1 MODEL

μ	::= wk	(Weak)	ς	::= cta	(Thread group)
	| rlx	(Relaxed)		| gpu	(Processor)
	| ra	(Release/Acquire)		| sys	(System)
	| sc	(Sequentially Consistent)			

Orders/Relations in model

- $\trianglelefteq$ is the old $\leq$ (without coherence stuff from F4 and P5B).
  This provides the NO-TAR axiom.
- $\leq$ is a the *happens-before* suborder, which only includes rf when they are morally strong.
  This serves as a cross-location transitive kernel for the per-location order.
- $\sqsubseteq$ is a per-location order that relates morally strong and poloc accesses
  This includes $\leq$ for morally strong accesses.
  This provides the SC-PER-LOC axiom.

Write $d \vartriangle e$ if they conflict (ie, read/write or write/write, same location).
Write $d \blacktriangle e$ if they conflict and are morally strong

Definition 1.1. A *pomset with preconditions* is a tuple $(E, \lambda, \leq, \trianglelefteq, \sqsubseteq)$ where

(M1) E is a set of *events*
(M2) $\lambda : E \longrightarrow (\Phi \times \mathcal{A})$ is a *labeling* from which we derive functions
- $\Phi : E \longrightarrow \Phi$ *(formulae)*
- $\mathcal{A} : E \longrightarrow \mathcal{A}$ *(actions)*

(M3) $\leq \subseteq (E \times E)$, $\trianglelefteq \subseteq (E \times E)$, and $\sqsubseteq \subseteq (E \times E)$ are partial orders
(M4) $\bigwedge_e \Phi(e)$ is satisfiable *(consistency)*
(M5) if $d \trianglelefteq e$ then $\Phi(e)$ implies $\Phi(d)$ *(causal strengthening)*
(M6) if $d \leq e$ then $d \trianglelefteq e$
(M7) if $d \leq e$ and d conflicts with e then $d \sqsubseteq e$

We say $d < e$ when $d \leq e$ and $d \neq e$, and similarly for $\vartriangleleft$ and $\sqsubset$.

Definition 1.2 (Strong fulfillment). We say $\mathcal{A}(d) = (\mathsf{W}xv)$ *fulfills* $\mathcal{A}(e) = (\mathsf{R}xv)$ if

(F3A) $d \vartriangleleft e$
(F3B) $d < e$ if d is morally strong with e
(F3C) $d \sqsubseteq e$ (if d is not morally strong with e)
(F4) $\forall \mathcal{A}(c) = (\mathsf{W}x..)$ either $c \sqsubseteq d$ or $e \sqsubseteq c$,

Definition 1.3 (Weak fulfillment). We say $\mathcal{A}(d) = (\mathsf{W}xv)$ *fulfills* $\mathcal{A}(e) = (\mathsf{R}xv)$ if

(F3A) $d \vartriangleleft e$
(F3B) $d < e$ if d is morally strong with e
(F3C) $e \not\sqsubseteq d$ (if d is not morally strong with e)
(F4) $\forall \mathcal{A}(c) = (\mathsf{W}x..)$ either $c \precsim d$ or $e \precsim c$, where

$$d \precsim e \text{ when } \begin{cases} d \sqsubseteq e & \text{if } d \text{ is morally strong with } e \\ e \not\sqsubseteq d & \text{otherwise} \end{cases}$$

If all accesses are morally strong with each other, weak fulfillment degenerates to

(F3) $d < e$
(F4) $\forall \mathcal{A}(c) = (\mathsf{W}x..)$ either $c \sqsubseteq d$ or $e \sqsubseteq c$

If no accesses are morally strong with each other, weak fulfillment degenerates to

(F3) $e \not\sqsubseteq d$
(F4) $\nexists \mathcal{A}(c) = (\mathsf{W}x..)$ both $d \sqsubset c$ and $c \sqsubset e$

Definition 1.4. 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) $\leq' \supseteq \leq$, $\trianglelefteq' \supseteq \trianglelefteq$, and $\sqsubseteq' \supseteq \sqsubseteq$
(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 $d = (\mathsf{R}vx)$ 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 conflicts with e then $d \sqsubseteq' e$
(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 $d \leq' e$

