C/C++ Concurrency and Program Transformations

Recap: Shared Memory Accesses

Non-atomic accesses: Read (R), Write (W)

Atomic accesses = operation + memory order

Operations:

- Read (R)
- Write (W)
- Atomic update (U)
- Fence (F)

Memory orders:

- Relaxed (rlx)
- Release (rel)
- Acquire (acq)
- Acquire-Release (acq_rel)
- Sequentially consistent (sc)

Example:

- X.load(memory order)
- X.store(val, memory order)
- X.CAS(oldval, nwval, success mem order)
- atomic_thread_fence(memory order)

Recap: Access Types

Non-atomic read. e.g. t = X;

- R_{na}

Non-atomic write. e.g. X := v;

- W_{na}

Atomic read. e.g. X.load(memory order)

- R_{rlx}, R_{acq}, R_{sc}

Atomic write. e.g. X.store(val, memory order)

- W_{rlx}, W_{rel}, W_{sc}

Atomic update. e.g. X.CAS(oldval, nwval, success mem order)

- U_{rlx}, U_{rel}, U_{acq}, U_{acq_rel}, U_{sc}

Fence. e.g. atomic_thread_fence(memory order)

- F_{rel}, F_{acq}, F_{acq_rel}, F_{sc}

Axioms

Happens-before (hb) is irreflexive. (irrHB)

Execution is coherent.

- rf; hb is irreflexive (CohHBRF)
- mo; rf; hb is irreflexive. (Coh-RW)
- mo; hb is irreflexive. (Coh-WW)
- mo; hb; rf^{-1} is irreflexive. (Coh-WR)
- mo; rf; hb; rf^{-1} is irreflexive. (Coh-RR)

$[\mathsf{U}] \cap (\mathsf{fr}; \mathsf{mo}) = \emptyset$ (Atomicity)

$(\mathsf{hb} \cup \mathsf{mo}) \cap ([SC] \times [SC]) \subseteq \mathsf{sc}$ (ConsSC)

$$\forall a, b.\ (a, b) \in \mathsf{rf}; [\mathsf{RU}_{\mathsf{sc}}] \implies \mathsf{sc}_{\mathsf{imm}}(a, b) \vee a \notin \mathsf{SC} \wedge \nexists x.\ \mathsf{hbloc}(a, x) \wedge x \in \mathsf{WU}_{\mathsf{sc}} \wedge \mathsf{sc}(x, b)$$

(SCread)

Challenges

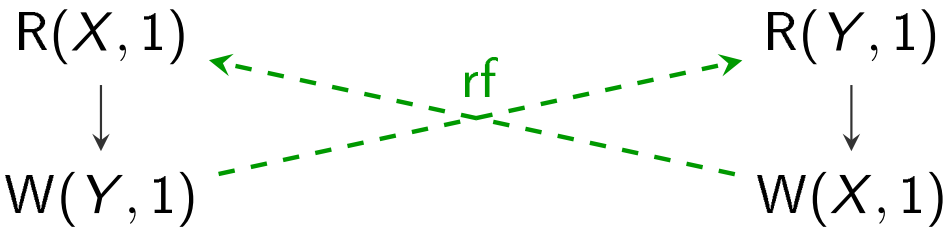
$$\begin{array}{l} a = X; \\ \text{if}(a == 1) \\ \qquad Y = 1; \end{array} \quad \Bigg\| \quad \begin{array}{l} b = Y; \\ \text{if}(b == 1) \\ \qquad X = 1; \end{array}$$

(A) $a = b = 1$ is not possible.

$$\begin{array}{l} a = X; \\ Y = 1; \end{array} \quad \Bigg\| \quad \begin{array}{l} b = Y; \\ \text{if}(b == 1) \\ \qquad X = 1; \end{array}$$

(B) $a = b = 1$ is possible.

Two programs share an execution.



Causality cycle: $(\text{po} \cup \text{rf})$.

Axioms

Causality cycle: $(\mathsf{po} \cup \mathsf{rf})$.

Allow causality cycle:

- Justifies outcome in (A) which is not possible.
- Violates data-race-freedom guarantee.
- Justitifes (B) outcome.
- Allows read-write reordering.

Disallow causality cycle:

- Restrict undesired outcome in (A).
- Provides data-race-freedom guarantee.
- Restricts (B) outcome.
- Restrict read-write reordering in compiler and architectures.

Tradeoff: Data-race-freedom guarantee vs efficient compilation.

Compilation.

