Owicki-Gries for Weak Memory Models

Ori Lahav

Max Planck Institute for Software Systems (MPI-SWS)

Joint work in progress with Viktor Vafeiadis

Weak Memory Models

- Sequential consistency (a.k.a. "interleaving semantics") is the standard memory model for reasoning about concurrency.
- However, in the presence of races, SC is invalidated by hardware implementations and compiler optimizations.

Example (Store Buffering)

Initially $x = y = 0$.

$$\begin{array}{l||l} x := 1 & y := 1 \\ a := y & b := x \end{array}$$

This can return $a = b = 0$ (observed on x86/Power/ARM).

- Weak memory models provide formal sound semantics for realistic high-performance concurrency.

Our Work

Goals:

- Verify concurrent programs under WM.
- Investigate what program logics are sound under WM.

Contributions:

- We show that the most basic technique, Owicki-Gries, is unsound for WM *(even without ghost variables and atomic blocks)*.
- We identify a simple weakening of OG that is sound for the Release/Acquire memory model.
- We demonstrate the usefulness of this simple program logic.

C11 Memory Model

- Introduced in the recent standard for C and C++ (ISO/IEC 14882:2011, ISO/IEC 9899:2011).
- Formalized in [Batty et al., POPL'11].

- Memory accesses are labeled with memory orders (e.g., SC, Release/Acquire, Relaxed, Non-Atomic).

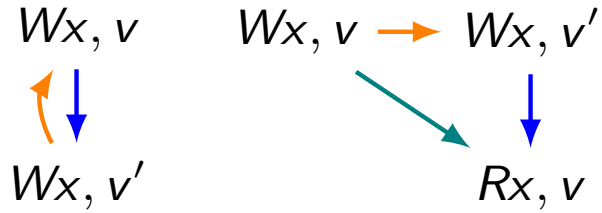
In this work we study the "Release/Acquire" fragment of C11.
(exhibits good balance between efficiency and sanity)

Release/Acquire Memory Model

- Each program is associated with a set of graphs (called: *executions*).
- An execution is *consistent* if it can be augmented with relations:
  - *reads-from*: associates each read with a corresponding write
  - *memory-order*: total order on all writes to the same location

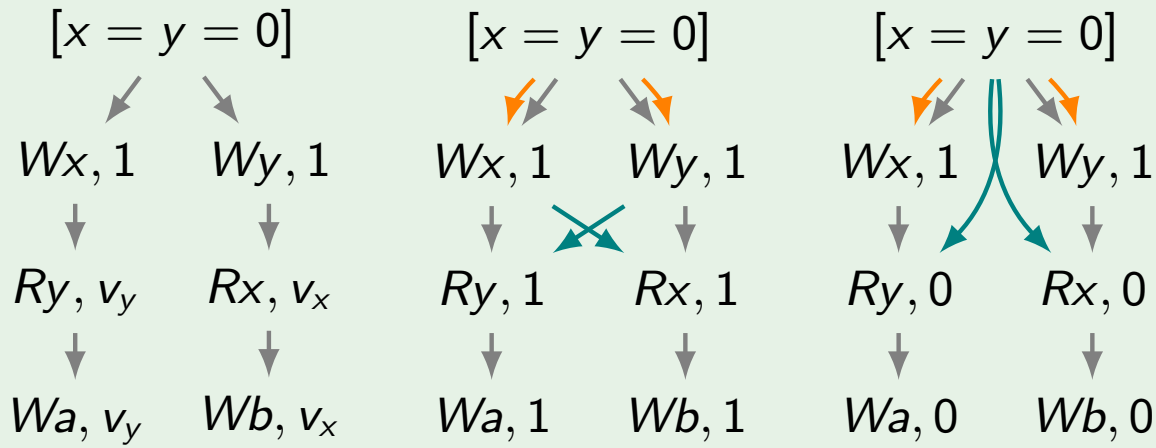
  such that *happens-before* $= (\textit{program-order} \cup \textit{reads-from})^*$ is acyclic

  and none of the following occurs:



Example (Store Buffering)

$$\begin{array}{c} x = y = 0 \\ x := 1 \;\|\; y := 1 \\ a := y \;\|\; b := x \end{array}$$



Owicki-Gries Method (1976)

OG = Hoare logic + rule for parallel composition

$$\frac{\{P_1\}\, c_1\, \{Q_1\} \qquad \{P_2\}\, c_2\, \{Q_2\} \qquad \{P_1\}\, c_1\, \{Q_1\} \text{ and } \{P_2\}\, c_2\, \{Q_2\} \text{ are } \textit{non-interfering}}{\{P_1 \wedge P_2\}\, c_1 \parallel c_2\, \{Q_1 \wedge Q_2\}}$$

