

Entire Proof System of rCOE

Sini Chen

April 3, 2025

1 Foundational Layer: Sequential Program Verification

1.1 Axioms from Hoare Logic and Separation Logic

Axiom 1 (Skip).

$$\{P\} \text{ skip } \{P\}$$

Axiom 2 (Consequence).

$$\frac{P \Rightarrow P' \quad \{P'\} C \{Q'\} \quad Q' \Rightarrow Q}{\{P\} C \{Q\}}$$

Axiom 3 (Sequential Composition).

$$\frac{\{P\} C_1 \{Q\} \quad \{Q\} C_2 \{R\}}{\{P\} C_1; C_2 \{R\}}$$

Axiom 4 (Conditional).

$$\frac{\{P \wedge B\} C \{Q\} \quad \{P \wedge \neg B\} C' \{Q\}}{\{P\} \text{ if } B \text{ then } C \text{ else } C' \{Q\}}$$

Axiom 5 (Loop).

$$\frac{\{P \wedge B\} C \{P\}}{\{P\} \text{ while } B \text{ do } C \{P \wedge \neg B\}}$$

Axiom 6 (Conjunction).

$$\frac{\{P_1\} C \{Q_1\} \quad \{P_2\} C \{Q_2\}}{\{P_1 \wedge P_2\} C \{Q_1 \wedge Q_2\}}$$

Axiom 7 (Disjunction).

$$\frac{\{P_1\} C \{Q\} \quad \{P_2\} C \{Q\}}{\{P_1 \vee P_2\} C \{Q\}}$$

Axiom 8 (Frame Rule).

$$\frac{\{P\} C \{Q\}}{\{P * R\} C \{Q * R\}}$$

1.2 Expression Typology

Rule 1 (Primitive Evaluation).

$$\frac{\text{prim} \in \{\text{bool}, \text{int}\}}{\vdash \{\text{true}\} \text{ prim } \{\mathbf{ret} = \llbracket \text{prim} \rrbracket\}}$$

Rule 2 (Reference Resolution).

$$\frac{\text{ref} \neq \perp}{\vdash \{\text{true}\} \text{ ref } \{\mathbf{ret} = \text{loc}(\text{ref})\}}$$

Rule 3 (Null Safety).

$$\vdash \{\text{true}\} \text{ null } \{\mathbf{ret} = \emptyset\}$$

Rule 4 (Contextual This).

$$\frac{\delta_{\text{local}} \vdash \text{ lEnv.top}() = o}{\vdash \{\text{true}\} \text{ this } \{\mathbf{ret} = \text{loc}(o)\}}$$

Rule 5 (Type Conversion).

$$\frac{\tau \preceq \tau_d(e)}{\vdash \{\text{true}\} \tau(e) \{\mathbf{ret} = \text{loc}(e) \wedge \tau_a = \tau\}}$$

$\tau_{\text{parent}} \preceq \tau_{\text{derived}}$ subtyping relations

1.3 Assignment Validation

Rule 6 (Store Assignment).

$$\frac{\vdash \{P\} e \{P \wedge \mathbf{ret} = v\}}{\vdash \{P[v/x]\} x := e; \{P\}}$$

where $x \in \text{dom}(s)$

Rule 7 (Heap Mutation).

$$\frac{\{P \wedge x \mapsto _ \} e \{P \wedge \mathbf{ret} = v\}}{\vdash \{P \wedge x \mapsto _ \} x := e; \{P \wedge x \mapsto v\}}$$

where $\text{loc}(x) \in \text{dom}(h)$

1.4 Foundational Declarations of Variables and Objects

Rule 8 (Primitive Declaration).

$$\frac{\tau \in \{\mathbf{int}, \mathbf{bool}\} \quad x \notin \text{fv}(P) \quad x \notin \text{dom}(s)}{\vdash \{P\} \tau x; \{\exists \nu \in \mathbb{V}_\tau. P \oplus (x \mapsto \nu)\}}$$

where $\mathbb{V}_\tau = \begin{cases} Z & \tau = \mathbf{int} \\ \{\mathbf{true}, \mathbf{false}\} & \tau = \mathbf{bool} \end{cases}$

Rule 9 (Reference Declaration (Phase I)).

$$\frac{x \notin \text{dom}(s)}{\vdash \{P\} \tau x; \{P \oplus \{x \mapsto (\tau_d, \tau_a, \perp)\}\}}$$

where $\tau_d = \tau_a = \tau, \perp : \text{null}$

Rule 10 (Object Instantiation (Phase II)).

$$\frac{\text{alloc}(h, \text{SecureZone}) = (a, h')}{\vdash \{P * G\} \text{new } C(); \{P \wedge \mathbf{ret} = a * G \oplus (a \mapsto (C, \text{init}(C)))\}}$$