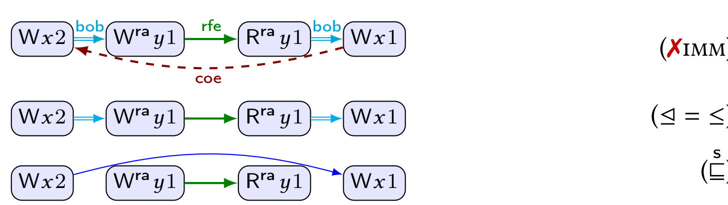
2 IMM EXAMPLES

Interpreting this definition for the IMM:

- No wk, default is rlx
- All threads in same cta (only one scope)
- Actions are morally strong when both are ra/sc, mimicking happens-before
- Strong fulfillment may do the right thing

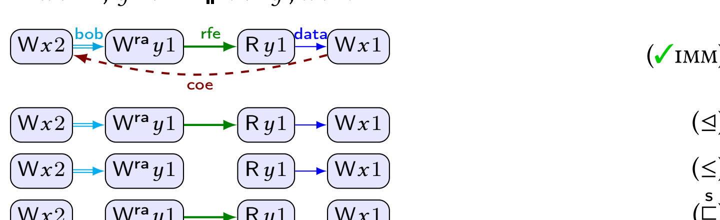
Disallowed by IMM:

$$x := 2;\ y^{\mathsf{ra}} := 1 \ \|\ r := y^{\mathsf{ra}};\ x := 1$$



Allowed by IMM, but not by Power/ARMv7/ARMv8/TSO:

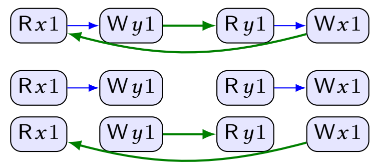
$$x := 2;\ y^{\mathsf{ra}} := 1 \ \|\ r := y;\ x := r$$



Anton says we could potentially disallow by adding an axiom to IMM forbidding cycles in co $\cup([W];\mathsf{rfe}^?;([R^{\mathsf{ra}}]\cup \mathsf{po};[FW^{\mathsf{ra}}]);\mathsf{ar}^*;[W])$

Need $\trianglelefteq$ to prevent thin air on rlx:

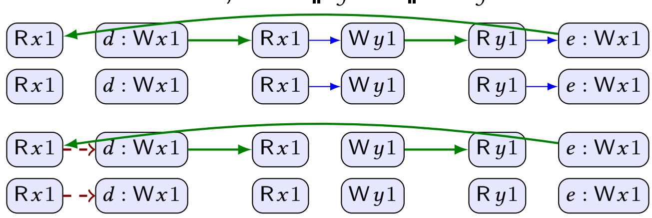
$$y := x \parallel x := y$$



($\trianglelefteq$)
($\leq$)
($\overset{\mathsf{s}}{\sqsubseteq}$)

Example from talk:

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

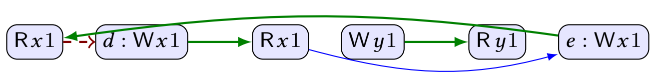


($\trianglelefteq$)
($\leq$)
($\overset{\mathsf{s}}{\sqsubseteq}$)
($\overset{\mathsf{w}}{\sqsubseteq}$)

Comment: Two order idea from OOPLSA talk does not work for this example. In this setting it corresponds to:

- Require: $d \sqsubseteq e$ when $d \trianglelefteq e$ and they conflict

Using this, we would have a cycle (weak/strong fulfillment not relevant here):



($\sqsubseteq$)

3 PTX EXAMPLES

Based on [Lustig et al. 2019; NVIDIA 201].

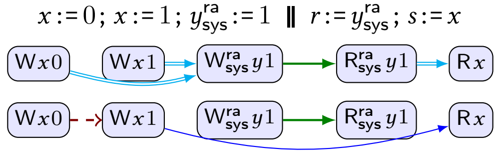
PTX requires weak fulfillment.

Default scope is cta. In examples, all threads in different ctas.

Default mode is wk.