Non-interference

$R \wedge P \vdash R\{u/x\}$ for every:

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

Owicki-Gries Method (1976)

OG = Hoare logic + rule for parallel composition

$$\frac{\{P_1\}\, c_1\, \{Q_1\} \qquad \{P_2\}\, c_2\, \{Q_2\} \qquad \{P_1\}\, c_1\, \{Q_1\} \text{ and } \{P_2\}\, c_2\, \{Q_2\} \text{ are } \textit{non-interfering}}{\{P_1 \wedge P_2\}\, c_1 \parallel c_2\, \{Q_1 \wedge Q_2\}}$$

Non-interference

$R \wedge P \vdash R\{u/x\}$ for every:

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

non-interference of ~~executions~~ proofs

Store Buffering Example

$$\{x = 0 \wedge b = 2\}$$

$$\begin{array}{l|l} x := 1 & y := 1 \\ a := y & b := x \end{array}$$

$$\{a = 1 \vee b = 1\}$$

Store Buffering Example

$$\{x = 0 \wedge b = 2\}$$

$$\begin{array}{l|l}
\{\top\} & \{b = 2, x \neq 2\} \\
x := 1 & y := 1 \\
\{x = 1\} & \{y = 1, x \neq 2\} \\
a := y & b := x \\
\{x = 1 \wedge (y = 1 \rightarrow a = 1 \vee b = 1 \vee b = 2)\} & \{y = 1, b \neq 2\}
\end{array}$$

$$\{a = 1 \vee b = 1\}$$

Store Buffering Example

$$\{x = 0 \wedge b = 2\}$$

$\{\top\}$	$\Vert$	$\{b = 2, x \neq 2\}$
$x := 1$	$\Vert$	$y := 1$
$\{x = 1\}$	$\Vert$	$\{y = 1, x \neq 2\}$
$a := y$	$\Vert$	$b := x$
$\{x = 1 \wedge (y = 1 \rightarrow a = 1 \vee b = 1 \vee b = 2)\}$	$\Vert$	$\{y = 1, b \neq 2\}$

$$\{a = 1 \vee b = 1\}$$

Store Buffering Example

$$\{x = 0 \wedge b = 2\}$$

$\{\top\}$	$\Vert$	$\{b = 2, x \neq 2\}$
$x := 1$	$\Vert$	$y := 1$
$\{x = 1\}$	$\Vert$	$\{y = 1, x \neq 2\}$
$a := y$	$\Vert$	$b := x$
$\{x = 1 \wedge (y = 1 \rightarrow a = 1 \vee b = 1 \vee b = 2)\}$	$\Vert$	$\{y = 1, b \neq 2\}$

$$\{a = 1 \vee b = 1\}$$

$\Longrightarrow$ Unsoundness for weak memory!

Stronger Non-interference Condition

$$\frac{\{P_1\}\, c_1\, \{Q_1\} \qquad \{P_2\}\, c_2\, \{Q_2\} \qquad \{P_1\}\, c_1\, \{Q_1\} \text{ and } \{P_2\}\, c_2\, \{Q_2\} \text{ are non-interfering}}{\{P_1 \wedge P_2\}\, c_1 \parallel c_2\, \{Q_1 \wedge Q_2\}}$$

Non-interference

$R \wedge P \vdash R\{v/x\}$ for every:

- assertion R in the proof outline of one thread
- assignment $x := u$ with precondition P in the proof outline of the other thread
- value v such that $P \wedge R' \not\vdash u \neq v$ for some assertion R' above R

Example: Message Passing

$$\{y = 0\}$$

$$\begin{array}{l} x := 42 \\ \\ y := 1 \end{array} \quad \Big\| \quad \begin{array}{l} \texttt{while } y = 0 \\ \\ \quad \texttt{skip} \\ \\ \\ a := x \end{array}$$

$$\{a = 42\}$$

Example: Message Passing

$$\{y = 0\}$$

$$\begin{array}{l} \{\top\} \\ x := 42 \\ \{x = 42\} \\ y := 1 \\ \{\top\} \end{array} \quad \Big\| \quad \begin{array}{l} \{y \neq 0 \to x = 42\} \\ \texttt{while } y = 0 \\ \quad \{y \neq 0 \to x = 42\} \\ \quad \texttt{skip} \\ \quad \{y \neq 0 \to x = 42\} \\ \{x = 42\} \\ a := x \\ \{a = 42\} \end{array}$$

$$\{a = 42\}$$

Example: Coherence

$$\{x = a = c = 0\}$$

$$x := 1 \quad \Big\| \quad x := 2 \quad \Big\| \quad \begin{array}{l} a := x \\ b := x \end{array} \quad \Big\| \quad \begin{array}{l} c := x \\ d := x \end{array}$$

