Reachability and Safety Games under TSO Semantics

SCooL / GandALF 2024 in Reykjavik

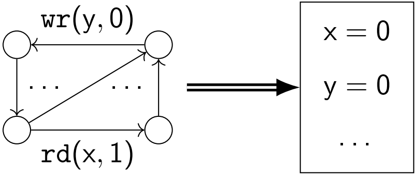
Stephan Spengler

Uppsala University, Sweden

20 June 2024

Reachability and Safety Games under TSO semantics

Reachability and Safety Games under TSO semantics

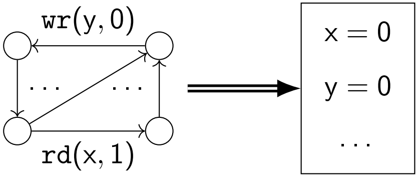


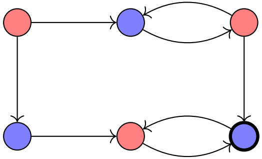
$x = 0$

$y = 0$

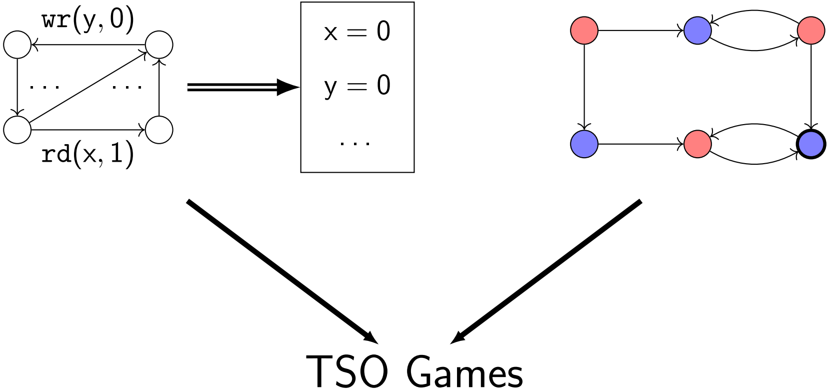
$\dots$

Reachability and Safety Games under TSO semantics





Reachability and Safety Games under TSO semantics



$x = 0$

$y = 0$

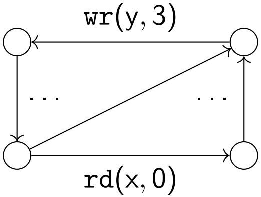
. . .

TSO Games

Total Store Order

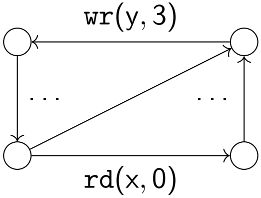
Total Store Order

Proc^1:

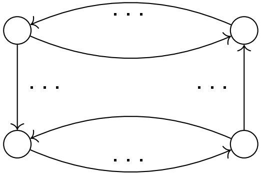


Total Store Order

Proc^1:



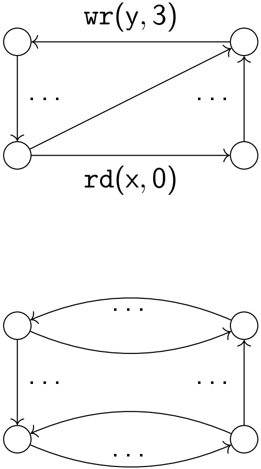
Proc^2:



Total Store Order

Proc^1:

Proc^2:



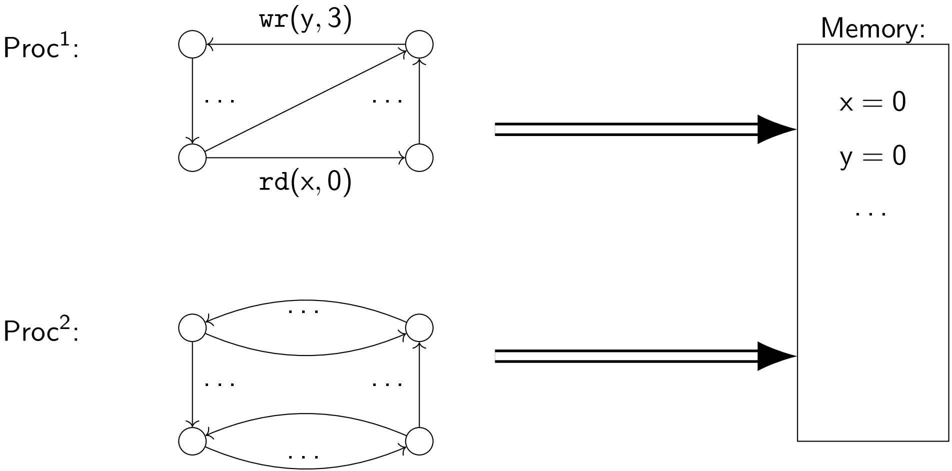
Memory:

$x = 0$

$y = 0$

. . .

Total Store Order



Proc^1:

$\texttt{wr}(y, 3)$

$\texttt{rd}(x, 0)$

Proc^2:

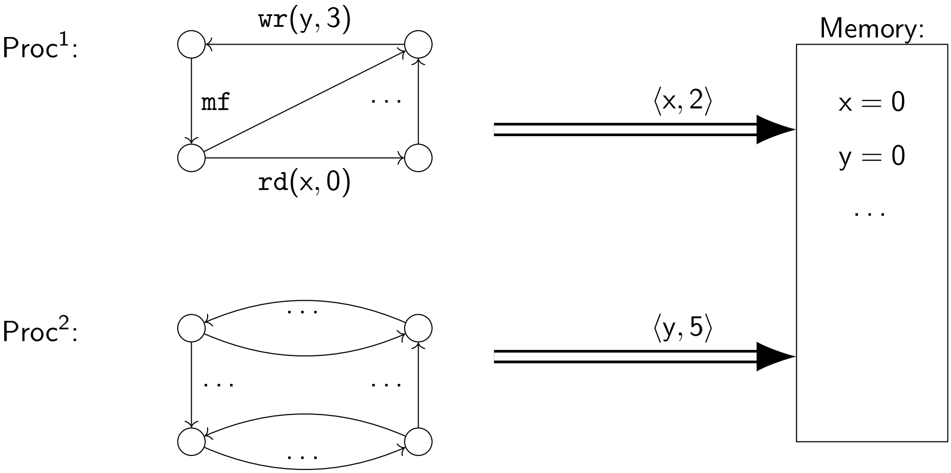
Memory:

$x = 0$

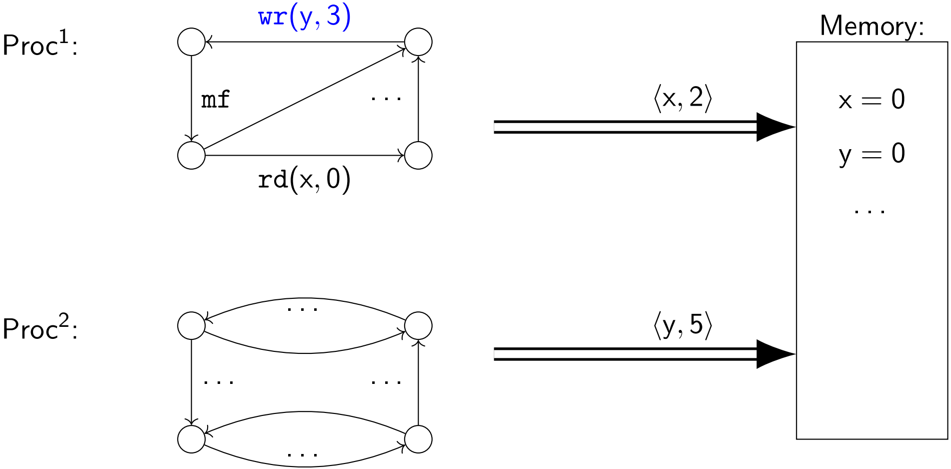
$y = 0$

$\dots$

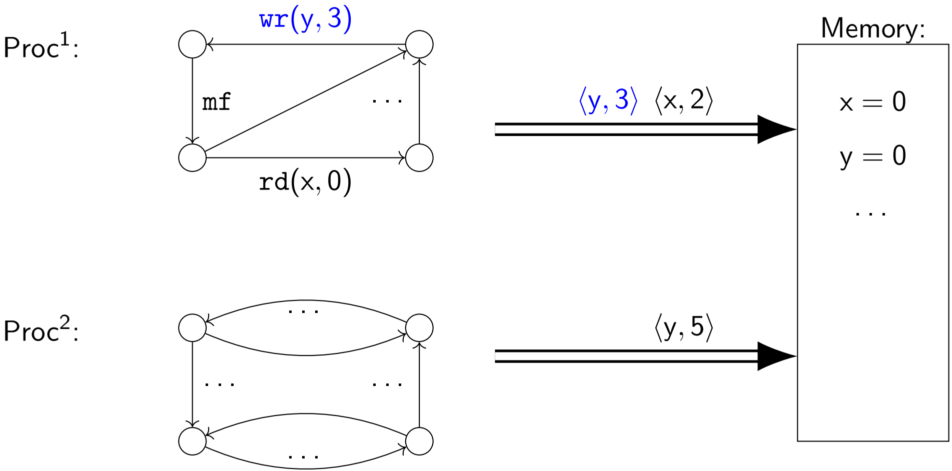
Total Store Order



Total Store Order



Total Store Order



Proc^1:

$\mathtt{wr}(y, 3)$

mf

. . .

$\mathtt{rd}(x, 0)$

$\langle y, 3\rangle$ $\langle x, 2\rangle$

Memory:

$x = 0$

$y = 0$

. . .

Proc^2:

. . .

