Modular Transactions: Bounding Mixed Races in Space and Time

Brijesh Dongol* Radha Jagadeesan† James Riely†

*University of Surrey, †DePaul University

PPoPP 2019

What can a programmer conclude about this code?

$$x \coloneqq 1;\ y \coloneqq 1;\ \text{atomic}\,\{\, F \coloneqq 1 \,\};\ z \coloneqq 1;\ \text{if}\ w\ \text{then}\ x \coloneqq 2$$
$$\parallel\ y \coloneqq 2;\ \text{atomic}\,\{\, r \coloneqq F \,\};\ z \coloneqq 2\,;\ \text{if}\ r\ \text{then}\ w \coloneqq y - y + x$$

What can a programmer conclude about this code?

$$x := 1;\ y := 1;\ \text{atomic}\,\{\,F := 1\,\};\ z := 1;\ \text{if}\ w\ \text{then}\ x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\,r := F\,\};\ z := 2;\ \text{if}\ r\ \text{then}\ w := y - y + x$$

$$\begin{tikzcd}
\mathrm{W}\,x\,1 \arrow[r] & \mathrm{W}\,y\,1 \arrow[r] & \boxed{\mathrm{W}\,F\,1} \arrow[r] \arrow[d, "\mathrm{xwr}"] & \mathrm{W}\,z\,1 \arrow[r] & \mathrm{R}\,w\,? \arrow[r] & \mathrm{W}\,x\,2 & & \\
& \mathrm{W}\,y\,2 \arrow[r] & \boxed{\mathrm{R}\,F\,1} \arrow[r] & \mathrm{W}\,z\,2 \arrow[r] & \mathrm{R}\,y\,? \arrow[r] & \mathrm{R}\,y\,? \arrow[r] & \mathrm{R}\,x\,? \arrow[r] & \mathrm{W}\,w\,? \arrow[ulll, "\mathrm{wr}"']
\end{tikzcd}$$

- Variables initially 0; Reads as shown
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\mathrm{wr}}$/$\xrightarrow{\mathrm{xwr}}$ Write-to-Read Dependency (Plain/Transactional)

What can a programmer conclude about this code?

$$
\begin{array}{l}
x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if}\ w\ \text{then}\ x := 2 \\
\parallel\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2;\ \text{if}\ r\ \text{then}\ w := y - y + x
\end{array}
$$

$$
\begin{tikzcd}
\mathrm{W}x1 \arrow[r] & \mathrm{W}y1 \arrow[r] & \boxed{\mathrm{W}F1} \arrow[r] \arrow[d, "\mathrm{xwr}"] & \mathrm{W}z1 \arrow[r] & \mathrm{R}w? \arrow[r] & \mathrm{W}x2 & & \\
& \mathrm{W}y2 \arrow[r] & \boxed{\mathrm{R}F1} \arrow[r] & \mathrm{W}z2 \arrow[r] & \mathrm{R}y? \arrow[r] & \mathrm{R}y? \arrow[r] & \mathrm{R}x? \arrow[r] & \mathrm{W}w? \arrow[ulll, "\mathrm{wr}"']
\end{tikzcd}
$$

- Variables initially 0; Reads as shown
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\mathrm{wr}}/\xrightarrow{\mathrm{xwr}}$ Write-to-Read Dependency (Plain/Transactional)
- Full of races when $\langle \mathrm{R}y?\rangle$, $\langle \mathrm{R}y?\rangle$, $\langle \mathrm{R}x?\rangle$ occur

What can a programmer conclude about this code?

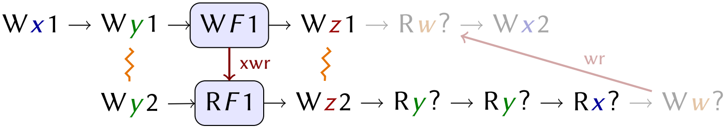
$$x := 1;\ y := 1;\ \text{atomic}\ \{\ F := 1\ \};\ z := 1;\ \text{if}\ \ w\ \text{then}\ \ x := 2$$
$$\|\ \ y := 2;\ \text{atomic}\ \{\ r := F\ \};\ z := 2;\ \text{if}\ \ r\ \text{then}\ \ w := y - y + x$$

$$\begin{tikzcd}
\mathrm{W}\,x\,1 \arrow[r] & \mathrm{W}\,y\,1 \arrow[r] \arrow[d, squiggly, no head] & \boxed{\mathrm{W}\,F\,1} \arrow[r] \arrow[d, "\mathrm{xwr}"] & \mathrm{W}\,z\,1 \arrow[r] & \mathrm{R}\,w\,? \arrow[r] & \mathrm{W}\,x\,2 & & \\
& \mathrm{W}\,y\,2 \arrow[r] & \boxed{\mathrm{R}\,F\,1} \arrow[r] & \mathrm{W}\,z\,2 \arrow[r] & \mathrm{R}\,y\,? \arrow[r] & \mathrm{R}\,y\,? \arrow[r] & \mathrm{R}\,x\,? \arrow[r] & \mathrm{W}\,w\,? \arrow[ulll, "\mathrm{wr}"']
\end{tikzcd}$$

- Variables initially 0; Reads as shown
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\mathrm{wr}}/\xrightarrow{\mathrm{xwr}}$ Write-to-Read Dependency (Plain/Transactional)
- Full of races when $\langle \mathrm{R}\,y\,?\rangle$, $\langle \mathrm{R}\,y\,?\rangle$, $\langle \mathrm{R}\,x\,?\rangle$ occur
  - Past race on y (should see same value on both reads)

What can a programmer conclude about this code?

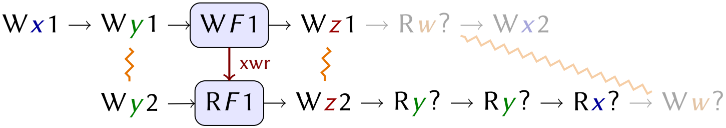
$$x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if } w \text{ then } x := 2$$
$$\|\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2;\ \text{if } r \text{ then } w := y - y + x$$



- Variables initially 0; Reads as shown
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\text{wr}}$/$\xrightarrow{\text{xwr}}$ Write-to-Read Dependency (Plain/Transactional)
- Full of races when $\langle \mathrm{R}\,y\,?\rangle$, $\langle \mathrm{R}\,y\,?\rangle$, $\langle \mathrm{R}\,x\,?\rangle$ occur
  - Past race on y (should see same value on both reads)
  - Current race on z (should not affect x or y)

What can a programmer conclude about this code?

$$x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if } w \text{ then } x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2;\ \text{if } r \text{ then } w := y - y + x$$



- Variables initially 0; Reads as shown
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\text{wr}}$/$\xrightarrow{\text{xwr}}$ Write-to-Read Dependency (Plain/Transactional)
- Full of races when $\langle \mathrm{R}\,y\,?\rangle$, $\langle \mathrm{R}\,y\,?\rangle$, $\langle \mathrm{R}\,x\,?\rangle$ occur
  - Past race on y (should see same value on both reads)
  - Current race on z (should not affect x or y)
  - Future race on w (should not affect x or y)

What can a programmer conclude about this code?

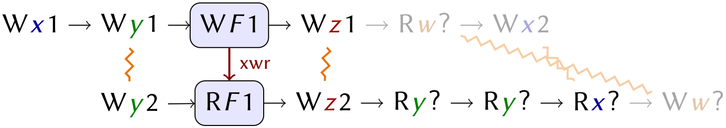
$$
\begin{array}{l}
\ {\color{blue}x} := 1;\ {\color{green}y} := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ {\color{red}z} := 1;\ \text{if}\ {\color{orange}w}\ \text{then}\ {\color{blue}x} := 2 \\
\|\ \ {\color{green}y} := 2;\ \text{atomic}\,\{\, r := F \,\};\ {\color{red}z} := 2;\ \text{if}\ r\ \text{then}\ {\color{orange}w} := {\color{green}y} - {\color{green}y} + {\color{blue}x}
\end{array}
$$

$$
\begin{tikzcd}
\mathrm{W}x1 \arrow[r] & \mathrm{W}y1 \arrow[r] \arrow[d, squiggly, no head] & \boxed{\mathrm{W}F1} \arrow[r] \arrow[d, "\mathrm{xwr}"] & \mathrm{W}z1 \arrow[r] \arrow[d, squiggly, no head] & \mathrm{R}w? \arrow[r] & \mathrm{W}x2 \arrow[dr, squiggly, no head] & & \\
& \mathrm{W}y2 \arrow[r] & \boxed{\mathrm{R}F1} \arrow[r] & \mathrm{W}z2 \arrow[r] & \mathrm{R}y? \arrow[r] & \mathrm{R}y? \arrow[r] & \mathrm{R}x? \arrow[r] & \mathrm{W}w?
\end{tikzcd}
$$

- Variables initially 0; Reads as shown
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\mathrm{wr}}/\xrightarrow{\mathrm{xwr}}$ Write-to-Read Dependency (Plain/Transactional)
- Full of races when $\langle \mathrm{R}y?\rangle$, $\langle \mathrm{R}y?\rangle$, $\langle \mathrm{R}x?\rangle$ occur
  - Past race on y (should see same value on both reads)
  - Current race on z (should not affect x or y)
  - Future race on w (should not affect x or y)
  - Future race on x (read should not see the future)

Sequential Consistency (SC)

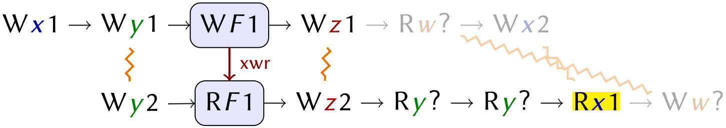
$$x := 1;\ y := 1;\ \text{atomic}\,\{\,F := 1\,\};\ z := 1;\ \text{if}\ w\ \text{then}\ x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\,r := F\,\};\ z := 2;\ \text{if}\ r\ \text{then}\ w := y - y + x$$



- Execution by interleaving, respecting orders

Sequential Consistency (SC)

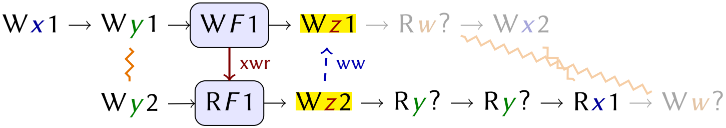
$$x \coloneqq 1;\ y \coloneqq 1;\ \text{atomic}\,\{\, F \coloneqq 1 \,\};\ z \coloneqq 1;\ \text{if } w \text{ then } x \coloneqq 2$$
$$\parallel\ y \coloneqq 2;\ \text{atomic}\,\{\, r \coloneqq F \,\};\ z \coloneqq 2;\ \text{if } r \text{ then } w \coloneqq y - y + x$$



