) \ =\ \mathrm{tr}\big(\big([\![C]\!]^*(B^{(s_2)})\big)\, \rho\big) \qquad (\rho \sqsupseteq 0),$$

so that reasoning "pushes back" postconditions through the program.

Forward reasoning and state transformation. Forward reasoning propagates information in the execution order: start from a known input configuration (often after a block of initializations), compute the resulting state/expectation after each command, and finally read off the desired postcondition.

Linking forward and backward: (R.Inner). Forward reasoning often produces a strong *post-state description* (a concrete $|\psi\rangle$ or a structured superposition), whereas the final correctness goal is phrased as an *expectation* of a postcondition effect (e.g. $|x\rangle\langle x|$ meaning "measure x"). The new rule (R.Inner) in CoqQ is designed to bridge this gap: it turns a forward-derived saturated judgment about reaching a specific state into a quantitative precondition for a given postcondition, via inner-product/overlap calculations.

8.3 Soundness w.r.t. denotational semantics

Theorem 8.4 (Soundness)**.** *Every derivable Hoare judgment in the system is valid with respect to the denotational semantics $[\![\cdot]\!]$ of Section 6 and the validity definitions in Definition 8.2. Concretely, for each choice of proof mode (total vs. partial correctness, and saturated vs. non-saturated),*

$$\vdash \{A\}\, C\, \{B\} \quad \Longrightarrow \quad \models \{A\}\, C\, \{B\}.$$

Proof. We only prove the soundness for several main rules.
Ax.Sk. Fix any input partial state $\rho \in \mathcal{D}_{\leq 1}(H)$. By the denotational semantics of skip, $[\![\mathsf{skip}]\!](\rho) = \rho$. Therefore $\mathrm{tr}\big(A\,\rho\big) = \mathrm{tr}\big(A\,[\![\mathsf{skip}]\!](\rho)\big)$. Moreover $\mathrm{tr}(\rho) - \mathrm{tr}([\![\mathsf{skip}]\!](\rho)) = \mathrm{tr}(\rho) - \mathrm{tr}(\rho) = 0$. Hence

$$\mathrm{tr}(A\rho) \;\leq\; \mathrm{tr}\big(A\,[\![\mathsf{skip}]\!](\rho)\big) + \big(\mathrm{tr}(\rho) - \mathrm{tr}([\![\mathsf{skip}]\!](\rho))\big),$$

which is exactly $\models_p \{A\}\ \mathsf{skip}\ \{A\}$ by Definition 8.2.
Ax.UTF Fix $\rho \in \mathcal{D}_{\leq 1}(H)$. Let U abbreviate the (globally lifted) unitary acting on $\mathrm{set}(x)$, i.e. $U := U[x]^{(\mathrm{set}(x))}$. By denotational semantics of unitary application,

$$[\![x := U[x]]\!](\rho) \;=\; U\,\rho\,U^\dagger.$$

Now evaluate the postcondition on the output:

$$\begin{aligned}
\mathrm{tr}\Big(\big(U[x]\,A\,U[x]^\dagger\big)\,[\![x := U[x]]\!](\rho)\Big) &= \mathrm{tr}\Big(\big(UAU^\dagger\big)\,\big(U\rho U^\dagger\big)\Big) \\
&= \mathrm{tr}\Big(UA(U^\dagger U)\rho U^\dagger\Big) \\
&= \mathrm{tr}\Big(UA\rho U^\dagger\Big) \\
&= \mathrm{tr}\big(A\rho\big),
\end{aligned}$$

where we used associativity and $U^\dagger U = \mathbf{1}$, then cyclicity of trace $\mathrm{tr}(UA\rho U^\dagger) = \mathrm{tr}(A\rho U^\dagger U) = \mathrm{tr}(A\rho)$. Also $\mathrm{tr}([\![x := U[x]]\!](\rho)) = \mathrm{tr}(U\rho U^\dagger) = \mathrm{tr}(\rho)$, so the trace-gap term is 0. Therefore the partial-correctness inequality holds with equality:

$$\begin{aligned}
\mathrm{tr}(A\rho) &= \mathrm{tr}\Big(\big(U[x]AU[x]^\dagger\big)\,[\![x := U[x]]\!](\rho)\Big) \\
&\leq \mathrm{tr}\Big(\big(U[x]AU[x]^\dagger\big)\,[\![x := U[x]]\!](\rho)\Big) + \big(\mathrm{tr}(\rho) - \mathrm{tr}([\![x := U[x]]\!](\rho))\big).
\end{aligned}$$

