Pomsets with Preconditions

A Simple Model of Relaxed Memory

Radha Jagadeesan[†] Alan Jeffrey[*] James Riely[†]

[†]DePaul University

[*]Mozilla Research and the Servo Project

OOPSLA
November 2020

Memory model

What value can be read for x?

$$x := 1\,;\, x := 2 \qquad \parallel \quad x := 3\,;\, x := 4\,; \qquad s := x\,;\, x := 5$$

$$\begin{tikzcd}
\boxed{Wx1} \arrow[r, dotted, "\mathsf{po}"] & \boxed{Wx2} & & \\
\boxed{Wx3} \arrow[r, dotted, "\mathsf{po}"] & \boxed{Wx4} \arrow[r, dotted, "\mathsf{po}"] & \boxed{Rx?} \arrow[r, dotted, "\mathsf{po}"] & \boxed{Wx5}
\end{tikzcd}$$

- x–z: shared locations, initially 0
- r–s: registers
- po: program order

- Concurrent: 1? 2?
- Sequential: 3? 4? 5?

Memory model

What value can be read for x?

$$x := 1;\ x := 2 \qquad \|\ \ x := 3;\ x := 4; \qquad s := x;\ x := 5$$

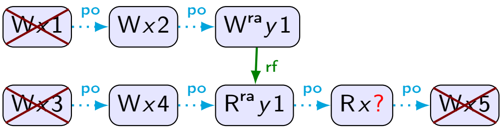
- x–z: shared locations, initially 0
- r–s: registers
- po: program order

- Concurrent: 1✓ 2✓
- Sequential: 3✗ 4✓ 5✗

Memory model

What value can be read for x?

$$x := 1\,;\; x := 2\,;\; y^{\mathsf{ra}} := 1 \;\parallel\; x := 3\,;\; x := 4\,;\; r := y^{\mathsf{ra}}\,;\; s := x\,;\; x := 5$$



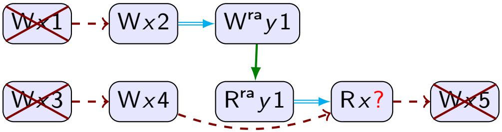
- x–z: shared locations, initially 0
- r–s: registers
- po: program order
- rf: reads-from
- W^{ra}: release
- R^{ra}: acquire

- Concurrent: 1✗ 2✓
- Sequential: 3✗ 4✓ 5✗

Memory model

What value can be read for x?

$$x := 1\,;\ x := 2\,;\ y^{\mathrm{ra}} := 1 \ \parallel\ x := 3\,;\ x := 4\,;\ r := y^{\mathrm{ra}}\,;\ s := x\,;\ x := 5$$

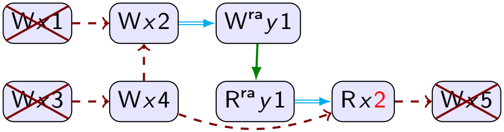


- Labelled partial order (Pomset)
  - $\dashrightarrow$ po $\cap\, x$
  - $\Rightarrow$ po in to release
  - $\Rightarrow$ po out of acquire
- Fulfillment
  - $\rightarrow$ $(\mathrm{W}\,x\,v)$ before $(\mathrm{R}\,x\,v)$
  - $\dashrightarrow$ Any other $(\mathrm{W}\,x)$ before $(\mathrm{W}\,x\,v)$ or after $(\mathrm{R}\,x\,v)$

Memory model

What value can be read for x?

$$x := 1\,;\ x := 2\,;\ y^{\mathsf{ra}} := 1 \ \parallel\ x := 3\,;\ x := 4\,;\ r := y^{\mathsf{ra}}\,;\ s := x\,;\ x := 5$$

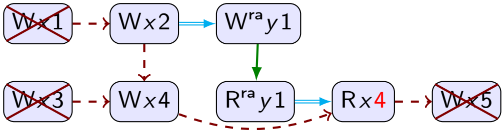


- Labelled partial order (Pomset)
  - po $\cap$ x
  - po in to release
  - po out of acquire
- Fulfillment
  - (W x v) before (R x v)
  - Any other (W x) before (W x v) or after (R x v)

Memory model

What value can be read for x?

$$x := 1\,;\ x := 2\,;\ y^{\mathsf{ra}} := 1 \ \parallel\ x := 3\,;\ x := 4\,;\ r := y^{\mathsf{ra}}\,;\ \color{red}{s := x}\,;\ x := 5$$

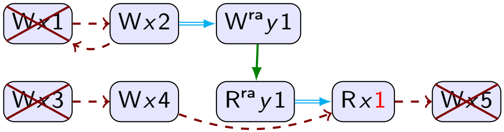


- Labelled partial order (Pomset)
  - $\dashrightarrow$ po $\cap\ x$
  - $\Rightarrow$ po in to release
  - $\Rightarrow$ po out of acquire
- Fulfillment
  - $\rightarrow$ (W$x\,v$) before (R$x\,v$)
  - $\dashrightarrow$ Any other (Wx) before (W$x\,v$) or after (R$x\,v$)

Memory model

What value can be read for x?

$$x := 1\,;\; x := 2\,;\; y^{\mathrm{ra}} := 1 \;\parallel\; x := 3\,;\; x := 4\,;\; r := y^{\mathrm{ra}}\,;\; s := x\,;\; x := 5$$

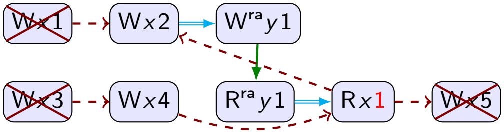


- Labelled partial order (Pomset)
  - po ∩ x
  - po in to release
  - po out of acquire
- Fulfillment
  - (W x v) before (R x v)
  - Any other (W x) before (W x v) or after (R x v)

Memory model

What value can be read for x?

$$x := 1\,;\, x := 2\,;\, y^{\mathrm{ra}} := 1 \;\parallel\; x := 3\,;\, x := 4\,;\, r := y^{\mathrm{ra}}\,;\, \color{red}{s := x}\,;\, x := 5$$



- Labelled partial order (Pomset)
  - po $\cap\, x$
  - po in to release
  - po out of acquire
- Fulfillment
  - (W x v) before (R x v)
  - Any other (W x) before (W x v) or after (R x v)


PRETTY
Image in the Public domain

Preconditions for dependency

Pomset = a single execution

$$s := x;\ y := s \ \parallel\ r := y;\ x := 1;\ z := r$$

$\boxed{\mathsf{R}x1}$ $\boxed{s{=}1 \mid \mathsf{W}y1}$ $\boxed{\mathsf{R}y1}$ $\boxed{\mathsf{true} \mid \mathsf{W}x1}$ $\boxed{r{=}1 \mid \mathsf{W}z1}$

- Writes have preconditions

Preconditions for dependency

Pomset = a single execution

$$s := x\,;\ y := s \ \parallel\ r := y\,;\ x := 1\,;\ z := r$$

$$\boxed{\mathsf{R}x1} \longrightarrow \boxed{1{=}1 \mid \mathsf{W}y1} \qquad \boxed{\mathsf{R}y1} \quad \boxed{\mathsf{true} \mid \mathsf{W}x1} \quad \boxed{r{=}1 \mid \mathsf{W}z1}$$

- Writes have preconditions
- Order from $s := (\mathsf{R}x1)$ substitutes $[1/s]$

Preconditions for dependency

Pomset = a single execution

$$s := x\,;\, y := s \;\parallel\; r := y\,;\, x := 1\,;\, z := r$$

$$\begin{tikzcd}
\boxed{\mathsf{R}x1} \arrow[r] & \boxed{1{=}1 \mid \mathsf{W}y1} & & \boxed{\mathsf{R}y1} \arrow[rr, bend right] & \boxed{\mathsf{true} \mid \mathsf{W}x1} & \boxed{1{=}1 \mid \mathsf{W}z1}
\end{tikzcd}$$

- Writes have preconditions
- Order from $s := (\mathsf{R}x1)$ substitutes $[1/s]$
- Order from $r := (\mathsf{R}y1)$ substitutes $[1/r]$

Preconditions for dependency

Pomset = a single execution

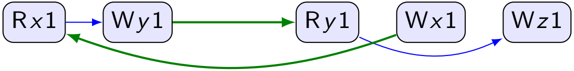
$$y := x \parallel r := y\,;\, x := 1\,;\, z := r$$

- Writes have preconditions
- Order from $s := (\mathsf{R}\,x\,1)$ substitutes $[1/s]$
- Order from $r := (\mathsf{R}\,y\,1)$ substitutes $[1/r]$
- We elide tautologies

Out of order

What value can be written to z?

$$y := x \parallel r := y;\ x := 1;\ z := r \qquad (\star) \quad ✓$$

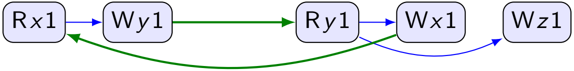


- Partial order
  - per-location
  - synchronization
  - reads-from
  - dependency
- May reorder $r := y$ and $x := 1$
  - in compiler
  - on ARM

Out of ~~order~~ thin air (OOTA)

What value can be written to z?

$$y := x \parallel r := y\,;\, x := \color{red}{r}\,;\, z := r \qquad \text{(OOTA1)} \ \color{red}{\text{✗}}$$

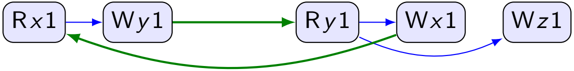


- Partial order
  - per-location
  - synchronization
  - reads-from
  - dependency
- ~~May reorder $r := y$ and $x := 1$~~
  - ~~in compiler~~
  - ~~on ARM~~

Out of ~~order~~ thin air (OOTA)

What value can be written to z?

$$y := x \;\|\; r := y\,;\; x := {\color{red}r}\,;\; z := r \qquad \text{(OOTA1)} \quad ✗$$

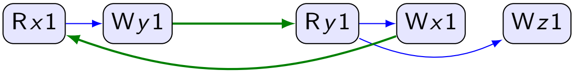


- Partial order
  - per-location
  - synchronization
  - reads-from
  - dependency
- ~~May reorder $r := y$ and $x := 1$~~
  - ~~in compiler~~
  - ~~on ARM~~
- Partial order prevents!

Out of ~~order~~ thin air (OOTA)

What value can be written to z?

$\mathtt{if}\,(x)\,\{y := 1\} \parallel r := y;\ \mathtt{if}\,(r)\,\{x := 1;\ z := r\}$ (OOTA2) ✗



- Partial order
  - per-location
  - synchronization
  - reads-from
  - dependency

- ~~May reorder $r := y$ and $x := 1$~~
  - ~~in compiler~~
  - ~~on ARM~~
- Partial order prevents!
- Control flow variant is DRF


VERY
PRETTY

Can this program write 1 to z?

$$y := x \parallel r := y; \qquad x := r; z := r \tag{OOTA1}$$

Can this program write 1 to z?

$$y := x \parallel r := y\,;\, \texttt{if}\,(r)\,\{x := r\,;\, z := r\}\,\texttt{else}\,\{x := 1\} \qquad (\star)$$

Can this program write 1 to z?

$$y := x \parallel r := y;\ \mathtt{if}\,(r)\,\{x := r;\ z := r\}\ \mathtt{else}\ \{x := 1\} \qquad (\star)$$

x and y can only be 0 or 1

$$y := x \parallel r := y;\ \mathtt{if}\,(r)\,\{x := 1;\ z := 1\}\ \mathtt{else}\ \{x := 1\}$$

Can this program write 1 to z?

$$y := x \;\|\; r := y;\, \texttt{if}\,(r)\,\{x := r;\, z := r\}\,\texttt{else}\,\{x := 1\} \qquad (\star)$$

x and y can only be 0 or 1

$$y := x \;\|\; r := y;\, \texttt{if}\,(r)\,\{x := 1;\, z := 1\}\,\texttt{else}\,\{x := 1\}$$

lift common code

$$y := x \;\|\; r := y;\, x := 1;\, \texttt{if}\,(r)\,\{z := 1\}$$

Can this program write 1 to z?