($\mathsf{R}x0$) must be forbidden. Before fulfilling the read:

$$x := 0;\ x := 1;\ y^{\mathsf{ra}}_{\mathsf{sys}} := 1 \parallel r := y^{\mathsf{ra}}_{\mathsf{sys}};\ s := x \qquad (\text{PUB1}_{\text{SYS}})$$

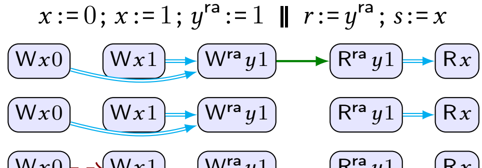


($\trianglelefteq\ =\ \leq$)
($\sqsubseteq$)

($\mathsf{W}x1$) $\sqsubsetneq$ ($\mathsf{R}x$) is required by M7, enforcing publication.

($\mathsf{R}x0$) must be allowed:

$$x := 0;\ x := 1;\ y^{\mathsf{ra}} := 1 \parallel r := y^{\mathsf{ra}};\ s := x \qquad (\text{PUB1}_{\text{CTA}})$$



($\trianglelefteq$)
($\leq$)
($\sqsubseteq$)

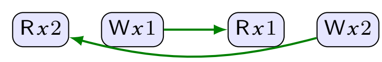
We do not have ($\mathsf{W}^{\mathsf{ra}}y1$) $\leq$ ($\mathsf{R}^{\mathsf{ra}}y1$) since F3 only requires order for things that are morally strong.

Another example that may be of interest (nothing morally strong). Can this $(\mathsf{R}\,x\,0)$?

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

PTX allows TC16 for events that are not mutually strong (TC16_{WK}), but disallows it when events are mutually strong (TC16_{SYS}). Note that $\leq$ imposes no requirements here. Fulfillment imposes no order. This example shows that F3C cannot be strengthened to require that $d \sqsubseteq e$.

$$r := x;\ x := 1 \ \parallel\ s := x;\ x := 2 \qquad (\text{TC16}_{\text{WK}})$$

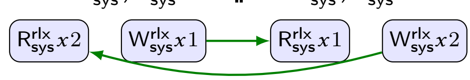


($\trianglelefteq$)

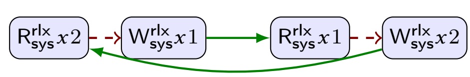
($\leq$)

($\sqsubseteq$)

$$r := x^{\mathsf{rlx}}_{\mathsf{sys}};\ x^{\mathsf{rlx}}_{\mathsf{sys}} := 1 \ \parallel\ s := x^{\mathsf{rlx}}_{\mathsf{sys}};\ x^{\mathsf{rlx}}_{\mathsf{sys}} := 2 \qquad (\text{TC16}_{\text{SYS}})$$



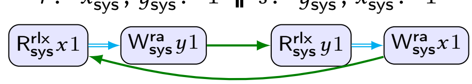
($\trianglelefteq\ =\ \leq$)



($\sqsubseteq$)

About Release-Acquire semantics. Not sure the status of this example in C11 and IMM:

$$r := x^{\mathsf{rlx}}_{\mathsf{sys}};\ y^{\mathsf{ra}}_{\mathsf{sys}} := 1 \ \parallel\ s := y^{\mathsf{rlx}}_{\mathsf{sys}};\ x^{\mathsf{ra}}_{\mathsf{sys}} := 1 \qquad (\text{LB-REL})$$

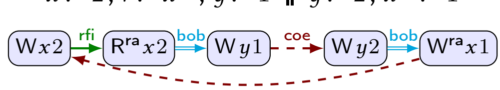


($\trianglelefteq\ =\ \leq$)

4 OOPSLA COUNTEREXAMPLES

Anton example 1 (Allowed by ARM) `[rfi-bob-coe-coe]`

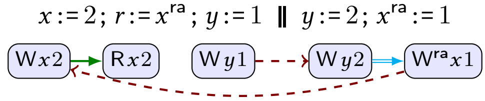
$$x := 2;\ r := x^{\mathsf{ra}};\ y := 1 \ \parallel\ y := 2;\ x^{\mathsf{ra}} := 1$$



(✓ARM8)

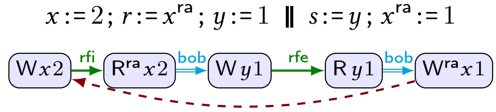
To allow this, weaken ra to rlx when read fulfilled by relaxed write of same thread (don't need to allow this when the write is part of an RMW).