Hence $\models_p \{A\}\ x := U[x]\ \{U[x]AU[x]^\dagger\}$.
Ax.InF Fix $\rho \in \mathcal{D}_{\leq 1}(H)$. Let $x^\star := \mathrm{set}(x)$ and write the global space as $H \cong H_{x^\star} \otimes H_{\overline{x^\star}}$. The denotation of initialization $x := |t\rangle$ is the reset map:

$$[\![x := |t\rangle]\!](\rho) = |t\rangle_x\langle t| \;\otimes\; \mathrm{tr}_{x^\star}(\rho).$$

This map is trace-preserving: $\mathrm{tr}(\llbracket x := |t\rangle \rrbracket(\rho)) = \mathrm{tr}(\rho)$, hence the trace-gap is 0. So it suffices to show the stronger inequality $\mathrm{tr}(A_s\rho) \le \mathrm{tr}\big((A_s \otimes |t\rangle_x\langle t|)\llbracket x := |t\rangle \rrbracket(\rho)\big)$, indeed we will show equality.

Because $s \cap x^\star = \emptyset$, the assertion A_s acts only on the complement factor $H_{\overline{x^\star}}$ (possibly together with other registers disjoint from $x^\star$). Using the defining property of partial trace ("testers commute with trace-out") we have:

$$\mathrm{tr}\big(A_s\,\rho\big) = \mathrm{tr}\big(A_s\,\mathrm{tr}_{x^\star}(\rho)\big). \tag{1}$$

(Precisely: $\mathrm{tr}((A_s \otimes \mathbf{1}_{H_{x^\star}})\rho) = \mathrm{tr}(A_s\,\mathrm{tr}_{x^\star}(\rho))$.)

Now evaluate the postcondition on the output:

$$\begin{aligned}
\mathrm{tr}\Big((A_s \otimes |t\rangle_x\langle t|)\ \llbracket x := |t\rangle \rrbracket(\rho)\Big) &= \mathrm{tr}\Big((A_s \otimes |t\rangle_x\langle t|)\ (|t\rangle_x\langle t| \otimes \mathrm{tr}_{x^\star}(\rho))\Big) \\
&= \mathrm{tr}\Big((A_s \otimes |t\rangle_x\langle t|)\ (|t\rangle_x\langle t| \otimes \mathrm{tr}_{x^\star}(\rho))\Big) \\
&= \mathrm{tr}\Big(A_s\,\mathrm{tr}_{x^\star}(\rho)\Big) \cdot \mathrm{tr}\big(|t\rangle\langle t| \cdot |t\rangle\langle t|\big) \\
&= \mathrm{tr}\Big(A_s\,\mathrm{tr}_{x^\star}(\rho)\Big) \cdot \mathrm{tr}\big(|t\rangle\langle t|\big) \\
&= \mathrm{tr}\Big(A_s\,\mathrm{tr}_{x^\star}(\rho)\Big).
\end{aligned}$$

Here we used that tensor traces factorize and $|t\rangle\langle t|$ is a rank-one projection: $(|t\rangle\langle t|)^2 = |t\rangle\langle t|$ and $\mathrm{tr}(|t\rangle\langle t|) = 1$.

Combining with (1) gives

$$\mathrm{tr}(A_s\rho) = \mathrm{tr}\Big((A_s \otimes |t\rangle_x\langle t|)\ \llbracket x := |t\rangle \rrbracket(\rho)\Big),$$

and since the trace gap is 0, the partial-correctness inequality holds. Hence the rule is sound.