$$y := x \parallel r := y;\ \mathtt{if}\,(r)\,\{x := r;\ z := r\}\ \mathtt{else}\,\{x := 1\} \qquad (\star)$$

x and y can only be 0 or 1

$$y := x \parallel r := y;\ \mathtt{if}\,(r)\,\{x := 1;\ z := 1\}\ \mathtt{else}\,\{x := 1\}$$

lift common code

$$y := x \parallel \color{red}{r := y};\ \color{red}{x := 1};\ \mathtt{if}\,(r)\,\{z := 1\}$$

commute independent statements

$$y := x \parallel \color{red}{x := 1};\ \color{red}{r := y};\ \mathtt{if}\,(r)\,\{z := 1\}$$

Can this program write 1 to z?

$$y := x \parallel r := y;\, \texttt{if}\,(r)\,\{x := r;\, z := r\}\,\texttt{else}\,\{x := 1\} \qquad (\star)$$

x and y can only be 0 or 1

$$y := x \parallel r := y;\, \texttt{if}\,(r)\,\{x := 1;\, z := 1\}\,\texttt{else}\,\{x := 1\}$$

lift common code

$$y := x \parallel r := y;\, x := 1;\, \texttt{if}\,(r)\,\{z := 1\}$$

commute independent statements

$$\color{red}{y := x} \parallel x := 1;\, r := y;\, \texttt{if}\,(r)\,\{z := 1\}$$

interleaving

$$x := 1;\, \color{red}{y := x};\, r := y;\, \texttt{if}\,(r)\,\{z := 1\}$$

Can this program write 1 to z?

$$y := x \parallel r := y;\ \texttt{if}\,(r)\,\{x := r;\ z := r\}\ \texttt{else}\ \{x := 2\} \qquad \text{(OOTA3)}$$

x and y can only be 0 or 2

$$y := x \parallel r := y;\ \texttt{if}\,(r)\,\{x := 2;\ z := 2\}\ \texttt{else}\ \{x := 2\}$$

lift common code

$$y := x \parallel r := y;\ x := 2;\ \texttt{if}\,(r)\,\{z := 2\}$$

commute independent statements

$$y := x \parallel x := 2;\ r := y;\ \texttt{if}\,(r)\,\{z := 2\}$$

interleaving

$$x := 2;\ y := x;\ r := y;\ \texttt{if}\,(r)\,\{z := 2\}$$

Can this program write 1 to z?

$$y := x \parallel r := y;\ \texttt{if}(r)\{x := r;\ z := r\}\ \texttt{else}\ \{x := 2\} \qquad \text{(OOTA3)}$$

~~x and y can only be 0 or 1~~

$$\cancel{y := x \parallel r := y;\ \texttt{if}(r)\{x := r;\ z := 1\}\ \texttt{else}\ \{x := 1\}}$$

~~lift common code~~

$$\cancel{y := x \parallel r := y;\ x := 1;\ \texttt{if}(r)\{z := 1\}}$$

~~commute independent statements~~

$$\cancel{y := x \parallel x := 1;\ r := y;\ \texttt{if}(r)\{z := 1\}}$$

~~interleaving~~

$$\cancel{x := 1;\ y := x;\ r := y;\ \texttt{if}(r)\{z := 1\}}$$

Can this program write 1 to z?

$$y := x \;\|\; r := y;\, \mathtt{if}\,(\textcolor{red}{b})\,\{x := r;\, z := r\}\,\mathtt{else}\,\{x := 1\} \;\|\; b := 1 \qquad \text{(OOTA4)}$$

~~x and y can only be 0 or 1~~

$$\cancel{y := x \;\|\; r := y;\, \mathtt{if}\,(r)\,\{x := r;\, z := 1\}\,\mathtt{else}\,\{x := 1\}}$$

~~lift common code~~

$$\cancel{y := x \;\|\; r := y;\, x := 1;\, \mathtt{if}\,(r)\,\{z := 1\}}$$

~~commute independent statements~~

$$\cancel{y := x \;\|\; x := 1;\, r := y;\, \mathtt{if}\,(r)\,\{z := 1\}}$$

~~interleaving~~

$$\cancel{x := 1;\, y := x;\, r := y;\, \mathtt{if}\,(r)\,\{z := 1\}}$$

Can this program write 1 to z?

$$y := x \parallel r := y;\ \mathtt{if}\,(r)\,\{x := r;\ z := r\}\,\mathtt{else}\,\{x := 1\} \tag{$\star$}$$

$$y := x \parallel r := y;\ \mathtt{if}\,(r)\,\{x := r;\ z := r\}\,\mathtt{else}\,\{x := 2\} \tag{OOTA3}$$

$$y := x \parallel r := y;\ \mathtt{if}\,(b)\,\{x := r;\ z := r\}\,\mathtt{else}\,\{x := 1\} \parallel b := 1 \tag{OOTA4}$$

Can this program write 1 to z?

$$y := x \parallel r := y; \texttt{if}(r)\{x := r; z := r\}\,\texttt{else}\,\{x := 1\} \qquad (\star)$$

$$y := x \parallel r := y; \texttt{if}(r)\{x := r; z := r\}\,\texttt{else}\,\{x := 2\} \qquad \text{(OOTA3)}$$

$$y := x \parallel r := y; \texttt{if}(b)\{x := r; z := r\}\,\texttt{else}\,\{x := 1\} \parallel b := 1 \qquad \text{(OOTA4)}$$

- Existential justification: ARM✓ ($\star$)✓ OOTA1-3✓ OOTA4✗
  - Commitment (JMM - Java Memory Model) [Manson, Pugh, Adve, 2005]
  - Speculation [Jagadeesan, Pitcher, Riely, 2010]
  - Promising [Kang, Hur, Lahav, Vafeiadis, Dreyer 2017]

Can this program write 1 to z?

$$y := x \parallel r := y; \texttt{if}\,(r)\,\{x := r; z := r\}\,\texttt{else}\,\{x := 1\} \qquad (\star)$$

$$y := x \parallel r := y; \texttt{if}\,(r)\,\{x := r; z := r\}\,\texttt{else}\,\{x := 2\} \qquad (\text{OOTA3})$$

$$y := x \parallel r := y; \texttt{if}\,(b)\,\{x := r; z := r\}\,\texttt{else}\,\{x := 1\} \parallel b := 1 \qquad (\text{OOTA4})$$

- Existential justification: ARM✓ $(\star)$✓ OOTA1-3✓ OOTA4✗
  - Commitment (JMM - Java Memory Model) [Manson, Pugh, Adve, 2005]
  - Speculation [Jagadeesan, Pitcher, Riely, 2010]
  - Promising [Kang, Hur, Lahav, Vafeiadis, Dreyer 2017]
- Universal justification: ARM✗ $(\star)$✓ OOTA1-3✓ OOTA4✓
  - Event Structures [Jeffrey, Riely, 2016]

Can this program write 1 to z?

$$y := x \parallel r := y\,;\ \mathtt{if}\,(r)\,\{x := r\,;\ z := r\}\,\mathtt{else}\,\{x := 1\} \tag{$\star$}$$

$$y := x \parallel r := y\,;\ \mathtt{if}\,(r)\,\{x := r\,;\ z := r\}\,\mathtt{else}\,\{x := 2\} \tag{OOTA3}$$

$$y := x \parallel r := y\,;\ \mathtt{if}\,(b)\,\{x := r\,;\ z := r\}\,\mathtt{else}\,\{x := 1\} \quad \parallel\ b := 1 \tag{OOTA4}$$

$$y := x \parallel r := y\,;\ \mathtt{if}\,(b)\,\{r := \mathtt{new}\,\mathtt{D}\}\,\mathtt{else}\,\{s := \mathtt{new}\,\mathtt{C}\}\,;\ x := r \ \parallel\ b := 1 \tag{OOTA5}$$

- In the JMM, OOTA5 *"is type correct if it declares x, y and r of type* `D`*. However, it has a legal execution where they reference a* `C` *object."* [Lochbihler 2013]

Can this program write 1 to z?

$$y := x \parallel r := y;\ \texttt{if}\,(r)\,\{x := r;\ z := r\}\,\texttt{else}\,\{x := 1\} \qquad (\star)$$

$$y := x \parallel r := y;\ \texttt{if}\,(r)\,\{x := r;\ z := r\}\,\texttt{else}\,\{x := 2\} \qquad \text{(OOTA3)}$$

$$y := x \parallel r := y;\ \texttt{if}\,(b)\,\{x := r;\ z := r\}\,\texttt{else}\,\{x := 1\} \parallel b := 1 \qquad \text{(OOTA4)}$$

$$y := x \parallel r := y;\ \texttt{if}\,(b)\,\{r := \texttt{new D}\}\,\texttt{else}\,\{s := \texttt{new C}\};\ x := r \parallel b := 1 \qquad \text{(OOTA5)}$$

- In the JMM, OOTA5 *"is type correct if it declares x, y and r of type* `D`. *However, it has a legal execution where they reference a* `C` *object."* [Lochbihler 2013]
  - To prove safety, Lochbihler partitions memory by type

Can this program write 1 to z?

$$y := x \parallel r := y\, ;\, \texttt{if}\,(r)\,\{x := r\, ;\, z := r\}\,\texttt{else}\,\{x := 1\} \qquad (\star)$$

$$y := x \parallel r := y\, ;\, \texttt{if}\,(r)\,\{x := r\, ;\, z := r\}\,\texttt{else}\,\{x := 2\} \qquad \text{(OOTA3)}$$

$$y := x \parallel r := y\, ;\, \texttt{if}\,(b)\,\{x := r\, ;\, z := r\}\,\texttt{else}\,\{x := 1\} \parallel b := 1 \qquad \text{(OOTA4)}$$

$$y := x \parallel r := y\, ;\, \texttt{if}\,(b)\,\{r := \texttt{new D}\}\,\texttt{else}\,\{s := \texttt{new C}\}\, ;\, x := r \parallel b := 1 \qquad \text{(OOTA5)}$$

- In the JMM, OOTA5 *"is type correct if it declares x, y and r of type* `D`*. However, it has a legal execution where they reference a* `C` *object."* [Lochbihler 2013]
  - To prove safety, Lochbihler partitions memory by type
    - ~~Realistic memory allocation~~
    - ~~Java Security Architecture (security-sensitive constants)~~

Can this program write 1 to z?

$$y := x \parallel r := y\,;\ \mathtt{if}\,(r)\,\{x := r\,;\ z := r\}\ \mathtt{else}\,\{x := 1\} \qquad (\star)$$

$$y := x \parallel r := y\,;\ \mathtt{if}\,(r)\,\{x := r\,;\ z := r\}\ \mathtt{else}\,\{x := 2\} \qquad \text{(OOTA3)}$$

$$y := x \parallel r := y\,;\ \mathtt{if}\,(b)\,\{x := r\,;\ z := r\}\ \mathtt{else}\,\{x := 1\} \parallel b := 1 \qquad \text{(OOTA4)}$$

$$y := x \parallel r := y\,;\ \mathtt{if}\,(b)\,\{r := \mathtt{new}\,\mathtt{D}\}\ \mathtt{else}\,\{s := \mathtt{new}\,\mathtt{C}\}\,;\ x := r \parallel b := 1 \qquad \text{(OOTA5)}$$

- In the JMM, OOTA5 *"is type correct if it declares x, y and r of type* `D`*. However, it has a legal execution where they reference a* `C` *object."* [Lochbihler 2013]
  - To prove safety, Lochbihler partitions memory by type
    - ~~Realistic memory allocation~~
    - ~~Java Security Architecture (security-sensitive constants)~~
- *Out Of Thin Air (OOTA)* is a *queasy feeling*

Can this program write 1 to z?

$$y := x \parallel r := y\,;\ \mathtt{if}\,(r)\,\{x := r\,;\ z := r\}\ \mathtt{else}\ \{x := 1\} \qquad (\star)$$

$$y := x \parallel r := y\,;\ \mathtt{if}\,(r)\,\{x := r\,;\ z := r\}\ \mathtt{else}\ \{x := 2\} \qquad \text{(OOTA3)}$$

$$y := x \parallel r := y\,;\ \mathtt{if}\,(b)\,\{x := r\,;\ z := r\}\ \mathtt{else}\ \{x := 1\} \parallel b := 1 \qquad \text{(OOTA4)}$$

$$y := x \parallel r := y\,;\ \mathtt{if}\,(b)\,\{r := \mathtt{new}\,\mathtt{D}\}\ \mathtt{else}\ \{s := \mathtt{new}\,\mathtt{C}\}\,;\ x := r \parallel b := 1 \qquad \text{(OOTA5)}$$

