Release-guarantee reasoning under weak memory consistency

Viktor Vafeiadis

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

8 September 2026

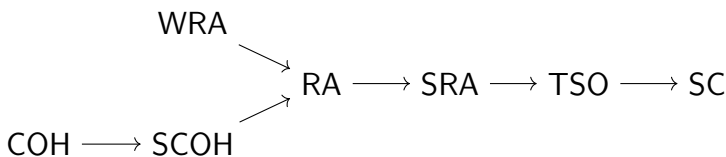
RGSep

My PhD (≈2007)

- RGSep = rely-guarantee + separation logic
- RGSep initially developed for SC
- Soundness proof was non-trivial at the time

New results (OOPSLA'26)

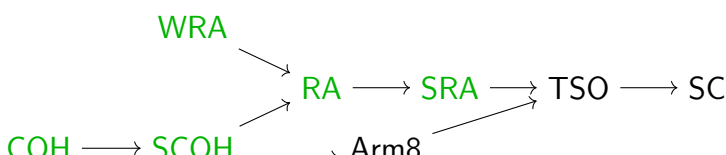
- RGSep is sound under RA (non-trivial proof)
- RGSep is unsound under WRA, SCOH



Independent Reads of Independent Writes (IRIW)

Initially, $x = y = 0$.

$$x := 1 \;\Big\|\; \begin{array}{ll} a := x & \text{// reads 1} \\ b := y & \text{// reads 0} \end{array} \;\Big\|\; \begin{array}{ll} c := y & \text{// reads 1} \\ d := x & \text{// reads 0} \end{array} \;\Big\|\; y := 1$$



Hoare logic

$\{P\}\ C\ \{Q\}$

P: precondition
C: program
Q: postcondition

$$\frac{}{\{P\}\ \textbf{skip}\ \{P\}} \qquad \frac{}{\{P[e/x]\}\ x := e\ \{P\}}$$

$$\frac{\{P\}\ C_1\ \{R\} \quad \{R\}\ C_2\ \{Q\}}{\{P\}\ C_1; C_2\ \{Q\}} \qquad \frac{\{B \wedge P\}\ C_1\ \{Q\} \quad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \textbf{if}\ B\ \textbf{then}\ C_1\ \textbf{else}\ C_2\ \{Q\}}$$

$$\frac{\{P \wedge B\}\ C\ \{P\}}{\{P\}\ \textbf{while}\ B\ \textbf{do}\ C\ \{P \wedge \neg B\}} \qquad \frac{P_1 \Rightarrow P_2 \quad \{P_2\}\ C\ \{Q_2\} \quad Q_2 \Rightarrow Q_1}{\{P_1\}\ c\ \{Q_1\}}$$

C. A. R. Hoare. An axiomatic basis for computer programming. CACM 12(10): 576-580 (1969)

Owicki-Gries method

Hoare logic & $\dfrac{\{P_1\}\ C_1\ \{Q_1\} \qquad \{P_2\}\ C_2\ \{Q_2\} \quad \text{the two proofs are } \textit{non-interfering}}{\{P_1 \wedge P_2\}\ C_1 \parallel C_2\ \{Q_1 \wedge Q_2\}}$

Non-interference

$R \wedge P \Rightarrow R[u/x]$ for every:

- assertion R in one proof outline
- assignment $x := u$ with precondition P in the other proof outline

$$\begin{array}{c} \vdots \\ \{P\} \\ x := u \\ \vdots \end{array} \Bigg\| \begin{array}{c} \vdots \\ \{R\} \\ \vdots \end{array}$$

S. S. Owicki, D. Gries. An axiomatic proof technique for parallel programs I. Acta Informatica 6: 319-340 (1976)

Rely-guarantee

- Compositional version of Owicki-Gries
- Introduce two components in specifications
- Rely R: interference caused by environment
- Guarantee G: interference caused by the program

$$\frac{\begin{array}{ll} R_1; G_1 \vdash \{P_1\}\ C_1\ \{Q_1\} & R \cup G_2 \subseteq R_1 \\ R_2; G_2 \vdash \{P_2\}\ C_2\ \{Q_2\} & R \cup G_1 \subseteq R_2 \end{array}}{R; G_1 \cup G_2 \vdash \{P_1 \wedge P_2\}\ C_1 \parallel C_2\ \{Q_1 \wedge Q_2\}}$$