R.SC We will prove if $\models_p \{A\}\ C_1\ \{B\}$ and $\models_p \{B\}\ C_2\ \{C\}$, then $\models_p \{A\}\ C_1; C_2\ \{C\}$.

Fix $\rho \in \mathcal{D}_{\le 1}(H)$ and write $\rho_1 := \llbracket C_1 \rrbracket(\rho)$. By $\models_p \{A\}C_1\{B\}$,

$$\mathrm{tr}(A\rho) \ \le\ \mathrm{tr}(B\rho_1) + \big(\mathrm{tr}(\rho) - \mathrm{tr}(\rho_1)\big). \tag{1}$$

By $\models_p \{B\}C_2\{C\}$ applied to ρ_1,

$$\mathrm{tr}(B\rho_1) \ \le\ \mathrm{tr}\big(C\,\llbracket C_2 \rrbracket(\rho_1)\big) + \big(\mathrm{tr}(\rho_1) - \mathrm{tr}(\llbracket C_2 \rrbracket(\rho_1))\big). \tag{2}$$

Substitute (2) into (1):

$$\begin{aligned}
\mathrm{tr}(A\rho) &\le \mathrm{tr}\big(C\,\llbracket C_2 \rrbracket(\rho_1)\big) + \big(\mathrm{tr}(\rho_1) - \mathrm{tr}(\llbracket C_2 \rrbracket(\rho_1))\big) + \big(\mathrm{tr}(\rho) - \mathrm{tr}(\rho_1)\big) \\
&= \mathrm{tr}\big(C\,\llbracket C_2 \rrbracket(\rho_1)\big) + \big(\mathrm{tr}(\rho) - \mathrm{tr}(\llbracket C_2 \rrbracket(\rho_1))\big).
\end{aligned}$$

Finally, by denotational semantics of sequencing,

$$\llbracket C_1; C_2 \rrbracket = \llbracket C_2 \rrbracket \circ \llbracket C_1 \rrbracket \quad \Rightarrow \quad \llbracket C_1; C_2 \rrbracket(\rho) = \llbracket C_2 \rrbracket(\rho_1).$$

So the inequality becomes exactly

$$\mathrm{tr}(A\rho) \ \le\ \mathrm{tr}\big(C\,\llbracket C_1; C_2 \rrbracket(\rho)\big) + \big(\mathrm{tr}(\rho) - \mathrm{tr}(\llbracket C_1; C_2 \rrbracket(\rho))\big),$$

i.e. $\models_p \{A\}\ C_1; C_2\ \{C\}$.

R.IF Assume that for every $t \in T$, $\models_p \{A_t\}\ C_t\ \{B\}$. Let $P_t := |t\rangle_x\langle t|$ be the computational-basis projector on x (lifted to H).

Fix $\rho \in \mathcal{D}_{\leq 1}(H)$. For each $t \in T$, define the (subnormalized) post-measurement state

$$\rho_t := P_t \, \rho \, P_t \sqsupseteq 0.$$

(So $\mathrm{tr}(\rho_t)$ is the probability mass of outcome t.)
Using linearity and cyclicity of trace,

$$\begin{aligned} \mathrm{tr}\Big(\Big(\sum_{t\in T} P_t A_t P_t\Big)\rho\Big) &= \sum_{t\in T} \mathrm{tr}(P_t A_t P_t \rho) \\ &= \sum_{t\in T} \mathrm{tr}(A_t P_t \rho P_t) \\ &= \sum_{t\in T} \mathrm{tr}(A_t \rho_t). \end{aligned} \tag{1}$$

For each t, apply $\models_p \{A_t\} C_t \{B\}$ to the input state ρ_t:

$$\mathrm{tr}(A_t \rho_t) \;\leq\; \mathrm{tr}\big(B\,[\![C_t]\!](\rho_t)\big) + \big(\mathrm{tr}(\rho_t) - \mathrm{tr}([\![C_t]\!](\rho_t))\big). \tag{2}$$