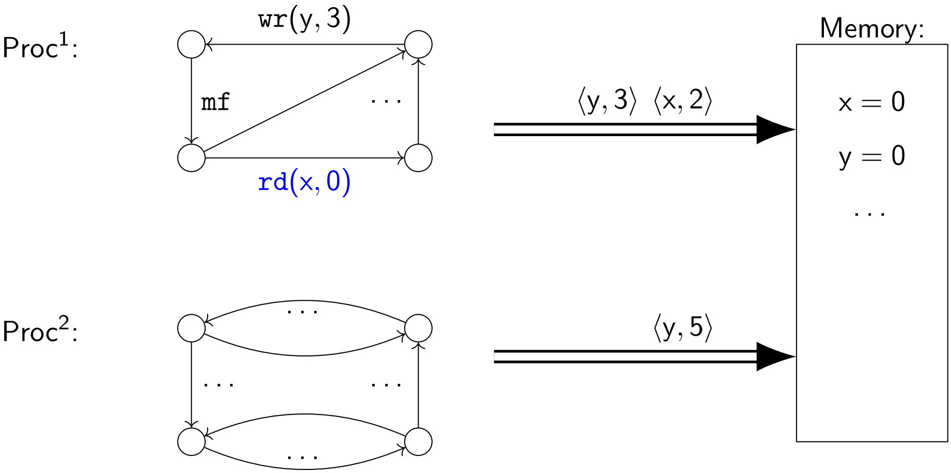
. . .

. . .

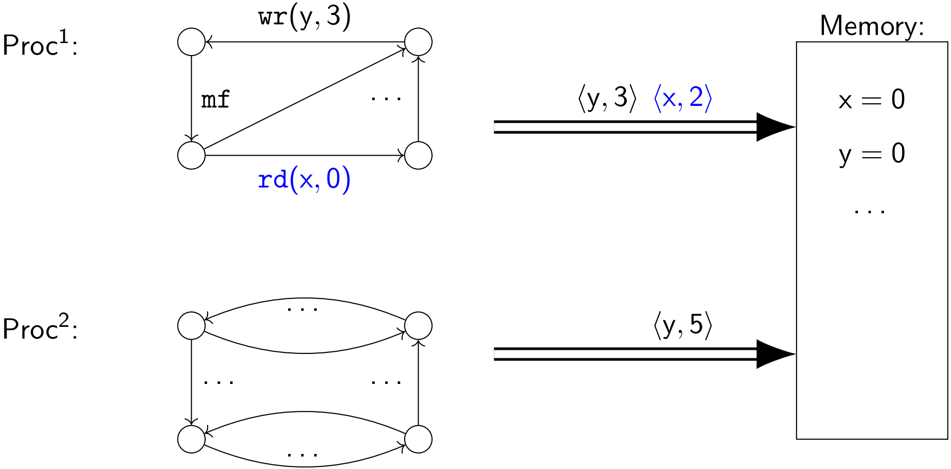
. . .

$\langle y, 5\rangle$

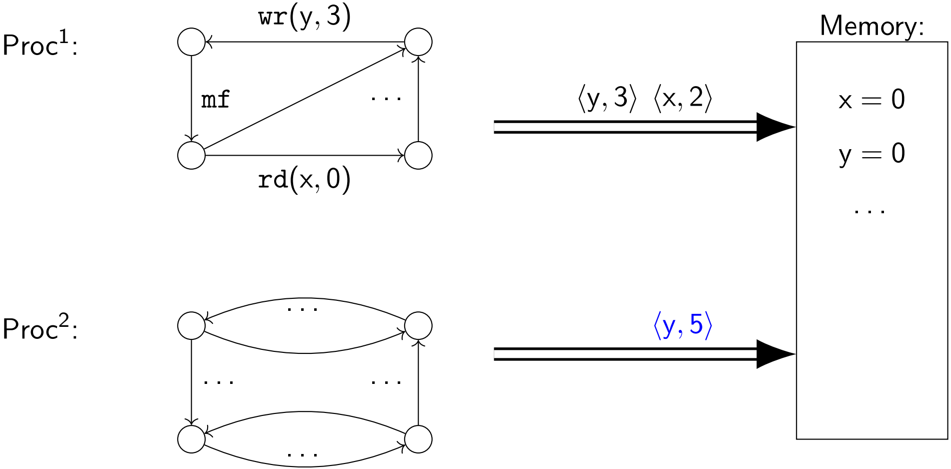
Total Store Order



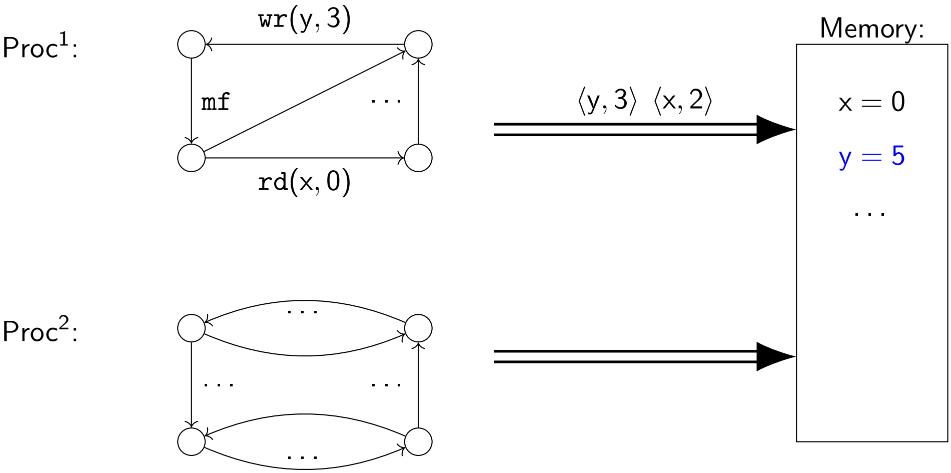
Total Store Order



Total Store Order



Total Store Order



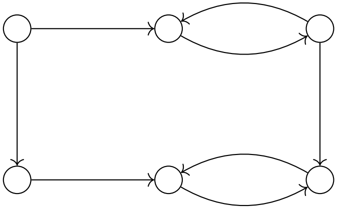
Games

Games

- players A and B

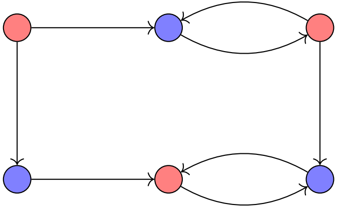
Games

- players A and B
- configurations C
- transition relation $\longrightarrow$



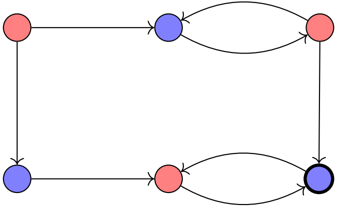
Games

- players A and B
- configurations $C = C_A \cup C_B$
- transition relation $\longrightarrow$
  - $\longrightarrow \subseteq (C_A \times C_B) \cup (C_B \times C_A)$



Games

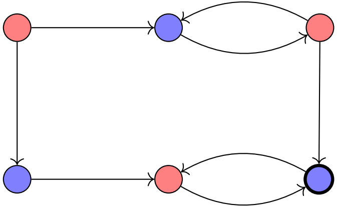
- players A and B
- configurations $C = C_A \cup C_B$
- transition relation $\longrightarrow$
  - $\longrightarrow \subseteq (C_A \times C_B) \cup (C_B \times C_A)$
- final configuration $c_F \in C$



Games

- players A and B
- configurations $C = C_A \cup C_B$
- transition relation $\longrightarrow$
  - $\longrightarrow \subseteq (C_A \times C_B) \cup (C_B \times C_A)$
- final configuration $c_F \in C$

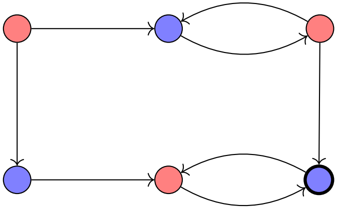
- reachability game:
  - A tries to reach C_F
  - B tries to avoid C_F



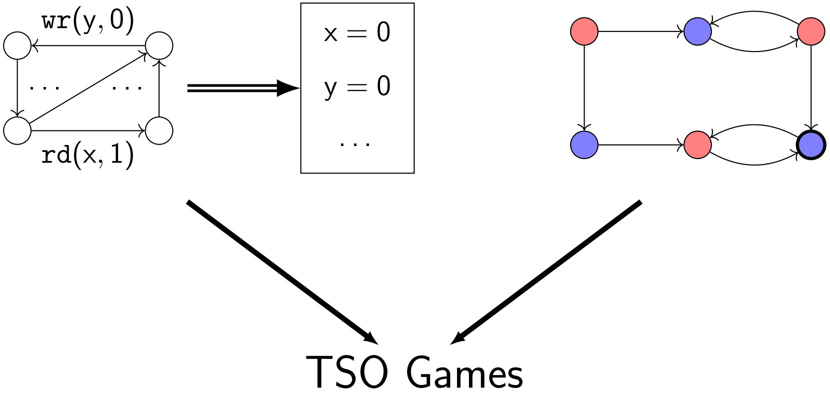
Games

- players A and B
- configurations $C = C_A \cup C_B$
- transition relation $\longrightarrow$
  - $\longrightarrow \subseteq (C_A \times C_B) \cup (C_B \times C_A)$
- final configuration $c_F \in C$

- reachability game:
  - A tries to reach C_F
  - B tries to avoid C_F
- safety game: reversed roles



TSO Games



$x = 0$

$y = 0$

$\ldots$

TSO Games

TSO Games

TSO Games

TSO (configuration graph): $c_1 \xrightarrow{\text{instr}} c_2 \xrightarrow{\mathtt{up}^*} c_3$

TSO Games

TSO (configuration graph): $c_1 \xrightarrow{\text{instr}} c_2 \xrightarrow{\text{up}^*} c_3$

TSO Game: $c_1 \xrightarrow{\hspace{3em}} c_2 \xrightarrow{\hspace{3em}} c_3$

TSO Games

TSO (configuration graph):