- Execution by interleaving, respecting orders
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\text{wr}} / \xrightarrow{\text{xwr}}$ Write-to-Read Dependency (Plain/Transactional)

Sequential Consistency (SC)

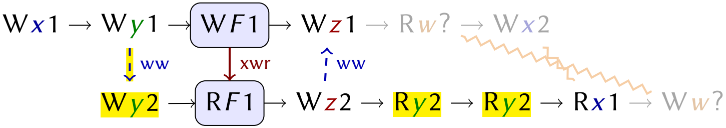
$$x := 1;\ y := 1;\ \text{atomic}\,\{\,F := 1\,\};\ z := 1;\ \text{if}\ w\ \text{then}\ x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\,r := F\,\};\ z := 2;\ \text{if}\ r\ \text{then}\ w := y - y + x$$



- Execution by interleaving, respecting orders
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\text{wr}}$ / $\xrightarrow{\text{xwr}}$ Write-to-Read Dependency (Plain/Transactional)
  - $\xrightarrow{\text{ww}}$ Write-to-Write Order

Sequential Consistency (SC)

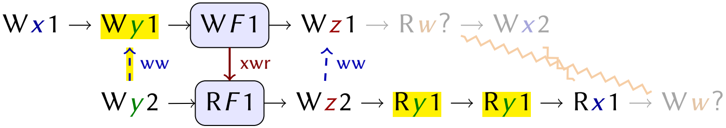
$$x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if } w \text{ then } x := 2$$
$$\|\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2;\ \text{if } r \text{ then } w := y - y + x$$



- Execution by interleaving, respecting orders
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\text{wr}}$/$\xrightarrow{\text{xwr}}$ Write-to-Read Dependency (Plain/Transactional)
  - $\dashrightarrow^{\text{ww}}$ Write-to-Write Order

Sequential Consistency (SC)

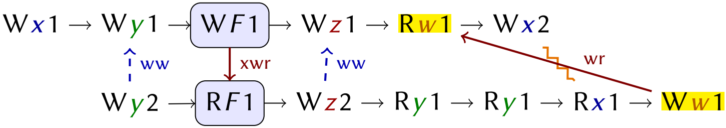
$$x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if } w \text{ then } x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2;\ \text{if } r \text{ then } w := y - y + x$$



- Execution by interleaving, respecting orders
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\text{wr}}$/$\xrightarrow{\text{xwr}}$ Write-to-Read Dependency (Plain/Transactional)
  - $\overset{\text{ww}}{\dashrightarrow}$ Write-to-Write Order

Sequential Consistency (SC)

$$x := 1;\ y := 1;\ \text{atomic}\,\{\,F := 1\,\};\ z := 1;\ \text{if}\ w\ \text{then}\ x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\,r := F\,\};\ z := 2;\ \text{if}\ r\ \text{then}\ w := y - y + x$$



- Execution by interleaving, respecting orders
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\text{wr}}/\xrightarrow{\text{xwr}}$ Write-to-Read Dependency (Plain/Transactional)
  - $\overset{\text{ww}}{\dashrightarrow}$ Write-to-Write Order

Sequential Consistency (SC)

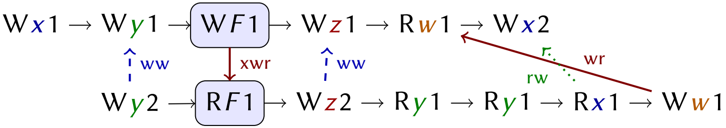
$$
\begin{array}{l}
x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if}\ w\ \text{then}\ x := 2 \\
\parallel\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2;\ \text{if}\ r\ \text{then}\ w := y - y + x
\end{array}
$$

$$
\begin{tikzcd}
\mathrm{W}x1 \arrow[r] & \mathrm{W}y1 \arrow[r] & \boxed{\mathrm{W}F1} \arrow[r] \arrow[d, "\mathrm{xwr}"] & \mathrm{W}z1 \arrow[r] & \mathrm{R}w1 \arrow[r] & \mathrm{W}x2 & & \\
& \mathrm{W}y2 \arrow[r] \arrow[u, dashed, "\mathrm{ww}"'] & \boxed{\mathrm{R}F1} \arrow[r] & \mathrm{W}z2 \arrow[r] \arrow[u, dashed, "\mathrm{ww}"'] & \mathrm{R}y1 \arrow[r] & \mathrm{R}y1 \arrow[r] & \mathrm{R}x1 \arrow[r] \arrow[ul, dotted, "\mathrm{rw}"] & \mathrm{W}w1 \arrow[ulll, "\mathrm{wr}"']
\end{tikzcd}
$$

- Execution by interleaving, respecting orders
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\mathrm{wr}}$ / $\xrightarrow{\mathrm{xwr}}$ Write-to-Read Dependency (Plain/Transactional)
  - $\overset{\mathrm{ww}}{\dashrightarrow}$ Write-to-Write Order
  - $\overset{\mathrm{rw}}{\cdots\!\!>}$ Read-to-Write Antidependency

Sequential Consistency (SC)

$$x := 1;\ y := 1;\ \text{atomic}\,\{\,F := 1\,\};\ z := 1;\ \text{if}\ w\ \text{then}\ x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\,r := F\,\};\ z := 2;\ \text{if}\ r\ \text{then}\ w := y - y + x$$



- Execution by interleaving, respecting orders
  - $\longrightarrow$ Program Order
  - $\xrightarrow{\text{wr}}$/$\xrightarrow{\text{xwr}}$ Write-to-Read Dependency (Plain/Transactional)
  - $\xrightarrow{\text{ww}}$ Write-to-Write Order
  - $\xrightarrow{\text{rw}}$ Read-to-Write Antidependency
- SC Declaratively:
  - Require union of orders acyclic (CAUSALITY)

Performance Relies On Reordering & Optimization

- Reordering performed in hardware

$x := 1;\ y := 1 \longrightarrow y := 1;\ x := 1$	Independent Writes
$r := x;\ q := y \longrightarrow q := y;\ r := x$	Independent Reads
$x := 1;\ q := y \longrightarrow q := y;\ x := 1$	Store Buffering
$r := x;\ q := y \longrightarrow q := y;\ r := x$	Load Buffering

Performance Relies On Reordering & Optimization

- Reordering performed in hardware

$x := 1;\ y := 1 \longrightarrow y := 1;\ x := 1$	Independent Writes
$r := x;\ q := y \longrightarrow q := y;\ r := x$	Independent Reads
$x := 1;\ q := y \longrightarrow q := y;\ x := 1$	Store Buffering
$r := x;\ q := y \longrightarrow q := y;\ r := x$	Load Buffering

- Peephole optimization + reordering enables common subexpression elimination, loop invariant code motion, etc

$r := x;\ q := x \longrightarrow r := x;\ q := x$	Redundant Load
$x := 1;\ q := x \longrightarrow x := 1;\ q := 1$	Store Forwarding
$x := 1;\ x := 2 \longrightarrow x := 2$	Dead Store

Store Buffering under SC

$$\begin{array}{r} x := 1;\ q := y \\ \parallel y := 1;\ r := x \end{array} \quad \xRightarrow{?} \quad \begin{array}{r} q := y;\ x := 1 \\ \parallel r := x;\ y := 1 \end{array}$$

- Delay write past nonconflicting read?
  - Performed by x86-TSO, ARMv8, etc

Store Buffering under SC

$$\begin{array}{r} x := 1;\ q := y \\ \parallel y := 1;\ r := x \end{array} \quad \stackrel{?}{\Longrightarrow} \quad \begin{array}{r} q := y;\ x := 1 \\ \parallel r := x;\ y := 1 \end{array}$$

- Delay write past nonconflicting read?
  - Performed by x86-TSO, ARMv8, etc
- *Correctness:* Rewrites should not introduce new behavior

Store Buffering under SC

$$\begin{array}{r} x := 1;\ q := y \\ \parallel y := 1;\ r := x \end{array} \quad \xRightarrow{?} \quad \begin{array}{r} q := y;\ x := 1 \\ \parallel r := x;\ y := 1 \end{array}$$

$$\begin{array}{c} \mathrm{W}\,x\,1 \longrightarrow \mathrm{R}\,y\,0 \\ \mathsf{rw} \quad \mathsf{rw} \\ \mathrm{W}\,y\,1 \longrightarrow \mathrm{R}\,x\,0 \end{array} \qquad ✗$$

- Delay write past nonconflicting read?
  - Performed by x86-TSO, ARMv8, etc
- *Correctness:* Rewrites should not introduce new behavior

Store Buffering under SC

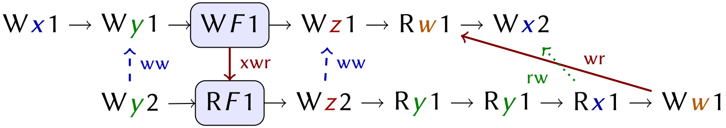
$$
\begin{array}{r} x := 1;\ q := y \\ \parallel y := 1;\ r := x \end{array}
\quad \stackrel{?}{\Longrightarrow} \quad
\begin{array}{r} q := y;\ x := 1 \\ \parallel r := x;\ y := 1 \end{array}
$$

$$
\begin{tikzcd}
\mathrm{W}\,x\,1 \arrow[r] & \mathrm{R}\,y\,0 \arrow[dl, dotted, "\mathrm{rw}"] \\
\mathrm{W}\,y\,1 \arrow[r] & \mathrm{R}\,x\,0 \arrow[ul, dotted, "\mathrm{rw}"']
\end{tikzcd}
\quad ✗
\qquad\qquad
\begin{tikzcd}
\mathrm{R}\,y\,0 \arrow[r] \arrow[dr, dotted, "\mathrm{rw}"'] & \mathrm{W}\,x\,1 \\
\mathrm{R}\,x\,0 \arrow[r] \arrow[ur, dotted, "\mathrm{rw}"] & \mathrm{W}\,y\,1
\end{tikzcd}
\quad ✓
$$

- Delay write past nonconflicting read?
  - Performed by x86-TSO, ARMv8, etc
- *Correctness:* Rewrites should not introduce new behavior

SC-DRF: A Contract Between Programmer & Implementor

$$x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if } w \text{ then } x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2;\ \text{if } r \text{ then } w := y - y + x$$



- *Happens-Before*
  - $\xrightarrow{\text{hb}}$ includes $\longrightarrow$ and $\xrightarrow{\text{xwr}}$ but not $\xrightarrow{\text{wr}}$, $\overset{\text{ww}}{\dashrightarrow}$ or $\overset{\text{rw}}{\cdots\rightarrow}$