$\text{init}(C) := \bigotimes_{f_i : \tau_i \in \text{fields}(C)} f_i \hookrightarrow \text{default}(\tau_i)$

Rule 11 (Reference Resolution (Phase III)).

$$\frac{s(x) = (\tau_d, \tau_a, \perp) \wedge C \preceq \tau_d \quad \{P * G\} \text{new } C(); \{P \wedge \mathbf{ret} = a * G'\}}{\vdash \{P\} x = \text{new } C(); \{P[s'(x) := (\tau_d, C, a)]\}}$$

2 Behavior Abstraction Layer: Method Declarations

Rule 12 (Virtual Method Verification).

$$\begin{array}{ll} \delta \vdash \{P_s\} \bar{s} \{Q_s[z/\mathbf{ret}]\} & (\text{Body verification}) \\ \delta \vdash \{P_s\} \{Q_s\} \Rightarrow \langle P_d \rangle \langle Q_d \rangle & (\text{Dynamic Dispatch}) \\ \hline \delta \vdash \text{virtual } C.m(\overline{D} \ \overline{x}) \bar{s} \vdash Sd \Rightarrow Ss & (\text{Virtual Method}) \end{array}$$

Rule 13 (Inherit Method Verification).

$$\begin{array}{ll} E <_1 F & (\text{Direct Inheritance}) \\ \delta \vdash \{P_{sF}\} \{Q_{sF}\} \Rightarrow \langle P_{dE} \rangle \langle Q_{dE} \rangle & (\text{Dynamic Dispatch}) \\ \hline \delta \vdash \text{virtual } C.m(\overline{D} \ \overline{x}) \bar{s} \vdash Sd & (\text{Inherit Method}) \end{array}$$

Rule 14 (Override Method Verification).

$$\begin{array}{ll} E <_1 F & (\text{Direct Inheritance}) \\ \delta \vdash \{P_s\} \bar{s} \{Q_s[z/\mathbf{ret}]\} & (\text{Body Verification}) \\ \delta \vdash \langle P_{dE} \rangle \langle Q_{dE} \rangle \Rightarrow \langle P_{dF} \rangle \langle Q_{dF} \rangle & (\text{Behavior Subtyping}) \\ \delta \vdash \{P_{sE}\} \{Q_{sF}\} \Rightarrow \langle P_{dE} \rangle \langle Q_{dE} \rangle & (\text{Dynamic Dispatch}) \\ \hline \delta \vdash \text{override } C.m(\overline{D} \ \overline{x}) \bar{s} \vdash Sd \Rightarrow Ss & (\text{Override Method}) \end{array}$$

Rule 15 (Class Declaration).

$$\frac{\forall_{M_i \in \overline{M}}. \delta \vdash M_i \text{ in } C}{\delta \vdash \mathbf{class } C \text{ extends } D \ \{\overline{\tau} \ \overline{x}; \overline{M}\}}$$

3 Concurrency Extension Layer

3.1 Axioms from Concurrent Separation Logic

Axiom 9 (Complete Program Initialization).

$$\frac{\{P\} \text{init} \ \{*\}_{1 \leq i \leq m} RI_{r_i} * P \quad \{P\} C_1 || \dots || C_n \{Q\}}{\{P\} \text{init}; \mathbf{resource} \ \overline{r}; C_1 || \dots || C_n \ \{*\}_{1 \leq i \leq m} RI_{r_i} * Q \}}$$

Axiom 10 (Parallel Composition).

$$\frac{\forall_{1 \leq i \leq n} \{P_i\} C_i \{Q_i\}}{\{*\}_{1 \leq i \leq n} P_i \ C_1 || \dots || C_n \ \{*\}_{1 \leq i \leq n} Q_i \}}$$

Axiom 11 (Critical Region).

$$\frac{\{P * RI_r \wedge B\} C \{Q * RI_r\}}{\{P\} \mathbf{with } r \text{ when } B \text{ do } C \{Q\}}$$

3.2 Program Equivalence and Method Invocation

Rule 16 (Method Invocation: Body).

$$\begin{array}{lcl}
\vdash \{P\}\{Q\} & & (\text{Local Environment}) \\
\delta \vdash \langle P_d \rangle \langle Q_d \rangle & & (\text{Specification Retrieval}) \\
\{P\}\{Q\} \Rightarrow \langle P_d[\bar{y}/\bar{x}] \rangle \langle Q_d[\bar{y}/\bar{x}] \rangle & & (\text{Control Flow Trans}) \\
\hline
\{P * P_d[\bar{y}/\bar{x}]\} o.m(\bar{y}) \{Q \wedge \mathbf{ret} = z * Q_d[\bar{y}/\bar{x}]\} & & (\text{Invocation})
\end{array}$$

Rule 17 (Logical Twin with Delay Abstraction).

