# 9 Quantum Incorrectness Reasoning

In this section we present Quantum Incorrectness Logic (QIL), an under-approximate program logic for static bug-catching in quantum programs. The key technical move is to replace (boolean) satisfaction of a quantum assertion by an under-approximation relation, and to interpret "achieving a postcondition" in terms of the mixture of reachable terminal states, rather than individual execution paths.

Throughout, assertions are *projective* quantum predicates (orthogonal projections on the fixed global Hilbert space $H$), as developed in Section 4.10 of the prerequisites chapter. We continue to use the qwhile language and the collecting/relational semantics from Section 7 (multiset denotation $[\![C]\!]_\epsilon$ and exit-conditioned mixture $\mathrm{Mix}_\epsilon(C, \rho)$).

A first idea for bug-catching using a correctness logic is to prove the negation of a Hoare triple:

$$\neg\big(\models \{P\}\, C\, \{Q\}\big),$$

hoping this would imply a "bug postcondition" like $\{P\}\, C\, \{\neg Q\}$. In general, even in the classical setting this implication fails, but the quantum situation is strictly worse because of the interaction between (i) mixtures and (ii) projection predicates.

If we use projection satisfaction (support inclusion) to specify erroneous outcomes, then mixtures can force false positives. Concretely, let $P_c = |0\rangle\langle 0|$ be the intended "correct" predicate and consider the mixed (erroneous) state

$$\rho_e = \tfrac{1}{2}\big(|0\rangle\langle 0| + |1\rangle\langle 1|\big),$$

which contains both a good component $|0\rangle\langle 0|$ and a bad component $|1\rangle\langle 1|$. If we try to describe $\rho_e$ by finding a projection $Q_e$ such that $\rho_e$ satisfies $Q_e$, then $Q_e$ must contain $\mathrm{supp}(\rho_e)$, hence it necessarily also contains the good state $|0\rangle\langle 0|$. Any such $Q_e$ therefore also "accepts" a correct state, which is a false positive.

This is the conceptual mismatch: satisfaction of a projection $P$ is a universal-style statement ("the whole state lies inside $P$"), but bug-catching is existential-style ("some bad behavior is possible").

Moreover, in classical IL, a result predicate can be understood as a set of states and "achieving" it means reaching each of its states from some initial state. If we try to transport that interpretation naively to projections (even with under-approximation), it becomes unreasonable: a projection typically under-approximates infinitely many density operators, while a finite program has only finitely many branch outcomes at each step. For example, measuring one qubit yields at most two post-states, so it cannot literally "reach all" density matrices under-approximated by $|0\rangle\langle 0|$.

QIL resolves both obstacles by: (i) replacing satisfaction by an under-approximation relation that can pinpoint "100% bad" directions without capturing good states; (ii) interpreting "achieving $Q$" as: $Q$ under-approximates the *probabilistic mixture* of reachable terminal states (with the chosen exit condition), rather than requiring reachability of every individual state described by $Q$.

## 9.1 Under-Approximation Relation

We use the under-approximation relation from the prerequisites chapter as the basic semantic notion of "evidence of a bug": Let $P$ be a projection on $H$ and let $\rho \in \mathcal{D}_{\leq 1}(H)$ be a (possibly subnormalized) density operator. We say that $P$ under-approximates $\rho$, written $P \preceq \rho$, if $\mathrm{Ran}(P) \subseteq \mathrm{supp}(\rho)$.

Intuitively, $P \preceq \rho$ means that $\rho$ has nonzero weight in every direction of the subspace $P$, so $P$ can be read as a "definite bad component" present in $\rho$. This is exactly the inverted direction compared to satisfaction-by-support-inclusion.

The main structural facts we will use are: (i) Monotonicity in the predicate: if $P \leq Q$ and $Q \preceq \rho$ then $P \preceq \rho$; (ii) Mixtures behave like disjunction: if $\rho = \rho_1 + \rho_2$, then $P \preceq \rho$ iff $P \leq (\Pi_{\rho_1} \vee \Pi_{\rho_2})$, i.e. $P$ sits inside the join of the component supports.

These properties explain why QIL uses projection connectives and why it reasons about mixtures of reachable states rather than single states.

### 9.2 Strong Validity

QIL triples are indexed by an exit condition $\epsilon \in \{\mathsf{ok}, \mathsf{er}\}$ and have the form

$$[P]\ C\ [\epsilon : Q],$$

where $P, Q$ are projections on $H$.

From the collecting semantics (Section 7), for each command $C$ and exit $\epsilon$, we have a multiset-valued denotation

$$[\![C]\!]_\epsilon : \mathcal{D}_{\leq 1}(H) \to \mathsf{Mset}(\mathcal{D}_{\leq 1}(H)),$$

and we form the exit-conditioned mixture by summing the multiset:

$$\mathrm{Mix}_\epsilon(C, \rho) \ := \ \sum\big([\![C]\!]_\epsilon(\rho)\big) \ \in \ \mathcal{D}_{\leq 1}(H).$$

**Definition 9.1** (Strong validity of QIL triples)**.** A QIL triple is (strongly) valid, written $\models [P]\, C\, [\epsilon : Q]$, if for all $\rho \in \mathcal{D}_{\leq 1}(H)$,

$$P \preceq \rho \quad \Longrightarrow \quad Q \preceq \mathrm{Mix}_\epsilon(C, \rho).$$

This definition formalizes the slogan: if the initial state contains at least the behaviors in $P$, then the mixture of reachable terminal states (with exit $\epsilon$) contains at least the behaviors in $Q$.

**Predicate-transformer view (post-image).** We will package strong validity via a post-image transformer on projections. Let $P$ be a projection. Choose any $\rho_P \in \mathcal{D}_{\leq 1}(H)$ with $\mathrm{supp}(\rho_P) = \mathrm{Ran}(P)$ (the paper uses $\rho_P = P/\mathrm{tr}(P)$ when $P \neq 0$, else 0). Define

$$\mathrm{post}_\epsilon(C)(P) \ := \ \mathrm{supp}\big(\mathrm{Mix}_\epsilon(C, \rho_P)\big),$$

viewed as the projection onto that support subspace. This is well-defined (independent of the choice of $\rho_P$ with support $P$). Then strong validity has the equivalent form

$$\models [P]\, C\, [\epsilon : Q] \quad \Longleftrightarrow \quad \mathrm{post}_\epsilon(C)(P) \ \supseteq \ Q.$$

**Duality with quantum Hoare reasoning.** For programs without $\mathsf{error}$, the same post transformer $\mathrm{post}_{\mathsf{ok}}$ is also the one used in (applied) quantum Hoare logic, but with the opposite inclusion direction:

$$\models \{P\}\, C\, \{Q\} \iff \mathrm{post}_{\mathsf{ok}}(C)(P) \subseteq Q, \qquad \models [P]\, C\, [\epsilon : Q] \iff \mathrm{post}_\epsilon(C)(P) \supseteq Q.$$

However, we chose to present CoqQ's reasoning style which did not use the post-image transformer.