SC-DRF: A Contract Between Programmer & Implementor

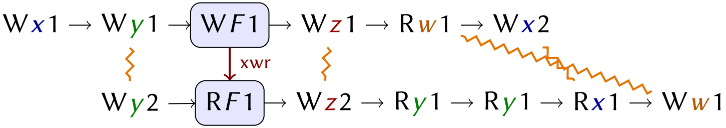
$$x \coloneqq 1;\ y \coloneqq 1;\ \text{atomic}\,\{\,F \coloneqq 1\,\};\ z \coloneqq 1;\ \text{if}\ w\ \text{then}\ x \coloneqq 2$$
$$\parallel\ y \coloneqq 2;\ \text{atomic}\,\{\,r \coloneqq F\,\};\ z \coloneqq 2;\ \text{if}\ r\ \text{then}\ w \coloneqq y - y + x$$

$$\begin{tikzcd}
\mathrm{W}\,x\,1 \arrow[r] & \mathrm{W}\,y\,1 \arrow[r] & \boxed{\mathrm{W}\,F\,1} \arrow[r] \arrow[d, "\mathrm{xwr}"] & \mathrm{W}\,z\,1 \arrow[r] & \mathrm{R}\,w\,1 \arrow[r] & \mathrm{W}\,x\,2 & & \\
& \mathrm{W}\,y\,2 \arrow[r] \arrow[u, dashed, "\mathrm{ww}"'] & \boxed{\mathrm{R}\,F\,1} \arrow[r] & \mathrm{W}\,z\,2 \arrow[r] \arrow[u, dashed, "\mathrm{ww}"'] & \mathrm{R}\,y\,1 \arrow[r] & \mathrm{R}\,y\,1 \arrow[r] & \mathrm{R}\,x\,1 \arrow[r] \arrow[ul, dotted, "\mathrm{rw}"] & \mathrm{W}\,w\,1 \arrow[ulll, "\mathrm{wr}"']
\end{tikzcd}$$

- *Happens-Before*
  - $\xrightarrow{hb}$ includes $\longrightarrow$ and $\xrightarrow{xwr}$ but not $\xrightarrow{wr}$, $\overset{ww}{\dashrightarrow}$ or $\overset{rw}{\dashrightarrow}$
- *Data race:* “Incorrect publication”
  - $\mathrm{W}x \overset{hb}{\not\leftrightarrow} \mathrm{W}x \qquad \mathrm{W}x \overset{hb}{\not\leftrightarrow} \mathrm{R}x$

SC-DRF: A Contract Between Programmer & Implementor

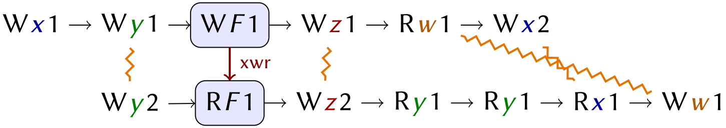
$$x := 1;\ y := 1;\ \text{atomic}\,\{\,F := 1\,\};\ z := 1;\ \text{if } w \text{ then } x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\,r := F\,\};\ z := 2;\ \text{if } r \text{ then } w := y - y + x$$



- *Happens-Before*
  - $\xrightarrow{\text{hb}}$ includes $\longrightarrow$ and $\xrightarrow{\text{xwr}}$ but not $\xrightarrow{\text{wr}}$, $\dashrightarrow^{\text{ww}}$ or $\overset{\text{rw}}{\cdots\rightarrow}$
- *Data race:* "Incorrect publication"
  - $\mathrm{W}\,x \overset{\text{hb}}{\not\leftrightarrow} \mathrm{W}\,x \qquad \mathrm{W}\,x \overset{\text{hb}}{\not\leftrightarrow} \mathrm{R}\,x$

SC-DRF: A Contract Between Programmer & Implementor

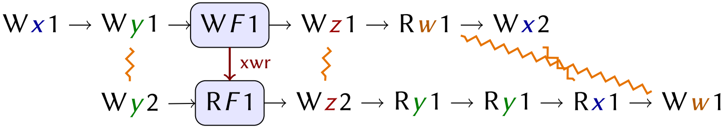
$$x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if } w \text{ then } x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2;\ \text{if } r \text{ then } w := y - y + x$$



- *Happens-Before*
  - $\xrightarrow{\text{hb}}$ includes $\longrightarrow$ and $\xrightarrow{\text{xwr}}$ but not $\xrightarrow{\text{wr}}$, $\xrightarrow{\text{ww}}$ or $\xrightarrow{\text{rw}}$
- *Data race:* “Incorrect publication”
  - $\mathsf{W}\,x \overset{\text{hb}}{\not\leftrightarrow} \mathsf{W}\,x \qquad \mathsf{W}\,x \overset{\text{hb}}{\not\leftrightarrow} \mathsf{R}\,x$
- *DRF program:* every SC execution is Data Race Free

SC-DRF: A Contract Between Programmer & Implementor

$$x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if}\ w\ \text{then}\ x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2;\ \text{if}\ r\ \text{then}\ w := y - y + x$$



- *Happens-Before*
  - $\xrightarrow{\text{hb}}$ includes $\longrightarrow$ and $\xrightarrow{\text{xwr}}$ but not $\xrightarrow{\text{wr}}$, $\overset{\text{ww}}{\dashrightarrow}$ or $\overset{\text{rw}}{\cdots\rightarrow}$
- *Data race:* "Incorrect publication"
  - $\mathsf{W}x \overset{\text{hb}}{\not\leftrightarrow} \mathsf{W}x \qquad \mathsf{W}x \overset{\text{hb}}{\not\leftrightarrow} \mathsf{R}x$
- *DRF program:* every SC execution is Data Race Free
- *SC-DRF:* DRF program $\Rightarrow$ SC behavior

SC-DRF: A Contract Between Programmer & Implementor

$$x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if } w \text{ then } x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2;\ \text{if } r \text{ then } w := y - y + x$$

$$\begin{tikzcd}
\mathrm{W}x1 \arrow[r] & \mathrm{W}y1 \arrow[r] & \boxed{\mathrm{W}F1} \arrow[r] \arrow[d, "\mathrm{xwr}"] & \mathrm{W}z1 \arrow[r] & \mathrm{R}w1 \arrow[r] & \mathrm{W}x2 & & \\
& \mathrm{W}y2 \arrow[r] & \boxed{\mathrm{R}F1} \arrow[r] & \mathrm{W}z2 \arrow[r] & \mathrm{R}y1 \arrow[r] & \mathrm{R}y1 \arrow[r] & \mathrm{R}x1 \arrow[r] & \mathrm{W}w1
\end{tikzcd}$$

- *Happens-Before*
  - $\xrightarrow{\text{hb}}$ includes $\longrightarrow$ and $\xrightarrow{\text{xwr}}$ but not $\xrightarrow{\text{wr}}$, $\xrightarrow{\text{ww}}$ or $\xrightarrow{\text{rw}}$
- *Data race:* “Incorrect publication”
  - $\mathrm{W}x \overset{\text{hb}}{\not\leftrightarrow} \mathrm{W}x \qquad \mathrm{W}x \overset{\text{hb}}{\not\leftrightarrow} \mathrm{R}x$
- *DRF program:* every SC execution is Data Race Free
- *SC-DRF:* DRF program $\Rightarrow$ SC behavior
  - ☺ *No SC data race ever $\Rightarrow$ everything correctly published always*

SC-DRF: A Contract Between Programmer & Implementor

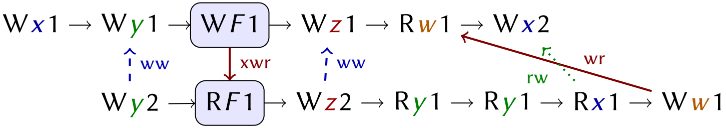
$$x := 1;\ y := 1;\ \text{atomic}\ \{\ F := 1\ \};\ z := 1;\ \text{if}\ \ w\ \ \text{then}\ \ x := 2$$
$$\parallel\ \ y := 2;\ \text{atomic}\ \{\ r := F\ \};\ z := 2;\ \text{if}\ \ r\ \ \text{then}\ \ w := y - y + x$$

$$\begin{tikzcd}
\mathrm{W}x1 \arrow[r] & \mathrm{W}y1 \arrow[r] \arrow[d, leftrightsquigarrow] & \boxed{\mathrm{W}F1} \arrow[r] \arrow[d, "\mathrm{xwr}"] & \mathrm{W}z1 \arrow[r] \arrow[d, leftrightsquigarrow] & \mathrm{R}w1 \arrow[r] & \mathrm{W}x2 \arrow[dr, leftrightsquigarrow] & & \\
& \mathrm{W}y2 \arrow[r] & \boxed{\mathrm{R}F1} \arrow[r] & \mathrm{W}z2 \arrow[r] & \mathrm{R}y1 \arrow[r] & \mathrm{R}y1 \arrow[r] & \mathrm{R}x1 \arrow[r] & \mathrm{W}w1
\end{tikzcd}$$

- *Happens-Before*
  - $\xrightarrow{\mathrm{hb}}$ includes $\longrightarrow$ and $\xrightarrow{\mathrm{xwr}}$ but not $\xrightarrow{\mathrm{wr}}$, $\overset{\mathrm{ww}}{\dashrightarrow}$ or $\overset{\mathrm{rw}}{\dashrightarrow}$
- *Data race:* “Incorrect publication”
  - $\mathrm{W}x \overset{\mathrm{hb}}{\not\leftrightarrow} \mathrm{W}x \qquad \mathrm{W}x \overset{\mathrm{hb}}{\not\leftrightarrow} \mathrm{R}x$
- *DRF program:* every SC execution is Data Race Free
- *SC-DRF:* DRF program ⇒ SC behavior
  - ☺ *No SC data race ever ⇒ everything correctly published always*
  - ☹ Any SC data race ever ⇒ relaxed values/undefined behavior

Local SC-DRF (Dolan, et al, 2018)

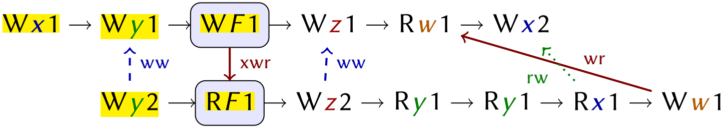
$$x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if } w \text{ then } x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2;\ \text{if } r \text{ then } w := y - y + x$$



- Let $L = \{x, y\}$ be a set of locations

Local SC-DRF (Dolan, et al, 2018)

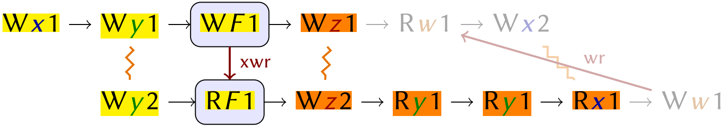
$x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if } w \text{ then } x := 2$
$\parallel\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2;\ \text{if } r \text{ then } w := y - y + x$