Outcome of execution:

$$O(X) \triangleq \{(e.\mathsf{Loc}, e.\mathsf{wval}) \mid e \in X.\mathsf{E} \cap \mathsf{WU}.\ [\{e\}]; X.\mathsf{mo} = \emptyset\}$$

Outcome of peogram P:

$$O(P) = \{O(X) \mid X \in [[P]]\}$$

Given a compilation

$$P_{\mathsf{C11}} \Rightarrow P_1 \Rightarrow P_2 \Rightarrow \cdots \Rightarrow P_{\mathsf{arch}}$$

Correctness requires:

$$O(P_{\mathsf{C11}}) \supseteq O(P_1) \supseteq O(P_2) \supseteq \cdots \supseteq O(P_{\mathsf{arch}})$$

Correct Compilation

For each consistent execution of the target program there exists a corresponding consistent execution of the source program with same outcome.

$$X \in [\![\mathbb{P}_{\mathsf{tgt}}]\!].\ \exists X' \in [\![\mathbb{P}_{\mathsf{src}}]\!].\ O(X) = O(X')$$

Transformations

Redundant access elimination such as

- $Y_{\mathsf{rlx}} = v; Y_{\mathsf{rlx}} = v'; \leadsto Y_{\mathsf{rlx}} = v';$ (OW)
- $Y_{\mathsf{rlx}} = v; a = Y_{\mathsf{rlx}}; \leadsto Y_{\mathsf{rlx}} = v; a = v;$ (RAW)
- $a = Y_{\mathsf{rlx}}; a = Y_{\mathsf{rlx}}; \leadsto a = Y_{\mathsf{rlx}};$ (RAR)

Transformations

Redundant access elimination such as

- $Y_{\mathsf{rlx}} = v; Y_{\mathsf{rlx}} = v'; \rightsquigarrow Y_{\mathsf{rlx}} = v';$ (OW)
- $Y_{\mathsf{rlx}} = v; a = Y_{\mathsf{rlx}}; \rightsquigarrow Y_{\mathsf{rlx}} = v; a = v;$ (RAW)
- $a = Y_{\mathsf{rlx}}; a = Y_{\mathsf{rlx}}; \rightsquigarrow a = Y_{\mathsf{rlx}};$ (RAR)

Fence Insertion such as

- $C_1; C_2 \rightsquigarrow C_1; \mathsf{F}_{\{\mathsf{rel},\mathsf{acq},\mathsf{sc}\}}; C_2$

Transformations

Redundant access elimination such as

- $Y_{\mathsf{rlx}} = v; Y_{\mathsf{rlx}} = v'; \rightsquigarrow Y_{\mathsf{rlx}} = v';$ (OW)
- $Y_{\mathsf{rlx}} = v; a = Y_{\mathsf{rlx}}; \rightsquigarrow Y_{\mathsf{rlx}} = v; a = v;$ (RAW)
- $a = Y_{\mathsf{rlx}}; a = Y_{\mathsf{rlx}}; \rightsquigarrow a = Y_{\mathsf{rlx}};$ (RAR)

Fence Insertion such as

- $C_1; C_2 \rightsquigarrow C_1; \mathsf{F}_{\{\mathsf{rel},\mathsf{acq},\mathsf{sc}\}}; C_2$

Access strengthening such as

- $Y_{\mathsf{rlx}} = v \rightsquigarrow Y_{\mathsf{rel}} = v$
- $a = Y_{\mathsf{rlx}} \rightsquigarrow a = Y_{\mathsf{acq}}$

Transformations

Redundant access elimination such as

- $Y_{\mathsf{rlx}} = v; Y_{\mathsf{rlx}} = v'; \leadsto Y_{\mathsf{rlx}} = v';$ (OW)
- $Y_{\mathsf{rlx}} = v; a = Y_{\mathsf{rlx}}; \leadsto Y_{\mathsf{rlx}} = v; a = v;$ (RAW)
- $a = Y_{\mathsf{rlx}}; a = Y_{\mathsf{rlx}}; \leadsto a = Y_{\mathsf{rlx}};$ (RAR)

Fence Insertion such as

- $C_1; C_2 \leadsto C_1; \mathsf{F}_{\{\mathsf{rel},\mathsf{acq},\mathsf{sc}\}}; C_2$

Access strengthening such as

- $Y_{\mathsf{rlx}} = v \leadsto Y_{\mathsf{rel}} = v$
- $a = Y_{\mathsf{rlx}} \leadsto a = Y_{\mathsf{acq}}$

Reordering of accesses such as

- $a = Y_{\mathsf{rlx}}; Z_{\mathsf{rlx}} = v \leadsto Z_{\mathsf{rlx}} = v; a = Y_{\mathsf{rlx}}$

Access Elimination

Can we prove the correctness of an access elimination?

Access Elimination

Can we prove the correctness of an access elimination?