### 9.3 Proof System

We write $\vdash [P]\, C\, [\epsilon : Q]$ for derivability in the QIL proof system.

$$\frac{}{\vdash [P]\ C\ [\epsilon : 0]}\ (\textsc{Empty})$$

$$\frac{}{\vdash [P]\ \mathsf{error}\ [\mathsf{ok} : 0]\ [\mathsf{er} : P]}\ (\textsc{Error}) \qquad \frac{}{\vdash [P]\ \mathsf{skip}\ [\mathsf{ok} : P]\ [\mathsf{er} : 0]}\ (\textsc{Skip})$$

$$\frac{}{\vdash [P]\ \mathsf{apply}\ U_s\ [\mathsf{ok} : U_s^{(s)}\, P\, (U_s^{(s)})^\dagger]\ [\mathsf{er} : 0]}\ (\textsc{Unitary})$$

Initialization in the QIL paper is the special case "prepare $|0\rangle$ on a qubit"; in our syntax this is $\mathsf{init}\ \rho_s$ with $\rho_s = |0\rangle\langle 0|$ and $s = \{q\}$. The rule is:

$$\frac{\vdash [P]\ \mathsf{init}\ \ |0\rangle\langle 0|_{\{q\}}\ \ [\mathsf{ok} : \mathrm{supp}(\textstyle\sum_n K_n^{(q)}\, P\, (K_n^{(q)})^\dagger)]\ [\mathsf{er} : 0]}{}\ (\textsc{Init})$$

where $\{K_n\}_n$ are the Kraus operators of the reset-to-$|0\rangle$ map on $q$ (as in the QIL paper, this is written concretely using $|0\rangle\langle n|$ operators, and one must take $\mathrm{supp}(\cdot)$ because the raw expression need not be a projection).

$$\frac{\vdash [P]\ C_1\ [\mathsf{ok} : R] \quad \vdash [R]\ C_2\ [\epsilon : Q]}{\vdash [P]\ C_1; C_2\ [\epsilon : Q]}\ (\textsc{Seq1}) \qquad \frac{\vdash [P]\ C_1\ [\mathsf{er} : Q]}{\vdash [P]\ C_1; C_2\ [\mathsf{er} : Q]}\ (\textsc{Seq2})$$

$$\frac{P \supseteq P' \quad \vdash [P']\ C\ [\epsilon : Q'] \quad Q' \supseteq Q}{\vdash [P]\ C\ [\epsilon : Q]}\ (\textsc{Order})$$

$$\frac{\vdash [P_1]\ C\ [\epsilon : Q_1] \quad \vdash [P_2]\ C\ [\epsilon : Q_2]}{\vdash [P_1 \vee P_2]\ C\ [\epsilon : Q_1 \vee Q_2]}\ (\textsc{Disjunction})$$

Let $M_s = \{(m, M_m)\}_{m \in \mathsf{Out}(M_s)}$ be a measurement on subsystem $s$ and recall the post-measurement state is $M_m^{(s)}\, \rho\, (M_m^{(s)})^\dagger$.

$$\frac{\forall m \in \mathsf{Out}(M_s),\ \vdash \left[\mathrm{supp}(M_m^{(s)}\, P\, (M_m^{(s)})^\dagger)\right]\ C_m\ [\epsilon : Q]}{\vdash [P]\ \mathsf{if}\ (\square m.\ M_s = m \to C_m)\ \mathsf{fi}\ [\epsilon : Q]}\ (\textsc{If})$$

(The support appears because measurement changes the state before the branch code runs.)

Let $W \equiv \mathsf{while}\ M_s' = 1\ \mathsf{do}\ C\ \mathsf{od}$ with $M_s' = \{M_0, M_1\}$.

$$\frac{\forall n.\ \vdash \left[\mathrm{supp}(M_1^{(s)}\, P_n\, (M_1^{(s)})^\dagger)\right]\ C\ [\mathsf{ok} : P_{n+1}]}{\vdash [P_0]\ W\ [\mathsf{ok} : \mathrm{supp}(M_0^{(s)}\, P_N\, (M_0^{(s)})^\dagger)]}\ (\textsc{While1})$$

$$\frac{\forall n.\ \vdash \left[\mathrm{supp}(M_1^{(s)}\, P_n\, (M_1^{(s)})^\dagger)\right]\ C\ [\mathsf{ok} : P_{n+1}] \quad \vdash \left[\mathrm{supp}(M_1^{(s)}\, P_N\, (M_1^{(s)})^\dagger)\right]\ C\ [\mathsf{er} : Q]}{\vdash [P_0]\ W\ [\mathsf{er} : Q]}\ (\textsc{While2})$$

These are exactly the loop-variant style rules from QIL: $P_n$ tracks what can be achieved after $n$ completed continue-iterations.

$$\frac{Q_{\mathsf{ok}} = \mathrm{supp}(RPR^\dagger) \quad Q_{\mathsf{er}} = \mathrm{supp}(R^\perp P (R^\perp)^\dagger)}{\vdash [P]\ \mathsf{assert}(R)\ [\mathsf{ok} : Q_{\mathsf{ok}}]\ [\mathsf{er} : Q_{\mathsf{er}}]}\ (\textsc{Assert})$$

where $\mathsf{assert}(R)$ is encoded as a measurement with outcomes true$\to$ skip and false$\to$ error (as in the QIL paper).

$$\frac{\forall m,\ \vdash \left[\mathrm{supp}(M_m^{(s)} P (M_m^{(s)})^\dagger)\right]\ C_m\ [\epsilon : Q_m]}{\vdash [P]\ \mathsf{if}\ (\square m.\ M_s = m \to C_m)\ \mathsf{fi}\ [\epsilon : \bigvee_m Q_m]}\ (\textsc{Derived If})$$

$$\frac{\forall n < N.\ \vdash \left[\mathrm{supp}(M_1^{(s)} P_n (M_1^{(s)})^\dagger)\right]\ C\ [\mathsf{ok} : P_{n+1}]}{\vdash [P_0]\ W\ [\mathsf{ok} : \bigvee_{i=0}^{N} \mathrm{supp}(M_0^{(s)} P_i (M_0^{(s)})^\dagger)]}\ (\textsc{Derived While})$$

## 9.4 Soundness of the Logic

**Theorem 9.2** (Soundness)**.** *For any qwhile command $C$, exit condition $\epsilon \in \{\mathsf{ok}, \mathsf{er}\}$, and projections $P, Q$,*

$$\vdash [P]\ C\ [\epsilon : Q] \quad \Longrightarrow \quad \models [P]\ C\ [\epsilon : Q].$$