Message passing

$$
\begin{array}{c}
\{x = 0 \wedge y = 0\} \\
\begin{array}{l|l}
\{\top\} & \{y = 1 \Rightarrow x = 1\} \\
x := 1 & a := y \\
\{x = 1\} & \{a = 1 \Rightarrow x = 1\} \\
y := 1 & b := x \\
\{x = 1\} & \{a = 1 \Rightarrow b = 1\}
\end{array} \\
\{a = 1 \Rightarrow b = 1\}
\end{array}
$$

Therefore, OG/RG is unsound under SCOH.

Coherence test

OG/RG can establish that:

$$
\begin{array}{c}
\{\top\} \\
\{\top\} \quad \| \quad \{\top\} \\
x := 1 \quad \| \quad x := 2 \\
\{x = 1 \vee x = 2\} \quad \| \quad \{x = 2 \vee x = 1\} \\
a := x \quad \| \quad b := x \\
\{a = 2 \Rightarrow x = 2\} \quad \| \quad \{b = 1 \Rightarrow x = 1\} \\
\{\neg(a = 2 \wedge b = 1)\}
\end{array}
$$

Therefore, OG/RG is unsound under WRA/CC.

Store buffering

OG/RG can establish that:

$$
\begin{array}{c}
\{a \neq 0\} \\
\begin{array}{l|l}
\{a \neq 0\} & \{\top\} \\
x := 1 & y := 1 \\
\{x \neq 0\} & \{y \neq 0\} \\
a := y & b := x \\
\{x \neq 0\} & \{y \neq 0 \wedge (a \neq 0 \vee b = x)\}
\end{array} \\
\{a \neq 0 \vee b \neq 0\}
\end{array}
$$

So OG/RG is unsound under TSO!

Store buffering

OG/RG can establish that:

$$
\begin{array}{c}
\{a \neq 0\} \\
\begin{array}{l|l}
\{a \neq 0\} & \{\top\} \\
x := 1 & y := 1 \\
\{x \neq 0\} & \{y \neq 0\} \\
a := y & b := x \\
\{x \neq 0\} & \{y \neq 0 \wedge (a \neq 0 \vee b = x)\}
\end{array} \\
\{a \neq 0 \vee b \neq 0\}
\end{array}
$$

So OG/RG is unsound under TSO!

Store buffering

OG/RG can establish that:

$$
\begin{array}{c}
\{a \neq 0\} \\
\begin{array}{l|l}
\{a \neq 0\} & \{\top\} \\
x := 1 & y := 1 \\
\{x \neq 0\} & \{y \neq 0\} \\
a := y & b := x \\
\{x \neq 0\} & \{y \neq 0 \wedge (a \neq 0 \vee b = x)\}
\end{array} \\
\{a \neq 0 \vee b \neq 0\}
\end{array}
$$

So OG/RG is unsound under TSO!

How come RGSep is sound?

RGSep places a key restriction on RG:

- Distinguish local vars (registers) and shared vars (memory locations).
- Specs cannot mention the local variables of the environment.

$$
\begin{array}{c}
\{a \neq 0\} \\
\begin{array}{l|l}
\{a \neq 0\} & \{\top\} \\
x := 1 & y := 1 \\
\{x \neq 0\} & \{y \neq 0\} \\
a := y & b := x \\
\{x \neq 0\} & \{y \neq 0 \wedge (a \neq 0 \vee b = x)\}
\end{array} \\
\{a \neq 0 \vee b \neq 0\}
\end{array}
$$

Release-acquire model (operationally)

State representation:

- Memory M is a set of messages of the form $\langle loc : val @ time, view \rangle$
- Each thread has a view $V \subseteq M$.