$$x := 2;\ r := x^{\mathsf{ra}};\ y := 1 \ \parallel\ y := 2;\ x^{\mathsf{ra}} := 1$$



RF variant `[rfi-bob-rfe-coe]`:

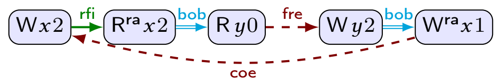
$$x := 2;\ r := x^{\mathsf{ra}};\ y := 1 \ \parallel\ s := y;\ x^{\mathsf{ra}} := 1$$



(✓ARM8)

TSO variant [rfi-bob-fre-coe]:

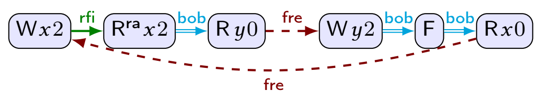
$$x := 2;\ r := x^{\mathsf{ra}};\ s := y \ \parallel\ y := 2;\ x^{\mathsf{ra}} := 1$$



(✓ARM8)

Double FRE variant [rfi-bob-fre-fre]:

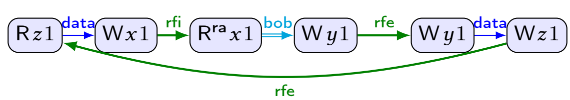
$$x := 2;\ r := x^{\mathsf{ra}};\ s := y \ \parallel\ y := 2;\ \mathsf{F};\ r := x$$



(✓ARM8)

It does not seem possible to do this only with rfe. ARM disallows this [data-rfi-bob-rfe-rfe]:

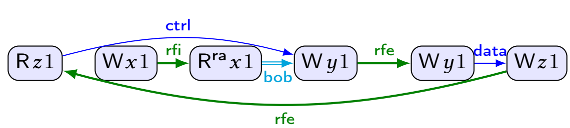
$$x := z;\ r := x^{\mathsf{ra}};\ y := 1 \ \parallel\ z := y$$



(✗ARM8)

It also disallows [ctrl-rfi-bob-rfe-rfe]:

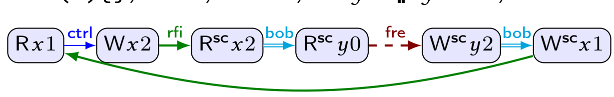
$$\mathtt{if}(z)\{\};\ x := 1;\ r := x^{\mathsf{ra}};\ y := 1 \ \parallel\ z := y$$



(✗ARM8)

ARM allows some counterintuitive results for SC access [ctrl-rfi-bob-fre-rfe]:

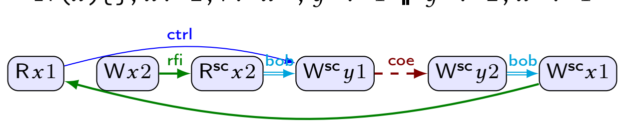
$$\mathtt{if}(x)\{\};\ x := 2;\ r := x^{\mathsf{sc}};\ s := y^{\mathsf{sc}} \ \parallel\ y^{\mathsf{sc}} := 2;\ x^{\mathsf{sc}} := 1$$



(✓ARM8)

Not possible with coe [ctrl-rfi-bob-coe-rfe]:

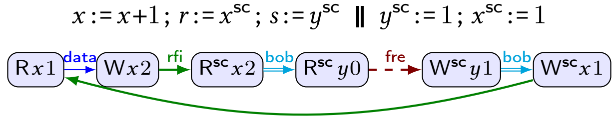
$$\mathtt{if}(x)\{\};\ x := 2;\ r := x^{\mathsf{sc}};\ y^{\mathsf{sc}} := 1 \ \parallel\ y^{\mathsf{sc}} := 2;\ x^{\mathsf{sc}} := 1$$



(✗ARM8)

This is not allowed with a data dependency instead of a control dependency [data-rfi-bob-fre-rfe]:

$$x := x{+}1;\ r := x^{\mathsf{sc}};\ s := y^{\mathsf{sc}} \ \parallel\ y^{\mathsf{sc}} := 1;\ x^{\mathsf{sc}} := 1$$

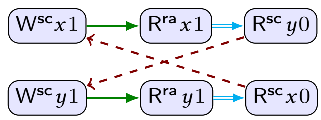


(✗ARM8)

5 SC EXAMPLES

This execution is allowed by trailing-sync compilation to power `[IRIW-aqc-sc]` from [Lahav et al. 2017, §A.2].