- Let $L = \{x, y\}$ be a set of locations
- Let σ be an *L-stable* point in an execution
  - No extension, in any execution, has an *L-race* with σ

Local SC-DRF (Dolan, et al, 2018)

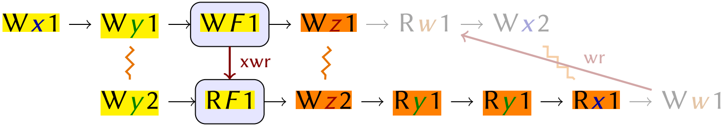
$$x := 1;\ y := 1;\ \text{atomic}\,\{\, F := 1 \,\};\ z := 1;\ \text{if }\ w\ \text{ then }\ x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\, r := F \,\};\ z := 2\,;\ \text{if }\ r\ \text{ then }\ w := y - y + x$$



- Let $L = \{x, y\}$ be a set of locations
- Let σ be an *L-stable* point in an execution
  - No extension, in any execution, has an *L-race* with σ
- Let ρ be an extension of σ in an execution

Local SC-DRF (Dolan, et al, 2018)

$$x := 1;\ y := 1;\ \text{atomic}\,\{\,F := 1\,\};\ z := 1;\ \text{if}\ w\ \text{then}\ x := 2$$
$$\parallel\ y := 2;\ \text{atomic}\,\{\,r := F\,\};\ z := 2;\ \text{if}\ r\ \text{then}\ w := y - y + x$$

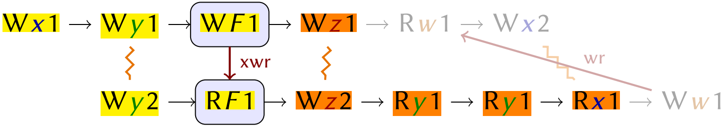


- Let $L = \{x, y\}$ be a set of locations
- Let σ be an *L-stable* point in an execution
  - No extension, in any execution, has an *L-race* with σ
- Let ρ be an extension of σ in an execution
- *No SC L-race in $\rho \Rightarrow L$ correctly published in ρ*

Local SC-DRF (Dolan, et al, 2018)

$x := 1;\ y := 1;\ \text{atomic}\,\{\,F := 1\,\};\ z := 1;\ \text{if}\ w\ \text{then}\ x := 2$

$\parallel\ y := 2;\ \text{atomic}\,\{\,r := F\,\};\ z := 2;\ \text{if}\ r\ \text{then}\ w := y - y + x$



- Let $L = \{x, y\}$ be a set of locations
- Let σ be an *L-stable* point in an execution
  - No extension, in any execution, has an *L-race* with σ
- Let ρ be an extension of σ in an execution
- *No SC L-race in $\rho \Rightarrow L$ correctly published in ρ*
- Ignore races outside L, in past (σ), in future (after ρ)

SC-LDRF: Reordering & Optimization

- Reordering performed in hardware

$x := 1;\ y := 1 \longrightarrow y := 1;\ x := 1$	Independent Writes ☺
$r := x;\ q := y \longrightarrow q := y;\ r := x$	Independent Reads ☺
$x := 1;\ q := y \longrightarrow q := y;\ x := 1$	Store Buffering ☺
$r := x;\ q := y \longrightarrow q := y;\ r := x$	Load Buffering ☹

- Peephole optimization + reordering enables common subexpression elimination, loop invariant code motion, etc

$r := x;\ q := x \longrightarrow r := x;\ q := x$	Redundant Load ☺
$x := 1;\ q := x \longrightarrow x := 1;\ q := 1$	Store Forwarding ☺
$x := 1;\ x := 2 \longrightarrow x := 2$	Dead Store ☺

Load Buffering

$$\begin{array}{r} q := y;\ x := 1 \\ \parallel r := x;\ y := 1 \end{array} \quad \stackrel{?}{\Longrightarrow} \quad \begin{array}{r} x := 1;\ q := y \\ \parallel y := 1;\ r := x \end{array}$$

$$\begin{tikzcd} \mathrm{R}\,y\,1 \arrow[r] & \mathrm{W}\,x\,1 \arrow[dl, "\mathrm{wr}"] \\ \mathrm{R}\,x\,1 \arrow[r] & \mathrm{W}\,y\,1 \arrow[ul, "\mathrm{wr}"'] \end{tikzcd} \quad ✗ \qquad\qquad \begin{tikzcd} \mathrm{W}\,x\,1 \arrow[r] \arrow[dr, "\mathrm{wr}"'] & \mathrm{R}\,y\,1 \\ \mathrm{W}\,y\,1 \arrow[r] \arrow[ur, "\mathrm{wr}"] & \mathrm{R}\,x\,1 \end{tikzcd} \quad ✓$$

- LDRF disables “reading the future”
  - Require $(\xrightarrow{\mathrm{hb}} \cup \xrightarrow{\mathrm{wr}})$ acyclic (CAUSALITY)

Load Buffering

$$\begin{array}{l} q := y;\ x := 1 \\ \parallel r := x;\ y := 1 \end{array} \quad \overset{?}{\Longrightarrow} \quad \begin{array}{l} x := 1;\ q := y \\ \parallel y := 1;\ r := x \end{array}$$

$$\begin{tikzcd} \mathrm{R}\,y\,1 \arrow[r] & \mathrm{W}\,x\,1 \arrow[dl, "\mathrm{wr}"] \\ \mathrm{R}\,x\,1 \arrow[r] & \mathrm{W}\,y\,1 \arrow[ul, "\mathrm{wr}"'] \end{tikzcd} \quad ✗ \qquad\qquad \begin{tikzcd} \mathrm{W}\,x\,1 \arrow[r] \arrow[dr, "\mathrm{wr}"'] & \mathrm{R}\,y\,1 \\ \mathrm{W}\,y\,1 \arrow[r] \arrow[ur, "\mathrm{wr}"] & \mathrm{R}\,x\,1 \end{tikzcd} \quad ✓$$

- LDRF disables “reading the future”
  - Require $(\xrightarrow{\mathrm{hb}} \cup \xrightarrow{\mathrm{wr}})$ acyclic (CAUSALITY)
  - ☹ Requires fences on ARMv8 and PowerPC

Load Buffering

$$\begin{array}{r} q := y;\; x := 1 \\ \parallel r := x;\; y := 1 \end{array} \overset{?}{\Longrightarrow} \begin{array}{l} x := 1;\; q := y \\ \parallel y := 1;\; r := x \end{array}$$

$$\begin{array}{c} \mathrm{R}\,y\,1 \longrightarrow \mathrm{W}\,x\,1 \\ \mathrm{R}\,x\,1 \longrightarrow \mathrm{W}\,y\,1 \end{array} \quad \text{✗} \qquad\qquad \begin{array}{c} \mathrm{W}\,x\,1 \longrightarrow \mathrm{R}\,y\,1 \\ \mathrm{W}\,y\,1 \longrightarrow \mathrm{R}\,x\,1 \end{array} \quad \text{✓}$$

- LDRF disables "reading the future"
  - Require $(\xrightarrow{\text{hb}} \cup \xrightarrow{\text{wr}})$ acyclic (Causality)
  - ☹ Requires fences on ARMv8 and PowerPC
  - ☺ $< 1\%$ overhead on ARMv8

Load Buffering

$$q := y;\ x := 1 \;\|\; r := x;\ y := 1 \quad \overset{?}{\Longrightarrow} \quad x := 1;\ q := y \;\|\; y := 1;\ r := x$$

$$\begin{tikzcd} \mathsf{R}\,y\,1 \arrow[r] & \mathsf{W}\,x\,1 \arrow[dl, "\mathsf{wr}"] \\ \mathsf{R}\,x\,1 \arrow[r] & \mathsf{W}\,y\,1 \arrow[ul, "\mathsf{wr}"'] \end{tikzcd} \quad ✗ \qquad\qquad \begin{tikzcd} \mathsf{W}\,x\,1 \arrow[r] \arrow[dr, "\mathsf{wr}"'] & \mathsf{R}\,y\,1 \\ \mathsf{W}\,y\,1 \arrow[r] \arrow[ur, "\mathsf{wr}"] & \mathsf{R}\,x\,1 \end{tikzcd} \quad ✓$$

- LDRF disables "reading the future"
  - Require $(\xrightarrow{\mathsf{hb}} \cup \xrightarrow{\mathsf{wr}})$ acyclic (CAUSALITY)
  - ☹ Requires fences on ARMv8 and PowerPC
  - ☺ $< 1\%$ overhead on ARMv8
  - ☺ Compiler optimization unaffected

Load Buffering

$$\begin{array}{r} q := y;\ x := 1 \\ \parallel r := x;\ y := 1 \end{array} \quad \overset{?}{\Longrightarrow} \quad \begin{array}{l} x := 1;\ q := y \\ \parallel y := 1;\ r := x \end{array}$$

$$\begin{tikzcd} \mathsf{R}\,y\,1 \arrow[r] & \mathsf{W}\,x\,1 \arrow[dl, "\mathsf{wr}"] \\ \mathsf{R}\,x\,1 \arrow[r] & \mathsf{W}\,y\,1 \arrow[ul, "\mathsf{wr}"'] \end{tikzcd} \quad ✗ \qquad\qquad \begin{tikzcd} \mathsf{W}\,x\,1 \arrow[r] \arrow[dr, "\mathsf{wr}"'] & \mathsf{R}\,y\,1 \\ \mathsf{W}\,y\,1 \arrow[r] \arrow[ur, "\mathsf{wr}"] & \mathsf{R}\,x\,1 \end{tikzcd} \quad ✓$$

- LDRF disables "reading the future"
  - Require $(\xrightarrow{\mathsf{hb}} \cup \xrightarrow{\mathsf{wr}})$ acyclic (CAUSALITY)
  - ☹ Requires fences on ARMv8 and PowerPC
  - ☺ $<$ 1% overhead on ARMv8
  - ☺ Compiler optimization unaffected
  - ☺ Understandable semantics (compare C11, Java)

Our Paper

- Local *Transactional* Race Freedom (LTRF)
  - *Extend LDRF* to handle transactions
  - *Transactional idioms* supported

Our Paper

- Local *Transactional* Race Freedom (LTRF)
  - *Extend LDRF* to handle transactions
  - *Transactional idioms* supported
- Inspiration:
  - Local Data Race Freedom (LDRF):
    - Dolan, Sivaramakrishnan, Madhavapeddy, PLDI 2018
  - Transactions in relaxed memory:
    - Dongol, Jagadeesan, Riely, POPL 2018
    - Chong, Sorensen, Wickerson, PLDI, 2018
  - Safe Privatization in Transactional Memory:
    - Khyzha, Attiya, Gotsman, Rinetzky PPoPP 2018