Before we prove the theorem, we present several useful lemmas:

**Lemma 9.3.** *For every $\rho \sqsupseteq 0$, we have $0 \preceq \rho$.*

*Proof.* $\mathrm{Ran}(0) = \{0\} \subseteq \mathrm{supp}(\rho)$ for all $\rho \sqsupseteq 0$. □

**Lemma 9.4.** *If $A, B \sqsupseteq 0$ and $A \sqsupseteq \alpha B$ for some $\alpha > 0$, then $\mathrm{supp}(B) \subseteq \mathrm{supp}(A)$. Equivalently, if $\Pi_B$ is the support projector of $B$, then $\Pi_B \preceq A$.*

*Proof.* From $A - \alpha B \sqsupseteq 0$, we get: if $Av = 0$ then $\langle v, Av\rangle = 0$ and $0 \le \alpha\langle v, Bv\rangle \le \langle v, Av\rangle = 0$, hence $\langle v, Bv\rangle = 0$. Since $B \sqsupseteq 0$, $\langle v, Bv\rangle = 0$ implies $Bv = 0$, so $\ker(A) \subseteq \ker(B)$. Taking orthogonal complements gives $\mathrm{supp}(B) = (\ker B)^\perp \subseteq (\ker A)^\perp = \mathrm{supp}(A)$. □

**Lemma 9.5.** *Let $P$ be a projection and $\rho \sqsupseteq 0$. If $P \preceq \rho$, then there exists $\alpha > 0$ such that $\rho \sqsupseteq \alpha P$.*

*Proof.* Let $S := \mathrm{Ran}(P)$. The condition $P \preceq \rho$ means $S \subseteq \mathrm{supp}(\rho) = (\ker \rho)^\perp$, so $\rho$ has trivial kernel on $S$; equivalently, the restriction $\rho|_S : S \to S$ is positive definite. Since $S$ is finite-dimensional, $\rho|_S$ has a smallest eigenvalue $\alpha > 0$, so $\langle v, \rho v\rangle \ge \alpha\|v\|^2$ for all $v \in S$. This inequality is exactly $\rho \sqsupseteq \alpha P$ on all of $H$, because $P$ is the orthogonal projector onto $S$. □

**Lemma 9.6** (Empty is sound)**.** *For every command $C$, every projection $P$, and every exit condition $\epsilon \in \{\mathsf{ok}, \mathsf{er}\}$,*

$$\models [P]\ C\ [\epsilon : 0].$$

*Proof.* Fix $\rho \in \mathcal{D}_{\le 1}(H)$ and assume $P \preceq \rho$. We must show $0 \preceq \mathrm{Mix}_\epsilon(C, \rho)$. But this holds for all outputs by Lemma 9.3. Hence the triple is strongly valid. □

**Lemma 9.7** (Error is sound)**.** *For every projection $P$,*

$$\models [P]\ \mathsf{error}\ [\mathsf{ok} : 0] \qquad \textit{and} \qquad \models [P]\ \mathsf{error}\ [\mathsf{er} : P].$$

*Proof.* Fix $\rho \in \mathcal{D}_{\le 1}(H)$ and assume $P \preceq \rho$.

*Ok-branch.* By the collecting semantics of $\mathsf{error}$, $[\![\mathsf{error}]\!]_{\mathsf{ok}}(\rho) = \{\{0\}\}$, hence $\mathrm{Mix}_{\mathsf{ok}}(\mathsf{error}, \rho) = 0$. Therefore $0 \preceq \mathrm{Mix}_{\mathsf{ok}}(\mathsf{error}, \rho)$ by Lemma 9.3.

*Er-branch.* Also $[\![\mathsf{error}]\!]_{\mathsf{er}}(\rho) = \{\{\rho\}\}$, hence $\mathrm{Mix}_{\mathsf{er}}(\mathsf{error}, \rho) = \rho$. Thus $P \preceq \mathrm{Mix}_{\mathsf{er}}(\mathsf{error}, \rho)$ holds immediately from the assumption $P \preceq \rho$. □

**Lemma 9.8** (Skip is sound)**.** *For every projection $P$,*

$$\models [P]\ \mathsf{skip}\ [\mathsf{ok} : P] \qquad \textit{and} \qquad \models [P]\ \mathsf{skip}\ [\mathsf{er} : 0].$$

*Proof.* Fix $\rho \in \mathcal{D}_{\leq 1}(H)$ and assume $P \preceq \rho$.

*Ok-branch.* $[\![\mathsf{skip}]\!]_{\mathsf{ok}}(\rho) = \{\{\rho\}\}$, hence $\mathrm{Mix}_{\mathsf{ok}}(\mathsf{skip}, \rho) = \rho$. Therefore $P \preceq \mathrm{Mix}_{\mathsf{ok}}(\mathsf{skip}, \rho)$ is exactly the assumption $P \preceq \rho$.

*Er-branch.* $[\![\mathsf{skip}]\!]_{\mathsf{er}}(\rho) = \{\{0\}\}$, hence $\mathrm{Mix}_{\mathsf{er}}(\mathsf{skip}, \rho) = 0$. Therefore $0 \preceq \mathrm{Mix}_{\mathsf{er}}(\mathsf{skip}, \rho)$ by Lemma 9.3. □

**Lemma 9.9** (UNITARY is sound)**.** *Let $U_s$ be unitary on subsystem $s$, and write $U := U_s^{(s)}$ for its cylindrical extension. Then for every projection $P$,*

$$\models [P]\ \mathsf{apply}\ U_s\ [\mathsf{ok} : UPU^\dagger] \qquad \textit{and} \qquad \models [P]\ \mathsf{apply}\ U_s\ [\mathsf{er} : 0].$$

*Proof.* Fix $\rho \in \mathcal{D}_{\leq 1}(H)$ and assume $P \preceq \rho$.

*Ok-branch.* The collecting semantics gives $[\![\mathsf{apply}\ U_s]\!]_{\mathsf{ok}}(\rho) = \{\{U\rho U^\dagger\}\}$, hence $\mathrm{Mix}_{\mathsf{ok}}(\mathsf{apply}\ U_s, \rho) = U\rho U^\dagger$. We must show $UPU^\dagger \preceq U\rho U^\dagger$, i.e.

$$\mathrm{Ran}(UPU^\dagger) \subseteq \mathrm{supp}(U\rho U^\dagger).$$

But $\mathrm{Ran}(UPU^\dagger) = U(\mathrm{Ran}(P))$, and by support calculus, $\mathrm{supp}(U\rho U^\dagger) = U(\mathrm{supp}(\rho))$. Since $P \preceq \rho$ means $\mathrm{Ran}(P) \subseteq \mathrm{supp}(\rho)$, applying $U$ yields $U(\mathrm{Ran}(P)) \subseteq U(\mathrm{supp}(\rho))$, hence $UPU^\dagger \preceq U\rho U^\dagger$.