- In the JMM, OOTA5 *"is type correct if it declares x, y and r of type* `D`*. However, it has a legal execution where they reference a* `C` *object."* [Lochbihler 2013]
  - To prove safety, Lochbihler partitions memory by type
    - ~~Realistic memory allocation~~
    - ~~Java Security Architecture (security-sensitive constants)~~
- *Out Of Thin Air (OOTA)* is a *queasy feeling*
  Compositionality of proof rules is a *verifiable property*

Can this program write 1 to z?

$$y := x \parallel r := y;\ \texttt{if}\,(r)\,\{x := r;\ z := r\}\ \texttt{else}\ \{x := 1\} \qquad (\star)$$

$$y := x \parallel r := y;\ \texttt{if}\,(r)\,\{x := r;\ z := r\}\ \texttt{else}\ \{x := 2\} \qquad \text{(OOTA3)}$$

$$y := x \parallel r := y;\ \texttt{if}\,(b)\,\{x := r;\ z := r\}\ \texttt{else}\ \{x := 1\} \parallel b := 1 \qquad \text{(OOTA4)}$$

$$y := x \parallel r := y;\ \texttt{if}\,(b)\,\{r := \texttt{new}\,\mathrm{D}\}\ \texttt{else}\ \{s := \texttt{new}\,\mathrm{C}\};\ x := r \parallel b := 1 \qquad \text{(OOTA5)}$$

- In OOTA4, every thread satisfies:
  - Wy1 must be preceded by Rx1
  - if Wz1, then Wx1 must be preceded by Ry1
  - In PLTL: $[\diamondminus \mathrm{W}y1 \Rightarrow \diamondminus \mathrm{R}x1] \wedge [\mathrm{W}z1 \Rightarrow (\diamondminus \mathrm{R}y1 \wedge \boxminus(\mathrm{W}x1 \Rightarrow \diamondminus \mathrm{R}y1))]$

Can this program write 1 to z?

$$y := x \parallel r := y;\ \mathtt{if}\,(r)\,\{x := r;\ z := r\}\ \mathtt{else}\ \{x := 1\} \qquad (\star)$$

$$y := x \parallel r := y;\ \mathtt{if}\,(r)\,\{x := r;\ z := r\}\ \mathtt{else}\ \{x := 2\} \qquad (\text{OOTA3})$$

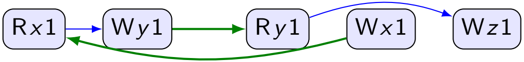
$$y := x \parallel r := y;\ \mathtt{if}\,(b)\,\{x := r;\ z := r\}\ \mathtt{else}\ \{x := 1\} \parallel b := 1 \qquad (\text{OOTA4})$$

$$y := x \parallel r := y;\ \mathtt{if}\,(b)\,\{r := \mathtt{new}\,\mathtt{D}\}\ \mathtt{else}\ \{s := \mathtt{new}\,\mathtt{C}\};\ x := r \parallel b := 1 \qquad (\text{OOTA5})$$

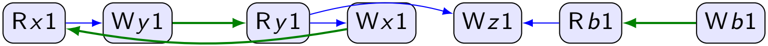
- In OOTA4, every thread satisfies:
  - $\mathrm{W}y1$ must be preceded by $\mathrm{R}x1$
  - if $\mathrm{W}z1$, then $\mathrm{W}x1$ must be preceded by $\mathrm{R}y1$
  - In PLTL: $[\diamondminus \mathrm{W}y1 \Rightarrow \diamondminus \mathrm{R}x1] \wedge [\mathrm{W}z1 \Rightarrow (\diamondminus \mathrm{R}y1 \wedge \boxminus(\mathrm{W}x1 \Rightarrow \diamondminus \mathrm{R}y1))]$
- Compositionality: every thread satisfies $\Longrightarrow$ program satisfies
  - $\exists$ justification: compositional for predicate logic OOTA1-3
  - $\forall$ justification: compositional for temporal logic OOTA4-5

Comparing attempted executions

$y := x \parallel r := y;\ \mathtt{if}\,(\textcolor{red}{r})\,\{x := r;\ z := r\}\ \mathtt{else}\,\{x := 1\}$ (⋆) ✓



$y := x \parallel r := y;\ \mathtt{if}\,(\textcolor{red}{b})\,\{x := r;\ z := r\}\ \mathtt{else}\,\{x := 1\} \parallel b := 1$ (OOTA4) ✗




STILL
PRETTY?
Image in the Public domain

It's hard...

- Programmer desiderata
  - Compositional/local reasoning
  - Sequential Consistency for Data Race Free programs (SC-DRF)

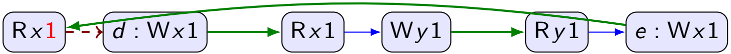
It's hard...

- Programmer desiderata
  - Compositional/local reasoning
  - Sequential Consistency for Data Race Free programs (SC-DRF)
- Implementer desiderata
  - Efficient on hardware
  - Compiler optimizations

It's hard...

- Programmer desiderata
  - Compositional/local reasoning
  - Sequential Consistency for Data Race Free programs (SC-DRF)
- Implementer desiderata
  - Efficient on ~~hardware~~ MCA hardware
  - Compiler optimizations
- Multi-copy atomicity (MCA)
  - When write published to one processor, published to all
  - ARM✓ x86-64✓ RISC-V✓ POWER✗
  - Prevents anomalies:

$$r := x;\ x := 1 \ \parallel\ y := x \ \parallel\ x := y$$



(MCA3) ✗

We've been trying for 20 years...

- Operational with alternate executions: ARM✓ OOTA✗
  - Java 1.1 fails CSE [Pugh 1999]
  - Commitment/Speculation/Promises [2005, 2010, 2017]
- Strong models: ARM✗ OOTA✓
  - SC [Singh, Narayanasamy, Marino, Millstein, Musuvathi 2012]
  - RC11 [Lahav, Vafeiadis 2016]
  - Event Structures [2016]

We've been trying for 20 years...

- Operational with alternate executions: ARM✓ OOTA✗
  - Java 1.1 fails CSE [Pugh 1999]
  - Commitment/Speculation/Promises [2005, 2010, 2017]
- Strong models: ARM✗ OOTA✓
  - SC [Singh, Narayanasamy, Marino, Millstein, Musuvathi 2012]
  - RC11 [Lahav, Vafeiadis 2016]
  - Event Structures [2016]
- Operational with compiler rewrites
  - [Ferreira, Feng, Shao 2010], [Pichon-Pharabod, Sewell 2016]
- Symbolic execution with multiple orders and acyclicly requirements
  - Hardware [Alglave 2010], [Alglave, Maranget, Tautschnig 2014]
  - C++ [Batty, Owens, Sarkar, Sewell, Weber 2011]
- Event Structures
  - WeakestMO [Chakraborty, Vafeiadis 2019]
  - Modularity [Paviotti, Cooksey, Paradis, Wright, Owens, Batty 2020]

PRINCIPLES

Image in the Public domain

Compositionality

Image in the Public domain


Compositionality
+ Construction
Image in the Public domain

Image in the Public domain


LOGIC
Image in the Public domain

Image in the Public domain

Image in the Public domain


1
ONE
ORDER
Image in the Public domain


4 SLIDES
Image in the Public domain

Semantic Domain

- A *pomset with preconditions* is a tuple $(E, \leq, \lambda)$ where
  - E is a set of *events*
  - $\leq \subseteq (E \times E)$ is a partial order
  - $\lambda : E \to (\Phi \times \mathcal{A})$ is a *labeling* from which we derive functions
    - $\Phi : E \to \Phi$ *(formulae)*
    - $\mathcal{A} : E \to \mathcal{A}$ *(actions)*

Semantic Domain

- A *pomset with preconditions* is a tuple $(E, \leq, \lambda)$ where
  - E is a set of *events*
  - $\leq \subseteq (E \times E)$ is a partial order
  - $\lambda : E \to (\Phi \times \mathcal{A})$ is a *labeling* from which we derive functions
    - $\Phi : E \to \Phi$ *(formulae)*
    - $\mathcal{A} : E \to \mathcal{A}$ *(actions)*
  - $\bigwedge_e \Phi(e)$ is satisfiable *(consistency)*
  - if $d \leq e$ then $\Phi(e)$ implies $\Phi(d)$ *(causal strengthening)*

Semantic Domain

- A *pomset with preconditions* is a tuple $(E, \leq, \lambda)$ where
  - E is a set of *events*
  - $\leq \subseteq (E \times E)$ is a partial order
  - $\lambda : E \to (\Phi \times \mathcal{A})$ is a *labeling* from which we derive functions
    - $\Phi : E \to \Phi$ *(formulae)*
    - $\mathcal{A} : E \to \mathcal{A}$ *(actions)*
  - $\bigwedge_e \Phi(e)$ is satisfiable *(consistency)*
  - if $d \leq e$ then $\Phi(e)$ implies $\Phi(d)$ *(causal strengthening)*
- We say $\mathcal{A}(d) = (\mathsf{W}x\,v)$ *fulfills* $\mathcal{A}(e) = (\mathsf{R}x\,v)$ if
  - $d < e$
  - if $\mathcal{A}(c) = (\mathsf{W}x..)$ then either $c \leq d$ or $e \leq c$

Semantic Domain

- A *pomset with preconditions* is a tuple $(E, \leq, \lambda)$ where
  - E is a set of *events*
  - $\leq \subseteq (E \times E)$ is a partial order
  - $\lambda : E \to (\Phi \times \mathcal{A})$ is a *labeling* from which we derive functions
    - $\Phi : E \to \Phi$ *(formulae)*
    - $\mathcal{A} : E \to \mathcal{A}$ *(actions)*
  - $\bigwedge_e \Phi(e)$ is satisfiable *(consistency)*
  - if $d \leq e$ then $\Phi(e)$ implies $\Phi(d)$ *(causal strengthening)*
- We say $\mathcal{A}(d) = (\mathsf{W}\,x\,v)$ *fulfills* $\mathcal{A}(e) = (\mathsf{R}\,x\,v)$ if
  - $d < e$
  - if $\mathcal{A}(c) = (\mathsf{W}\,x..)$ then either $c \leq d$ or $e \leq c$
- A pomset is x*-closed* if
  - every $\mathcal{A}(e) = (\mathsf{R}\,x..)$ is fulfilled
  - every $\Phi(e)$ is independent of x: $(\forall v.\ \Phi(e) \vDash \Phi(e)[v/x] \vDash \Phi(e))$

Semantic Operations (1/2)

- Let $P \in (\nu x.\mathcal{P})$ when $P \in \mathcal{P}$ and P is x-closed

Semantic Operations (1/2)

- Let $P \in (\nu x.\mathcal{P})$ when $P \in \mathcal{P}$ and P is x-closed
- Let $P \in (\phi \rhd \mathcal{P})$ when $P \in \mathcal{P}$ and $(\forall e \in E)\ \Phi(e)$ implies ϕ

Semantic Operations (1/2)

- Let $P \in (\nu x.\mathcal{P})$ when $P \in \mathcal{P}$ and P is x-closed
- Let $P \in (\phi \triangleright \mathcal{P})$ when $P \in \mathcal{P}$ and $(\forall e \in E)\ \Phi(e)$ implies ϕ
- Let $P' \in (\mathcal{P}[M/x])$ when $(\exists P \in \mathcal{P})$
  $E' = E$, $\leq' = \leq$, $\mathcal{A}' = \mathcal{A}$, and $(\forall e \in E')\ \Phi'(e) = \Phi(e)[M/x]$

Semantic Operations (1/2)