Our Paper

- Local *Transactional* Race Freedom (LTRF)
  - *Extend LDRF* to handle transactions
  - *Transactional idioms* supported
- Inspiration:
  - Local Data Race Freedom (LDRF):
    - Dolan, Sivaramakrishnan, Madhavapeddy, PLDI 2018
  - Transactions in relaxed memory:
    - Dongol, Jagadeesan, Riely, POPL 2018
    - Chong, Sorensen, Wickerson, PLDI, 2018
  - Safe Privatization in Transactional Memory:
    - Khyzha, Attiya, Gotsman, Rinetzky PPoPP 2018
- This talk:
  - *Implementation* model

Our Paper

- Local *Transactional* Race Freedom (LTRF)
  - *Extend LDRF* to handle transactions
  - *Transactional idioms* supported
- Inspiration:
  - Local Data Race Freedom (LDRF):
    - Dolan, Sivaramakrishnan, Madhavapeddy, PLDI 2018
  - Transactions in relaxed memory:
    - Dongol, Jagadeesan, Riely, POPL 2018
    - Chong, Sorensen, Wickerson, PLDI, 2018
  - Safe Privatization in Transactional Memory:
    - Khyzha, Attiya, Gotsman, Rinetzky PPoPP 2018
- This talk:
  - *Implementation* model
  - *Privatization idiom*

Our Paper

- Local *Transactional* Race Freedom (LTRF)
  - *Extend LDRF* to handle transactions
  - *Transactional idioms* supported
- Inspiration:
  - Local Data Race Freedom (LDRF):
    - Dolan, Sivaramakrishnan, Madhavapeddy, PLDI 2018
  - Transactions in relaxed memory:
    - Dongol, Jagadeesan, Riely, POPL 2018
    - Chong, Sorensen, Wickerson, PLDI, 2018
  - Safe Privatization in Transactional Memory:
    - Khyzha, Attiya, Gotsman, Rinetzky PPoPP 2018
- This talk:
  - *Implementation* model
  - *Privatization idiom*
    - Example of a *mixed* race:

$\boxed{\mathrm{W}x} \overset{\mathrm{hb}}{\not\leftrightarrow} \mathrm{W}x \qquad \boxed{\mathrm{W}x} \overset{\mathrm{hb}}{\not\leftrightarrow} \mathrm{R}x \qquad \boxed{\mathrm{R}x} \overset{\mathrm{hb}}{\not\leftrightarrow} \mathrm{W}x$

Our Paper

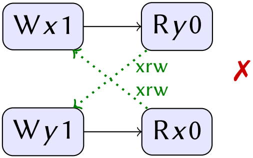
- Local *Transactional* Race Freedom (LTRF)
  - *Extend LDRF* to handle transactions
  - *Transactional idioms* supported
- Inspiration:
  - Local Data Race Freedom (LDRF):
    - Dolan, Sivaramakrishnan, Madhavapeddy, PLDI 2018
  - Transactions in relaxed memory:
    - Dongol, Jagadeesan, Riely, POPL 2018
    - Chong, Sorensen, Wickerson, PLDI, 2018
  - Safe Privatization in Transactional Memory:
    - Khyzha, Attiya, Gotsman, Rinetzky PPoPP 2018
- This talk:
  - *Implementation* model
  - *Privatization idiom*
    - Example of a *mixed* race:

$$\boxed{\mathrm{W}x} \overset{\mathrm{hb}}{\not\leftrightarrow} \mathrm{W}x \qquad \boxed{\mathrm{W}x} \overset{\mathrm{hb}}{\not\leftrightarrow} \mathrm{R}x \qquad \boxed{\mathrm{R}x} \overset{\mathrm{hb}}{\not\leftrightarrow} \mathrm{W}x$$

  - *Programmer* model

Synchronization Via Transactions (Store Buffering)

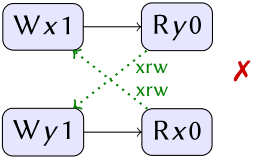
$$\begin{array}{rl} & \text{atomic}\ \{\ x \coloneqq 1\ \};\ \text{atomic}\ \{\ q \coloneqq y\ \} \\ \parallel & \text{atomic}\ \{\ y \coloneqq 1\ \};\ \text{atomic}\ \{\ r \coloneqq x\ \} \end{array}$$



- Strong Serializability
  - Transactions appear sequential
  - Respect program order ("real" time)

Synchronization Via Transactions (Store Buffering)

$$\begin{array}{rl} & \text{atomic}\,\{\, x := 1 \,\};\ \text{atomic}\,\{\, q := y \,\} \\ \parallel & \text{atomic}\,\{\, y := 1 \,\};\ \text{atomic}\,\{\, r := x \,\} \end{array}$$



- Rules:
  - $\xrightarrow{\text{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\text{xwr}})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{\text{hb}} \cup \xrightarrow{\text{xrw}} \cup \xrightarrow{\text{wr}})$ acyclic (CAUSALITY)

Synchronization Via Transactions (Store Buffering)

$$
\begin{array}{ccc}
 & x := 1 \;; & q := y \\
\parallel & y := 1 \;; & r := x
\end{array}
$$

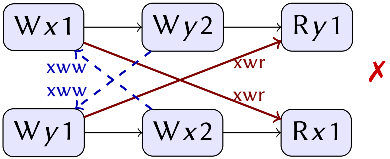
$$
\begin{tikzcd}
\mathsf{W}x1 \arrow[r] & \mathsf{R}y0 \arrow[dl, dotted, "\mathsf{rw}"] \\
\mathsf{W}y1 \arrow[r] & \mathsf{R}x0 \arrow[ul, dotted, "\mathsf{rw}"']
\end{tikzcd}
\quad ✓
$$

- Rules:
  - $\xrightarrow{\mathsf{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\mathsf{xwr}})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{\mathsf{hb}} \cup \overset{\mathsf{xrw}}{\dashrightarrow} \cup \xrightarrow{\mathsf{wr}})$ acyclic (CAUSALITY)
  - $(\xrightarrow{\mathsf{hb}} \;;\; \overset{\mathsf{rw}}{\dashrightarrow})$ irreflexive (OBSERVATION)

    Prevents $\begin{tikzcd} \mathsf{W}x1 \arrow[r] & \mathsf{W}x2 \arrow[r] & \mathsf{R}x1 \arrow[ll, dotted, bend left, "\mathsf{rw}"] \end{tikzcd}$ ✗

2+2W Litmus Test

$$
\begin{array}{rl}
 & \text{atomic}\,\{\, x := 1 \,\};\ \text{atomic}\,\{\, y := 2 \,\};\ \text{atomic}\,\{\, q := y \,\} \\
\parallel & \text{atomic}\,\{\, y := 1 \,\};\ \text{atomic}\,\{\, x := 2 \,\};\ \text{atomic}\,\{\, r := x \,\}
\end{array}
$$



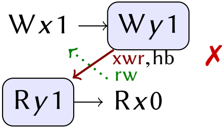
- Rules:
  - $\xrightarrow{\text{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\text{xwr}} \cup \overset{\text{xww}}{\dashrightarrow})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{\text{hb}} \cup \overset{\text{xrw}}{\dashrightarrow} \cup \xrightarrow{\text{wr}})$ acyclic (Causality)
  - $(\xrightarrow{\text{hb}}\ ;\ \overset{\text{rw}}{\dashrightarrow})$ irreflexive (Observation)
  - $(\xrightarrow{\text{hb}}\ ;\ \overset{\text{ww}}{\dashrightarrow})$ irreflexive (Coherence)

  Prevents $Wx1 \underset{\text{ww}}{\rightleftarrows} Wx2$ ✗

Publication

✓*By Dependency*

$x := 1;\ \text{atomic}\ \{\ y := 1\ \}$
$\parallel \text{atomic}\ \{\ q := y\ \};\ r := x$

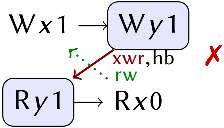


- Rules:
  - $\xrightarrow{\text{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\text{xwr}} \cup \xrightarrow{\text{xww}})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{\text{hb}} \cup \xrightarrow{\text{xrw}} \cup \xrightarrow{\text{wr}})$ acyclic (CAUSALITY)
  - $(\xrightarrow{\text{hb}}\ ;\ \xrightarrow{\text{rw}})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{\text{hb}}\ ;\ \xrightarrow{\text{ww}})$ irreflexive (COHERENCE)

Publication

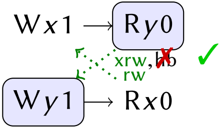
✓*By Dependency*

$x := 1;\ \text{atomic}\ \{\ y := 1\ \}$
$\parallel \text{atomic}\ \{\ q := y\ \};\ r := x$



✗*By Antidependency*

$x := 1;\ \text{atomic}\ \{\ q := y\ \}$
$\parallel \text{atomic}\ \{\ y := 1\ \};\ r := x$



- Rules:
  - $\xrightarrow{\text{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\text{xwr}} \cup \xrightarrow{\text{xww}})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{\text{hb}} \cup \xrightarrow{\text{xrw}} \cup \xrightarrow{\text{wr}})$ acyclic (CAUSALITY)
  - $(\xrightarrow{\text{hb}}\ ;\ \xrightarrow{\text{rw}})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{\text{hb}}\ ;\ \xrightarrow{\text{ww}})$ irreflexive (COHERENCE)

Implementation Model

- Rules:
  - $\xrightarrow{\text{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\text{xwr}} \cup \dashrightarrow^{\text{xww}})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{\text{hb}} \cup \overset{\text{xrw}}{\dashrightarrow} \cup \xrightarrow{\text{wr}})$ acyclic (Causality)
  - $(\xrightarrow{\text{hb}} \; ; \; \overset{\text{rw}}{\dashrightarrow})$ irreflexive (Observation)
  - $(\xrightarrow{\text{hb}} \; ; \; \overset{\text{ww}}{\dashrightarrow})$ irreflexive (Coherence)

Implementation Model

- Rules:
  - $\xrightarrow{\text{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\text{xwr}} \cup \xrightarrow{\text{xww}})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{\text{hb}} \cup \xrightarrow{\text{xrw}} \cup \xrightarrow{\text{wr}})$ acyclic (CAUSALITY)
  - $(\xrightarrow{\text{hb}} \; ; \xrightarrow{\text{rw}})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{\text{hb}} \; ; \xrightarrow{\text{ww}})$ irreflexive (COHERENCE)
- ☺ Satisfies SC-LTRF
- ☺ Validates many transactional idioms
  - Eg, Publication
- ☺ Does not overconstrain implementation
  - Eg, No publication by antidependency