*Er-branch.* $[\![\mathsf{apply}\ U_s]\!]_{\mathsf{er}}(\rho) = \{\{0\}\}$, so $\mathrm{Mix}_{\mathsf{er}}(\mathsf{apply}\ U_s, \rho) = 0$ and $0 \preceq 0$ by Lemma 9.3. □

**Lemma 9.10** (INIT is sound)**.** *Let $q$ be a register with basis $\{|n\rangle\}_{n=0}^{d-1}$ and consider initialization to $|0\rangle$:*

$$C_{\mathrm{init}} \ \equiv\ \mathsf{init}\ \ |0\rangle\langle 0|_{\{q\}}\,.$$

*Define Kraus operators on $H$ by*

$$K_n := \big(|0\rangle\langle n|\big)^{(q)} \qquad (n = 0, \ldots, d-1),$$

*and for any projection $P$ define the (not-necessarily-projective) operator*

$$A_P := \sum_{n=0}^{d-1} K_n\, P\, K_n^\dagger, \qquad Q := \Pi_{A_P} \quad \textit{(the support projector of } A_P\textit{)}.$$

*Then for every projection $P$,*

$$\models [P]\ C_{\mathrm{init}}\ [\mathsf{ok} : Q] \qquad \textit{and} \qquad \models [P]\ C_{\mathrm{init}}\ [\mathsf{er} : 0].$$

*Proof.* Fix $\rho \in \mathcal{D}_{\leq 1}(H)$ and assume $P \preceq \rho$.

*Step 1:* Initialization always terminates normally, so $[\![C_{\mathrm{init}}]\!]_{\mathsf{er}}(\rho) = \{\{0\}\}$ and $\mathrm{Mix}_{\mathsf{er}}(C_{\mathrm{init}}, \rho) = 0$. On the normal branch, the standard Kraus form of reset gives

$$\mathrm{Mix}_{\mathsf{ok}}(C_{\mathrm{init}}, \rho) \ =\ \sum_{n=0}^{d-1} K_n\, \rho\, K_n^\dagger.$$

(Equivalently, this equals $|0\rangle\langle 0|_{\{q\}} \otimes \mathrm{tr}_q(\rho)$, but we keep the Kraus form since the postcondition is defined from $\sum_n K_n P K_n^\dagger$.)

*Step 2:* We must show

$$Q \preceq \mathrm{Mix}_{\mathsf{ok}}(C_{\mathrm{init}}, \rho), \quad \text{i.e.} \quad \mathrm{supp}(A_P) \subseteq \mathrm{supp}\Big(\sum_n K_n \rho K_n^\dagger\Big),$$

because $\mathrm{Ran}(Q) = \mathrm{supp}(A_P)$ by definition of the support projector.

*Step 3:* By Lemma 9.5, from $P \preceq \rho$ we get some $\alpha > 0$ such that

$$\rho \sqsupseteq \alpha P.$$

*Step 4:* The map $E(X) := \sum_n K_n X K_n^\dagger$ is completely positive, hence positive and monotone: if $X \sqsupseteq Y \sqsupseteq 0$ then $E(X) \sqsupseteq E(Y)$. Applying $E$ to $\rho \sqsupseteq \alpha P$ yields

$$\sum_n K_n \rho K_n^\dagger \;\sqsupseteq\; \alpha \sum_n K_n P K_n^\dagger \;=\; \alpha A_P.$$

*Step 5: conclude support inclusion.* By Lemma 9.4, from $\sum_n K_n \rho K_n^\dagger \sqsupseteq \alpha A_P$ with $\alpha > 0$ we conclude

$$\operatorname{supp}(A_P) \subseteq \operatorname{supp}\Big(\sum_n K_n \rho K_n^\dagger\Big).$$

Thus $Q = \Pi_{A_P}$ under-approximates the normal-branch mixture: $Q \preceq \mathrm{Mix}_{\mathsf{ok}}(C_{\mathrm{init}}, \rho)$.
*Er-branch.* As observed in Step 1, $\mathrm{Mix}_{\mathsf{er}}(C_{\mathrm{init}}, \rho) = 0$, hence $0 \preceq 0$ by Lemma 9.3. □

**Lemma 9.11.** *For multisets $\nu_1, \nu_2 \in \mathsf{Mset}(\mathcal{D}_{\leq 1}(H))$,*

$$\sum(\nu_1 \uplus \nu_2) = \sum(\nu_1) + \sum(\nu_2).$$

*Proof.* By definition,

$$\sum(\nu_1 \uplus \nu_2) = \sum_\sigma (\nu_1(\sigma) + \nu_2(\sigma))\,\sigma = \sum_\sigma \nu_1(\sigma)\sigma + \sum_\sigma \nu_2(\sigma)\sigma = \sum(\nu_1) + \sum(\nu_2).$$

□

**Lemma 9.12.** *For any commands $C_1, C_2$ and any $\rho \in \mathcal{D}_{\leq 1}(H)$,*

$$\mathrm{Mix}_{\mathsf{ok}}(C_1; C_2, \rho) = [\![C_2]\!]\big(\mathrm{Mix}_{\mathsf{ok}}(C_1, \rho)\big),$$

*and*

$$\mathrm{Mix}_{\mathsf{er}}(C_1; C_2, \rho) = \mathrm{Mix}_{\mathsf{er}}(C_1, \rho) \;+\; \mathrm{Mix}_{\mathsf{er}}\big(C_2, \mathrm{Mix}_{\mathsf{ok}}(C_1, \rho)\big).$$

*Proof.* We use the collecting sequencing clauses (Section 7):

$$[\![C_1; C_2]\!]_{\mathsf{ok}} := [\![C_2]\!]_{\mathsf{ok}} \circ [\![C_1]\!]_{\mathsf{ok}}, \qquad [\![C_1; C_2]\!]_{\mathsf{er}} := [\![C_1]\!]_{\mathsf{er}} \uplus \big([\![C_2]\!]_{\mathsf{er}} \circ [\![C_1]\!]_{\mathsf{ok}}\big),$$

and the definition $\mathrm{Mix}_\epsilon(C, \rho) = \sum([\![C]\!]_\epsilon(\rho))$.
*Normal exit.*

$$\mathrm{Mix}_{\mathsf{ok}}(C_1; C_2, \rho) = \sum\big([\![C_1; C_2]\!]_{\mathsf{ok}}(\rho)\big) = \sum\big(([\![C_2]\!]_{\mathsf{ok}} \circ [\![C_1]\!]_{\mathsf{ok}})(\rho)\big).$$

