Entire Proof System of rCOE

May 11, 2025

1 Foundational Layer: Sequential Program Verification

1.1 Axioms from Hoare Logic and Separation Logic

Axiom 1 (**Skip**).
$$\{P\}\ \texttt{skip}\ \{P\}$$

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**).
$$\frac{\{P \wedge B\}\, C\, \{Q\} \quad \{P \wedge \neg B\}\, C'\, \{Q\}}{\{P\}\ \mathit{if}\ B\ \mathit{then}\ C\ \mathit{else}\ C'\ \{Q\}}$$

Axiom 5 (**Loop**).
$$\frac{\{P \wedge B\}\, C\, \{P\}}{\{P\}\ \mathit{while}\ B\ \mathit{do}\ C\ \{P \wedge \neg B\}}$$

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**).
$$\frac{\mathit{prim} \in \{\mathit{bool},\ \mathit{int}\}}{\vdash \{\mathit{true}\}\ \mathit{prim}\ \{\mathbf{ret} = [\![\mathit{prim}]\!]\}}$$

Rule 2 (**Reference Resolution**).
$$\frac{\mathit{ref} \neq \bot}{\vdash \{\mathit{true}\}\ \mathit{ref}\ \{\mathbf{ret} = \mathit{loc}(\mathit{ref})\}}$$

Rule 3 (**Null Safety**).
$$\vdash \{\mathit{true}\}\ \mathit{null}\ \{\mathbf{ret} = \emptyset\}$$

Rule 4 (**Contextual This**).
$$\frac{\delta_{\mathit{local}} \vdash \mathit{lEnv.top}() = o}{\vdash \{\mathit{true}\}\ \mathit{this}\ \{\mathbf{ret} = \mathit{loc}(o)\}}$$

Rule 5 (**Type Conversion**).
$$\frac{\tau \preccurlyeq \tau_d(e)}{\vdash \{\mathit{true}\}\ \tau(e)\ \{\mathbf{ret} = \mathit{loc}(e)\ \wedge\ \tau_a = \tau\}}$$

$\tau_{\mathit{parent}} \preccurlyeq \tau_{\mathit{derived}}$ *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 $x \in \mathit{dom}(s)$

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 $\mathit{loc}(x) \in \mathit{dom}(h)$

1.4 Foundational Declarations of Variables and Objects

Rule 8 (**Primitive Declaration**).

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

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

Rule 9 (**Reference Declaration (Phase I)**).

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

$$\text{where } \tau_\mathrm{d} = \tau_\mathrm{a} = \tau,\ \bot : \mathrm{null}$$

Rule 10 (**Object Instantiation (Phase II)**).

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

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

Rule 11 (**Reference Resolution (Phase III)**).

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

2 Behavior Abstraction Layer: Method Declarations

Rule 12 (**Virtual Method Verification**).

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

Rule 13 (**Inherit Method Verification**).

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

Rule 14 (**Override Method Verification**).

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

Rule 15 (**Class Declaration**).

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

3 Concurrency Extension Layer

3.1 Axioms from Concurrent Separation Logic

Axiom 9 (**Complete Program Initialization**).

$$\frac{\{P\}\ init\ \{*_{1 \le i \le m} RI_{r_i} * P\} \qquad \{P\}\ C_1 || \ldots || C_n\ \{Q\}}{\{P\}\ init; \mathbf{resource}\ \overline{r}; C_1 || \ldots || C_n\ \{*_{1 \le i \le m} RI_{r_i} * Q\}\}}$$

Axiom 10 (**Parallel Composition**).

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

Axiom 11 (**Critical Region**).

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

3.2 Program Equivalence and Method Invocation

Rule 16 (**Method Invocation: Body**).

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

Rule 17 (**Logical Twin with Delay Abstraction**).

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

$$upd(\pi_o) = \begin{cases}\pi_o \mapsto 1 & if\ synchronized\\ \pi_o \mapsto v + (1-v)/2 & if\ non-synchronized\end{cases}$$

$$rst(\pi_o) = \begin{cases}\pi_o \mapsto 0 & if\ synchronized\\ \pi_o \mapsto 2 * v - 1 & if\ non-synchronized\end{cases}$$

3.3 Scheduling Rules and Operations

Rule 18 (**Parallel Local Operations**).

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

Rule 19 (**Parallel Method Call with Local Operation**).

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

Rule 20 (**Parallel Calling Distinct Objects**).

$$\frac{\{P_1 * G_1\}\ C_1\ \{Q_1 * G_1'\} \qquad \{P_2 * G_2\}\ C_2\ \{Q_2 * G_2'\}}{\{P_1 * P_2 * G_1 * G_2\}\ C_1 \parallel C_2\ \{Q_1 * Q_2 * G_1' * G_2'\}}$$

Rule 21 (**Parallel Synchronized Method Calls**).

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

Rule 22 (**Parallel Non-Synchronized Method Calls**).

$$\frac{\{P_1 * G_1\}\ C_1\ \{Q_1 * G_1'\} \qquad \{P_2 * G_2\}C_2\{Q_2 * G_2'\}}{\{P_1 * P_2 * G_1 \wedge G_2\}\ C_1\ \parallel\ C_2\ \{Q_1 * Q_2 * G_1' \wedge G_2'\}}$$

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

Rule 23 (**Parallel Guard Condition with Local Operation**).

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

where $[\![guard]\!] = true.$

Rule 24 (**Parallel Guard with Method Invocation**).

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

where $[\![guard]\!] = true.$

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

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

Rule 26 (**Time Elapse**).

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

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