- Let $P \in (\nu x.\mathcal{P})$ when $P \in \mathcal{P}$ and P is x-closed
- Let $P \in (\phi \rhd \mathcal{P})$ when $P \in \mathcal{P}$ and $(\forall e \in E)\ \Phi(e)$ implies ϕ
- Let $P' \in (\mathcal{P}[M/x])$ when $(\exists P \in \mathcal{P})$
  $E' = E$, $\leq' = \leq$, $\mathcal{A}' = \mathcal{A}$, and $(\forall e \in E')\ \Phi'(e) = \Phi(e)[M/x]$
- Let $P' \in (\mathcal{P}^1 \parallel \mathcal{P}^2)$ when $(\exists P^1 \in \mathcal{P}^1)\ (\exists P^2 \in \mathcal{P}^2)$
  $E' = E^1 \cup E^2$, $\leq' \supseteq \leq^1 \cup \leq^2$, and $(\forall e \in E')$ either

$$
\begin{array}{c}
e \notin E^2,\ \mathcal{A}'(e) = \mathcal{A}^1(e) \text{ and } \Phi'(e) \text{ implies } \Phi^1(e), \\
e \notin E^1,\ \mathcal{A}'(e) = \mathcal{A}^2(e) \text{ and } \Phi'(e) \text{ implies } \Phi^2(e),\ \text{or} \\
\mathcal{A}'(e) = \mathcal{A}^1(e) = \mathcal{A}^2(e) \text{ and } \Phi'(e) \text{ implies } \Phi^1(e) \vee \Phi^2(e)
\end{array}
$$

Semantic Operations (2/2)

- Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})$ $(\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p2) $\leq' \supseteq \leq$

(p3a) $\mathcal{A}'(e) = \mathcal{A}(e)$

(p3b) $\mathcal{A}'(d) = a$

(p4a) $\Phi'(d)$ implies $\phi \wedge (d \notin E \vee \Phi(d))$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}\,v\,x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

(p5b) if d and e are conflicting actions then $d \leq' e$

(p5c) if d is an acquire or e is a release then $d \leq' e$

(p5d) if d is an SC write and e is an SC read then $d \leq' e$

(p5e) if d reads, and e is an acquiring fence, then $d \leq' e$

(p5f) if d is a releasing fence, and e writes, then $d \leq' e$

Semantic Operations (2/2)

- Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})$ $(\forall e \in E)$
  - (p1) $E' = E \cup \{d\}$
  - (p2) $\leq' \supseteq \leq$
  - (p3a) $\mathcal{A}'(e) = \mathcal{A}(e)$
  - (p3b) $\mathcal{A}'(d) = a$
  - (p4a) $\Phi'(d)$ implies $\phi \wedge (d \notin E \vee \Phi(d))$
  - (p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$
  - (p4c) if $d = (\mathsf{R}\,v\,x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$
  - (p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$
  - (p5b) if d and e are conflicting actions then $d \leq' e$
  - (p5c) if d is an acquire or e is a release then $d \leq' e$
  - (p5d) if d is an SC write and e is an SC read then $d \leq' e$
  - (p5e) if d reads, and e is an acquiring fence, then $d \leq' e$
  - (p5f) if d is a releasing fence, and e writes, then $d \leq' e$

Language (See paper for RMWs and address calculation)

$$
\begin{aligned}
[\![\texttt{skip}]\!] &\triangleq \{\checkmark\} \\
[\![r := M;\ C]\!] &\triangleq [\![C]\!][M/r] \\
[\![r := x^{\mu};\ C]\!] &\triangleq \textstyle\bigcup_v\ (\mathsf{R}^{\mu}\,x\,v) \Rightarrow [\![C]\!][x/r] \\
[\![x^{\mu} := M;\ C]\!] &\triangleq \textstyle\bigcup_v\ (M = v \mid \mathsf{W}^{\mu}\,x\,v) \Rightarrow [\![C]\!][M/x] \\
[\![\mathtt{F}^{\kappa};\ C]\!] &\triangleq (\mathsf{F}^{\kappa}) \Rightarrow [\![C]\!] \\
[\![\texttt{if}\,(M)\,\{C\}\,\texttt{else}\,\{D\}]\!] &\triangleq \big(M \triangleright [\![C]\!]\big) \parallel \big(\neg M \triangleright [\![D]\!]\big) \\
[\![C \parallel D]\!] &\triangleq [\![C]\!] \parallel [\![D]\!] \\
[\![\texttt{var}\,x;\ C]\!] &\triangleq \nu x\,.\,[\![C]\!]
\end{aligned}
$$

μ ::= rlx	(Relaxed)	κ ::= rel	(Release)
| ra	(Release/Acquire)	| acq	(Acquire)
| sc	(Sequentially Consistent)	| sc	(SC)

Language (See paper for RMWs and address calculation)

$$
\begin{aligned}
[\![\texttt{skip}]\!] &\triangleq \{\checkmark\} \\
[\![r := M;\ C]\!] &\triangleq [\![C]\!][M/r] \\
[\![r := x^{\mu};\ C]\!] &\triangleq \textstyle\bigcup_v\ (\mathsf{R}^{\mu}\,x\,v) \Rightarrow [\![C]\!][x/r] \\
[\![x^{\mu} := M;\ C]\!] &\triangleq \textstyle\bigcup_v\ (M = v \mid \mathsf{W}^{\mu}\,x\,v) \Rightarrow [\![C]\!][M/x] \\
[\![\mathtt{F}^{\kappa};\ C]\!] &\triangleq (\mathsf{F}^{\kappa}) \Rightarrow [\![C]\!] \\
[\![\texttt{if}\,(M)\,\{C\}\,\texttt{else}\,\{D\}]\!] &\triangleq \big(M \rhd [\![C]\!]\big) \parallel \big(\neg M \rhd [\![D]\!]\big) \\
[\![C \parallel D]\!] &\triangleq [\![C]\!] \parallel [\![D]\!] \\
[\![\texttt{var}\,x;\ C]\!] &\triangleq \nu x\,.\,[\![C]\!]
\end{aligned}
$$

μ	::=	rlx	(Relaxed)	κ	::=	rel	(Release)
	|	ra	(Release/Acquire)		|	acq	(Acquire)
	|	sc	(Sequentially Consistent)		|	sc	(SC)

Language (See paper for RMWs and address calculation)

$$
\begin{aligned}
[\![\texttt{skip}]\!] &\triangleq \{\checkmark\} \\
[\![r := M\,;\; C]\!] &\triangleq [\![C]\!][M/r] \\
[\![r := x^{\mu}\,;\; C]\!] &\triangleq \textstyle\bigcup_{v}\,(\mathsf{R}^{\mu}\,x\,v) \Rightarrow [\![C]\!][x/r] \\
[\![x^{\mu} := M\,;\; C]\!] &\triangleq \textstyle\bigcup_{v}\,(M = v \mid \mathsf{W}^{\mu}\,x\,v) \Rightarrow [\![C]\!][M/x] \\
[\![\mathtt{F}^{\kappa}\,;\; C]\!] &\triangleq (\mathsf{F}^{\kappa}) \Rightarrow [\![C]\!] \\
[\![\texttt{if}\,(M)\,\{C\}\,\texttt{else}\,\{D\}]\!] &\triangleq \big(M \triangleright [\![C]\!]\big) \parallel \big(\neg M \triangleright [\![D]\!]\big) \\
[\![C \;\parallel\; D]\!] &\triangleq [\![C]\!] \parallel [\![D]\!] \\
[\![\texttt{var}\,x\,;\; C]\!] &\triangleq \nu x\,.\,[\![C]\!]
\end{aligned}
$$

$$
\begin{aligned}
\mu \;::=\;& \mathsf{rlx} && \text{(Relaxed)} \\
\mid\;& \mathsf{ra} && \text{(Release/Acquire)} \\
\mid\;& \mathsf{sc} && \text{(Sequentially Consistent)}
\end{aligned}
$$

$$
\begin{aligned}
\kappa \;::=\;& \mathsf{rel} && \text{(Release)} \\
\mid\;& \mathsf{acq} && \text{(Acquire)} \\
\mid\;& \mathsf{sc} && \text{(SC)}
\end{aligned}
$$

Language (See paper for RMWs and address calculation)

$$
\begin{aligned}
[\![\texttt{skip}]\!] &\triangleq \{\checkmark\} \\
[\![r := M\,;\ C]\!] &\triangleq [\![C]\!][M/r] \\
[\![r := x^{\mu}\,;\ C]\!] &\triangleq \textstyle\bigcup_v\ (\mathsf{R}^{\mu}\,x\,v) \Rightarrow [\![C]\!][x/r] \\
[\![x^{\mu} := M\,;\ C]\!] &\triangleq \textstyle\bigcup_v\ (M = v \mid \mathsf{W}^{\mu}\,x\,v) \Rightarrow [\![C]\!][M/x] \\
[\![\mathtt{F}^{\kappa}\,;\ C]\!] &\triangleq (\mathsf{F}^{\kappa}) \Rightarrow [\![C]\!] \\
[\![\texttt{if}\,(M)\,\{C\}\,\texttt{else}\,\{D\}]\!] &\triangleq \big(M \triangleright [\![C]\!]\big) \parallel \big(\neg M \triangleright [\![D]\!]\big) \\
[\![C \parallel D]\!] &\triangleq [\![C]\!] \parallel [\![D]\!] \\
[\![\texttt{var}\,x\,;\ C]\!] &\triangleq \nu x\,.\,[\![C]\!]
\end{aligned}
$$

$$
\begin{aligned}
\mu &::= \mathsf{rlx} && \text{(Relaxed)} \\
&\mid \mathsf{ra} && \text{(Release/Acquire)} \\
&\mid \mathsf{sc} && \text{(Sequentially Consistent)}
\end{aligned}
\qquad
\begin{aligned}
\kappa &::= \mathsf{rel} && \text{(Release)} \\
&\mid \mathsf{acq} && \text{(Acquire)} \\
&\mid \mathsf{sc} && \text{(SC)}
\end{aligned}
$$

Language (See paper for RMWs and address calculation)

$$
\begin{aligned}
[\![\texttt{skip}]\!] &\triangleq \{\checkmark\} \\
[\![r := M\,;\ C]\!] &\triangleq [\![C]\!][M/r] \\
[\![r := x^{\mu}\,;\ C]\!] &\triangleq \textstyle\bigcup_v\ (\mathsf{R}^{\mu}\,x\,v) \Rightarrow [\![C]\!][x/r] \\
[\![x^{\mu} := M\,;\ C]\!] &\triangleq \textstyle\bigcup_v\ (M = v \mid \mathsf{W}^{\mu}\,x\,v) \Rightarrow [\![C]\!][M/x] \\
[\![\mathtt{F}^{\kappa}\,;\ C]\!] &\triangleq (\mathsf{F}^{\kappa}) \Rightarrow [\![C]\!] \\
[\![\texttt{if}\,(M)\,\{C\}\,\texttt{else}\,\{D\}]\!] &\triangleq \big(M \triangleright [\![C]\!]\big) \parallel \big(\neg M \triangleright [\![D]\!]\big) \\
[\![C \parallel D]\!] &\triangleq [\![C]\!] \parallel [\![D]\!] \\
[\![\texttt{var}\,x\,;\ C]\!] &\triangleq \nu x\,.\,[\![C]\!]
\end{aligned}
$$

μ	::=	rlx	(Relaxed)	κ	::=	rel	(Release)
	|	ra	(Release/Acquire)		|	acq	(Acquire)
	|	sc	(Sequentially Consistent)		|	sc	(SC)

Language (See paper for RMWs and address calculation)

$$
\begin{aligned}
[\![\texttt{skip}]\!] &\triangleq \{\checkmark\} \\
[\![r := M;\ C]\!] &\triangleq [\![C]\!][M/r] \\
[\![r := x^{\mu};\ C]\!] &\triangleq \textstyle\bigcup_v\ (\mathsf{R}^{\mu}\,x\,v) \Rightarrow [\![C]\!][x/r] \\
[\![x^{\mu} := M;\ C]\!] &\triangleq \textstyle\bigcup_v\ (M = v \mid \mathsf{W}^{\mu}\,x\,v) \Rightarrow [\![C]\!][M/x] \\
[\![\mathtt{F}^{\kappa};\ C]\!] &\triangleq (\mathsf{F}^{\kappa}) \Rightarrow [\![C]\!] \\
[\![\texttt{if}\,(M)\,\{C\}\,\texttt{else}\,\{D\}]\!] &\triangleq \big(M \triangleright [\![C]\!]\big) \parallel \big(\neg M \triangleright [\![D]\!]\big) \\
[\![C \parallel D]\!] &\triangleq [\![C]\!] \parallel [\![D]\!] \\
[\![\texttt{var}\,x;\ C]\!] &\triangleq \nu x \,.\, [\![C]\!]
\end{aligned}
$$

μ	::=	rlx	(Relaxed)	κ	::=	rel	(Release)
	|	ra	(Release/Acquire)		|	acq	(Acquire)
	|	sc	(Sequentially Consistent)		|	sc	(SC)