By construction, $[\![C_2]\!]$ (the super-operator semantics of Section 6) coincides with the normal-mixture semantics: $[\![C_2]\!](\sigma) = \mathrm{Mix}_{\mathsf{ok}}(C_2, \sigma)$ for all inputs $\sigma$, because $[\![\cdot]\!]$ collapses branching by summation and treats abnormal termination as contributing 0. Therefore, running $C_2$ after collecting the normal outcomes of $C_1$ yields exactly $[\![C_2]\!](\mathrm{Mix}_{\mathsf{ok}}(C_1, \rho))$, giving the first equation.
*Abnormal exit.*

$$\begin{aligned}
\mathrm{Mix}_{\mathsf{er}}(C_1; C_2, \rho) &= \sum\big([\![C_1; C_2]\!]_{\mathsf{er}}(\rho)\big) \\
&= \sum\Big([\![C_1]\!]_{\mathsf{er}}(\rho) \uplus ([\![C_2]\!]_{\mathsf{er}} \circ [\![C_1]\!]_{\mathsf{ok}})(\rho)\Big) \\
&= \sum\big([\![C_1]\!]_{\mathsf{er}}(\rho)\big) \;+\; \sigma\big(([\![C_2]\!]_{\mathsf{er}} \circ [\![C_1]\!]_{\mathsf{ok}})(\rho)\big) \qquad \text{(Lemma 9.11)} \\
&= \mathrm{Mix}_{\mathsf{er}}(C_1, \rho) \;+\; \mathrm{Mix}_{\mathsf{er}}\big(C_2, \mathrm{Mix}_{\mathsf{ok}}(C_1, \rho)\big),
\end{aligned}$$

where the second summand is "errors that occur in $C_2$ after $C_1$ terminates normally" (consistent with the operational reading of sequencing). □

**Lemma 9.13** (SEQ1 is sound)**.** *Assume*

$$\models [P]\ C_1\ [\mathsf{ok} : R] \qquad \textit{and} \qquad \models [R]\ C_2\ [\epsilon : Q],$$

*where* $\epsilon \in \{\mathsf{ok}, \mathsf{er}\}$*. Then*

$$\models [P]\ (C_1; C_2)\ [\epsilon : Q].$$

*Proof.* Fix an arbitrary $\rho \in \mathcal{D}_{\leq 1}(H)$ and assume $P \preceq \rho$. From $\models [P]\ C_1\ [\mathsf{ok} : R]$ we obtain

$$R \preceq \mathrm{Mix}_{\mathsf{ok}}(C_1, \rho).$$

Now apply $\models [R]\ C_2\ [\epsilon : Q]$ to the input state $\sigma := \mathrm{Mix}_{\mathsf{ok}}(C_1, \rho)$: since $R \preceq \sigma$, we get

$$Q \preceq \mathrm{Mix}_{\epsilon}(C_2, \sigma) = \mathrm{Mix}_{\epsilon}\big(C_2, \mathrm{Mix}_{\mathsf{ok}}(C_1, \rho)\big).$$

*Case* $\epsilon = \mathsf{ok}$. By Lemma 9.12,

$$\mathrm{Mix}_{\mathsf{ok}}(C_1; C_2, \rho) = [\![C_2]\!]\big(\mathrm{Mix}_{\mathsf{ok}}(C_1, \rho)\big) = \mathrm{Mix}_{\mathsf{ok}}\big(C_2, \mathrm{Mix}_{\mathsf{ok}}(C_1, \rho)\big),$$

so $Q \preceq \mathrm{Mix}_{\mathsf{ok}}(C_1; C_2, \rho)$ as required.
*Case* $\epsilon = \mathsf{er}$. By Lemma 9.12,

$$\mathrm{Mix}_{\mathsf{er}}(C_1; C_2, \rho) = \mathrm{Mix}_{\mathsf{er}}(C_1, \rho) + \mathrm{Mix}_{\mathsf{er}}\big(C_2, \mathrm{Mix}_{\mathsf{ok}}(C_1, \rho)\big).$$

Since $\mathrm{Mix}_{\mathsf{er}}\big(C_2, \mathrm{Mix}_{\mathsf{ok}}(C_1, \rho)\big) \sqsupseteq 0$, we have

$$\mathrm{supp}\Big(\mathrm{Mix}_{\mathsf{er}}\big(C_2, \mathrm{Mix}_{\mathsf{ok}}(C_1, \rho)\big)\Big) \subseteq \mathrm{supp}\big(\mathrm{Mix}_{\mathsf{er}}(C_1; C_2, \rho)\big).$$

Then by monotonicity of under-approximation w.r.t. support inclusion (Lemma 4.54(ii) in the prerequisites), from $Q \preceq \mathrm{Mix}_{\mathsf{er}}(C_2, \mathrm{Mix}_{\mathsf{ok}}(C_1, \rho))$ we conclude $Q \preceq \mathrm{Mix}_{\mathsf{er}}(C_1; C_2, \rho)$.

In both cases, $Q \preceq \mathrm{Mix}_{\epsilon}(C_1; C_2, \rho)$ follows. Since $\rho$ was arbitrary, the triple is strongly valid. □

**Lemma 9.14** (SEQ2 is sound)**.** *Assume* $\models [P]\ C_1\ [\mathsf{er} : Q]$*. Then*

$$\models [P]\ (C_1; C_2)\ [\mathsf{er} : Q]$$

*for any command* $C_2$*.*

*Proof.* Fix $\rho \in \mathcal{D}_{\leq 1}(H)$ and assume $P \preceq \rho$. By assumption, $Q \preceq \mathrm{Mix}_{\mathsf{er}}(C_1, \rho)$.

By Lemma 9.12,

$$\mathrm{Mix}_{\mathsf{er}}(C_1; C_2, \rho) = \mathrm{Mix}_{\mathsf{er}}(C_1, \rho) + \mathrm{Mix}_{\mathsf{er}}\big(C_2, \mathrm{Mix}_{\mathsf{ok}}(C_1, \rho)\big).$$

Hence $\mathrm{supp}(\mathrm{Mix}_{\mathsf{er}}(C_1, \rho)) \subseteq \mathrm{supp}(\mathrm{Mix}_{\mathsf{er}}(C_1; C_2, \rho))$. Using Lemma 4.54(ii) (monotonicity under support inclusion), from $Q \preceq \mathrm{Mix}_{\mathsf{er}}(C_1, \rho)$ we conclude $Q \preceq \mathrm{Mix}_{\mathsf{er}}(C_1; C_2, \rho)$. □

**Lemma 9.15** (ORDER is sound)**.** *Assume* $P \supseteq P'$*,* $Q' \supseteq Q$*, and* $\models [P']\ C\ [\epsilon : Q']$*. Then* $\models [P]\ C\ [\epsilon : Q]$*.*

*Proof.* Fix $\rho \in \mathcal{D}_{\leq 1}(H)$ and assume $P \preceq \rho$.