Implementation Model

- Rules:
  - $\xrightarrow{hb}$ includes $(\longrightarrow \cup \xrightarrow{xwr} \cup \xrightarrow{xww})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{hb} \cup \xrightarrow{xrw} \cup \xrightarrow{wr})$ acyclic (Causality)
  - $(\xrightarrow{hb}\ ;\ \xrightarrow{rw})$ irreflexive (Observation)
  - $(\xrightarrow{hb}\ ;\ \xrightarrow{ww})$ irreflexive (Coherence)
- ☺ Satisfies SC-LTRF
- ☺ Validates many transactional idioms
  - Eg, Publication
- ☺ Does not overconstrain implementation
  - Eg, No publication by antidependency
- ☺ Validates reorderings & optimizations (except Load Buffering)
- ☺ Efficient compilation to x86-TSO and ARMv8

Implementation Model

- Rules:
  - $\xrightarrow{hb}$ includes $(\longrightarrow \cup \xrightarrow{xwr} \cup \xrightarrow{xww})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{hb} \cup \xrightarrow{xrw} \cup \xrightarrow{wr})$ acyclic (Causality)
  - $(\xrightarrow{hb}\ ;\ \xrightarrow{rw})$ irreflexive (Observation)
  - $(\xrightarrow{hb}\ ;\ \xrightarrow{ww})$ irreflexive (Coherence)
- ☺ Satisfies SC-LTRF
- ☺ Validates many transactional idioms
  - Eg, Publication
- ☺ Does not overconstrain implementation
  - Eg, No publication by antidependency
- ☺ Validates reorderings & optimizations (except Load Buffering)
- ☺ Efficient compilation to x86-TSO and ARMv8
- ☹ Does not validate *privatization*

Proof Case For SC-LTRF: Switching Reads

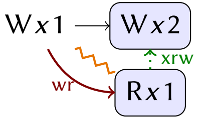
$$\begin{array}{l} \;\; x := 1;\ \text{atomic}\ \{\ x := 2\ \} \\ \parallel \text{atomic}\ \{\ r := x\ \} \end{array}$$

$$\mathsf{W}\,x\,1 \longrightarrow \boxed{\mathsf{W}\,x\,2}$$

- Let ρ be execution of top thread

Proof Case For SC-LTRF: Switching Reads

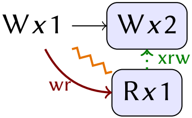
$$x \coloneqq 1;\ \mathsf{atomic}\ \{\ x \coloneqq 2\ \}$$
$$\|\ \mathsf{atomic}\ \{\ r \coloneqq x\ \}$$



- Let ρ be execution of top thread, then add bottom read

Proof Case For SC-LTRF: Switching Reads

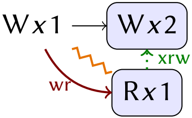
$$x := 1;\ \text{atomic}\ \{\ x := 2\ \}$$
$$\|\ \text{atomic}\ \{\ r := x\ \}$$



- Let ρ be execution of top thread, then add bottom read
- SC-LTRF requires that we find a sequential action with race

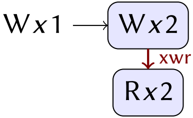
Proof Case For SC-LTRF: Switching Reads

$$x := 1;\ \text{atomic}\ \{\ x := 2\ \}$$
$$\|\ \text{atomic}\ \{\ r := x\ \}$$



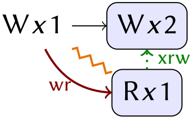
- Let ρ be execution of top thread, then add bottom read
- SC-LTRF requires that we find a sequential action with race

☹ No Race After ρ



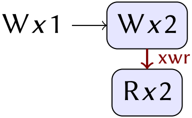
Proof Case For SC-LTRF: Switching Reads

$x := 1$; atomic { $x := 2$ }
$\|$ atomic { $r := x$ }

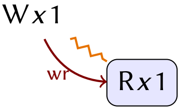


- Let ρ be execution of top thread, then add bottom read
- SC-LTRF requires that we find a sequential action with race

☹ No Race After ρ



☺ Race After Prefix



Privatization

$$\begin{aligned} &\text{atomic}\,\{\ \text{if } !y \text{ then cheap}(x)\ \} \\ \| \,&\text{atomic}\,\{\ y := 1\ \};\ \text{expensive}(x) \end{aligned}$$

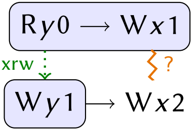
Privatization

$$\begin{array}{l} \quad \text{atomic}\,\{\, \text{if } !y \text{ then } x := 1 \,\} \\ \| \,\text{atomic}\,\{\, y := 1 \,\};\; x := 2 \end{array}$$

- Considered race free

Privatization

$$\begin{array}{l} \,\text{atomic}\,\{\,\text{if}\ !y\ \text{then}\ x \coloneqq 1\,\} \\ \|\,\text{atomic}\,\{\,y \coloneqq 1\,\};\ x \coloneqq 2 \end{array}$$



- Considered race free

Privatization

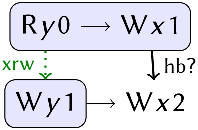
$$\begin{array}{l} \text{atomic}\,\{\,\text{if } !y \text{ then } x := 1\,\} \\ \|\,\text{atomic}\,\{\,y := 1\,\};\ x := 2 \end{array}$$

$$\begin{tikzcd} R\,y0 \arrow[r] \arrow[d, dotted, "\mathrm{xrw}"'] & W\,x1 \arrow[d, leftrightarrow, "\mathrm{hb?}"] \\ W\,y1 \arrow[r] & W\,x2 \end{tikzcd}$$

- Considered race free
- Rules:
  - $\xrightarrow{\mathrm{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\mathrm{xwr}} \cup \dashrightarrow^{\mathrm{xww}})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{\mathrm{hb}} \cup \overset{\mathrm{xrw}}{\dashrightarrow} \cup \xrightarrow{\mathrm{wr}})$ acyclic (Causality)
  - $(\xrightarrow{\mathrm{hb}}\ ;\ \overset{\mathrm{rw}}{\dashrightarrow})$ irreflexive (Observation)
  - $(\xrightarrow{\mathrm{hb}}\ ;\ \overset{\mathrm{ww}}{\dashrightarrow})$ irreflexive (Coherence)

Privatization

$$\begin{array}{l} \text{atomic}\,\{\,\text{if }!y\text{ then }x := 1\,\} \\ \|\,\text{atomic}\,\{\,y := 1\,\};\ x := 2 \end{array}$$



- Considered race free
- Rules:
  - $\xrightarrow{\text{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\text{xwr}} \cup \xrightarrow{\text{xww}})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{\text{hb}} \cup \xrightarrow{\text{xrw}} \cup \xrightarrow{\text{wr}})$ acyclic (CAUSALITY)
  - $(\xrightarrow{\text{hb}}\ ;\ \xrightarrow{\text{rw}})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{\text{hb}}\ ;\ \xrightarrow{\text{ww}})$ irreflexive (COHERENCE)

Privatization

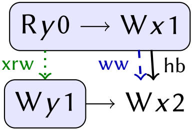
$$\begin{array}{l} \text{atomic}\,\{\ \text{if } !y \text{ then } x := 1\ \} \\ \|\ \text{atomic}\,\{\ y := 1\ \};\ x := 2 \end{array}$$

$$\begin{tikzcd} \boxed{\mathrm{R}\,y\,0 \longrightarrow \mathrm{W}\,x\,1} \arrow[d, dotted, "\mathrm{xrw}"'] \arrow[d, shift left=20, "\cancel{\ }\ \mathrm{hb?}"] \\ \boxed{\mathrm{W}\,y\,1} \longrightarrow \mathrm{W}\,x\,2 \end{tikzcd}$$

- Considered race free
- Rules:
  - $\xrightarrow{\mathrm{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\mathrm{xwr}} \cup \xrightarrow{\mathrm{xww}})$ (HB$_{\mathrm{BASE}}$)
  - $(\xrightarrow{\mathrm{hb}} \cup \xrightarrow{\mathrm{xrw}} \cup \xrightarrow{\mathrm{wr}})$ acyclic (Causality)
  - $(\xrightarrow{\mathrm{hb}}\ ;\ \xrightarrow{\mathrm{rw}})$ irreflexive (Observation)
  - $(\xrightarrow{\mathrm{hb}}\ ;\ \xrightarrow{\mathrm{ww}})$ irreflexive (Coherence)

Privatization

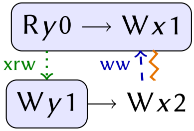
$$\begin{array}{l} \quad \text{atomic}\,\{\,\text{if } !y \text{ then } x := 1\,\} \\ \|\,\text{atomic}\,\{\,y := 1\,\};\ x := 2 \end{array}$$



- Considered race free
- Rules:
  - $\xrightarrow{\text{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\text{xwr}} \cup \xrightarrow{\text{xww}})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{\text{hb}} \cup \xrightarrow{\text{xrw}} \cup \xrightarrow{\text{wr}})$ acyclic (CAUSALITY)
  - $(\xrightarrow{\text{hb}}\ ;\ \xrightarrow{\text{rw}})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{\text{hb}}\ ;\ \xrightarrow{\text{ww}})$ irreflexive (COHERENCE)
  - $\xrightarrow{\text{hb}}$ includes $(\xrightarrow{\text{ww}} \cap (\xrightarrow{\text{xrw}}\ ;\ \xrightarrow{\text{hb}}))$ (HB$_{\text{WW}}$)

Privatization

$$\text{atomic}\,\{\,\text{if } !y \text{ then } x := 1\,\}$$
$$\|\,\text{atomic}\,\{\,y := 1\,\};\ x := 2$$



- Considered race free
- Rules:
  - $\xrightarrow{\text{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\text{xwr}} \cup \xrightarrow{\text{xww}})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{\text{hb}} \cup \xrightarrow{\text{xrw}} \cup \xrightarrow{\text{wr}})$ acyclic (CAUSALITY)
  - $(\xrightarrow{\text{hb}}\ ;\ \xrightarrow{\text{rw}})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{\text{hb}}\ ;\ \xrightarrow{\text{ww}})$ irreflexive (COHERENCE)
  - $\xrightarrow{\text{hb}}$ includes $(\xrightarrow{\text{ww}} \cap (\xrightarrow{\text{xrw}}\ ;\ \xrightarrow{\text{hb}}))$ (HB$_{\text{WW}}$)
- SC-LTRF requires we find SC execution with a race