$$\{a = 1 \wedge b = 2 \wedge c = 2 \rightarrow d \neq 1\}$$

Example: Coherence

$$\{x = a = c = 0\}$$

$$\begin{array}{l|l|l|l}
\{x \neq 1 \wedge a \neq 1\} & \{x \neq 2 \wedge c \neq 2\} & \{\top\} & \{\top\} \\
x := 1 & x := 2 & a := x & c := x \\
\{\top\} & \{\top\} & \{\top\} & \{\top\} \\
 & & b := x & d := x \\
 & & \{a = 1 \wedge b = 2 \rightarrow x = 2\} & \{c = 2 \wedge d = 1 \rightarrow x = 1\}
\end{array}$$

$$\{a = 1 \wedge b = 2 \wedge c = 2 \rightarrow d \neq 1\}$$

Soundness Proof

Challenges in a weak memory setting:

- No intuitive operational semantics
- No notion of global state

Main proof steps:

- Introduce a notion of a local state that is visible at a given edge of the execution.
- Study properties of visibility under the release/acquire model.
- Show that edges of consistent executions can be annotated with the assertions from the Hoare proof, such that every state that is visible at some edge satisfies its annotation.

Related Works

- C11 formalizations: Sewell et al. (POPL'11,POPL'12, PLDI'12).
- Separation logic based approaches: Relaxed Separation Logic, Vafeiadis,Narayan (OOPSLA'13); GPS, Turon,Vafeiadis,Dreyer (OOPSLA'14).
- Other program logics: Rely/guarantee for TSO, Ridge (VSTTE'10); Verifying TSO programs, Jacobs (2014); Coherent Causal Memory, Cohen (coRR 2014).

Further work:

- Study other realistic examples (e.g., RCU)
- Support fences
- Support ghost variables
- Completeness?
- Investigate rely/guarantee

Related Works

- C11 formalizations: Sewell et al. (POPL'11,POPL'12, PLDI'12).
- Separation logic based approaches: Relaxed Separation Logic, Vafeiadis,Narayan (OOPSLA'13); GPS, Turon,Vafeiadis,Dreyer (OOPSLA'14).
- Other program logics: Rely/guarantee for TSO, Ridge (VSTTE'10); Verifying TSO programs, Jacobs (2014); Coherent Causal Memory, Cohen (coRR 2014).

Further work:

- Study other realistic examples (e.g., RCU)
- Support fences
- Support ghost variables
- Completeness?
- Investigate rely/guarantee

Thank you!

Backup Slide: Rely/Guarantee Presentation of OG

$$\frac{P \vdash Q}{\langle P, \texttt{skip}, Q, \{P, Q\}, \emptyset\rangle} \qquad \frac{P \vdash Q\{u/x\}}{\langle P, x := u, Q, \{P, Q\}, \{\langle P, x := u\rangle\}\rangle}$$

$$\frac{\langle P, c_1, R, \mathcal{A}_1, \mathcal{B}_1\rangle \qquad \langle R, c_2, Q, \mathcal{A}_2, \mathcal{B}_2\rangle}{\langle P, c_1; c_2, Q, \mathcal{A}_1 \cup \mathcal{A}_2, \mathcal{B}_1 \cup \mathcal{B}_2\rangle}$$

$$\frac{\begin{array}{c}\langle P_1, c_1, Q_1, \mathcal{A}_1, \mathcal{B}_1\rangle \qquad \langle P_2, c_2, Q_2, \mathcal{A}_2, \mathcal{B}_2\rangle \\ P \vdash P_1 \wedge P_2 \qquad Q_1 \wedge Q_2 \vdash Q \\ \langle P_1, c_1, Q_1, \mathcal{A}_1, \mathcal{B}_1\rangle \text{ and } \langle P_2, c_2, Q_2, \mathcal{A}_2, \mathcal{B}_2\rangle \text{ are non-interfering}\end{array}}{\langle P, c_1 \parallel c_2, Q, \mathcal{A}_1 \cup \mathcal{A}_2 \cup \{P, Q\}, \mathcal{B}_1 \cup \mathcal{B}_2\rangle}$$

Non-interference

$\langle P_1, c_1, Q_1, \mathcal{A}_1, \mathcal{B}_1\rangle$ and $\langle P_2, c_2, Q_2, \mathcal{A}_2, \mathcal{B}_2\rangle$ are non-interfering if $R \wedge P \vdash R\{u/x\}$ for every ($R \in \mathcal{A}_1$ and $\langle P, x := u\rangle \in \mathcal{B}_2$) or ($R \in \mathcal{A}_2$ and $\langle P, x := u\rangle \in \mathcal{B}_1$).