Since $P \supseteq P'$ means $P' \leq P$ (range inclusion), by monotonicity in the predicate (Lemma 4.54(i)) we have $P' \preceq \rho$. Now apply $\models [P']\ C\ [\epsilon : Q']$ to get $Q' \preceq \mathrm{Mix}_{\epsilon}(C, \rho)$.

Finally, $Q' \supseteq Q$ means $Q \leq Q'$, and again by Lemma 4.54(i), from $Q' \preceq \mathrm{Mix}_{\epsilon}(C, \rho)$ we conclude $Q \preceq \mathrm{Mix}_{\epsilon}(C, \rho)$. Thus $\models [P]\ C\ [\epsilon : Q]$. □

**Lemma 9.16** (DISJUNCTION is sound)**.** *Assume* $\models [P_1]\ C\ [\epsilon : Q_1]$ *and* $\models [P_2]\ C\ [\epsilon : Q_2]$*. Then*

$$\models [P_1 \vee P_2]\ C\ [\epsilon : Q_1 \vee Q_2].$$

*Proof.* Fix $\rho \in \mathcal{D}_{\leq 1}(H)$ and assume $(P_1 \vee P_2) \preceq \rho$. Under-approximation treats $\vee$ as conjunction:

$$(P_1 \vee P_2) \preceq \rho \quad \Longleftrightarrow \quad (P_1 \preceq \rho)\ \wedge\ (P_2 \preceq \rho)$$

(Lemma 4.56(ii)). Hence $P_1 \preceq \rho$ and $P_2 \preceq \rho$.

Applying the two premises yields

$$Q_1 \preceq \mathrm{Mix}_\epsilon(C, \rho) \qquad \text{and} \qquad Q_2 \preceq \mathrm{Mix}_\epsilon(C, \rho).$$

Using the same connective law again,

$$(Q_1 \vee Q_2) \preceq \mathrm{Mix}_\epsilon(C, \rho) \quad \Longleftrightarrow \quad (Q_1 \preceq \mathrm{Mix}_\epsilon(C, \rho))\ \wedge\ (Q_2 \preceq \mathrm{Mix}_\epsilon(C, \rho)),$$

so we conclude $(Q_1 \vee Q_2) \preceq \mathrm{Mix}_\epsilon(C, \rho)$. Thus $\models [P_1 \vee P_2]\ C\ [\epsilon : Q_1 \vee Q_2]$. □

**Lemma 9.17.** *If* $X, Y \sqsupseteq 0$ *and* $X \sqsubseteq Y$ *(i.e.* $Y - X \sqsupseteq 0$*), then* $\mathrm{supp}(X) \subseteq \mathrm{supp}(Y)$*. Consequently, if* $Q$ *is a projection and* $Q \preceq X$*, then* $Q \preceq Y$*.*

*Proof.* If $X \sqsubseteq Y$ then $\ker(Y) \subseteq \ker(X)$: indeed, $Yv = 0$ implies $\langle v, Yv\rangle = 0$ and $0 \leq \langle v, Xv\rangle \leq \langle v, Yv\rangle = 0$, hence $Xv = 0$. Taking orthogonal complements yields $\mathrm{supp}(X) \subseteq \mathrm{supp}(Y)$. If $Q \preceq X$, then $\mathrm{Ran}(Q) \subseteq \mathrm{supp}(X) \subseteq \mathrm{supp}(Y)$, so $Q \preceq Y$. □

**Lemma 9.18.** *Let* $K$ *be any linear operator on* $H$*. For every projection* $P$ *and every* $\rho \sqsupseteq 0$*,*

$$P \preceq \rho \quad \Longrightarrow \quad \Pi_{KPK^\dagger} \preceq K\rho K^\dagger.$$

*Proof.* Assume $P \preceq \rho$. By Lemma 9.5, $\rho \sqsupseteq \alpha P$ for some $\alpha > 0$. Conjugating by $K$ preserves the Löwner order, so

$$K\rho K^\dagger \sqsupseteq \alpha\, KPK^\dagger.$$

By Lemma 9.17, $\mathrm{supp}(KPK^\dagger) \subseteq \mathrm{supp}(K\rho K^\dagger)$, i.e. $\Pi_{KPK^\dagger} \preceq K\rho K^\dagger$. □

**Lemma 9.19.** *Let* $(Q_i)_{i\in I}$ *be any family of projections on* $H$*, and let* $\rho \sqsupseteq 0$*. If* $Q_i \preceq \rho$ *for all* $i \in I$*, then* $\bigvee_{i\in I} Q_i \preceq \rho$*.*

*Proof.* Each $Q_i \preceq \rho$ means $\mathrm{Ran}(Q_i) \subseteq \mathrm{supp}(\rho)$. Therefore $\mathrm{span}\big(\bigcup_i \mathrm{Ran}(Q_i)\big) \subseteq \mathrm{supp}(\rho)$. But $\mathrm{Ran}(\bigvee_i Q_i) = \mathrm{span}(\bigcup_i \mathrm{Ran}(Q_i))$, hence $\bigvee_i Q_i \preceq \rho$. □

**Lemma 9.20.** *Let* $C \equiv \mathsf{if}\ (\square m.\ M_s = m \to C_m)\ \mathsf{fi}$ *and* $M_s = \{(m, M_m)\}_{m\in \mathrm{Out}(M_s)}$*. Then for each* $\epsilon \in \{\mathsf{ok}, \mathsf{er}\}$ *and each* $\rho \in \mathcal{D}_{\leq 1}(H)$*,*

$$\mathrm{Mix}_\epsilon(C, \rho) = \sum_{m\in \mathrm{Out}(M_s)} \mathrm{Mix}_\epsilon\Big(C_m,\ M_m^{(s)}\, \rho\, \big(M_m^{(s)}\big)^\dagger\Big).$$

*Proof.* By the collecting semantics of conditionals,

$$[\![C]\!]_\epsilon(\rho) = \biguplus_{m\in \mathrm{Out}(M_s)} [\![C_m]\!]_\epsilon\Big(M_m^{(s)}\, \rho\, \big(M_m^{(s)}\big)^\dagger\Big).$$

Applying $\sum$ and using distributivity over multiset union yields the stated sum, and each summand is exactly $\mathrm{Mix}_\epsilon(C_m, \cdot)$ by definition. □

**Lemma 9.21** (IF is sound)**.** *Let* $C \equiv \mathsf{if}\ (\square m.\ M_s = m \to C_m)\ \mathsf{fi}$ *and let* $M_s = \{(m, M_m)\}_{m \in \mathrm{Out}(M_s)}$. *Fix* $\epsilon \in \{\mathsf{ok}, \mathsf{er}\}$ *and a projection* $Q$. *Assume that for all* $m \in \mathrm{Out}(M_s)$,

$$\models \left[\Pi_{M_m^{(s)}\, P \left(M_m^{(s)}\right)^\dagger}\right]\ C_m\ [\epsilon : Q].$$