Privatization

atomic { if !y then $x := 1$ }
‖ atomic { $y := 1$ }; $x := 2$

$$\begin{tikzcd}
\boxed{R\,y\,0 \longrightarrow W\,x\,1} \arrow[d, dotted, "\mathrm{xrw}"'] & \\
\boxed{W\,y\,1} \arrow[r] & W\,x\,2 \arrow[ul, dashed, "\mathrm{ww}"]
\end{tikzcd}$$

- Considered race free
- Rules:
  - $\xrightarrow{\mathrm{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\mathrm{xwr}} \cup \xrightarrow{\mathrm{xww}})$ (HB$_{\text{BASE}}$)
  - $(\xrightarrow{\mathrm{hb}} \cup \xrightarrow{\mathrm{xrw}} \cup \xrightarrow{\mathrm{wr}})$ acyclic (CAUSALITY)
  - $(\xrightarrow{\mathrm{hb}}\ ;\ \xrightarrow{\mathrm{rw}})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{\mathrm{hb}}\ ;\ \xrightarrow{\mathrm{ww}})$ irreflexive (COHERENCE)
  - $\xrightarrow{\mathrm{hb}}$ includes $(\xrightarrow{\mathrm{ww}} \cap (\xrightarrow{\mathrm{xrw}}\ ;\ \xrightarrow{\mathrm{hb}}))$ (HB$_{\text{WW}}$)
- SC-LTRF requires we find SC execution with a race
  $R\,y\,0 \xrightarrow{\mathrm{xrw}} W\,y\,1 \Longrightarrow W\,x\,1 \xrightarrow{\mathrm{ww}} W\,x\,2$

Privatization

$$\begin{array}{l} \text{atomic}\,\{\,\text{if } !y \text{ then } x := 1\,\} \\ \|\,\text{atomic}\,\{\,y := 1\,\};\ x := 2 \end{array}$$

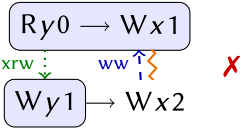
$$\begin{tikzcd} \boxed{\mathsf{R}y0 \longrightarrow \mathsf{W}x1} \arrow[d, dotted, "\mathsf{xrw}"'] & \\ \boxed{\mathsf{W}y1} \arrow[r] & \mathsf{W}x2 \arrow[ul, dashed, "\mathsf{ww}"'] \end{tikzcd}$$

- Considered race free
- Rules:
  - $\xrightarrow{\mathsf{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\mathsf{xwr}} \cup \dashrightarrow^{\mathsf{xww}})$ ($\mathrm{HB_{BASE}}$)
  - $(\xrightarrow{\mathsf{hb}} \cup \overset{\mathsf{xrw}}{\dashrightarrow} \cup \xrightarrow{\mathsf{wr}})$ acyclic (CAUSALITY)
  - $(\xrightarrow{\mathsf{hb}}\ ;\ \overset{\mathsf{rw}}{\dashrightarrow})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{\mathsf{hb}}\ ;\ \overset{\mathsf{ww}}{\dashrightarrow})$ irreflexive (COHERENCE)
  - $\xrightarrow{\mathsf{hb}}$ includes $(\overset{\mathsf{ww}}{\dashrightarrow} \cap (\overset{\mathsf{xrw}}{\dashrightarrow}\ ;\ \xrightarrow{\mathsf{hb}}))$ ($\mathrm{HB_{WW}}$)

- SC-LTRF requires we find SC execution with a race
  $\mathsf{R}y0 \overset{\mathsf{xrw}}{\dashrightarrow} \mathsf{W}y1 \Longrightarrow \mathsf{W}x1 \overset{\mathsf{ww}}{\dashrightarrow} \mathsf{W}x2 \Longrightarrow \mathsf{W}x1 \xrightarrow{\mathsf{hb}} \mathsf{W}x2$ ☹

Privatization

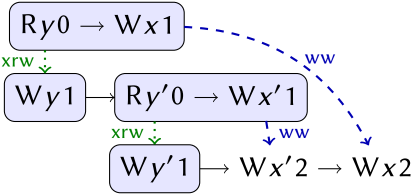
$$\text{atomic}\,\{\,\text{if } !y \text{ then } x := 1\,\}$$
$$\|\,\text{atomic}\,\{\,y := 1\,\};\ x := 2$$



- Considered race free
- Rules:
  - $\xrightarrow{\text{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\text{xwr}} \cup \xrightarrow{\text{xww}})$ $\quad(\text{HB}_{\text{BASE}})$
  - $(\xrightarrow{\text{hb}} \cup \xrightarrow{\text{xrw}} \cup \xrightarrow{\text{wr}})$ acyclic $\quad(\text{CAUSALITY})$
  - $(\xrightarrow{\text{hb}}\ ;\ \xrightarrow{\text{rw}})$ irreflexive $\quad(\text{OBSERVATION})$
  - $(\xrightarrow{\text{hb}}\ ;\ \xrightarrow{\text{ww}})$ irreflexive $\quad(\text{COHERENCE})$
  - $\xrightarrow{\text{hb}}$ includes $(\xrightarrow{\text{ww}} \cap (\xrightarrow{\text{xrw}}\ ;\ \xrightarrow{\text{hb}}))$ $\quad(\text{HB}_{\text{WW}})$
  - $(\xrightarrow{\text{xrw}}; \xrightarrow{\text{hb}}; \xrightarrow{\text{ww}})$ irreflexive $\quad(\text{ANTI}_{\text{WW}})$
- SC-LTRF requires we find SC execution with a race
  $\mathsf{R}y0 \xrightarrow{\text{xrw}} \mathsf{W}y1 \Longrightarrow \mathsf{W}x1 \xrightarrow{\text{ww}} \mathsf{W}x2 \Longrightarrow \mathsf{W}x1 \xrightarrow{\text{hb}} \mathsf{W}x2$ ☹

Privatization: Order Can Cascade

$$
\begin{array}{ll}
 & \mathsf{atomic}\,\{\,\mathsf{if}\ !y\ \mathsf{then}\ x := 1\,\} \\
\parallel & \mathsf{atomic}\,\{\,y := 1\,\};\ \mathsf{atomic}\,\{\,\mathsf{if}\ !y'\ \mathsf{then}\ x' := 1\,\} \\
\parallel & \mathsf{atomic}\,\{\,y' := 1\,\};\ x' := 2;\ x := 2
\end{array}
$$



- Rules:
  - $\xrightarrow{\mathsf{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\mathsf{xwr}} \cup \xrightarrow{\mathsf{xww}})$ (HB$_{\mathsf{BASE}}$)
  - $\xrightarrow{\mathsf{hb}}$ includes $(\xrightarrow{\mathsf{ww}} \cap (\xrightarrow{\mathsf{xrw}}\ ;\ \xrightarrow{\mathsf{hb}}))$ (HB$_{\mathsf{WW}}$)
  - $(\xrightarrow{\mathsf{hb}} \cup \xrightarrow{\mathsf{xrw}} \cup \xrightarrow{\mathsf{wr}})$ acyclic (CAUSALITY)
  - $(\xrightarrow{\mathsf{hb}}\ ;\ \xrightarrow{\mathsf{rw}})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{\mathsf{hb}}\ ;\ \xrightarrow{\mathsf{ww}})$ irreflexive (COHERENCE)
  - $(\xrightarrow{\mathsf{xrw}};\xrightarrow{\mathsf{hb}};\xrightarrow{\mathsf{ww}})$ irreflexive (ANTI$_{\mathsf{WW}}$)

Programmer Model

- Rules:
  - $\xrightarrow{hb}$ includes $(\longrightarrow \cup \xrightarrow{xwr} \cup \xrightarrow{xww})$ (HB$_{\text{BASE}}$)
  - $\xrightarrow{hb}$ includes $(\xrightarrow{ww} \cap (\xrightarrow{xrw} \; ; \; \xrightarrow{hb}))$ (HB$_{\text{WW}}$)
  - $(\xrightarrow{hb} \cup \xrightarrow{xrw} \cup \xrightarrow{wr})$ acyclic (CAUSALITY)
  - $(\xrightarrow{hb} \; ; \; \xrightarrow{rw})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{hb} \; ; \; \xrightarrow{ww})$ irreflexive (COHERENCE)
  - $(\xrightarrow{xrw} ; \xrightarrow{hb} ; \xrightarrow{ww})$ irreflexive (ANTI$_{\text{WW}}$)

Programmer Model

- Rules:
  - $\xrightarrow{hb}$ includes $(\longrightarrow \cup \xrightarrow{xwr} \cup \xrightarrow{xww})$ (HB$_{\text{BASE}}$)
  - $\xrightarrow{hb}$ includes $(\xrightarrow{ww} \cap (\xrightarrow{xrw}\ ;\ \xrightarrow{hb}))$ (HB$_{\text{WW}}$)
  - $(\xrightarrow{hb} \cup \xrightarrow{xrw} \cup \xrightarrow{wr})$ acyclic (CAUSALITY)
  - $(\xrightarrow{hb}\ ;\ \xrightarrow{rw})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{hb}\ ;\ \xrightarrow{ww})$ irreflexive (COHERENCE)
  - $(\xrightarrow{xrw}; \xrightarrow{hb}; \xrightarrow{ww})$ irreflexive (ANTI$_{\text{WW}}$)
- Satisfies SC-LTRF
- Validates ~~many~~ *more* transactional idioms
  - Eg, Publication, *Privatization*
- Does not overconstrain implementation
  - Eg, No publication by antidependency

Programmer Model

- Rules:
  - $\xrightarrow{hb}$ includes $(\longrightarrow \cup \xrightarrow{xwr} \cup \xrightarrow{xww})$ (HB$_{\text{BASE}}$)
  - $\xrightarrow{hb}$ includes $(\xrightarrow{ww} \cap (\xrightarrow{xrw} \ ; \ \xrightarrow{hb}))$ (HB$_{\text{WW}}$)
  - $(\xrightarrow{hb} \cup \xrightarrow{xrw} \cup \xrightarrow{wr})$ acyclic (CAUSALITY)
  - $(\xrightarrow{hb} \ ; \ \xrightarrow{rw})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{hb} \ ; \ \xrightarrow{ww})$ irreflexive (COHERENCE)
  - $(\xrightarrow{xrw}; \xrightarrow{hb}; \xrightarrow{ww})$ irreflexive (ANTI$_{\text{WW}}$)
- Satisfies SC-LTRF
- Validates ~~many~~ *more* transactional idioms
  - Eg, Publication, *Privatization*
- Does not overconstrain implementation
  - Eg, No publication by antidependency
- Overtuned to one idiom?