Language (See paper for RMWs and address calculation)

$$
\begin{aligned}
[\![\texttt{skip}]\!] &\triangleq \{\checkmark\} \\
[\![r := M;\ C]\!] &\triangleq [\![C]\!][M/r] \\
[\![r := x^{\mu};\ C]\!] &\triangleq \textstyle\bigcup_v\ (\mathsf{R}^{\mu}\, x\, v) \Rightarrow [\![C]\!][x/r] \\
[\![x^{\mu} := M;\ C]\!] &\triangleq \textstyle\bigcup_v\ (M = v \mid \mathsf{W}^{\mu}\, x\, v) \Rightarrow [\![C]\!][M/x] \\
[\![\mathtt{F}^{\kappa};\ C]\!] &\triangleq (\mathsf{F}^{\kappa}) \Rightarrow [\![C]\!] \\
[\![\texttt{if}\,(M)\,\{C\}\,\texttt{else}\,\{D\}]\!] &\triangleq \big(M \triangleright [\![C]\!]\big) \parallel \big(\neg M \triangleright [\![D]\!]\big) \\
[\![C \parallel D]\!] &\triangleq [\![C]\!] \parallel [\![D]\!] \\
[\![\texttt{var}\, x;\ C]\!] &\triangleq \nu x \,.\, [\![C]\!]
\end{aligned}
$$

$$
\begin{aligned}
\mu \ ::= &\ \mathsf{rlx} && \text{(Relaxed)} \\
\mid &\ \mathsf{ra} && \text{(Release/Acquire)} \\
\mid &\ \mathsf{sc} && \text{(Sequentially Consistent)}
\end{aligned}
\qquad
\begin{aligned}
\kappa \ ::= &\ \mathsf{rel} && \text{(Release)} \\
\mid &\ \mathsf{acq} && \text{(Acquire)} \\
\mid &\ \mathsf{sc} && \text{(SC)}
\end{aligned}
$$


4 SLIDES
Image in the Public domain

Sequential Semantics: Preconditions

$$z := r$$

$$\boxed{r{=}0 \mid \mathsf{W}z0}$$

Precondition records dependence on r

$[\![r := x^{\mu};\, C]\!] \triangleq \bigcup_v\, (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M;\, C]\!] \triangleq \bigcup_v\, (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R} v x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$z := r$

$\boxed{r{=}1 \mid \mathsf{W}\,z\,1}$

One pomset for each value

$[\![r := x^{\mu};\, C]\!] \triangleq \bigcup_v \,(\mathsf{R}^{\mu} x\, v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M;\, C]\!] \triangleq \bigcup_v \,(M = v \mid \mathsf{W}^{\mu} x\, v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})$ $(\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}\,v\,x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$z := r$

$\boxed{r{=}2 \mid \mathsf{W}z2}$

One pomset for each value

$[\![r := x^{\mu};\, C]\!] \triangleq \bigcup_v\, (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M;\, C]\!] \triangleq \bigcup_v\, (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R} v x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$$x := r\,;\, z := r$$

$\boxed{r{=}0 \mid \mathsf{W}x0}$ $\boxed{r{=}0 \mid \mathsf{W}z0}$

Two writes = two events per pomset

$[\![r := x^\mu\,;\, C]\!] \triangleq \bigcup_v\, (\mathsf{R}^\mu x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^\mu := M\,;\, C]\!] \triangleq \bigcup_v\, (M = v \mid \mathsf{W}^\mu x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$x := r\,;\, z := r$

$\boxed{r{=}1 \mid \mathsf{W}x1}$ $\boxed{r{=}1 \mid \mathsf{W}z1}$

Two writes = two events per pomset

$[\![r := x^{\mu}\,;\, C]\!] \triangleq \bigcup_v (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M\,;\, C]\!] \triangleq \bigcup_v (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$$x := r\,;\ z := r$$

$\boxed{r{=}1 \mid \mathsf{W}x1}$ $\boxed{r{=}2 \mid \mathsf{W}z2}$

Preconditions must be consistent

Pomset = a single execution

$[\![r := x^{\mu}\,;\ C]\!] \triangleq \bigcup_v\, (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M\,;\ C]\!] \triangleq \bigcup_v\, (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$x := r\,;\ z := r$

$\cancel{r{=}1 \mid \mathsf{W}x1}$ $\cancel{r{=}2 \mid \mathsf{W}z2}$

Preconditions must be consistent

Pomset = a single execution

$[\![r := x^{\mu}\,;\ C]\!] \triangleq \bigcup_v\ (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M\,;\ C]\!] \triangleq \bigcup_v\ (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R} v x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$x := r\,;\, z := r$

$\boxed{r{=}1 \mid \mathsf{W}x1}$ $\boxed{r{=}1 \mid \mathsf{W}z1}$

$x := 1$

$\boxed{1{=}1 \mid \mathsf{W}x1}$

A separate program

$[\![r := x^{\mu}\,;\, C]\!] \triangleq \bigcup_v (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M\,;\, C]\!] \triangleq \bigcup_v (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1)	$E' = E \cup \{d\}$	
(p4b)	if $d \neq (\mathsf{R}..)$	then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$
(p4c)	if $d = (\mathsf{R}vx)$	then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$
(p5a)	if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$	then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$\texttt{if}\,(r)\{x := r\,;\, z := r\}$

$\boxed{r{\neq}0 \wedge r{=}1 \mid \mathsf{W}x1}$ $\boxed{r{\neq}0 \wedge r{=}1 \mid \mathsf{W}z1}$

$\texttt{if}\,(\neg r)\{x := 1\}$

$\boxed{r{=}0 \wedge 1{=}1 \mid \mathsf{W}x1}$

Adding control

$[\![r := x^{\mu}\,;\, C]\!] \triangleq \bigcup_v \, (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M\,;\, C]\!] \triangleq \bigcup_v \, (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$\texttt{if}(r)\{x := r\,;\, z := r\}$

$\boxed{r{=}1 \mid \mathsf{W}x1}$ $\boxed{r{=}1 \mid \mathsf{W}z1}$

$\texttt{if}(\neg r)\{x := 1\}$

$\boxed{r{=}0 \mid \mathsf{W}x1}$

Simplifying

$[\![r := x^{\mu}\,;\, C]\!] \triangleq \bigcup_v (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M\,;\, C]\!] \triangleq \bigcup_v (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}\,v\,x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$\mathtt{if}\,(r)\,\{x := r;\, z := r\}$

$\boxed{r{=}1 \mid \mathsf{W}x1}$ $\boxed{r{=}1 \mid \mathsf{W}z1}$

$\mathtt{if}\,(\neg r)\,\{x := 1\}$

$\boxed{r{=}0 \mid \mathsf{W}x1}$

Combining both sides

$\mathtt{if}\,(r)\,\{x := r;\, z := r\}\,\mathtt{else}\,\{x := 1\}$

$\boxed{r{=}1 \vee r{=}0 \mid \mathsf{W}x1}$ $\boxed{r{=}1 \mid \mathsf{W}z1}$

We can coalesce the writes to x

$[\![r := x^{\mu};\, C]\!] \triangleq \bigcup_v (\mathsf{R}^{\mu}xv) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M;\, C]\!] \triangleq \bigcup_v (M = v \mid \mathsf{W}^{\mu}xv) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})$ $(\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$\texttt{if}\,(r)\,\{x := r\,;\, z := r\}$

$r{=}1 \mid \mathrm{W}x1$ $\quad$ $r{=}1 \mid \mathrm{W}z1$

$\texttt{if}\,(\neg r)\,\{x := 2\}$

$r{=}0 \mid \mathrm{W}x2$

Combining both sides

$\texttt{if}\,(r)\,\{x := r\,;\, z := r\}\,\texttt{else}\,\{x := 2\}$

$r{=}1 \mid \mathrm{W}x1$ $\quad$ $r{=}1 \mid \mathrm{W}z1$ $\quad$ $r{=}0 \mid \mathrm{W}x2$

We *cannot* coalesce the writes to x

$[\![r := x^\mu;\, C]\!] \triangleq \bigcup_v\, (\mathrm{R}^\mu x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^\mu := M;\, C]\!] \triangleq \bigcup_v\, (M = v \mid \mathrm{W}^\mu x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathrm{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathrm{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathrm{R}..)$, $e = (\mathrm{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$\texttt{if}\,(r)\,\{x := r\,;\; z := r\}$

$\boxed{r{=}1 \mid \mathsf{W}x1}$ $\boxed{r{=}1 \mid \mathsf{W}z1}$

$\texttt{if}\,(\neg r)\,\{x := 2\}$

$\boxed{r{=}0 \mid \mathsf{W}x2}$

Combining both sides

$\texttt{if}\,(r)\,\{x := r\,;\; z := r\}\;\texttt{else}\;\{x := 2\}$

$\boxed{\xcancel{r{=}1 \mid \mathsf{W}x1}}$ $\boxed{\xcancel{r{=}1 \mid \mathsf{W}z1}}$ $\boxed{\xcancel{r{=}0 \mid \mathsf{W}x2}}$

Preconditions must be consistent

$[\![r := x^{\mu};\, C]\!] \triangleq \bigcup_v\, (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M;\, C]\!] \triangleq \bigcup_v\, (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R} v x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$\texttt{if}\,(r)\,\{x := r\,;\, z := r\}\,\texttt{else}\,\{x := 1\}$

$\boxed{r{=}1 \lor r{=}0 \mid \mathsf{W}x1}$ $\boxed{r{=}1 \mid \mathsf{W}z1}$

Let's start fresh here

$[\![r := x^{\mu}\,;\, C]\!] \triangleq \bigcup_v\, (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M\,;\, C]\!] \triangleq \bigcup_v\, (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})$ $(\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}\, v x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$$r := y\,;\ \texttt{if}\,(r)\,\{x := r\,;\ z := r\}\ \texttt{else}\ \{x := 1\}$$

$\boxed{y{=}1 \vee y{=}0 \mid \mathsf{W}x1}$ $\boxed{y{=}1 \mid \mathsf{W}z1}$

To prepend, $r := y$, we first substitute $[y/r]$

$[\![r := x^{\mu}\,;\ C]\!] \triangleq \bigcup_v\,(\mathsf{R}^{\mu}xv) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M\,;\ C]\!] \triangleq \bigcup_v\,(M = v \mid \mathsf{W}^{\mu}xv) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$$r := y\,;\ \mathtt{if}\,(r)\,\{x := r\,;\ z := r\}\ \mathtt{else}\ \{x := 1\}$$

$\boxed{\mathsf{R}\,y\,1}$ $\boxed{(y{=}1 \vee y{=}0) \wedge (1{=}1 \vee 1{=}0) \mid \mathsf{W}\,x\,1}$ $\boxed{(y{=}1) \wedge (1{=}1) \mid \mathsf{W}\,z\,1}$

We then add the action, and its constraint: $\phi' \vDash \phi[v/y]$

$[\![r := x^{\mu}\,;\ C]\!] \triangleq \bigcup_v\ (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M\,;\ C]\!] \triangleq \bigcup_v\ (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$$r := y\,;\ \texttt{if}\,(r)\,\{x := r\,;\ z := r\}\ \texttt{else}\,\{x := 1\}$$

$\boxed{\mathsf{R}\,y\,2}$ $\boxed{(y{=}1 \lor y{=}0) \land (2{=}1 \lor 2{=}0) \mid \mathsf{W}\,x\,1}$ $\boxed{(y{=}1) \land (2{=}1) \mid \mathsf{W}\,z\,1}$

Incompatible reads are disallowed

$[\![r := x^\mu\,;\ C]\!] \triangleq \bigcup_v\ (\mathsf{R}^\mu x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^\mu := M\,;\ C]\!] \triangleq \bigcup_v\ (M = v \mid \mathsf{W}^\mu x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1)	$E' = E \cup \{d\}$	
(p4b)	if $d \neq (\mathsf{R}..)$	then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$
(p4c)	if $d = (\mathsf{R}vx)$	then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$
(p5a)	if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$	then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$$r := y\,;\ \mathtt{if}\,(r)\,\{x := r\,;\ z := r\}\ \mathtt{else}\,\{x := 1\}$$