*Then*

$$\models [P]\ C\ [\epsilon : Q].$$

*Proof.* Fix $\rho \in \mathcal{D}_{\leq 1}(H)$ and assume $P \preceq \rho$. For each $m \in \mathrm{Out}(M_s)$, define the post-measurement state

$$\rho_m := M_m^{(s)}\, \rho\, \left(M_m^{(s)}\right)^\dagger.$$

By Lemma 9.18 (with $K = M_m^{(s)}$), from $P \preceq \rho$ we obtain

$$\Pi_{M_m^{(s)}\, P \left(M_m^{(s)}\right)^\dagger} \preceq \rho_m.$$

Apply the premise (strong validity of the $m$-th branch) to $\rho_m$ to get

$$Q \preceq \mathrm{Mix}_\epsilon(C_m, \rho_m).$$

Now by Lemma 9.20,

$$\mathrm{Mix}_\epsilon(C, \rho) = \sum_{m \in \mathrm{Out}(M_s)} \mathrm{Mix}_\epsilon(C_m, \rho_m).$$

For each fixed $m$, we have $\mathrm{Mix}_\epsilon(C_m, \rho_m) \sqsubseteq \sum_k \mathrm{Mix}_\epsilon(C_k, \rho_k) = \mathrm{Mix}_\epsilon(C, \rho)$, so by Lemma 9.17, from $Q \preceq \mathrm{Mix}_\epsilon(C_m, \rho_m)$ we conclude $Q \preceq \mathrm{Mix}_\epsilon(C, \rho)$. Since $\rho$ was arbitrary, $\models [P]\ C\ [\epsilon : Q]$ holds. □

**Lemma 9.22.** *Let* $W \equiv \mathsf{while}\ M'_s = 1\ \mathsf{do}\ C\ \mathsf{od}$ *with* $M'_s = \{M_0, M_1\}$. *Define a sequence of partial states* $(\sigma_n)_{n \geq 0}$ *by*

$$\sigma_0 := \rho, \qquad \sigma_{n+1} := \mathrm{Mix}_{\mathsf{ok}}\Big(C,\ M_1^{(s)}\, \sigma_n\, \left(M_1^{(s)}\right)^\dagger\Big).$$

*Then:*

$$\mathrm{Mix}_{\mathsf{ok}}(W, \rho) = \sum_{n \geq 0} M_0^{(s)}\, \sigma_n\, \left(M_0^{(s)}\right)^\dagger,$$

*and*

$$\mathrm{Mix}_{\mathsf{er}}(W, \rho) = \sum_{n \geq 0} \mathrm{Mix}_{\mathsf{er}}\Big(C,\ M_1^{(s)}\, \sigma_n\, \left(M_1^{(s)}\right)^\dagger\Big).$$

*Proof.* This is a direct unpacking of the collecting loop semantics in the QIL style: normal termination consists of $n$ continue rounds (outcome 1 then execute $C$ normally), followed by a stop round (outcome 0); abnormal termination consists of $n$ normal iterations followed by a continue outcome 1 whose subsequent body execution exits abnormally. Summing (mixing) the multiset denotations yields the two series above. □

**Lemma 9.23** (WHILE1 is sound)**.** *Let* $W \equiv \mathsf{while}\ M'_s = 1\ \mathsf{do}\ C\ \mathsf{od}$ *with* $M'_s = \{M_0, M_1\}$. *Let* $(P_n)_{n \geq 0}$ *be a sequence of projections. Assume that for all* $n \geq 0$,

$$\models \left[\Pi_{M_1^{(s)}\, P_n \left(M_1^{(s)}\right)^\dagger}\right]\ C\ [\mathsf{ok} : P_{n+1}].$$

*Define the projection*

$$Q_{\mathsf{ok}} := \bigvee_{n \geq 0} \Pi_{M_0^{(s)}\, P_n \left(M_0^{(s)}\right)^\dagger}.$$

*Then*

$$\models [P_0]\ W\ [\mathsf{ok} : Q_{\mathsf{ok}}].$$

*Proof.* Fix $\rho \in \mathcal{D}_{\leq 1}(H)$ and assume $P_0 \preceq \rho$.

*Define the loop-head mixtures $\sigma_n$ and show $P_n \preceq \sigma_n$ for all $n$.* Let $(\sigma_n)$ be as in Lemma 9.22: $\sigma_0 = \rho$ and $\sigma_{n+1} = \mathrm{Mix}_{\mathsf{ok}}(C, M_1^{(s)}\sigma_n(M_1^{(s)})^\dagger)$. We prove by induction on $n$ that

$$P_n \preceq \sigma_n.$$

Base $n = 0$: $\sigma_0 = \rho$, so $P_0 \preceq \sigma_0$ is the assumption.

Inductive step: assume $P_n \preceq \sigma_n$. By Lemma 9.18 with $K = M_1^{(s)}$,

$$\Pi_{M_1^{(s)}\, P_n \left(M_1^{(s)}\right)^\dagger} \preceq M_1^{(s)}\, \sigma_n \left(M_1^{(s)}\right)^\dagger.$$

Apply the premise (strong validity for iteration $n$) to the input $M_1^{(s)}\sigma_n(M_1^{(s)})^\dagger$ to obtain

$$P_{n+1} \preceq \mathrm{Mix}_{\mathsf{ok}}\Big(C,\ M_1^{(s)}\, \sigma_n \left(M_1^{(s)}\right)^\dagger\Big) = \sigma_{n+1}.$$

Thus $P_{n+1} \preceq \sigma_{n+1}$.

*Show each termination component under-approximates the corresponding stop-support.* For each $n$, define the $n$-th stopping contribution

$$\tau_n := M_0^{(s)}\, \sigma_n \left(M_0^{(s)}\right)^\dagger.$$

From $P_n \preceq \sigma_n$ and Lemma 9.18 with $K = M_0^{(s)}$ we get

$$\Pi_{M_0^{(s)}\, P_n \left(M_0^{(s)}\right)^\dagger} \preceq \tau_n.$$

Moreover, by Lemma 9.22,

$$\mathrm{Mix}_{\mathsf{ok}}(W, \rho) = \sum_{k \geq 0} \tau_k,$$

so $\tau_n \sqsubseteq \mathrm{Mix}_{\mathsf{ok}}(W, \rho)$ for each $n$, and by Lemma 9.17,

$$\Pi_{M_0^{(s)}\, P_n \left(M_0^{(s)}\right)^\dagger} \preceq \mathrm{Mix}_{\mathsf{ok}}(W, \rho) \qquad \text{for all } n.$$