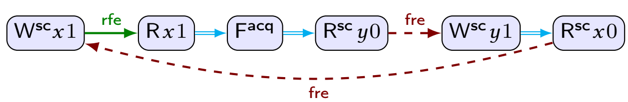
$$x^{\mathsf{sc}} := 1 \parallel x^{\mathsf{sc}} := 1 \parallel r := x^{\mathsf{ra}};\, s := y^{\mathsf{sc}} \parallel r := y^{\mathsf{ra}};\, s := x^{\mathsf{sc}}$$



(✓POWER)

[Lahav et al. 2017, §A.2] claims that ARM8 allows this `[RWC+acq+sc]`, but herd7 rejects it. Reason: they are citing the flowing/pop model [Flur et al. 2016] rather than [Pulte et al. 2018].

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



(✗ARM8)

6 MORE MODEL

These definitions need to be updated to include the additional orders.

Definition 6.1. A pomset is x-*closed* if

- every $\mathcal{A}(e) = (\mathsf{R}x..)$ is fulfilled
- every $\Phi(e)$ is independent of x: $\big(\forall v.\ \Phi(e) \vDash \Phi(e)[v/x] \vDash \Phi(e)\big)$

Definition 6.2. Let $P \in (\nu x . \mathcal{P})$ when $P \in \mathcal{P}$ and P is x-closed

Definition 6.3. Let $P \in (\phi \rhd \mathcal{P})$ when $P \in \mathcal{P}$ and $(\forall e \in E)\ \Phi(e)$ implies ϕ

Definition 6.4. Let $P' \in (\mathcal{P}[M/x])$ when $(\exists P \in \mathcal{P})$
$E' = E$, $\leq' = \leq$, $\mathcal{A}' = \mathcal{A}$, and $(\forall e \in E')\ \Phi'(e) = \Phi(e)[M/x]$

Definition 6.5. 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

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

Language

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

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

A citation: [Jagadeesan et al. 2020]

REFERENCES

Shaked Flur, Kathryn E. Gray, Christopher Pulte, Susmit Sarkar, Ali Sezgin, Luc Maranget, Will Deacon, and Peter Sewell. 2016. Modelling the ARMv8 architecture, operationally: concurrency and ISA. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016*, Rastislav Bodík and Rupak Majumdar (Eds.). ACM, 608–621. https://doi.org/10.1145/2837614.2837615

Radha Jagadeesan, Alan Jeffrey, and James Riely. 2020. Pomsets with Preconditions: A Simple Model of Relaxed Memory. *Proc. ACM Program. Lang.* 4, OOPSLA, Article 194 (Nov. 2020), 30 pages. https://doi.org/10.1145/3428262

Ori Lahav, Viktor Vafeiadis, Jeehoon Kang, Chung-Kil Hur, and Derek Dreyer. 2017. Repairing sequential consistency in C/C++11. In *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2017, Barcelona, Spain, June 18-23, 2017*, Albert Cohen and Martin T. Vechev (Eds.). ACM, 618–632. https://doi.org/10.1145/3062341.3062352

Daniel Lustig, Sameer Sahasrabuddhe, and Olivier Giroux. 2019. A Formal Analysis of the NVIDIA PTX Memory Consistency Model. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 2019, Providence, RI, USA, April 13-17, 2019*, Iris Bahar, Maurice Herlihy, Emmett Witchel, and Alvin R. Lebeck (Eds.). ACM, 257–270. https://doi.org/10.1145/3297858.3304043

NVIDIA. 201?. CUDA Toolkit Documentation v11.1.1. https://docs.nvidia.com/cuda/parallel-thread-execution/index.html#memory-consistency-model.

Christopher Pulte, Shaked Flur, Will Deacon, Jon French, Susmit Sarkar, and Peter Sewell. 2018. Simplifying ARM concurrency: multicopy-atomic axiomatic and operational models for ARMv8. *PACMPL* 2, POPL (2018), 19:1–19:29. https://doi.org/10.1145/3158107