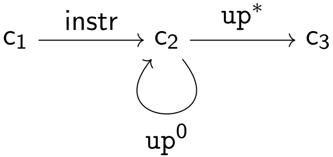
$$\begin{tikzcd}
c_1 \arrow[r, "\text{instr}"] & c_2 \arrow[r, "\mathrm{up}^*"] \arrow[loop below, "\mathrm{up}^0"] & c_3
\end{tikzcd}$$

TSO Game:

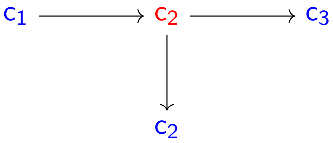
$$\begin{tikzcd}
c_1 \arrow[r] & c_2 \arrow[r] & c_3
\end{tikzcd}$$

TSO Games

TSO (configuration graph):



TSO Game:



TSO Games

TSO (configuration graph):

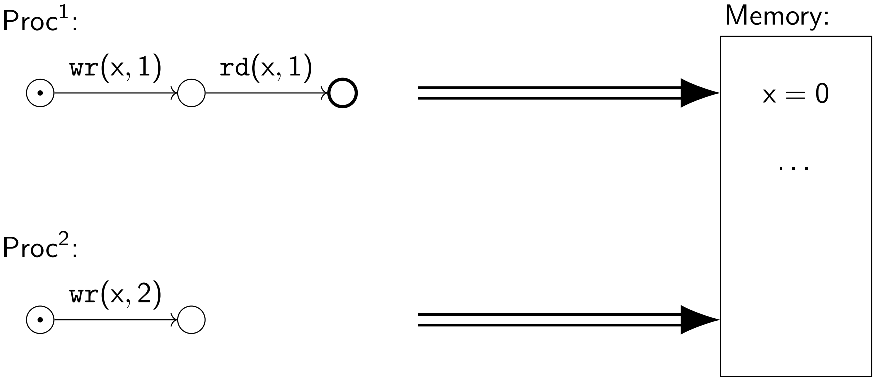
$$\begin{tikzcd} c_1 \arrow[r, "\mathsf{instr}"] & c_2 \arrow[r, "\mathtt{up}^*"] \arrow[loop below, "\mathtt{up}^0"] & c_3 \end{tikzcd}$$

TSO Game:

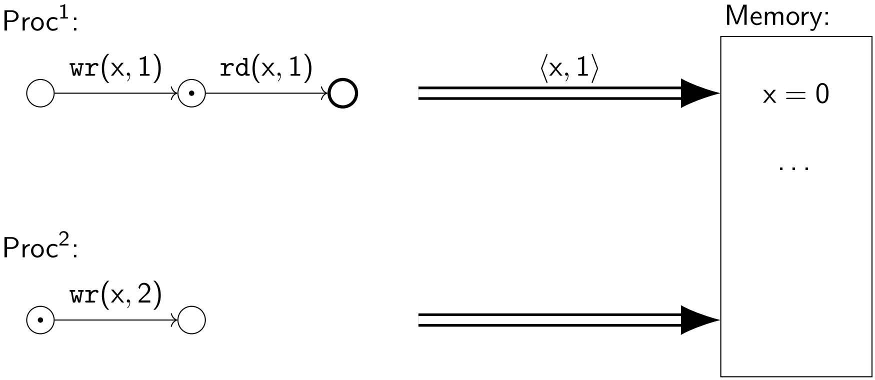
$$\begin{tikzcd} c_1 \arrow[r] & c_2 \arrow[r] \arrow[d] & c_3 \\ & c_2 & \end{tikzcd}$$