*Take the join.* Since each join-component under-approximates $\mathrm{Mix}_{\mathsf{ok}}(W, \rho)$, Lemma 9.19 yields

$$Q_{\mathsf{ok}} = \bigvee_{n \geq 0} \Pi_{M_0^{(s)}\, P_n \left(M_0^{(s)}\right)^\dagger} \preceq \mathrm{Mix}_{\mathsf{ok}}(W, \rho).$$

This is exactly the desired strong validity for the loop on the normal exit. □

**Lemma 9.24** (WHILE2 is sound)**.** *Let $W \equiv \mathsf{while}\ M_s' = 1\ \mathsf{do}\ C\ \mathsf{od}$ with $M_s' = \{M_0, M_1\}$. Let $(P_n)_{n \geq 0}$ be a sequence of projections and fix $N \geq 0$. Assume:*

$$\forall n \geq 0.\ \models \left[\Pi_{M_1^{(s)}\, P_n \left(M_1^{(s)}\right)^\dagger}\right]\ C\ [\mathsf{ok} : P_{n+1}]$$

*and*

$$\models \left[\Pi_{M_1^{(s)}\, P_N \left(M_1^{(s)}\right)^\dagger}\right]\ C\ [\mathsf{er} : Q].$$

*Then*

$$\models [P_0]\ W\ [\mathsf{er} : Q].$$

*Proof.* Fix $\rho \in \mathcal{D}_{\leq 1}(H)$ and assume $P_0 \preceq \rho$. Let $(\sigma_n)$ be defined as in Lemma 9.22.

*Step 1: reach $P_N$ at loop head.* By the same induction as in the proof of Lemma 9.23, the first assumption implies

$$P_n \preceq \sigma_n \qquad \text{for all } n \geq 0,$$

hence in particular $P_N \preceq \sigma_N$.

*Step 2: enter the error-triggering iteration.* From $P_N \preceq \sigma_N$ and Lemma 9.18 with $K = M_1^{(s)}$ we obtain

$$\Pi_{M_1^{(s)}} P_N \left(M_1^{(s)}\right)^\dagger \preceq M_1^{(s)} \sigma_N \left(M_1^{(s)}\right)^\dagger.$$

Apply the second assumption (the error premise at level $N$) to the input $M_1^{(s)}\sigma_N(M_1^{(s)})^\dagger$ to get

$$Q \preceq \mathrm{Mix}_{\mathsf{er}}\Big(C,\ M_1^{(s)} \sigma_N \left(M_1^{(s)}\right)^\dagger\Big).$$

*Step 3: embed into the loop's error mixture.* By Lemma 9.22,

$$\mathrm{Mix}_{\mathsf{er}}(W, \rho) = \sum_{n \geq 0} \mathrm{Mix}_{\mathsf{er}}\Big(C,\ M_1^{(s)} \sigma_n \left(M_1^{(s)}\right)^\dagger\Big),$$

so the $n = N$ summand is $\sqsubseteq$-below $\mathrm{Mix}_{\mathsf{er}}(W, \rho)$. By Lemma 9.17,

$$Q \preceq \mathrm{Mix}_{\mathsf{er}}(W, \rho).$$

This establishes strong validity of the While2 conclusion. □

## 9.5 Completeness of the Logic

**Theorem 9.25** (Completeness)**.** *For any qwhile command $C$, exit condition $\epsilon \in \{\mathsf{ok}, \mathsf{er}\}$, and projections $P, Q$,*

$$\models [P]\ C\ [\epsilon : Q] \quad \Longrightarrow \quad \vdash [P]\ C\ [\epsilon : Q].$$

Although Theorem 9.25 is often presented as an "absolute" completeness statement, its proper interpretation is the same as in classical Incorrectness Logic (IL): it is a *relative* (or "semantic") completeness result, where the proof system is complete *relative to* the chosen assertion domain and the availability of an oracle for semantic side conditions.

The key point is that QIL does not take assertions to be a syntactic language of formulas with an independent proof theory (as classical Hoare logic typically does). Instead, assertions are *semantic objects*—projections (subspaces) on the fixed Hilbert space $H$—and entailment between assertions is simply subspace inclusion (equivalently, the Löwner order on projections). Thus, completeness should be read as:

> Every semantically valid under-approximate claim expressible *at the projection/support level* can be derived using the QIL rules, assuming we can discharge the linear-algebraic side conditions (projection inclusion, support computations, and measurement-induced support updates).

This is exactly analogous to the classical IL completeness story: IL is complete once one assumes that (i) assertions are expressive enough to denote the relevant sets of reachable states, and (ii) the underlying entailment checks are available (often idealized as an oracle). QIL simply makes this "oracle" aspect more concrete: in finite dimension, these checks reduce to linear algebra.

There are three distinct "relativizations" built into the QIL completeness claim:

*(1) Relative to the assertion domain (projections).* QIL predicates capture only *support* information: whether a state has nonzero component in a subspace. They intentionally forget quantitative weights. Consequently, QIL completeness does not mean "all bugs" are derivable; it means: all bugs that can be characterized as *support-level* facts are derivable. In particular, if two programs produce output states with the same support but different probabilities, QIL cannot distinguish them, and no completeness theorem can change this because it is an expressiveness limitation of the assertion domain, not of the proof rules.

*(2) Relative to semantic side conditions (entailment and support computation).* Even with projections as assertions, several rules require side conditions of the form $P' \supseteq P$ or $Q \supseteq Q'$, and the operational meaning of conditionals/loops requires computing supports of mixtures (e.g. $\mathrm{supp}(MPM^\dagger)$ and $\mathrm{supp}(\mathrm{Mix}_\epsilon(C, \rho_P))$). A completeness theorem implicitly assumes that such inclusions and support computations are available (or can be carried out within the metatheory). This is the direct analogue of assuming an "assertion theory" in Cook-style completeness: the proof system is complete once these semantic checks are not the bottleneck.

*(3) Relative to the loop model and finiteness assumptions.* The while rules in QIL are formulated using an $\omega$-sequence of variants $(P_n)$, reflecting that a loop may iterate arbitrarily many times. In classical program logics this typically leads to completeness statements that are non-constructive (they quantify over an infinite family). A major contribution of the QIL development is to show that, in finite-dimensional quantum systems, this infinite family can be replaced by a *bounded* family without losing completeness: the increasing chain of reachable support projections must stabilize once it cannot increase in rank, and the rank is bounded by $\dim(H)$. Thus, for finite-dimensional $H$, one can replace While1/While2 by bounded versions (with $N \leq \dim(H)$) and retain soundness and completeness. This is best viewed as a *relative completeness with a finiteness hypothesis*: the bounded system is complete relative to the assumption that the program's state space is finite-dimensional (as it is for standard qubit-register programs).