$$\begin{array}{l}
\Delta_m = \text{calc}(m(\bar{y})) \\
\delta \vdash \{P * P_d\} o.m(\bar{x}) \{Q \wedge \mathbf{ret} = v * Q_d\} \\
\vdash \forall_{o_i \text{ in } \text{requires}(o.m)} \pi_{o_i} \mapsto v_{o_i} \\
\hline
\{P * t = t_0 *_{\forall_i} \text{upd}(\pi_{o_i})\} \# \Delta_m \{Q \wedge \mathbf{ret} = v * t = t_0 + \Delta_m *_{\forall_i} \text{rst}(\pi_{o_i})\}
\end{array}$$

$$\begin{array}{l}
\text{upd}(\pi_o) = \begin{cases} \pi_o \mapsto 1 & \text{if synchronized} \\ \pi_o \mapsto v + (1 - v)/2 & \text{if non-synchronized} \end{cases} \\
\text{rst}(\pi_o) = \begin{cases} \pi_o \mapsto 0 & \text{if synchronized} \\ \pi_o \mapsto 2 * v - 1 & \text{if non-synchronized} \end{cases}
\end{array}$$

3.3 Scheduling Rules and Operations

Rule 18 (Parallel Local Operations).

$$\frac{\{P_1\} C_1 \{Q_1\} \quad \{P_2\} C_2 \{Q_2\}}{\{P_1 * P_2\} C_1 \parallel C_2 \{Q_1 * Q_2\}}$$

Rule 19 (Parallel Method Call with Local Operation).

$$\frac{\{P_1\} C_1 \{Q_1\} \quad \{P_2 * G\} C_2 \{Q_2 * G'\}}{\{P_1 * P_2 * G\} C_1 \parallel C_2 \{Q_1 * Q_2 * G'\}}$$

Rule 20 (Parallel Calling Distinct Objects).

$$\frac{\{P_1 * G_1\} C_1 \{Q_1 * G'_1\} \quad \{P_2 * G_2\} C_2 \{Q_2 * G'_2\}}{\{P_1 * P_2 * G_1 * G_2\} C_1 \parallel C_2 \{Q_1 * Q_2 * G'_1 * G'_2\}}$$

Rule 21 (Parallel Synchronized Method Calls).

$$\frac{\{P_1 * G\} C_1 \{Q_1 * G'\} \quad \{P_2 * G'\} C_2 \{Q_2 * G''\}}{\{P_1 * P_2 * G\} C_1 \parallel C_2 \{Q_1 * Q_2 * G''\}}$$

Rule 22 (Parallel Non-Synchronized Method Calls).

$$\frac{\{P_1 * G_1\} C_1 \{Q_1 * G'_1\} \quad \{P_2 * G_2\} C_2 \{Q_2 * G'_2\}}{\{P_1 * P_2 * G_1 \wedge G_2\} C_1 \parallel C_2 \{Q_1 * Q_2 * G'_1 \wedge G'_2\}}$$

where $\delta \vdash C_1$ and C_2 are interference-free

Rule 23 (Parallel Guard Condition with Local Operation).

$$\frac{\{P_1\} C_1 \{Q_1\} \quad \{true\} @\langle guard \rangle \{Q_2\}}{\{P_1\} C_1 \parallel @\langle guard \rangle \{Q_1 * Q_2\}}$$

where $\llbracket guard \rrbracket = true$.

Rule 24 (Parallel Guard with Method Invocation).

$$\frac{\{P_1 * G\} C_1 \{Q_1 * G'\} \quad \{true\} @\langle guard \rangle \{Q_2\}}{\{P_1 * G * true\} C_1 \parallel @\langle guard \rangle \{Q_1 * G' * Q_2\}}$$

where $\llbracket guard \rrbracket = true$.

Rule 25 (General Parallelism: Time Elapse with Atomics).

$$\frac{\{P_1\} C_1 \{Q_1\} \quad \{P_2\} \# \Delta \{Q_2\}}{\{P_1 * P_2\} C_1 \parallel \# \Delta \{Q_1 * P_2\}}$$

Rule 26 (Time Elapse).

$$\begin{array}{l}
\{P_1 * now = t_1\} \# \Delta_1 \{Q_1 * now = t_1 + \Delta_1\} \\
\{P_2 * now = t_2\} \# \Delta_2 \{Q_2 * now = t_2 + \Delta_2\} \\
\hline
\{P_1 * P_2 * now = t\} \# \Delta_1 \parallel \# \Delta_2 \{Q_1 * P_2 * now = t_1 + \Delta_1\}
\end{array} \tag{1}$$

where $\max(t_1, t_2) \leq t \leq \min(t_1 + \Delta_1, t_2 + \Delta_2)$
 $\min(t_1 + \Delta_1, t_2 + \Delta_2) = t_1 + \Delta_1$