$\cancel{\mathsf{R}y2}$ $\quad$ $\cancel{(y{=}1 \lor y{=}0) \land (2{=}1 \lor 2{=}0) \mid \mathsf{W}x1}$ $\quad$ $\cancel{(y{=}1) \land (2{=}1) \mid \mathsf{W}z1}$

Incompatible reads are disallowed

$[\![r := x^{\mu}\,;\ C]\!] \triangleq \bigcup_v\ (\mathsf{R}^{\mu}x\,v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M\,;\ C]\!] \triangleq \bigcup_v\ (M = v \mid \mathsf{W}^{\mu}x\,v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1)	$E' = E \cup \{d\}$	
(p4b)	if $d \neq (\mathsf{R}..)$	then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$
(p4c)	if $d = (\mathsf{R}\,v\,x)$	then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$
(p5a)	if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$	then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$$r := y\,;\ \mathtt{if}\,(r)\,\{x := r\,;\ z := r\}\ \mathtt{else}\,\{x := 1\}$$

$\mathsf{R}\,y\,1$ $\quad$ $(y{=}1 \vee y{=}0) \wedge (1{=}1 \vee 1{=}0) \mid \mathsf{W}\,x\,1$ $\quad$ $(y{=}1) \wedge (1{=}1) \mid \mathsf{W}\,z\,1$

Order may be introduced

$[\![r := x^{\mu}\,;\ C]\!] \triangleq \bigcup_v\ (\mathsf{R}^{\mu}\,x\,v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M\,;\ C]\!] \triangleq \bigcup_v\ (M = v \mid \mathsf{W}^{\mu}\,x\,v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}\,v\,x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$$r := y\,;\ \mathtt{if}\,(r)\,\{x := r\,;\ z := r\}\ \mathtt{else}\,\{x := 1\}$$

$\boxed{\mathsf{R}\,y\,1}$ $\boxed{(y{=}1 \vee y{=}0) \wedge (1{=}1 \vee 1{=}0) \mid \mathsf{W}\,x\,1}$ $\boxed{(1{=}1) \wedge (1{=}1) \mid \mathsf{W}\,z\,1}$

Order enables substitution $[v/y]$

$[\![r := x^{\mu}\,;\ C]\!] \triangleq \bigcup_v\ (\mathsf{R}^{\mu}\,x\,v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M\,;\ C]\!] \triangleq \bigcup_v\ (M = v \mid \mathsf{W}^{\mu}\,x\,v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}\,v\,x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$y := 0;\ r := y;\ \texttt{if}\,(r)\,\{x := r;\ z := r\}\ \texttt{else}\,\{x := 1\}$

$0{=}0 \mid \mathsf{W}y0$ $\quad$ $\mathsf{R}y1$ $\quad$ $(0{=}1 \vee 0{=}0) \wedge (1{=}1 \vee 1{=}0) \mid \mathsf{W}x1$ $\quad$ $(1{=}1) \wedge (1{=}1) \mid \mathsf{W}z1$

Prepending $y := 0$ substitutes $[0/y]$

Order imposed by sub? Write✗ Read✓

$[\![r := x^{\mu};\ C]\!] \triangleq \bigcup_v\ (\mathsf{R}^{\mu}xv) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M;\ C]\!] \triangleq \bigcup_v\ (M = v \mid \mathsf{W}^{\mu}xv) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

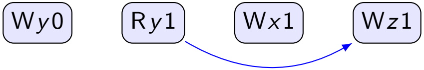
(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Sequential Semantics: Preconditions

$y := 0;\ r := y;\ \mathtt{if}\,(r)\,\{x := r;\ z := r\}\ \mathtt{else}\ \{x := 1\}$



Simplifying tautologies

$[\![r := x^{\mu};\, C]\!] \triangleq \bigcup_v (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M;\, C]\!] \triangleq \bigcup_v (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})$ $(\forall e \in E)$

- (p1) $E' = E \cup \{d\}$
- (p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$
- (p4c) if $d = (\mathsf{R} v x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$
- (p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

Concurrent Semantics: Fulfillment

$$y := 0\,;\ r := y\,;\ \texttt{if}\,(r)\,\{x := r\,;\ z := r\}\ \texttt{else}\,\{x := 1\}$$

$$\begin{tikzcd} \mathsf{W}y0 \arrow[r, dashed] & \mathsf{R}y1 \arrow[rr, bend right] & \mathsf{W}x1 & \mathsf{W}z1 \end{tikzcd}$$

Must preserve order on *conflicting* accesses (WW, WR)

$[\![r := x^{\mu}\,;\, C]\!] \triangleq \bigcup_v\, (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

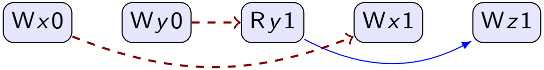
$[\![x^{\mu} := M\,;\, C]\!] \triangleq \bigcup_v\, (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

- (p1) $E' = E \cup \{d\}$
- (p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$
- (p4c) if $d = (\mathsf{R} v x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$
- (p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$
- (p5b) if d and e are conflicting actions then $d \leq' e$

Concurrent Semantics: Fulfillment

$x := 0;\ y := 0;\ r := y;\ \texttt{if}\,(r)\,\{x := r;\ z := r\}\ \texttt{else}\,\{x := 1\}$



Initializing x

$[\![r := x^{\mu};\ C]\!] \triangleq \bigcup_v (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

$[\![x^{\mu} := M;\ C]\!] \triangleq \bigcup_v (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

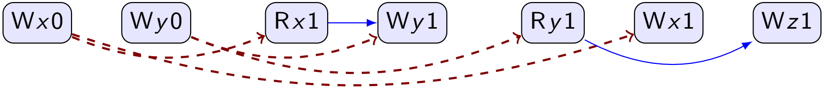
(p4c) if $d = (\mathsf{R} v x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

(p5b) if d and e are conflicting actions then $d \leq' e$

Concurrent Semantics: Fulfillment

$x := 0;\ y := 0;\ (y := x \parallel r := y;\ \texttt{if}\,(r)\,\{x := r;\ z := r\}\ \texttt{else}\ \{x := 1\})$



Introducing $y := x$

$[\![r := x^{\mu};\ C]\!] \triangleq \bigcup_v (\mathsf{R}^{\mu} x v) \Rightarrow [\![C]\!][x/r]$

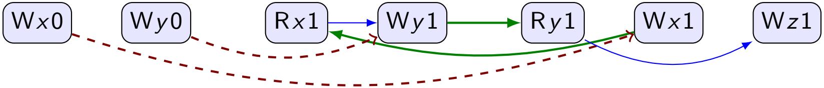
$[\![x^{\mu} := M;\ C]\!] \triangleq \bigcup_v (M = v \mid \mathsf{W}^{\mu} x v) \Rightarrow [\![C]\!][M/x]$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

- (p1) $E' = E \cup \{d\}$
- (p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$
- (p4c) if $d = (\mathsf{R} v x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$
- (p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$
- (p5b) if d and e are conflicting actions then $d \leq' e$

Concurrent Semantics: Fulfillment

$x := 0;\, y := 0;\, (y := x \parallel r := y;\, \texttt{if}\,(r)\,\{x := r;\, z := r\}\,\texttt{else}\,\{x := 1\})$ $(\star)$ ✓



Fulfillment requirements

We say $\mathcal{A}(d) = (\mathsf{W}xv)$ *fulfills* $\mathcal{A}(e) = (\mathsf{R}xv)$ if

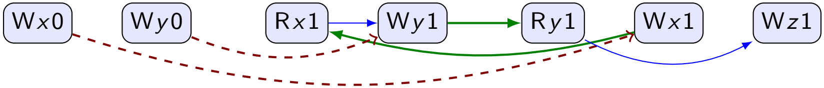
- $d < e$
- if $\mathcal{A}(c) = (\mathsf{W}x..)$ then either $c \le d$ or $e \le c$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$
(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$
(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$
(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \le' e$
(p5b) if d and e are conflicting actions then $d \le' e$

Concurrent Semantics: Fulfillment

$x := 0;\ y := 0;\ (y := x \parallel r := y;\ \texttt{if}\,(r)\,\{x := r;\ z := r\}\ \texttt{else}\ \{x := 1\})$ $(\star)$ ✓



Allowed!

We say $\mathcal{A}(d) = (\mathsf{W}xv)$ *fulfills* $\mathcal{A}(e) = (\mathsf{R}xv)$ if

- $d < e$
- if $\mathcal{A}(c) = (\mathsf{W}x..)$ then either $c \leq d$ or $e \leq c$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

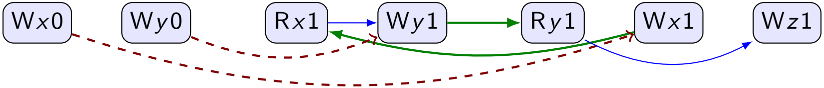
(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

(p5b) if d and e are conflicting actions then $d \leq' e$

Concurrent Semantics: Fulfillment

$$x := 0;\ y := 0;\ (y := x \parallel r := y;\ \mathtt{if}\,(r)\,\{x := r;\ z := r\}\ \mathtt{else}\ \{x := 1\}) \qquad (\star) \quad ✓$$



Allowed!
Partial order

We say $\mathcal{A}(d) = (\mathsf{W}xv)$ *fulfills* $\mathcal{A}(e) = (\mathsf{R}xv)$ if

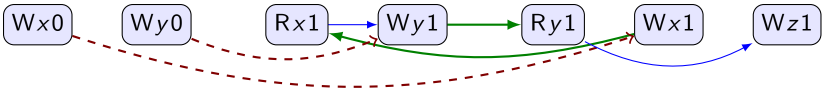
- $d < e$
- if $\mathcal{A}(c) = (\mathsf{W}x..)$ then either $c \le d$ or $e \le c$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$
(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$
(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$
(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \le' e$
(p5b) if d and e are conflicting actions then $d \le' e$

Concurrent Semantics: Fulfillment

$x := 0;\ y := 0;\ (y := x \parallel r := y;\ \texttt{if}\,(r)\,\{x := r;\ z := r\}\ \texttt{else}\ \{x := 1\})$ $(\star)$ ✓



Allowed!

Partial order

All reads fulfilled

We say $\mathcal{A}(d) = (\mathsf{W}xv)$ *fulfills* $\mathcal{A}(e) = (\mathsf{R}xv)$ if

- $d < e$
- if $\mathcal{A}(c) = (\mathsf{W}x..)$ then either $c \leq d$ or $e \leq c$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

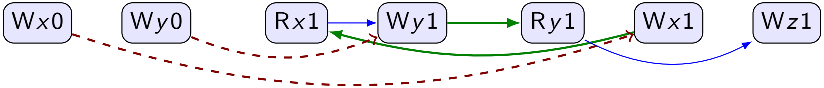
(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

(p5b) if d and e are conflicting actions then $d \leq' e$

Concurrent Semantics: Fulfillment

$x := 0;\ y := 0;\ (y := x \parallel r := y;\ \texttt{if}\,(r)\,\{x := r;\ z := r\}\ \texttt{else}\ \{x := 1\})$ $(\star)$ ✓



Allowed!

Partial order

All reads fulfilled

All preconditions tautologies

We say $\mathcal{A}(d) = (\mathsf{W}xv)$ *fulfills* $\mathcal{A}(e) = (\mathsf{R}xv)$ if

- $d < e$
- if $\mathcal{A}(c) = (\mathsf{W}x..)$ then either $c \leq d$ or $e \leq c$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})\ (\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

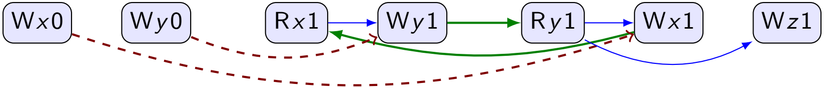
(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$

(p5b) if d and e are conflicting actions then $d \leq' e$

Concurrent Semantics: Fulfillment

$x := 0;\ y := 0;\ (y := x \parallel r := y;\ \texttt{if}\,(r)\,\{x := r;\ z := r\}\ \texttt{else}\,\{x := 2\})$ (OOTA3) ✗



Disallowed!