process player / *update player*

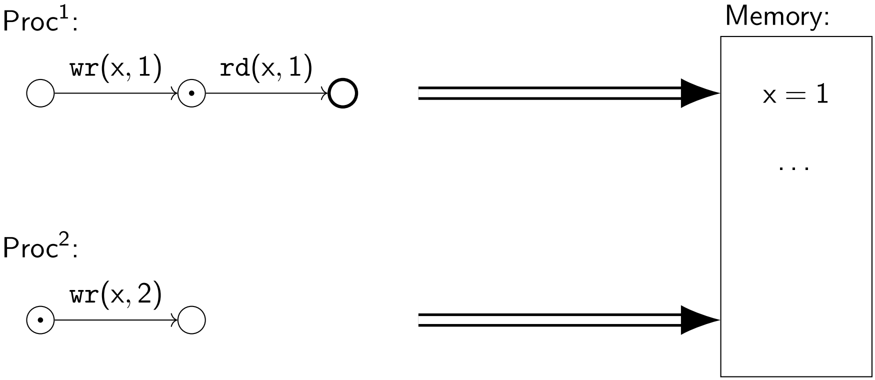
TSO Games - Reachability Problem



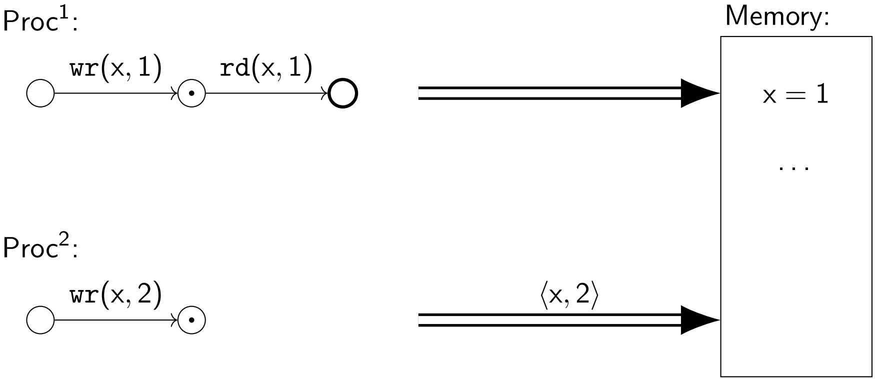
TSO Games - Reachability Problem



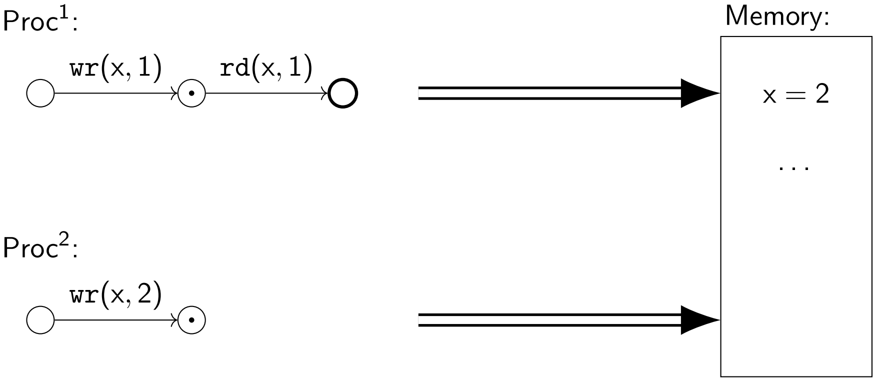
TSO Games - Reachability Problem



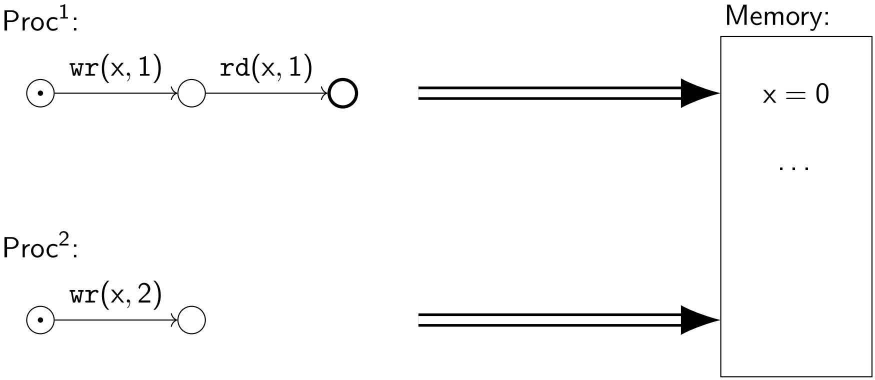
TSO Games - Reachability Problem



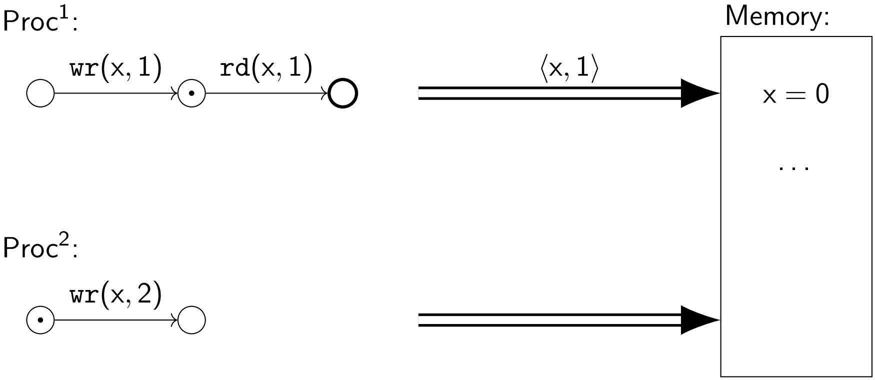
TSO Games - Reachability Problem



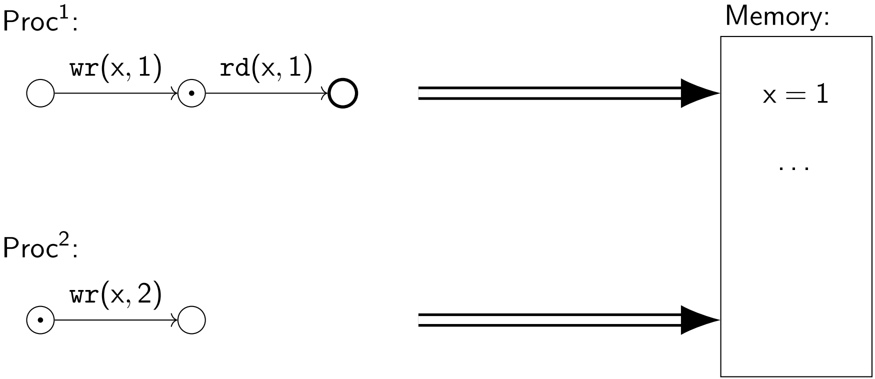
TSO Games - Reachability Problem



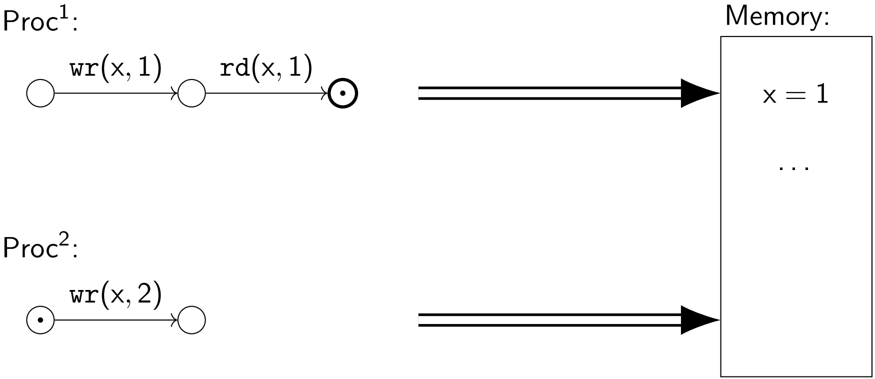
TSO Games - Reachability Problem



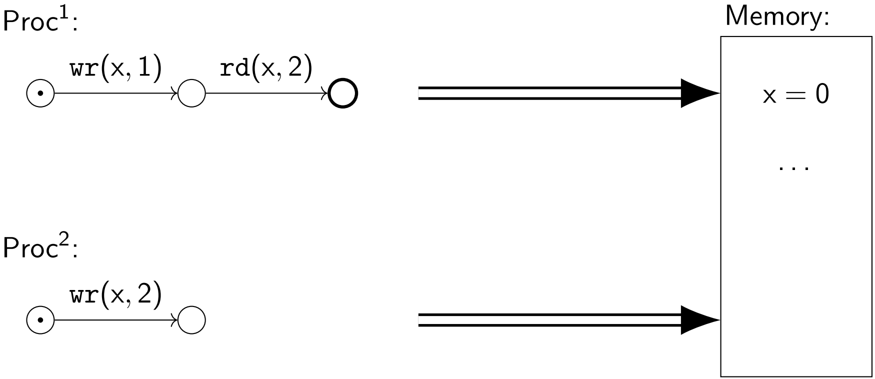
TSO Games - Reachability Problem



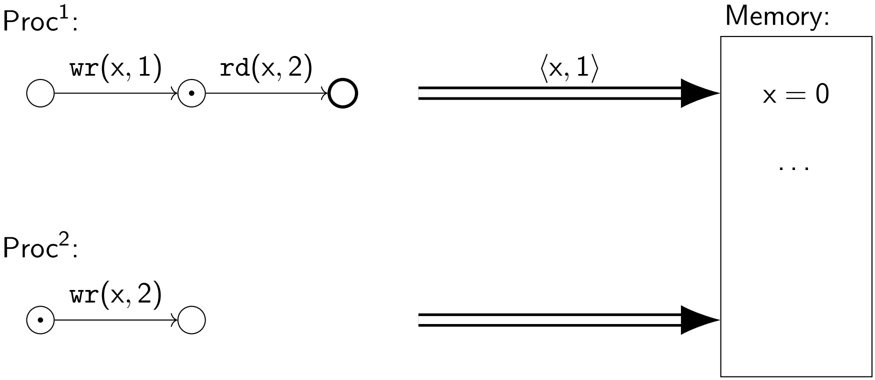
TSO Games - Reachability Problem



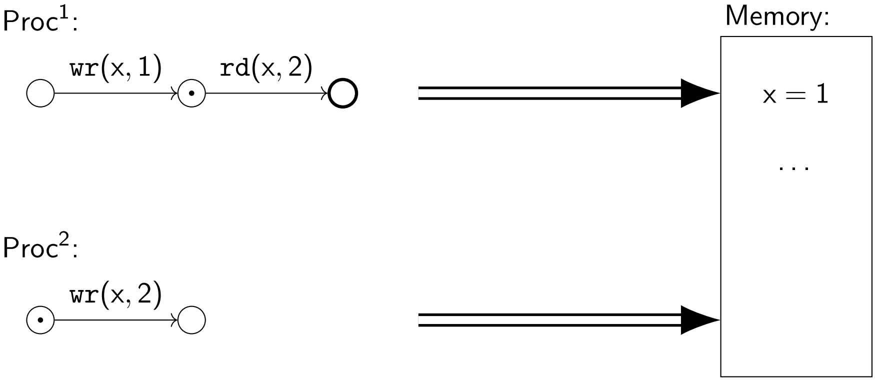
TSO Games - Reachability Problem



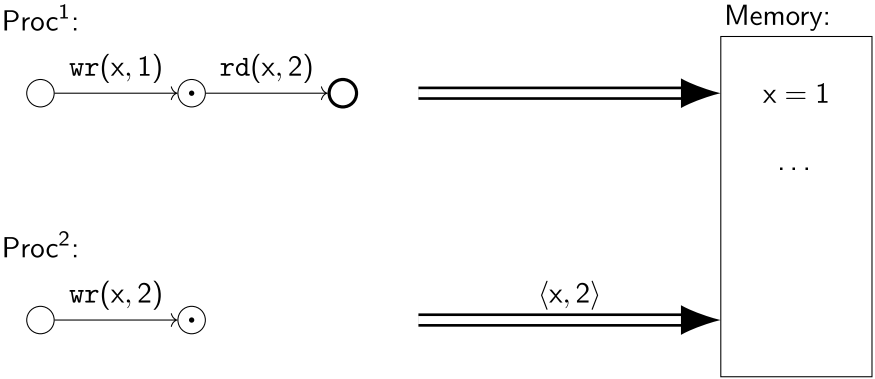
TSO Games - Reachability Problem



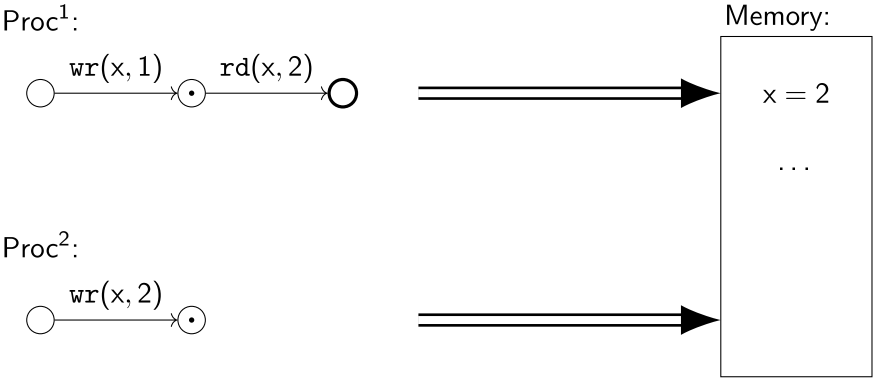
TSO Games - Reachability Problem



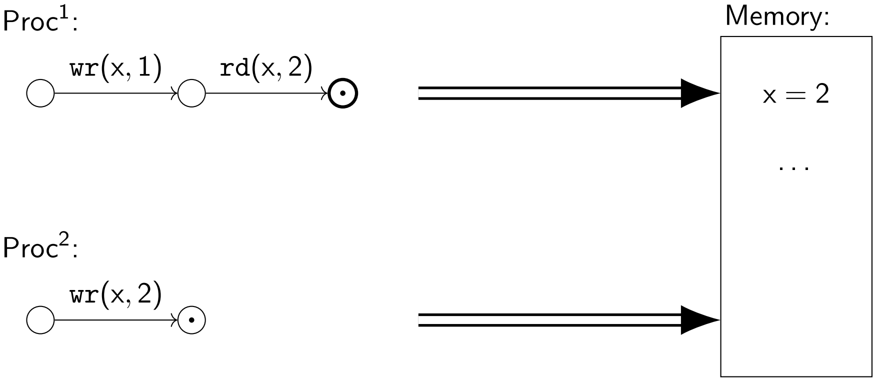
TSO Games - Reachability Problem



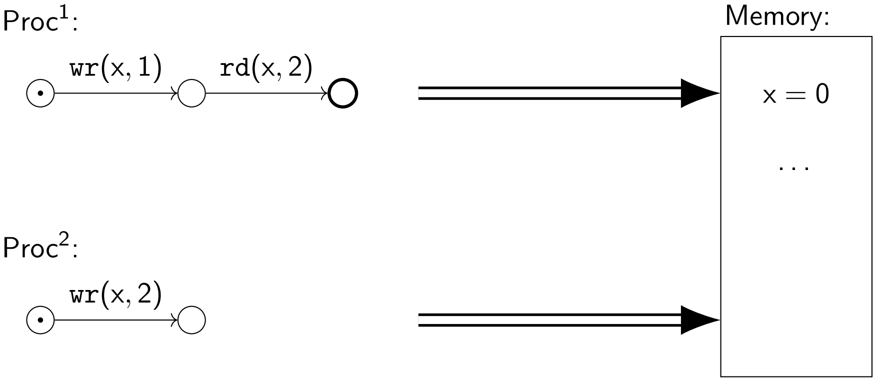
TSO Games - Reachability Problem



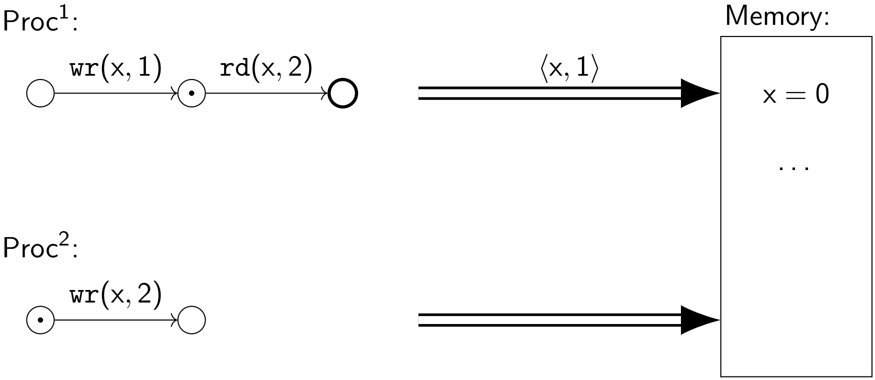
TSO Games - Reachability Problem



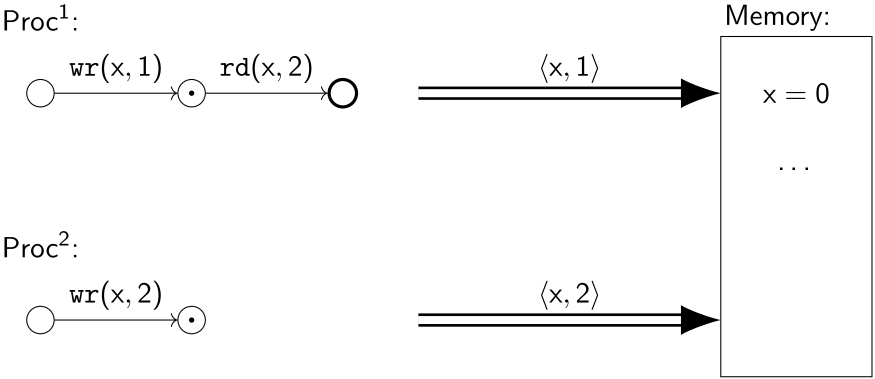
TSO Games - Reachability Problem



TSO Games - Reachability Problem



TSO Games - Reachability Problem



TSO Games - Reachability Problem

TSO Games - Reachability Problem

- Proc$^\iota$ can reach final state **without** help from other processes: winning strategy for process player: only play in Proc$^\iota$

TSO Games - Reachability Problem

- Proc^{ι} can reach final state **without** help from other processes: winning strategy for process player: only play in Proc^{ι}

- Proc^{ι} can reach final state **only with** help from other processes: winning strategy for update player: do not update any message

TSO Games - Reachability Problem

- Proc^{ι} can reach final state **without** help from other processes: winning strategy for process player: only play in Proc^{ι}
- Proc^{ι} can reach final state **only with** help from other processes: winning strategy for update player: do not update any message
- similar for *safety* games

TSO Games - Reachability Problem

- Proc^ι can reach final state **without** help from other processes: winning strategy for process player: only play in Proc^ι
- Proc^ι can reach final state **only with** help from other processes: winning strategy for update player: do not update any message
- similar for *safety* games
- analysis reduces to single-process programs (finite behaviour)

TSO Games - Reachability Problem

- Proc^{ι} can reach final state **without** help from other processes: winning strategy for process player: only play in Proc^{ι}