Remember the theorem:

$$X \in [\![\mathbb{P}_{\mathsf{tgt}}]\!].\ \exists X' \in [\![\mathbb{P}_{\mathsf{src}}]\!].\ O(X) = O(X')$$

Access Elimination

Can we prove the correctness of an access elimination?

Remember the theorem:

$$X \in [\![\mathbb{P}_{\mathsf{tgt}}]\!].\ \exists X' \in [\![\mathbb{P}_{\mathsf{src}}]\!].\ O(X) = O(X')$$

Proof Strategy:

1. For each consistent execution X' of target program $\mathbb{P}_{\mathsf{tgt}}$, define an execution X of $\mathbb{P}_{\mathsf{src}}$.
2. Show that X is consistent.
3. Show that $O(X) = O(X')$

OW Elimination

Let's consider OW transformations i.e.

$$Y_o = v; Y_{o'} = v' \rightsquigarrow Y_{o'} = v' \qquad \text{(OW)}$$

where $o, o' \in \{\mathsf{na}, \mathsf{rlx}, \mathsf{rel}, \mathsf{sc}\}$

Theorem

(OW) is correct when $o \sqsubseteq o'$.

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. For each consistent execution X' of target program $\mathbb{P}_{\mathsf{tgt}}$, define an execution X of $\mathbb{P}_{\mathsf{src}}$.

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. For each consistent execution X' of target program $\mathbb{P}_{\text{tgt}}$, define an execution X of $\mathbb{P}_{\text{src}}$.

Let X' be a target execution where $Y_{o'} = v'$ results in event $e' = \mathrm{W}_{o'}(Y, v')$.
We define source execution X where $Y_o = v$ results in event $e = \mathrm{W}_o(Y, v)$ and

- $X.\mathrm{E} = X'.\mathrm{E} \cup \{e\}$
- $X.\mathrm{po} = X'.\mathrm{po} \cup \{(e, e')\} \cup \{(a, e) \mid X'.\mathrm{po}(a, e')\} \cup \{(e, a) \mid X'.\mathrm{po}(e', a)\}$
- $X.\mathrm{rf} = X'.\mathrm{rf}$
- $X.\mathrm{mo} = X'.\mathrm{mo} \cup \{(e, e')\} \cup \{(a, e) \mid X'.\mathrm{mo}(a, e')\} \cup \{(e, a) \mid X'.\mathrm{mo}(e', a)\}$
- $X.\mathrm{sc} = X'.\mathrm{sc} \cup \{(e, e') \mid o = \mathrm{sc}\} \cup \{(a, e) \mid X'.(a, e') \wedge o = \mathrm{sc}\} \cup \{(e, a) \mid X'.\mathrm{sc}(e', a) \wedge o = \mathrm{sc}\}$

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. Show that X is consistent.

We show X is consistent by contradiction.

Note that
$X.\mathsf{hb} = X'.\mathsf{hb} \cup \{(e, e')\} \cup \{(a, e) \mid X'.\mathsf{hb}(a, e')\} \cup \{(e, a) \mid X'.\mathsf{hb}(e', a)\}$

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. Show that X is consistent.

We show X is consistent by contradiction.

Note that
$X.\mathsf{hb} = X'.\mathsf{hb} \cup \{(e, e')\} \cup \{(a, e) \mid X'.\mathsf{hb}(a, e')\} \cup \{(e, a) \mid X'.\mathsf{hb}(e', a)\}$

(irrHB)
Assume there exists a $X.\mathsf{hb}$ cycle.
It implies that there exists a $(X.\mathsf{po} \cup X.\mathsf{sw})^+$ cycle.
It implies that the $(X.\mathsf{po} \cup X.\mathsf{sw})^+$ cycle passes through event e.
However, from definition, there is no incoming/outgoing $X.\mathsf{sw}$ edge to/from event e.
Therefore $X.\mathsf{hb}$ through e contains $X.\mathsf{hb}(-, e)$ and $X.\mathsf{hb}(e, -)$.
It implies $X'.\mathsf{hb}$ is a cycle which is a contradiction.
Therefore $X.\mathsf{hb}$ satisfies (irrHB).

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. Show that X is consistent.

We show X is consistent by contradiction.

Note that
$X.\mathsf{hb} = X'.\mathsf{hb} \cup \{(e, e')\} \cup \{(a, e) \mid X'.\mathsf{hb}(a, e')\} \cup \{(e, a) \mid X'.\mathsf{hb}(e', a)\}$

Assume X is not coherent.

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. Show that X is consistent.

We show X is consistent by contradiction.