Cycle

We say $\mathcal{A}(d) = (\mathsf{W}xv)$ *fulfills* $\mathcal{A}(e) = (\mathsf{R}xv)$ if

- $d < e$
- if $\mathcal{A}(c) = (\mathsf{W}x..)$ then either $c \le d$ or $e \le c$

Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})$ $(\forall e \in E)$

(p1) $E' = E \cup \{d\}$

(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$

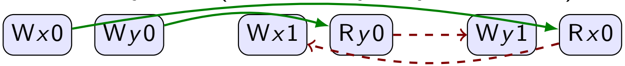
(p4c) if $d = (\mathsf{R}vx)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$

(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \le' e$

(p5b) if d and e are conflicting actions then $d \le' e$

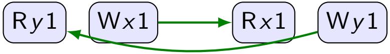
Buffering

$x := 0;\ y := 0;\ (x := 1;\ r := y \ \parallel\ y := 1;\ r := x)$



(SB) ✓

$r := y;\ x := 1 \ \parallel\ r := x;\ y := 1$



(LB) ✓

Synchronization and Fences

- Let $P' \in (\phi \mid a) \Rightarrow \mathcal{P}$ when $(\exists P \in \mathcal{P})$ $(\forall e \in E)$

(p1) $E' = E \cup \{d\}$
(p2) $\leq' \supseteq \leq$
(p3a) $\mathcal{A}'(e) = \mathcal{A}(e)$
(p3b) $\mathcal{A}'(d) = a$
(p4a) $\Phi'(d)$ implies $\phi \wedge (d \notin E \vee \Phi(d))$
(p4b) if $d \neq (\mathsf{R}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$
(p4c) if $d = (\mathsf{R}\,v\,x)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)[v/x]$
(p5a) if $d = (\mathsf{R}..)$, $e = (\mathsf{W}..)$ then $e = d$ or $\Phi'(e)$ implies $\Phi(e)$ or $d \leq' e$
(p5b) if d and e are conflicting actions then $d \leq' e$
(p5c) if d is an acquire or e is a release then $d \leq' e$
(p5d) if d is an SC write and e is an SC read then $d \leq' e$
(p5e) if d reads, and e is an acquiring fence, then $d \leq' e$
(p5f) if d is a releasing fence, and e writes, then $d \leq' e$

Publication

$$x := 0\,;\ x := 1\,;\ y^{\mathsf{ra}} := 1 \ \parallel\ r := y^{\mathsf{ra}}\,;\ s := x$$

$$\begin{tikzcd} \mathsf{W}\,x\,0 \arrow[r, dashed] & \mathsf{W}\,x\,1 \arrow[r, Rightarrow] & \mathsf{W}^{\mathsf{ra}}\,y\,1 \arrow[r] & \mathsf{R}^{\mathsf{ra}}\,y\,1 \arrow[r, Rightarrow] & \mathsf{R}\,x\,0 \end{tikzcd}$$

(PUB1) ✗

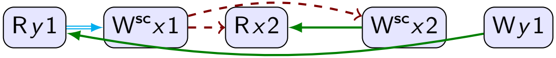
$$x := 0\,;\ x := 1\,;\ \mathtt{F}^{\mathsf{rel}}\,;\ y := 1 \ \parallel\ r := y\,;\ \mathtt{F}^{\mathsf{acq}}\,;\ s := x$$

$$\begin{tikzcd} \mathsf{W}\,x\,0 \arrow[r, dashed] & \mathsf{W}\,x\,1 \arrow[r, Rightarrow] & \mathsf{F}^{\mathsf{rel}} \arrow[r, Rightarrow] & \mathsf{W}\,y\,1 \arrow[r] & \mathsf{R}\,y\,1 \arrow[r, Rightarrow] & \mathsf{F}^{\mathsf{acq}} \arrow[r, Rightarrow] & \mathsf{R}\,x\,0 \end{tikzcd}$$

(PUB2) ✗

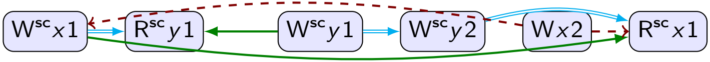
SC Access/Fences

$$r := y;\ x^{\mathrm{sc}} := 1;\ s := x \parallel x^{\mathrm{sc}} := 2;\ y := 1$$



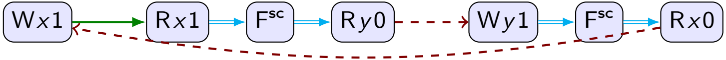
(SC1) ✓

$$x^{\mathrm{sc}} := 1;\ r := y^{\mathrm{sc}} \parallel y^{\mathrm{sc}} := 1;\ y^{\mathrm{sc}} := 2;\ x := 2;\ s := x^{\mathrm{sc}}$$



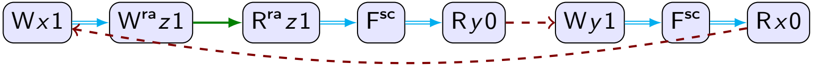
(SC2) ✓

$$x := 1 \parallel r := x;\ \mathrm{F}^{\mathrm{sc}};\ r := y \parallel y := 1;\ \mathrm{F}^{\mathrm{sc}};\ r := x$$



(SC3) ✗

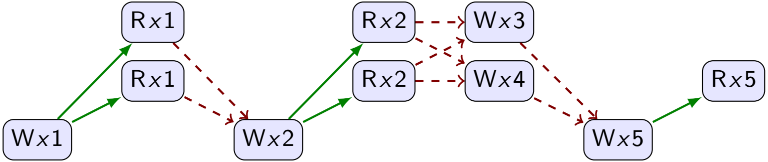
$$x := 1;\ z^{\mathrm{ra}} := 1; \parallel r^{\mathrm{ra}} := z;\ \mathrm{F}^{\mathrm{sc}};\ r := y \parallel y := 1;\ \mathrm{F}^{\mathrm{sc}};\ r := x$$



(SC4) ✗

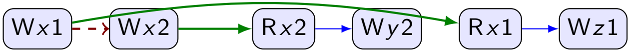
Coherence

$x := 1 \parallel x := 2 \parallel x := 3 \parallel x := 4 \parallel x := 5 \parallel r := x;\, r := x;\, r := x;\, r := x;\, r := x$



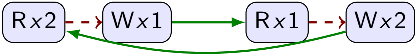
(CO1) ✓

$x := 1;\, x := 2 \parallel y := x;\, z := x$



(CO2) ✓

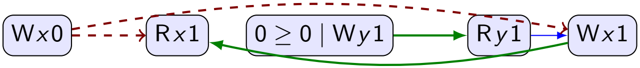
$r := x;\, x := 1 \parallel s := x;\, x := 2$



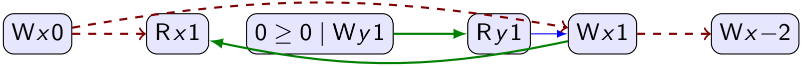
(TC16) ✗

Internal Reads

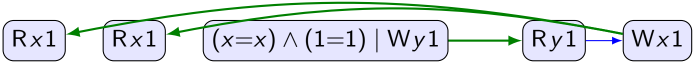
$x := 0;\ (r := x;\ \texttt{if}\,(r \geq 0)\{y := 1\} \parallel x := y)$ (TC1) ✓



$x := 0;\ (r := x;\ \texttt{if}\,(r \geq 0)\{y := 1\} \parallel x := y \parallel x := -2)$ (TC9) ✓

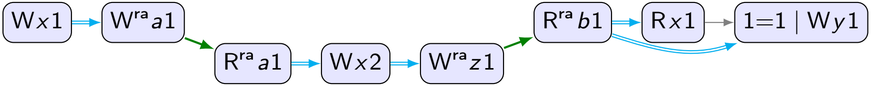


$r := x;\ s := x;\ \texttt{if}\,(r = s)\{y := 1\} \parallel x := y$ (TC2) ✓



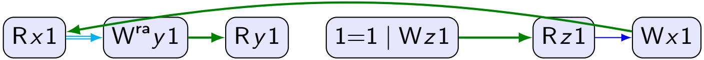
Internal Reads+Synchronization

$$x := 1;\ a^{\mathsf{ra}} := 1;\ \mathtt{if}\,(z^{\mathsf{ra}})\{y := x\} \parallel \mathtt{if}\,(a^{\mathsf{ra}})\{x := 2;\ z^{\mathsf{ra}} := 1\}$$



(IN1) ✗

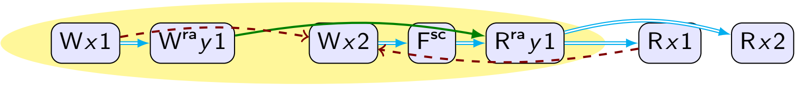
$$r := x;\ y^{\mathsf{ra}} := 1;\ s := y;\ z := s \parallel x := z$$



(IN2) ✓

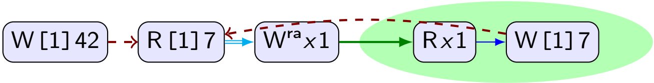
Local data race freedom

$$(x := 1;\ y^{\mathsf{ra}} := 1) \parallel (x := 2;\ \mathtt{F}^{\mathsf{sc}};\ \mathtt{if}\,(y^{\mathsf{ra}})\,\{r := x;\ s := x\})$$



(past) ✗

$$(r := 1;\ [r] := 42;\ s := [r];\ x^{\mathsf{ra}} := r) \parallel (r := x;\ [r] := 7)$$



(future) ✗

Valid Transformations

$$
\begin{array}{rcll}
[\![r := x\,;\ s := y\,;\ C]\!] & = & [\![s := y\,;\ r := x\,;\ C]\!] & \text{if } r \neq s \\
[\![x := M\,;\ y := N\,;\ C]\!] & = & [\![y := N\,;\ x := M\,;\ C]\!] & \text{if } x \neq y \\
[\![x := M\,;\ s := y\,;\ C]\!] & = & [\![s := y\,;\ x := M\,;\ C]\!] & \text{if } x \neq y \text{ and } s \notin \mathrm{id}(M) \\
[\![x^{\mu} := M\,;\ s := y\,;\ C]\!] & \supseteq & [\![s := y\,;\ x^{\mu} := M\,;\ C]\!] & \text{if } x \neq y \text{ and } s \notin \mathrm{id}(M) \\
[\![x := M\,;\ s := y^{\mu}\,;\ C]\!] & \supseteq & [\![s := y^{\mu}\,;\ x := M\,;\ C]\!] & \text{if } x \neq y \text{ and } s \notin \mathrm{id}(M) \\
[\![\mathrm{F}^{\mu}\,;\ \mathrm{F}^{\mu}\,;\ C]\!] & \supseteq & [\![\mathrm{F}^{\mu}\,;\ s := r\,;\ C]\!] & \\
[\![r := x^{\mu}\,;\ s := x^{\mu}\,;\ C]\!] & \supseteq & [\![r := x^{\mu}\,;\ s := r\,;\ C]\!] & \\
[\![r_1 := x\,;\ s := y\,;\ r_2 := x\,;\ C]\!] & \supseteq & [\![r_1 := x\,;\ r_2 := r_1\,;\ s := y\,;\ C]\!] & \text{if } r_2 \neq s
\end{array}
$$

- Reordering: W $\leftrightarrow$ W, R $\leftrightarrow$ R, W $\leftrightarrow$ R
- Roach motel: Relaxed $\leftarrow$ Acquire, Relaxed $\rightarrow$ Release
- Elimination: Redundant fence, Redundant read, Common Subexpression

Valid Transformations

$$
\begin{aligned}
[\![C \parallel \texttt{var}\, x;\, D]\!] &= [\![\texttt{var}\, x;\, (C \parallel D)]\!] && \text{if } x \notin \mathrm{id}(C)\\
[\![\texttt{if}\,(M)\,\{C\}\,\texttt{else}\,\{C\}]\!] &\supseteq [\![C]\!] && \\
[\![\texttt{if}\,(M)\,\{C\}\,\texttt{else}\,\{C\}]\!] &\subseteq^{\star} [\![C]\!] && \\
[\![\texttt{if}\,(M)\,\{C\}\,\texttt{else}\,\{D\}]\!] &= [\![C]\!] && \text{if } M \text{ is a tautology}\\
[\![x := M;\, x := N;\, C]\!] &\supseteq^{\star} [\![x := N;\, C]\!] && \\
[\![x^{\mu} := M;\, r := x;\, C]\!] &\supseteq^{\star} [\![x^{\mu} := M;\, r := M;\, C]\!] && \\
[\![r := x;\, C]\!] &\supseteq^{\star} [\![C]\!] && \text{if } r \notin \mathrm{id}(C)
\end{aligned}
$$