Transitions:

- A thread can increase its view: $V' = V \cup \{m\}$ provided that $m.\text{view} \subseteq V$.
- Read: return the value of the maximum message in V.
- Write: add a message to M and V with
  - message view $= V$, and
  - timestamp $>$ the timestamps of all same-location messages in V.

What is the meaning of RG specs? (1)

C satisfies (P, R, G, Q) iff for all $n \in \mathbb{N}$, all M, all $V \subseteq M$, if $V \models P$, then $\mathit{Safe}(n, C, M, V, R, G, Q)$.

$\mathit{Safe}(0, C, M, V, R, G, Q)$ holds always.
$\mathit{Safe}(n+1, C, M, V, R, G, Q)$ holds if the following hold:

- If $C = \textbf{skip}$ then $V \models Q$.
- $\langle C, M, V \rangle \not\to \textbf{abort}$.
- If m satisfies R, then $\mathit{Safe}(n, C, M \cup \{m\}, V, R, G, Q)$.
- Whenever $\langle C, M, V \rangle \to \langle C', M', V' \rangle$, then $(M' \setminus M)$ satisfies G and $\mathit{Safe}(n, C', M', V', R, G, Q)$.

Annotations

Annotation map $\mathcal{A}$:
Annotate each message with precondition P, rely R, guarantee G, such that

- the message view satisfies the precondition P,
- the precondition P is stable under R, and
- the message effect along with P imply G.

Key invariant (Message annotation compatibility):
For all concurrent messages $m_1, m_2 \in M$ (i.e. $m_1 \notin m_2.\text{view}$ and $m_2 \notin m_1.\text{view}$), $\mathcal{A}(m_1).\text{guar} \subseteq \mathcal{A}(m_2).\text{rely}$,

Consequence: Annotated message precondition holds at all larger views, and the message effect remains in its guarantee at all larger views.

What is the meaning of RG specs? (2)

C satisfies (P, R, G, Q) holds iff for all n, M, V, if $V \subseteq M$ and $V \models P$, then there exists $\mathcal{A}$, $\mathit{Safe}(n, C, M, V, \mathcal{A}, R, G, Q)$.

$\mathit{Safe}(0, C, M, V, \mathcal{A}, R, G, Q)$ holds always.
$\mathit{Safe}(n+1, C, M, V, \mathcal{A}, R, G, Q)$ holds if the following hold:

- If $C = \textbf{skip}$, then $V \models Q$.
- $\langle C, M, V\rangle \not\to \textbf{abort}$.
- If $\alpha.\mathrm{guar} \subseteq R$ and $G \subseteq \alpha.\mathrm{rely}$, then $\mathit{Safe}(n, C, M \uplus \{m\}, V, \mathcal{A}[m \mapsto \alpha], R, G, Q)$.
- Whenever $\langle C, M, V\rangle \to \langle C', M, V'\rangle$, then $\mathit{Safe}(n, C', M, V', \mathcal{A}, R, G, Q)$
- Whenever $\langle C, M, V\rangle \to \langle C', M \uplus \{m\}, V'\rangle$, then there exists α s.t. $R \subseteq \alpha.\mathrm{rely}$ and $\alpha.\mathrm{guar} \subseteq G$ and $\mathit{Safe}(n, C', M \uplus \{m\}, V', \mathcal{A}[m \mapsto \alpha], R, G, Q)$.

Auxiliary variable elimination is generally unsound?

$$\frac{C \text{ sat } (P, R, G, Q) \quad X \notin \text{vars}(P, R, G, Q)}{\mathit{erase}_X(C) \text{ sat } (P, R, G, Q)}$$

- Unsurprising, since auxiliary variables can encode SC.
- Howver,limited forms of auxiliary variables are sound.
- Accesses to the same auxiliary variable need to be related by happens-before.

What's more? What's next?

- RGSep supports local reasoning and ownership transfer.
  - local $\rightarrow$ shared
  - shared $\rightarrow$ local
- RGSep linearizability proofs using auxiliary variables.
- Logical principles for other memory models.
- Completeness.