Note that
$X.\mathsf{hb} = X'.\mathsf{hb} \cup \{(e, e')\} \cup \{(a, e) \mid X'.\mathsf{hb}(a, e')\} \cup \{(e, a) \mid X'.\mathsf{hb}(e', a)\}$

Assume X is not coherent.
Assume there exists a $X.\mathsf{rf}; X.\mathsf{hb}$ cycle.
It involves event e.
However, from definition, we know that $X.\mathsf{rf} = X.\mathsf{rf}$.
Therefore a $X.\mathsf{rf}; X.\mathsf{hb}$ cycle implies that there already exists a a $X'.\mathsf{rf}; X'.\mathsf{hb}$ cycle which is a contradiction.
Therefore X satisfies (CohHBRF).

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. Show that X is consistent.

We show X is consistent by contradiction.

Note that
$X.\mathsf{hb} = X'.\mathsf{hb} \cup \{(e, e')\} \cup \{(a, e) \mid X'.\mathsf{hb}(a, e')\} \cup \{(e, a) \mid X'.\mathsf{hb}(e', a)\}$

Assume X is not coherent.

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. Show that X is consistent.

We show X is consistent by contradiction.

Note that
$X.\mathsf{hb} = X'.\mathsf{hb} \cup \{(e, e')\} \cup \{(a, e) \mid X'.\mathsf{hb}(a, e')\} \cup \{(e, a) \mid X'.\mathsf{hb}(e', a)\}$
Assume X is not coherent.
Assume there exists a $X.\mathsf{mo}; X.\mathsf{rf}; X.\mathsf{hb}$ cycle.
It involves event e.
However, from definition, we know that
$X.\mathsf{mo}(a, e) \implies X.\mathsf{mo}(a, e')$,
$X.\mathsf{mo}(e, a) \implies a = e' \vee X'.\mathsf{mo}(e', a)$,
$X.\mathsf{po}(a, e) \implies X.\mathsf{po}(a, e')$, $X.\mathsf{po}(e, a) \implies a = e' \vee X'.\mathsf{po}(e', a)$.
Therefore a $X.\mathsf{mo}; X.\mathsf{rf}; X.\mathsf{hb}$ cycle implies that there already exists a a $X'.\mathsf{mo}; X'.\mathsf{rf}; X'.\mathsf{hb}$ cycle which is a contradiction.
Therefore X satisfies (Coh-RW).

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. Show that X is consistent.

We show X is consistent by contradiction.

Note that
$X.\mathsf{hb} = X'.\mathsf{hb} \cup \{(e, e')\} \cup \{(a, e) \mid X'.\mathsf{hb}(a, e')\} \cup \{(e, a) \mid X'.\mathsf{hb}(e', a)\}$

Assume X is not coherent.

Exercise: Prove rest of the coherence axioms.

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. Show that X is consistent.

We show X is consistent by contradiction.

(Atomicity)
Assume X' does not preserve (Atomicity).
However, from definition, it implies X' violates (Atomicity) which is a contradiction.
Therefore X' preserves atomicity.

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. Show that X is consistent.

We show X is consistent by contradiction.

(ConsSC)
Two possibilities:
Case $X.\text{sc} = X'.\text{sc}$ when $o \neq \text{sc}$
In this case X preserves (ConsSC) as X' preserves (ConsSC).
Case $X'.\text{sc} \subseteq X.\text{sc}$ when $o = \text{sc}$
Assume X' does not preserve (ConsSC).
However, from definition, it implies X' violates (ConsSC) which is a contradiction.
Therefore X preserves (ConsSC).

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. Show that X is consistent.

We show X is consistent by contradiction.

(SCread)
Assume X does not preserve (SCread).
Two possibilities:
Case $o = \mathsf{sc}$
Suppose $(a, b) \in X.\mathsf{rf}; [\mathsf{RU}_{\mathsf{sc}}]$ and $X.\mathsf{sc}_{\mathsf{imm}}(e, b)$ holds.
However, from definition we know that $\mathsf{sc}_{\mathsf{imm}}(e, e')$ and $e' \notin \mathsf{RU}_{\mathsf{sc}}$.
Considering the other condition if e violates the condition in X then e' already violates the condition in X'.
Hence a contradiction and (SCread) holds.
Case $o \neq \mathsf{sc}$
From the definition X preserves (ConsSC).

Therefore source execution X is consistent.

OW Elimination: Proof of Correctness

Remember the proof strategy steps:

1. Show that $O(X) = O(X')$

From definition, $[\{e\}]; X.\mathsf{mo} \neq \emptyset$ as $X.\mathsf{mo}(e, e')$ holds.
Hence $O(X) = O(X')$.

Therefore the transformation is correct.

Exercise

Show that the following transformation is not safe.

$Y_{\mathsf{sc}} = v; \mathsf{CAS}_{\mathsf{sc}}(Y, v_1, v') \rightsquigarrow \mathsf{CAS}_{\mathsf{sc}}(Y, v_1, v')$.

Hint. Check the (SCread) axiom.