- Scope extrusion
- Code lifting, Case analysis* (*See paper)
- Elimination: Dead code, Dead store*, Store forwarding*, Irrelevant read*

Other Transformations

- Valid✓ Proof✗
  - Redundant write after read elimination
- Sound observationally✓ Valid✗
  - Access mode strengthening (rlx $\rightarrow$ ra $\rightarrow$ sc)
  - Commuting/Eliminating some synchronizations
  - Implementing synchronizations using fences
  - Implementing locks using synchronizations
- Sound observationally? Valid✗
  - Access mode weakening, eg (sc $\rightarrow$ ra $\rightarrow$ rlx)
  - Lock elision
- Sound observationally✗
  - Thread inlining
  - Relevant read introduction
  - Write introduction

Other results and limitations

- Results
  - Compositional proof rule for PLTL
  - Efficient ARM implementation
  - Local data race freedom

Other results and limitations

- Results
  - Compositional proof rule for PLTL
  - Efficient ARM implementation
  - Local data race freedom
- Limitations
  - No: loops or functions, sequential composition
  - No: mixed size access, separate address dependencies
  - Limited: optimizations, logic for OOTA

Other results and limitations

- Results
  - Compositional proof rule for PLTL
  - Efficient ARM implementation
  - Local data race freedom
- Limitations
  - No: loops or functions, sequential composition
  - No: mixed size access, separate address dependencies
  - Limited: optimizations, logic for OOTA
  - MCA: ARM✓ x86-64✓ RISC-V✓ POWER✗

Other results and limitations

- Results
  - Compositional proof rule for PLTL
  - Efficient ARM implementation
  - Local data race freedom
- Limitations
  - No: loops or functions, sequential composition
  - No: mixed size access, separate address dependencies
  - Limited: optimizations, logic for OOTA
  - MCA: ARM✓ x86-64✓ RISC-V✓ POWER✗
    - Add per-location partial order: $\sqsubseteq$
    - Require *observation:* if d/e conflict and $d \leq e$ then $d \sqsubseteq e$
    - Prefixing (p5b): if e conflicts then $d \sqsubseteq' e$
    - Fulfillment (f4): if c conflicts then $c \sqsubseteq d$ or $e \sqsubseteq c$


PRINCIPLES
Image in the Public domain


Compositionality
Image in the Public domain

Compositionality

+ Construction

Image in the Public domain

Compositionality

+ Construction
+ Reasoning
+ Local races
+ Safety properties

Image in the Public domain


LOGIC
Image in the Public domain

LOGIC

+ PRECONDITIONS

Image in the Public domain

LOGIC

+ PRECONDITIONS
+ TEMPORAL SAFETY

Image in the Public domain


1
ONE
ORDER
Image in the Public domain

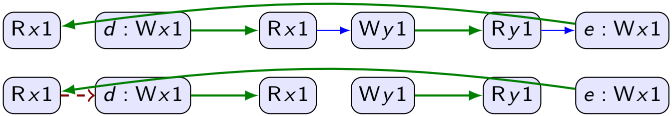

THANKS
Image in the Public domain

Non-MCA Architectures?

- Two partial orders:
  - $\leq$ causal order, as before
  - $\sqsubseteq$ per-location order
- Require: $d \sqsubseteq e$ when $d \leq e$ and they conflict (same location)
- When prefixing d:

  (p5b) if d and e conflict then $d \sqsubseteq' e$,
- When d *fulfills* e (on x):

  (f4) for every conflicting write c, either $c \sqsubseteq d$ or $e \sqsubseteq c$.
- Example: $r := x;\, x := 1 \parallel y := x \parallel x := y$

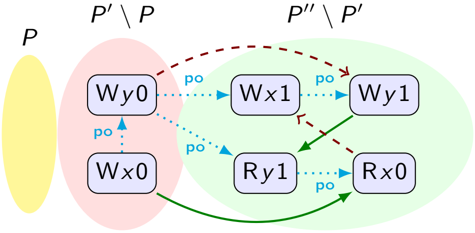


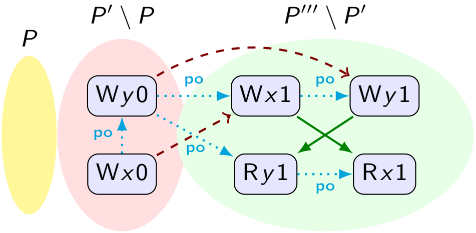
($\leq$)

($\sqsubseteq$)

LDRF: Reads from the Past

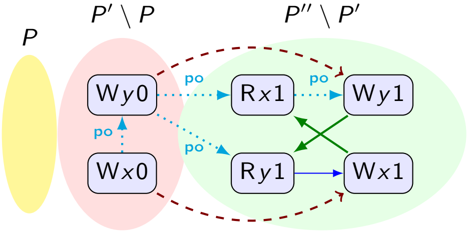
$$x := 0;\, y := 0;\, (x := 1;\, y := 1 \parallel \mathtt{if}(y)\{r := x\})$$

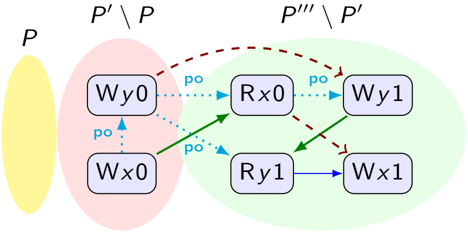




LDRF: Reads from the Future

$$x := 0;\ y := 0;\ (r := x;\ y := 1 \parallel s := y;\ x := s)$$





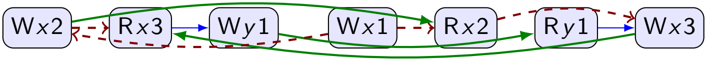
More OOTA

$$(y := x + 1 \parallel x := y)$$

(OOTA6) ✗

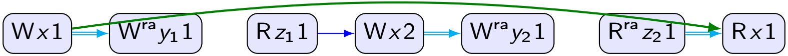
$$x := 2;\ \texttt{if}\,(x \neq 2)\,\{y := 1\} \parallel x := 1;\ r := x;\ \texttt{if}\,(y)\,\{x := 3\}$$

(OOTA7) ✗



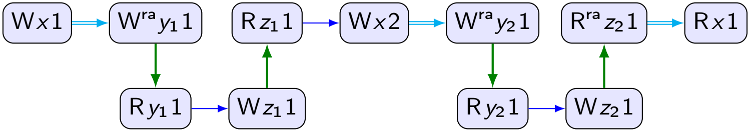
Blockers

$$\texttt{var}\, x;\ (x := 1;\ y_1^{\mathsf{ra}} := 1 \ \parallel\ \texttt{if}\,(z_1)\,\{x := 2\};\ y_2^{\mathsf{ra}} := 1 \ \parallel\ r := z_2^{\mathsf{ra}};\ s := x)$$



✗

Context $z_1 := y_1 \ \parallel\ z_2 := y_2 \ \parallel\ [-]$



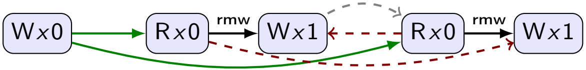
✗

RMWs

$\xrightarrow{\text{rmw}} \subseteq \leq$:

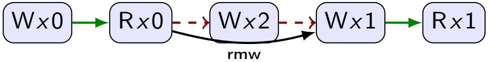
- If $(\mathsf{R}x2) \xrightarrow{\text{rmw}} (\mathsf{W}x3)$ and $\color{red}{(\mathsf{W}x1)} \leq (\mathsf{W}x3)$ then $\color{red}{(\mathsf{W}x1)} \leq (\mathsf{R}x2)$
- If $(\mathsf{R}x2) \xrightarrow{\text{rmw}} (\mathsf{W}x3)$ and $(\mathsf{R}x2) \leq \color{red}{(\mathsf{W}x3)}$ then $(\mathsf{W}x3) \leq \color{red}{(\mathsf{W}x3)}$
- $\xrightarrow{\text{rmw}}$ does not coalesce in $\parallel$ or prefixing

$$x := 0;\ \big(\mathtt{FADD}^{\mathsf{rlx},\mathsf{rlx}}(x,1)\ \parallel\ \mathtt{FADD}^{\mathsf{rlx},\mathsf{rlx}}(x,1)\big)$$



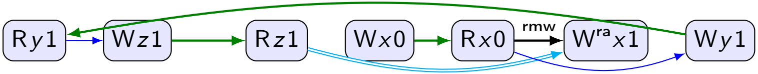
(RMW0) ✗

$$x := 0;\ s := \mathtt{FADD}^{\mathsf{rlx},\mathsf{rlx}}(x,1)\ \parallel\ x := 2;\ s := x$$



(RMW1) ✗

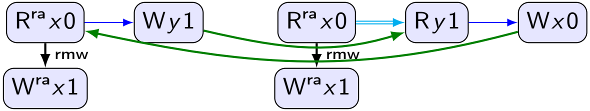
$$r := y;\ z := r\ \parallel\ r := z;\ x := 0;\ s := \mathtt{FADD}^{\mathsf{rlx},\mathsf{ra}}(x,1);\ y := s{+}1$$



(RMW2) ✓

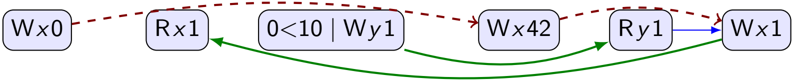
PS2.0 Examples

$$r := \texttt{FADD}^{\mathsf{ra},\mathsf{ra}}(x, 1);\ \texttt{if}\,(r{=}0)\,\{y := 1\} \ \parallel\ r := \texttt{FADD}^{\mathsf{ra},\mathsf{ra}}(x, 1);\ \texttt{if}\,(r{=}0)\,\{\texttt{if}\,(y)\,\{x := 0\}\}$$



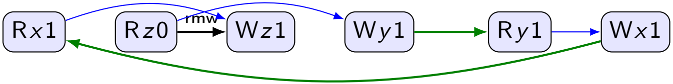
(CDRF) ✗

$$x := 0;\ \big(r := \texttt{CAS}^{\mathsf{rlx},\mathsf{rlx}}(x, 0, 1);\ \texttt{if}\,(r{<}10)\,\{y := 1\} \ \parallel\ x := 42;\ x := y\big)$$



(GA+E) ✓

$$r := x;\ s := \texttt{FADD}^{\mathsf{rlx},\mathsf{rlx}}(z, r);\ y := s{+}1 \ \parallel\ x := y$$

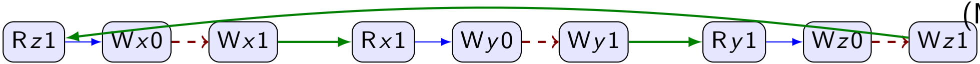


(RP) ✓

MCA Examples

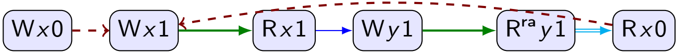
$\texttt{if}\,(z)\,\{x := 0\};\ x := 1 \parallel \texttt{if}\,(x)\,\{y := 0\};\ y := 1 \parallel \texttt{if}\,(y)\,\{z := 0\};\ z := 1$

(MCA1) ✗



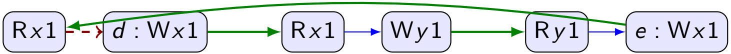
$x := 0;\ x := 1 \ \parallel\ y := x \ \parallel\ r := y^{\mathsf{ra}};\ s := x$

(MCA2) ✗



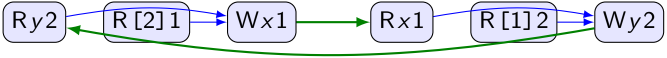
$r := x;\ x := 1 \ \parallel\ y := x \ \parallel\ x := y$

(MCA3) ✗



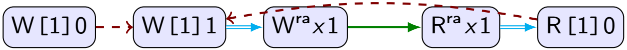
Address Calculation Examples

$$r := y\,;\ s := [r]\,;\ x := s \ \parallel\ r := x\,;\ s := [r]\,;\ y := s$$



(ADDR1) ✗

$$(r := 1;\ [r] := 0;\ [r] := 1;\ x^{\mathsf{ra}} := r) \ \parallel\ (r := x^{\mathsf{ra}};\ s := [r])$$



(ADDR2) ✗