- Proc^{ι} can reach final state **only with** help from other processes: winning strategy for update player: do not update any message

- similar for *safety* games

- analysis reduces to single-process programs (finite behaviour)

- complexity: PSPACE-complete

TSO Games - Adding Fairness

TSO Games - Adding Fairness

- Proc^{ι} can reach final state **without** help from other processes: winning strategy for process player: only play in Proc^{ι}

TSO Games - Adding Fairness

- Proc^{ι} can reach final state **without** help from other processes: winning strategy for process player: only play in Proc^{ι}

  Process Fairness:
  Every enabled process must be executed infinitely often.

TSO Games - Adding Fairness

- Proc^ι can reach final state **without** help from other processes: winning strategy for process player: only play in Proc^ι

  Process Fairness:
  Every enabled process must be executed infinitely often.

- Proc^ι can reach final state **only with** help from other processes: winning strategy for update player: do not update any message

TSO Games - Adding Fairness

- Proc^ι can reach final state **without** help from other processes: winning strategy for process player: only play in Proc^ι

  Process Fairness:
  Every enabled process must be executed infinitely often.

- Proc^ι can reach final state **only with** help from other processes: winning strategy for update player: do not update any message

  Update Fairness:
  Eventually, every buffer message must be updated to the memory.

Update Fairness

Eventually, every buffer message must be updated to the memory.

Update Fairness

Eventually, every buffer message must be updated to the memory.

safety games?

Update Fairness

Eventually, every buffer message must be updated to the memory.

~~safety games?~~ ⟶ reachability games!

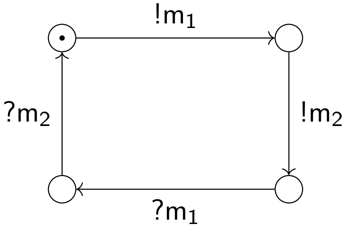
Update Fairness

Eventually, every buffer message must be updated to the memory.

~~safety games?~~ ⟶ reachability games!

Idea: Reduction from *Perfect Channel Systems*

- nondeterministic finite state automata augmented by FIFO *channel*



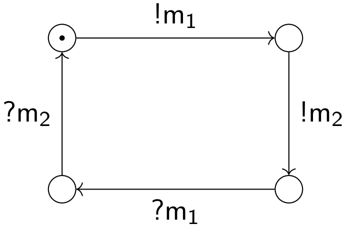
Update Fairness

Eventually, every buffer message must be updated to the memory.

~~safety games?~~ ⟶ reachability games!

Idea: Reduction from *Perfect Channel Systems*

- nondeterministic finite state automata augmented by FIFO *channel*
- use TSO buffer to simulate channel



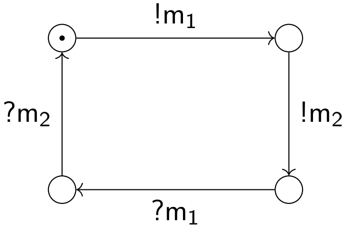
Update Fairness

Eventually, every buffer message must be updated to the memory.

~~safety games?~~ ⟶ reachability games!

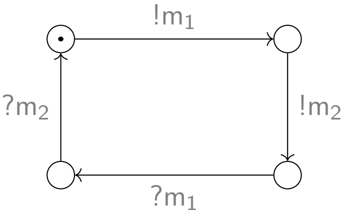
Idea: Reduction from *Perfect Channel Systems*

- nondeterministic finite state automata augmented by FIFO *channel*
- use TSO buffer to simulate channel
- reduce PCS reachability (undecidable) to TSO reachability game