Summing (2) over $t \in T$ and using (1),

$$\mathrm{tr}\Big(\Big(\sum_t P_t A_t P_t\Big)\rho\Big) \leq \sum_t \mathrm{tr}\big(B\,[\![C_t]\!](\rho_t)\big) + \sum_t \mathrm{tr}(\rho_t) - \sum_t \mathrm{tr}([\![C_t]\!](\rho_t)). \tag{3}$$

By the denotational semantics of conditionals (measurement-controlled branching),

$$[\![\mathsf{if}\ \mathsf{meas}[x] = t \to C_t\ \mathsf{fi}]\!](\rho) = \sum_{t\in T} [\![C_t]\!](\rho_t). \tag{4}$$

Therefore, by linearity of trace,

$$\sum_t \mathrm{tr}\big(B\,[\![C_t]\!](\rho_t)\big) = \mathrm{tr}\Big(B \sum_t [\![C_t]\!](\rho_t)\Big) = \mathrm{tr}\Big(B\,[\![\mathsf{if}\cdots\mathsf{fi}]\!](\rho)\Big). \tag{5}$$

Also,

$$\sum_t \mathrm{tr}([\![C_t]\!](\rho_t)) = \mathrm{tr}\Big(\sum_t [\![C_t]\!](\rho_t)\Big) = \mathrm{tr}\Big([\![\mathsf{if}\cdots\mathsf{fi}]\!](\rho)\Big). \tag{6}$$

Finally, because $\{P_t\}_{t\in T}$ is a projective measurement with $\sum_t P_t = \mathbf{1}$,

$$\sum_t \mathrm{tr}(\rho_t) = \sum_t \mathrm{tr}(P_t \rho P_t) = \mathrm{tr}\Big(\Big(\sum_t P_t\Big)\rho\Big) = \mathrm{tr}(\rho). \tag{7}$$

Substituting (5)–(7) into (3) yields

$$\mathrm{tr}\Big(\Big(\sum_t P_t A_t P_t\Big)\rho\Big) \;\leq\; \mathrm{tr}\Big(B\,[\![\mathsf{if}\cdots\mathsf{fi}]\!](\rho)\Big) + \big(\mathrm{tr}(\rho) - \mathrm{tr}([\![\mathsf{if}\cdots\mathsf{fi}]\!](\rho))\big),$$

which is exactly $\models_p$ for the conditional.
R.OR Fix $\rho \in \mathcal{D}_{\leq 1}(H)$. Since $A \sqsubseteq A'$ and $\rho \sqsupseteq 0$, Lemma 4.10 gives

$$\mathrm{tr}(A\rho) \;\leq\; \mathrm{tr}(A'\rho). \tag{1}$$

By $\models_p \{A'\} C \{B'\}$,

$$\mathrm{tr}(A'\rho) \;\leq\; \mathrm{tr}\big(B'\,[\![C]\!](\rho)\big) + \big(\mathrm{tr}(\rho) - \mathrm{tr}([\![C]\!](\rho))\big). \tag{2}$$

Finally, since $B' \sqsubseteq B$ and $[\![C]\!](\rho) \sqsupseteq 0$, again by Lemma 4.10,

$$\mathrm{tr}\big(B'\,[\![C]\!](\rho)\big) \;\le\; \mathrm{tr}\big(B\,[\![C]\!](\rho)\big). \tag{3}$$

Combining (1)–(3) yields

$$\mathrm{tr}(A\rho) \;\le\; \mathrm{tr}\big(B\,[\![C]\!](\rho)\big) + \big(\mathrm{tr}(\rho) - \mathrm{tr}([\![C]\!](\rho))\big),$$

which is exactly $\models_p \{A\}C\{B\}$.