Programmer Model

- Rules:
  - $\xrightarrow{\text{hb}}$ includes $(\longrightarrow \cup \xrightarrow{\text{xwr}} \cup \xrightarrow{\text{xww}})$ (HB$_{\text{BASE}}$)
  - $\xrightarrow{\text{hb}}$ includes $(\xrightarrow{\text{ww}} \cap (\xrightarrow{\text{xrw}} \,;\, \xrightarrow{\text{hb}}))$ (HB$_{\text{WW}}$)
  - $(\xrightarrow{\text{hb}} \cup \xrightarrow{\text{xrw}} \cup \xrightarrow{\text{wr}})$ acyclic (CAUSALITY)
  - $(\xrightarrow{\text{hb}} \,;\, \xrightarrow{\text{rw}})$ irreflexive (OBSERVATION)
  - $(\xrightarrow{\text{hb}} \,;\, \xrightarrow{\text{ww}})$ irreflexive (COHERENCE)
  - $(\xrightarrow{\text{xrw}}; \xrightarrow{\text{hb}}; \xrightarrow{\text{ww}})$ irreflexive (ANTI$_{\text{WW}}$)

Satisfies SC-LTRF

Validates ~~many~~ *more* transactional idioms

- Eg, Publication, *Privatization*

Does not overconstrain implementation

- Eg, No publication by antidependency

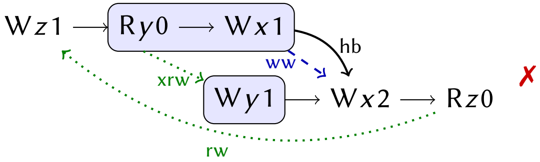
Overtuned to one idiom?

Validates reorderings & optimizations (except Load Buffering)

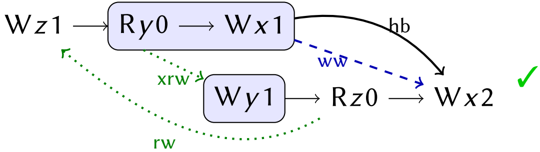
Efficient compilation to x86-TSO and ARMv8

Programmer Model Invalidates Store Buffering

$z := 1;\ \text{atomic}\ \{\ \text{if}\ !y\ \text{then}\ x := 1\ \}$
$\parallel\ \text{atomic}\ \{\ y := 1\ \};\ x := 2;\ r := z$

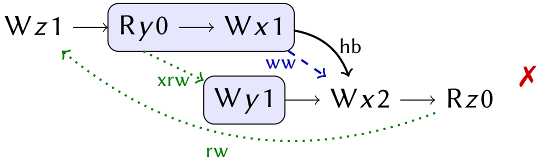


$z := 1;\ \text{atomic}\ \{\ \text{if}\ !y\ \text{then}\ x := 1\ \}$
$\parallel\ \text{atomic}\ \{\ y := 1\ \};\ r := z;\ x := 2$

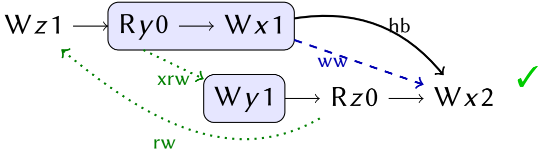


Programmer Model Invalidates Store Buffering

$$
\begin{aligned}
&z := 1;\ \mathsf{atomic}\,\{\,\mathsf{if}\ !y\ \mathsf{then}\ x := 1\,\}\\
\|\ &\mathsf{atomic}\,\{\,y := 1\,\};\ x := 2;\ r := z
\end{aligned}
$$



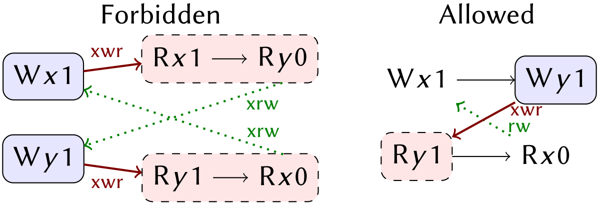
$$
\begin{aligned}
&z := 1;\ \mathsf{atomic}\,\{\,\mathsf{if}\ !y\ \mathsf{then}\ x := 1\,\}\\
\|\ &\mathsf{atomic}\,\{\,y := 1\,\};\ r := z;\ x := 2
\end{aligned}
$$



In Paper

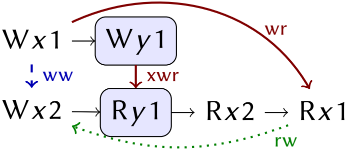
- Details
  - Lifting
  - Aborted/Live transactions
- Programmer Model $\Rightarrow$ Implementation Model
  - Quiescent Fences
- Variant Programmer Models

Aborted Transactions

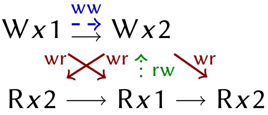


Coherence

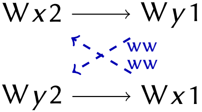
✗ Us, ✓ Java



✓ Us, ✗ C11



✓ Us, ✓ Java, ✓ C11

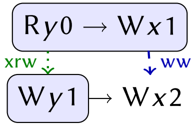


WW Variants

$$\xrightarrow{\mathsf{hb}} \text{ includes } \overset{\mathsf{ww}}{\dashrightarrow} \cap \left(\overset{\mathsf{xrw}}{\dashrightarrow}; \xrightarrow{\mathsf{hb}}\right) \qquad (\mathrm{HB}_{\mathsf{ww}})$$

$$\left(\overset{\mathsf{xrw}}{\dashrightarrow}; \xrightarrow{\mathsf{hb}}; \overset{\mathsf{ww}}{\dashrightarrow}\right) \text{ is irreflexive.} \qquad (\textsc{Anti}_{\mathsf{ww}})$$

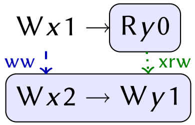
$$\begin{aligned} &\mathsf{atomic}\,\{\, r := y;\ x := 1 \,\} \\ \parallel\ &\mathsf{atomic}\,\{\, y := 1 \,\};\ x := 2 \end{aligned}$$



$$\xrightarrow{\mathsf{hb}} \text{ includes } \overset{\mathsf{ww}}{\dashrightarrow} \cap \left(\xrightarrow{\mathsf{hb}}; \overset{\mathsf{xrw}}{\dashrightarrow}\right) \qquad (\mathrm{HB}'_{\mathsf{ww}})$$

$$\left(\xrightarrow{\mathsf{hb}}; \overset{\mathsf{xrw}}{\dashrightarrow}; \overset{\mathsf{ww}}{\dashrightarrow}\right) \text{ is irreflexive.} \qquad (\textsc{Anti}'_{\mathsf{ww}})$$

$$\begin{aligned} &x := 1;\ \mathsf{atomic}\,\{\, r := y \,\} \\ \parallel\ &\mathsf{atomic}\,\{\, x := 2;\ y := 1 \,\} \end{aligned}$$



RW Variants

$$\xrightarrow{\mathsf{hb}} \text{ includes } \overset{\mathsf{rw}}{\dashrightarrow} \cap \left(\overset{\mathsf{xrw}}{\dashrightarrow}; \xrightarrow{\mathsf{hb}}\right) \tag{$\mathrm{HB}_{\mathrm{RW}}$}$$

$$\left(\overset{\mathsf{xrw}}{\dashrightarrow}; \xrightarrow{\mathsf{hb}}; \overset{\mathsf{rw}}{\dashrightarrow}\right) \text{ is irreflexive} \tag{$\textsc{Anti}_{\mathrm{RW}}$}$$

$$\begin{aligned} &\text{atomic}\,\{\, r := y;\ q := x \,\} \\ \parallel\ &\text{atomic}\,\{\, y := 1 \,\};\ x := 1 \end{aligned}$$

```tikzcd
\mathrm{R}y0 \arrow[r] \arrow[d, dashed, "\mathsf{xrw}"'] & \mathrm{R}x0 \arrow[d, dashed, "\mathsf{rw}"] \\
\mathrm{W}y1 \arrow[r] & \mathrm{W}x1
```

$$\xrightarrow{\mathsf{hb}} \text{ includes } \overset{\mathsf{rw}}{\dashrightarrow} \cap \left(\xrightarrow{\mathsf{hb}}; \overset{\mathsf{xrw}}{\dashrightarrow}\right) \tag{$\mathrm{HB}'_{\mathrm{RW}}$}$$

$$\left(\xrightarrow{\mathsf{hb}}; \overset{\mathsf{xrw}}{\dashrightarrow}; \overset{\mathsf{rw}}{\dashrightarrow}\right) \text{ is irreflexive.} \tag{$\textsc{Anti}'_{\mathrm{RW}}$}$$

$$\begin{aligned} &q := x;\ \text{atomic}\,\{\, r := y \,\} \\ \parallel\ &\text{atomic}\,\{\, x := 1;\ y := 1 \,\} \end{aligned}$$

```tikzcd
\mathrm{R}x0 \arrow[r] \arrow[d, dashed, "\mathsf{rw}"'] & \mathrm{R}y0 \arrow[d, dashed, "\mathsf{xrw}"] \\
\mathrm{W}x1 \arrow[r] & \mathrm{W}y1
```

WR Variants

$$\xrightarrow{\mathsf{hb}} \text{ includes } \xrightarrow{\mathsf{wr}} \cap \left(\overset{\mathsf{xrw}}{\dashrightarrow} ; \xrightarrow{\mathsf{hb}}\right) \qquad (\mathrm{HB}_{\mathrm{WR}})$$

$$\begin{array}{l} \ \text{atomic}\,\{\, r := y;\ x := 1 \,\} \\ \parallel\ \text{atomic}\,\{\, y := 1 \,\};\ q := x \end{array}$$

$$\begin{tikzcd} \mathrm{R}\,y\,0 \arrow[r] \arrow[d, dotted, "\mathsf{xrw}"'] & \mathrm{W}\,x\,1 \arrow[d, "\mathsf{wr}"] \\ \boxed{\mathrm{W}\,y\,1} \arrow[r] & \mathrm{R}\,x\,1 \end{tikzcd}$$

$$\xrightarrow{\mathsf{hb}} \text{ includes } \xrightarrow{\mathsf{wr}} \cap \left(\xrightarrow{\mathsf{hb}} ; \overset{\mathsf{xrw}}{\dashrightarrow}\right) \qquad (\mathrm{HB}'_{\mathrm{WR}})$$

$$\begin{array}{l} \ x := 1;\ \text{atomic}\,\{\, r := y \,\} \\ \parallel\ \text{atomic}\,\{\, q := x;\ y := 1 \,\} \end{array}$$

$$\begin{tikzcd} \mathrm{W}\,x\,1 \arrow[r] \arrow[d, "\mathsf{wr}"'] & \boxed{\mathrm{R}\,y\,0} \arrow[d, dotted, "\mathsf{xrw}"] \\ \mathrm{R}\,x\,1 \arrow[r] & \mathrm{W}\,y\,1 \end{tikzcd}$$