R.LP.P Let $W \equiv \mathsf{while}\ \mathsf{meas}[x] = b\ \mathsf{do}\ C\ \mathsf{od}$. Write $P_b := |b\rangle_x\langle b|$ and $P_{\neg b} := |\neg b\rangle_x\langle \neg b|$ (lifted to H). Assume $A \sqsubseteq \mathbf{1}$, $B \sqsubseteq \mathbf{1}$, and define

$$R \;:=\; P_b A P_b \;+\; P_{\neg b} B P_{\neg b}.$$

Fix an arbitrary $\rho \in \mathcal{D}_{\le 1}(H)$. Let

$$\rho_b := P_b \rho P_b, \qquad \rho_{\neg b} := P_{\neg b} \rho P_{\neg b}.$$

Then $\rho_b, \rho_{\neg b} \sqsupseteq 0$ and $\mathrm{tr}(\rho) = \mathrm{tr}(\rho_b) + \mathrm{tr}(\rho_{\neg b})$ because $P_b + P_{\neg b} = \mathbf{1}$.

We will use the denotational while semantics via syntactic approximants $W^{(n)}$ (Section 6):

$$W^{(0)} := \mathsf{abort}, \qquad W^{(n+1)} := \mathsf{if}\ \big(\mathsf{meas}[x] = \neg b \to \mathsf{skip} \;\square\; b \to C; W^{(n)}\big)\ \mathsf{fi},$$

and

$$[\![W]\!](\rho) = \bigsqcup_{n \ge 0} [\![W^{(n)}]\!](\rho) = \lim_{n\to\infty} [\![W^{(n)}]\!](\rho).$$

(Here $[\![\mathsf{abort}]\!](\rho) = 0$.)

Step 1: Since $A \sqsubseteq \mathbf{1}$ and $B \sqsubseteq \mathbf{1}$,

$$P_b A P_b \;\sqsubseteq\; P_b \mathbf{1} P_b = P_b, \qquad P_{\neg b} B P_{\neg b} \;\sqsubseteq\; P_{\neg b}.$$

Adding gives

$$R = P_b A P_b + P_{\neg b} B P_{\neg b} \;\sqsubseteq\; P_b + P_{\neg b} = \mathbf{1}. \tag{1}$$

Therefore for every $\sigma \sqsupseteq 0$, Lemma 4.10 yields

$$\mathrm{tr}(R\sigma) \;\le\; \mathrm{tr}(\sigma). \tag{2}$$

Step 2: Let $E_n := [\![W^{(n)}]\!]$. We claim that for every $n \ge 0$ and every $\rho \sqsupseteq 0$,

$$\mathrm{tr}(R\rho) \;\le\; \mathrm{tr}\big(B\,E_n(\rho)\big) \;+\; \big(\mathrm{tr}(\rho) - \mathrm{tr}(E_n(\rho))\big). \tag{$\star_n$}$$

We prove $(\star_n)$ by induction on n.

Base $n = 0$. $E_0(\rho) = [\![\mathsf{abort}]\!](\rho) = 0$. Hence the RHS of $(\star_0)$ is $\mathrm{tr}(B \cdot 0) + (\mathrm{tr}(\rho) - \mathrm{tr}(0)) = \mathrm{tr}(\rho)$. So $(\star_0)$ reduces to $\mathrm{tr}(R\rho) \le \mathrm{tr}(\rho)$, which holds by (2).

Step $n \to n+1$. Assume $(\star_n)$ holds for all inputs. We must prove $(\star_{n+1})$.

First, unfold $E_{n+1}(\rho)$ using the denotation of if and sequencing: the guard measurement has two outcomes $\neg b$ (terminate) and b (continue), so

$$E_{n+1}(\rho) = \rho_{\neg b} \;+\; E_n\big([\![C]\!](\rho_b)\big), \tag{3}$$

and therefore

$$\mathrm{tr}(E_{n+1}(\rho)) = \mathrm{tr}(\rho_{\neg b}) \;+\; \mathrm{tr}\Big(E_n\big([\![C]\!](\rho_b)\big)\Big). \tag{4}$$

Also, expand the precondition trace using the definition of R:

$$\begin{aligned}\operatorname{tr}(R\rho) &= \operatorname{tr}(P_b A P_b \rho) + \operatorname{tr}(P_{\neg b} B P_{\neg b} \rho) \\ &= \operatorname{tr}(A\rho_b) + \operatorname{tr}(B\rho_{\neg b}),\end{aligned} \tag{5}$$

where we used cyclicity and $P_b^2 = P_b$, $P_{\neg b}^2 = P_{\neg b}$.

Now we establish $(\star_{n+1})$ by bounding the $A\rho_b$ term and letting the $B\rho_{\neg b}$ term cancel on both sides.

(i) Use the premise $\models_p \{A\}C\{R\}$ on input ρ_b. Since $\rho_b \sqsupseteq 0$, the premise gives:

$$\operatorname{tr}(A\rho_b) \;\le\; \operatorname{tr}\big(R\,[\![C]\!](\rho_b)\big) \;+\; \big(\operatorname{tr}(\rho_b) - \operatorname{tr}([\![C]\!](\rho_b))\big). \tag{6}$$

(ii) Use the induction hypothesis $(\star_n)$ on input $[\![C]\!](\rho_b)$. Apply $(\star_n)$ with $\sigma := [\![C]\!](\rho_b) \sqsupseteq 0$:

$$\operatorname{tr}\big(R\,[\![C]\!](\rho_b)\big) \;\le\; \operatorname{tr}\Big(B\,E_n([\![C]\!](\rho_b))\Big) + \Big(\operatorname{tr}([\![C]\!](\rho_b)) - \operatorname{tr}(E_n([\![C]\!](\rho_b)))\Big). \tag{7}$$

(iii) Combine (6) and (7). Substitute (7) into (6); the $\operatorname{tr}([\![C]\!](\rho_b))$ terms cancel:

$$\operatorname{tr}(A\rho_b) \;\le\; \operatorname{tr}\Big(B\,E_n([\![C]\!](\rho_b))\Big) + \Big(\operatorname{tr}(\rho_b) - \operatorname{tr}(E_n([\![C]\!](\rho_b)))\Big). \tag{8}$$

(iv) Add $\operatorname{tr}(B\rho_{\neg b})$ to both sides and rewrite using (3)–(5). Using (5) on the left and linearity of trace on the right,

$$\operatorname{tr}(R\rho) = \operatorname{tr}(A\rho_b) + \operatorname{tr}(B\rho_{\neg b}) \;\le\; \operatorname{tr}(B\rho_{\neg b}) + \operatorname{tr}\Big(B\,E_n([\![C]\!](\rho_b))\Big) + \operatorname{tr}(\rho_b) - \operatorname{tr}(E_n([\![C]\!](\rho_b))).$$

By (3),

$$\operatorname{tr}(B\rho_{\neg b}) + \operatorname{tr}\Big(B\,E_n([\![C]\!](\rho_b))\Big) = \operatorname{tr}\Big(B\,E_{n+1}(\rho)\Big). \tag{9}$$

And by (4),

$$\operatorname{tr}(\rho_b) - \operatorname{tr}(E_n([\![C]\!](\rho_b))) = \operatorname{tr}(\rho) - \operatorname{tr}(E_{n+1}(\rho)). \tag{10}$$

Substituting (9)–(10) proves $(\star_{n+1})$. Thus $(\star_n)$ holds for all $n \ge 0$.

Step 3: By definition, $[\![W]\!](\rho) = \lim_{n\to\infty} E_n(\rho)$ (in finite dimension). Taking $n \to \infty$ in $(\star_n)$ and using continuity of trace (and linearity of $\operatorname{tr}(B\,\cdot)$), we obtain:

$$\operatorname{tr}(R\rho) \;\le\; \operatorname{tr}\big(B\,[\![W]\!](\rho)\big) \;+\; \big(\operatorname{tr}(\rho) - \operatorname{tr}([\![W]\!](\rho))\big).$$

This is exactly $\models_p \{R\}\ W\ \{B\}$ by Definition 8.2.

□