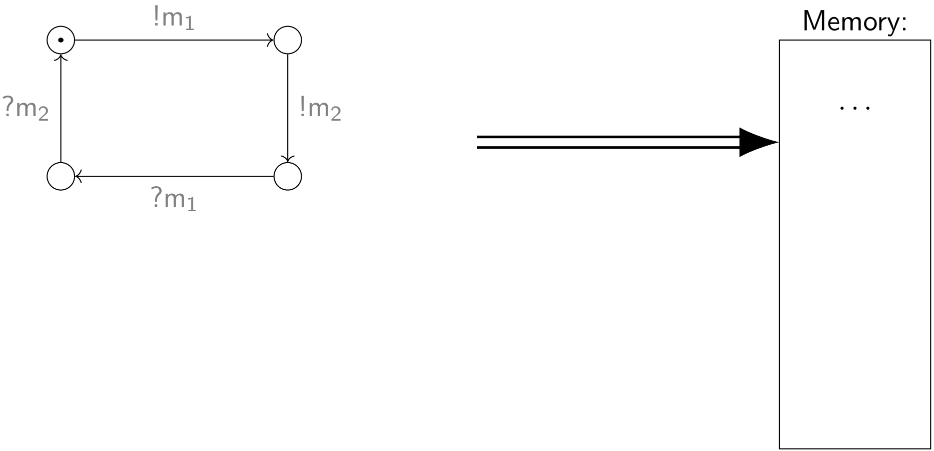


Update Fairness - PCS Reduction

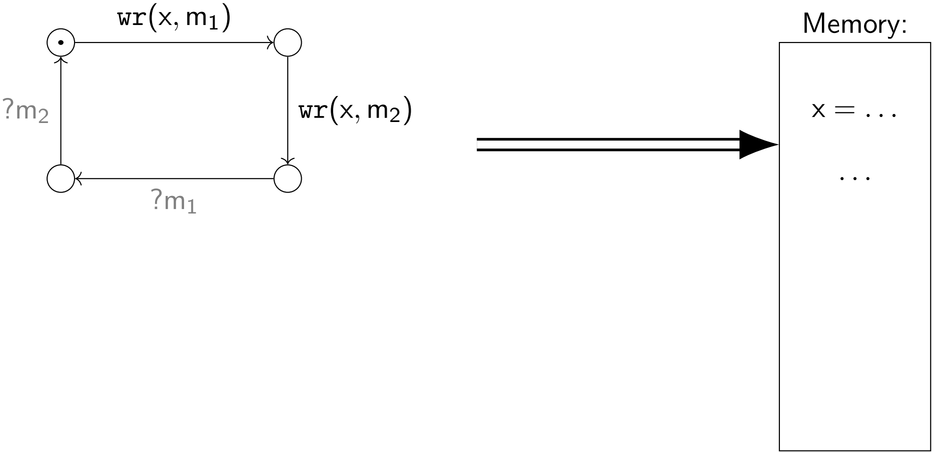
Update Fairness - PCS Reduction



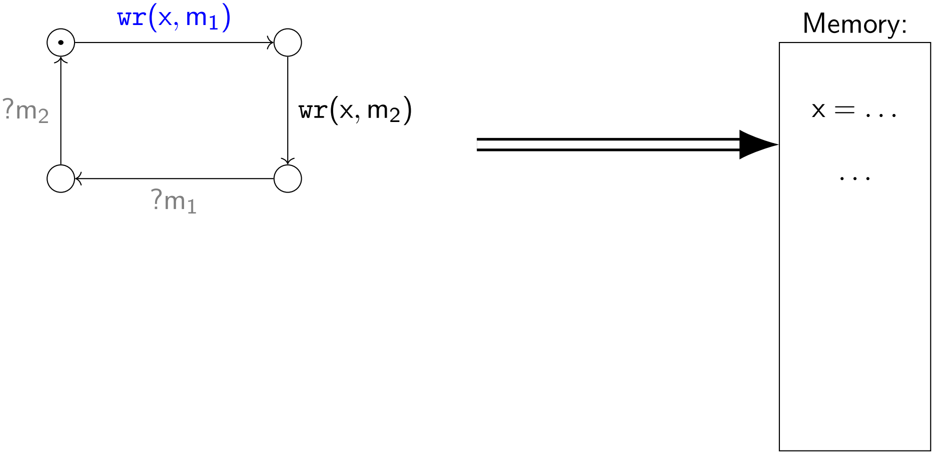
Update Fairness - PCS Reduction



Update Fairness - PCS Reduction



Update Fairness - PCS Reduction

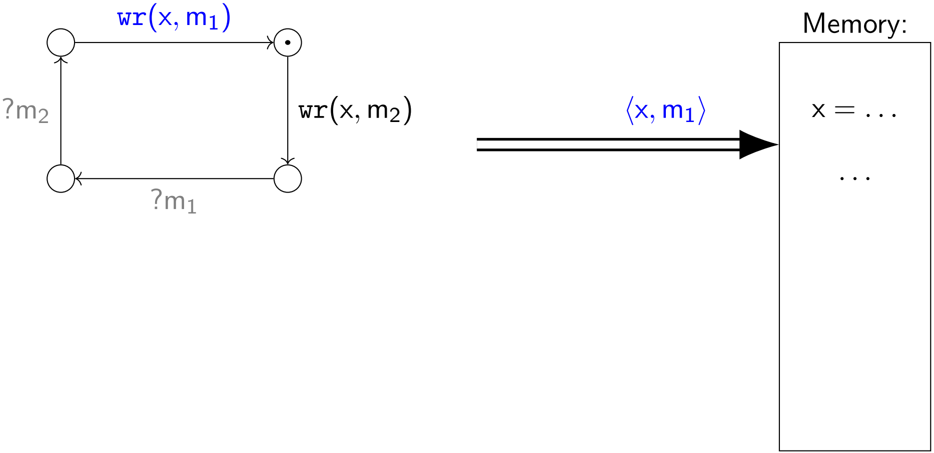


Memory:

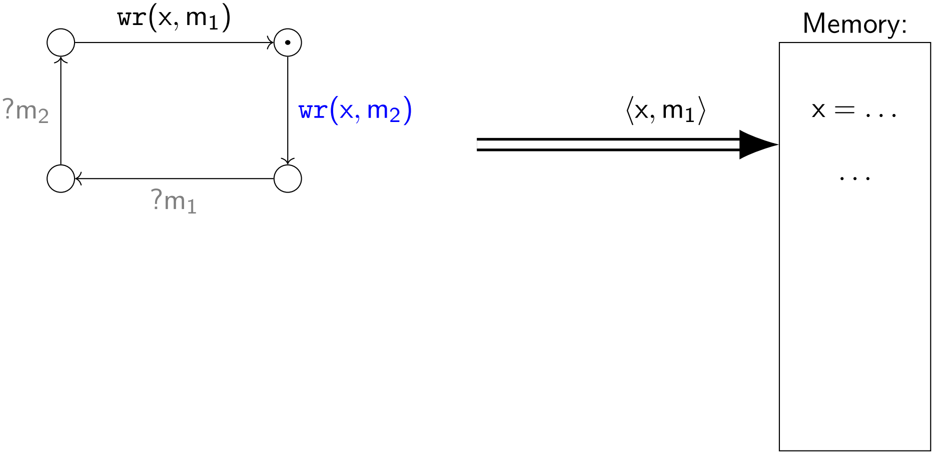
$x = \ldots$

$\ldots$

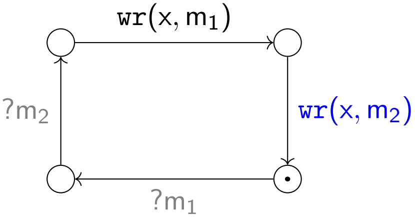
Update Fairness - PCS Reduction

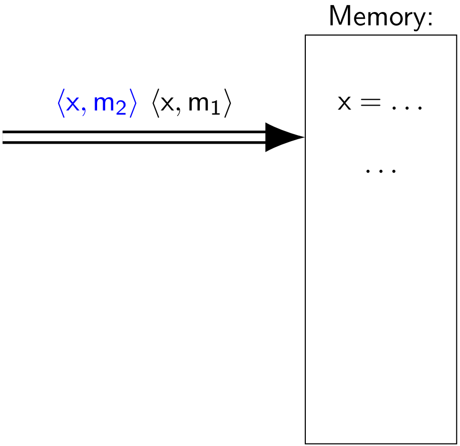


Update Fairness - PCS Reduction



Update Fairness - PCS Reduction





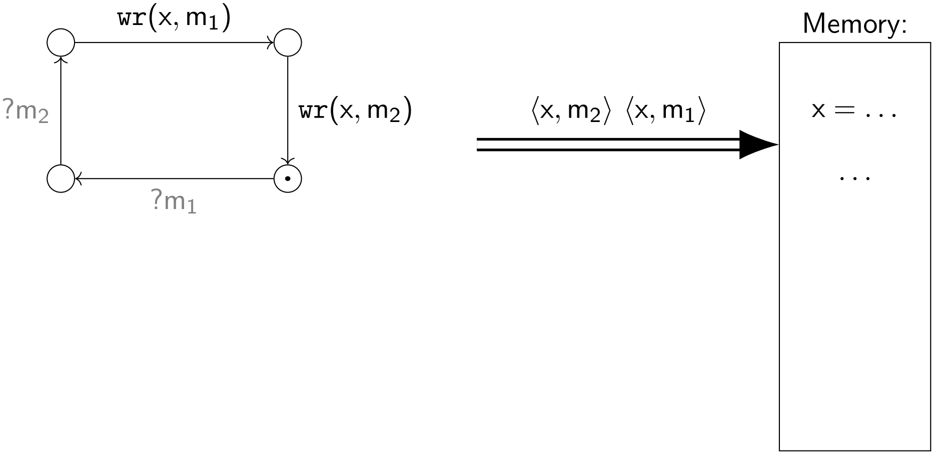
$\langle x, m_2 \rangle$ $\langle x, m_1 \rangle$

Memory:

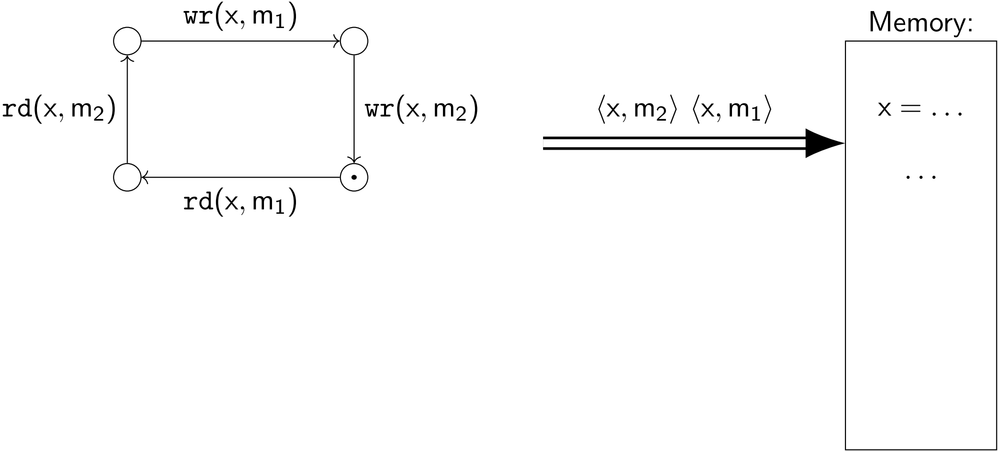
$x = \ldots$

$\ldots$

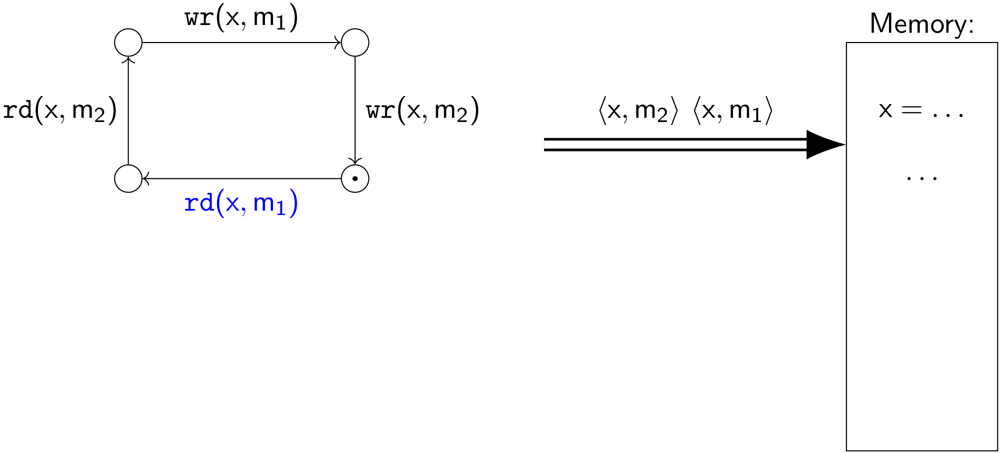
Update Fairness - PCS Reduction



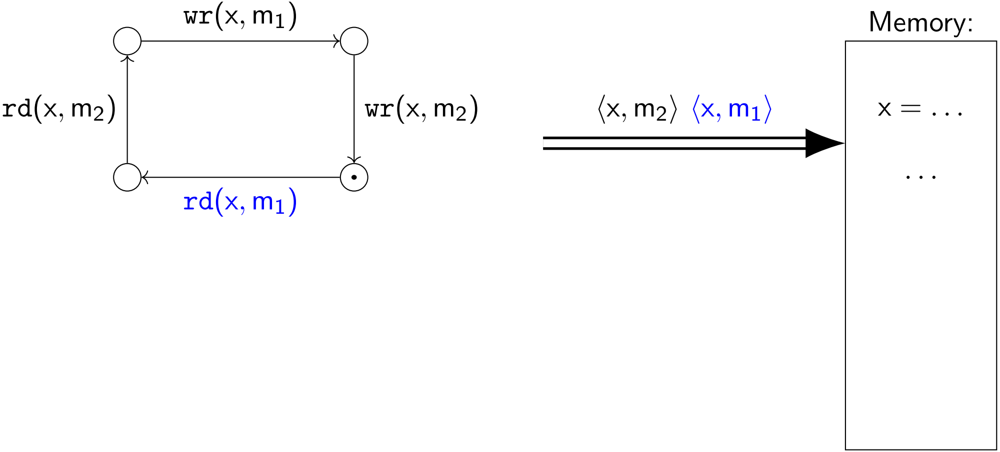
Update Fairness - PCS Reduction



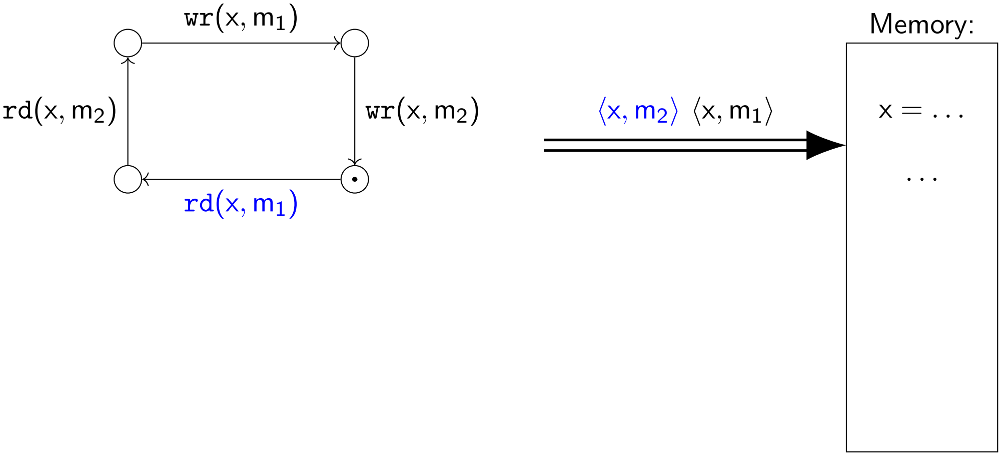
Update Fairness - PCS Reduction



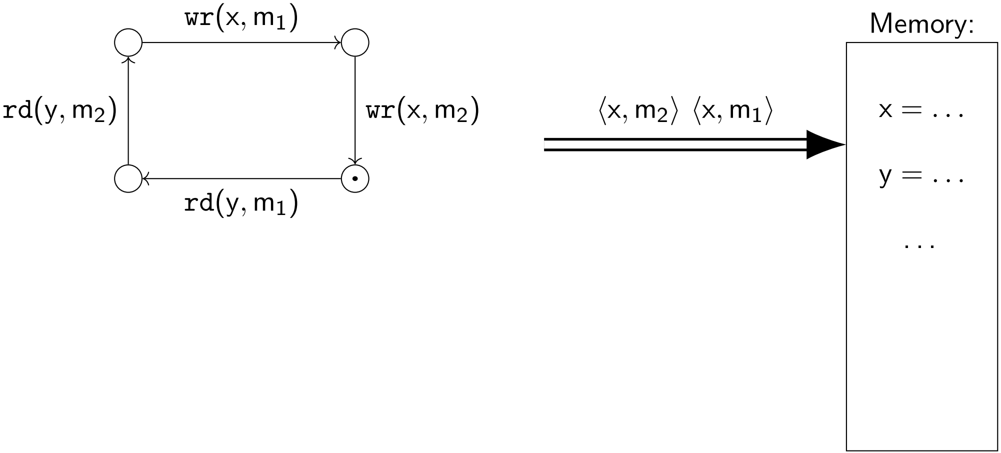
Update Fairness - PCS Reduction



Update Fairness - PCS Reduction



Update Fairness - PCS Reduction



$\mathtt{wr}(x, m_1)$

$\mathtt{rd}(y, m_2)$

$\mathtt{wr}(x, m_2)$

$\mathtt{rd}(y, m_1)$

$\langle x, m_2 \rangle \ \langle x, m_1 \rangle$

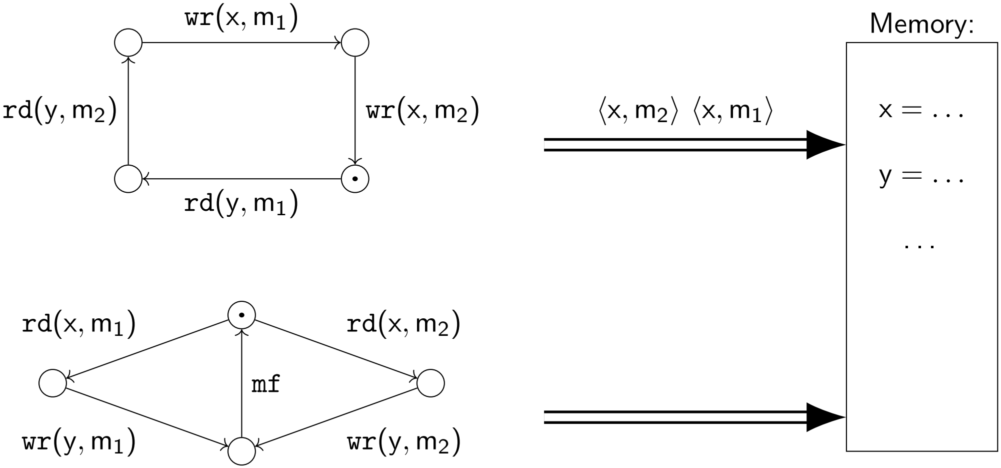
Memory:

$x = \ldots$

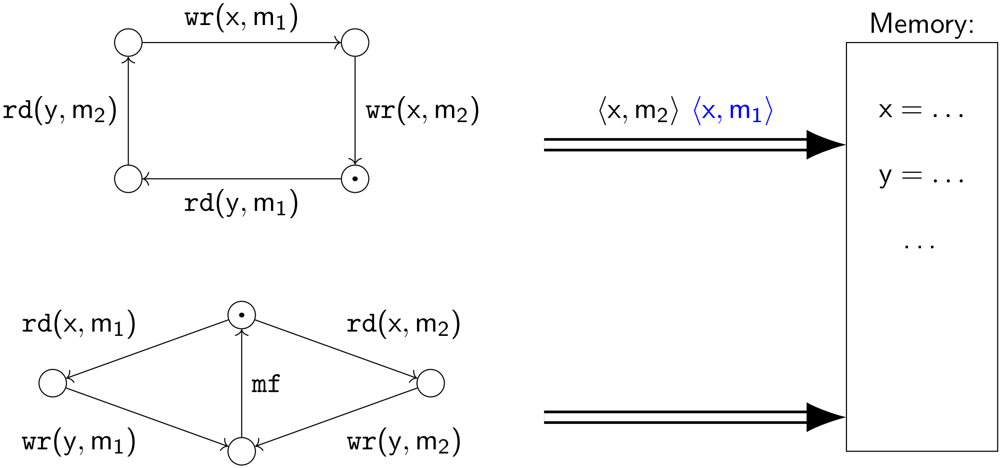
$y = \ldots$

$\ldots$

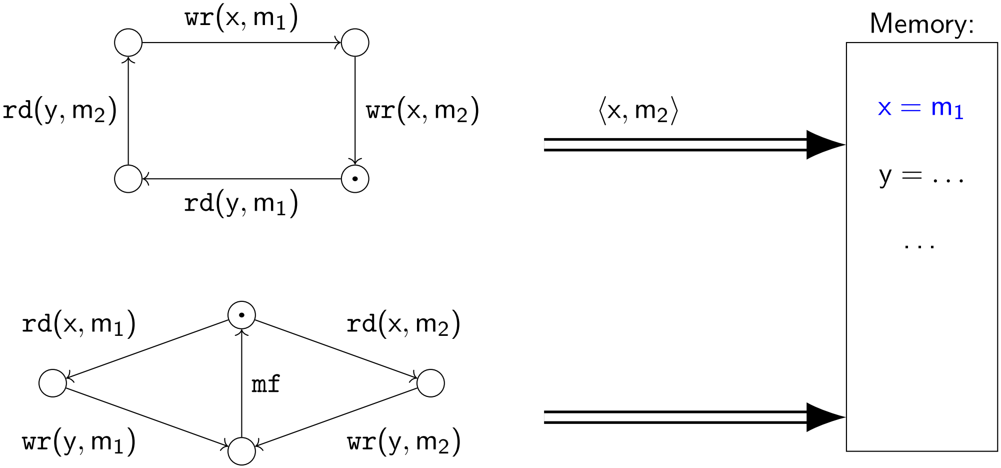
Update Fairness - PCS Reduction



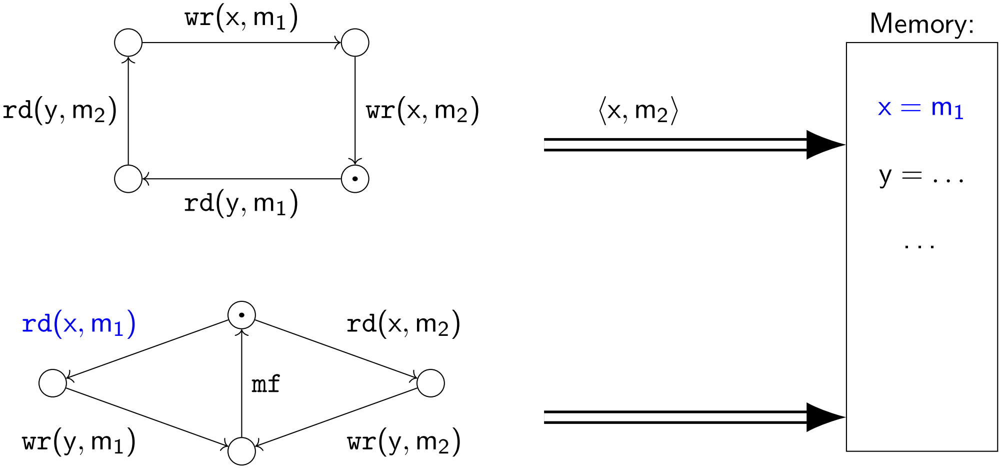
Update Fairness - PCS Reduction



Update Fairness - PCS Reduction



Update Fairness - PCS Reduction



$\mathtt{wr}(x, m_1)$

$\mathtt{rd}(y, m_2)$

$\mathtt{wr}(x, m_2)$

$\mathtt{rd}(y, m_1)$

$\langle x, m_2 \rangle$

Memory:

$x = m_1$

$y = \ldots$

$\ldots$

$\mathtt{rd}(x, m_1)$

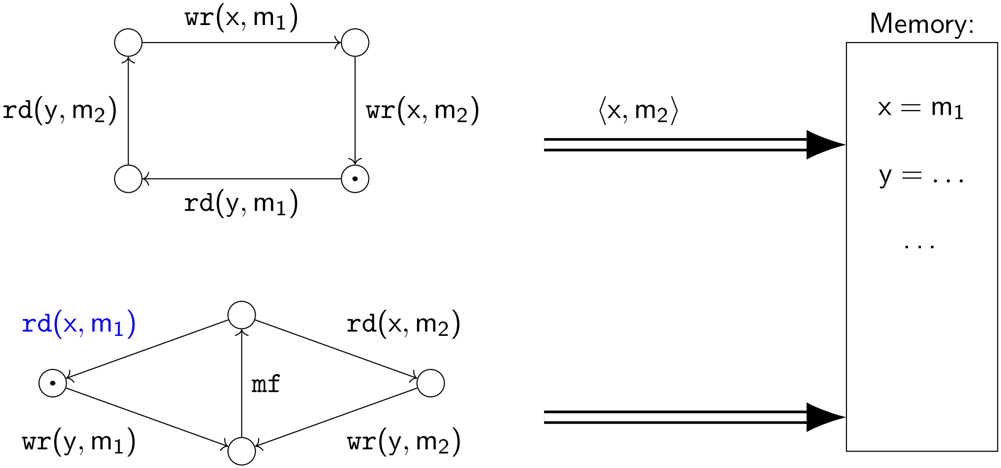
$\mathtt{rd}(x, m_2)$

$\mathtt{mf}$

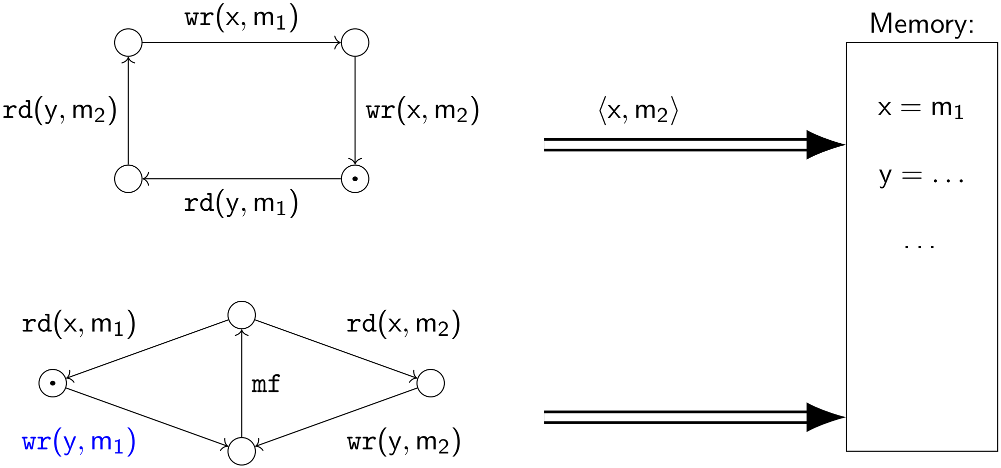
$\mathtt{wr}(y, m_1)$

$\mathtt{wr}(y, m_2)$

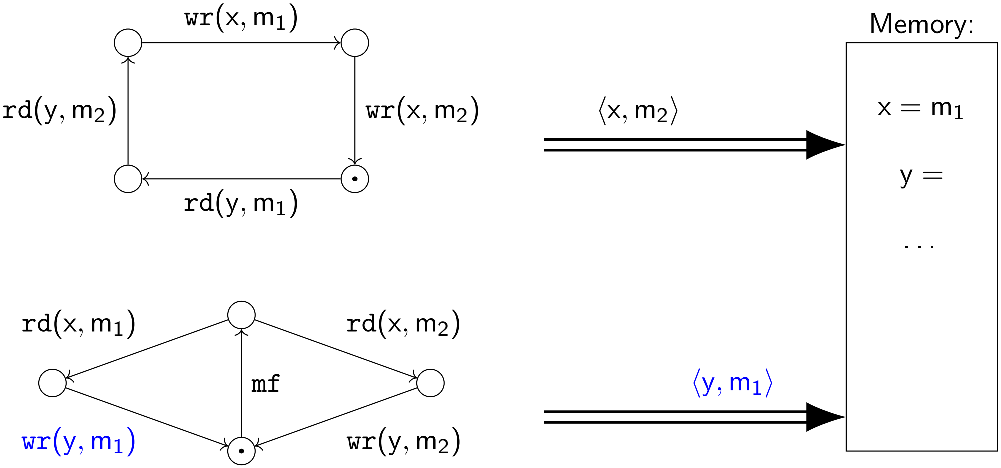
Update Fairness - PCS Reduction



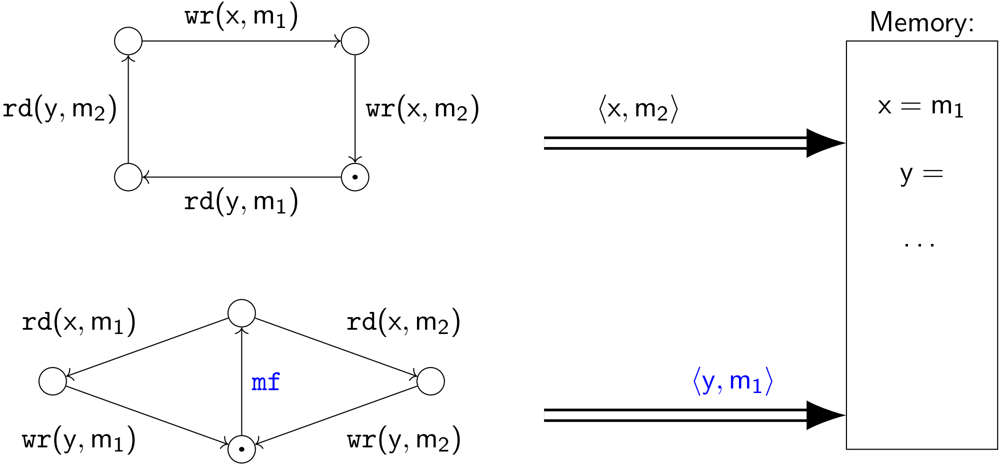
Update Fairness - PCS Reduction



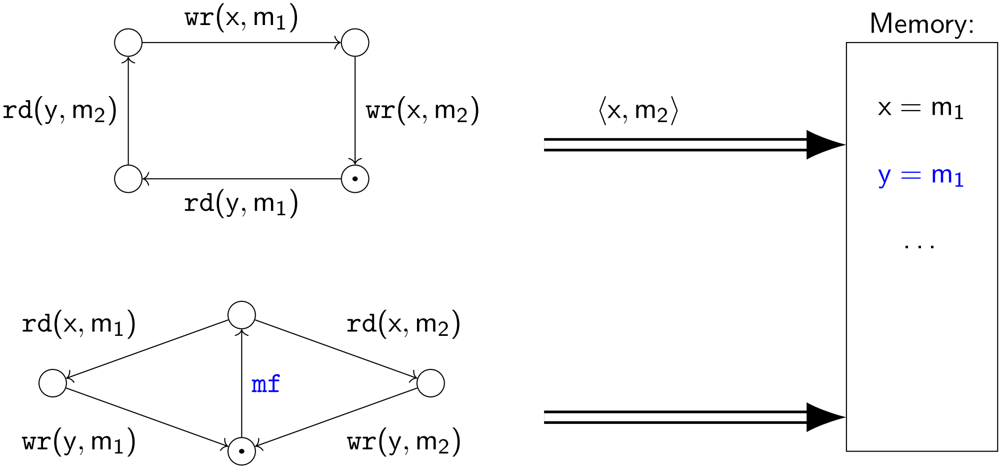
Update Fairness - PCS Reduction



Update Fairness - PCS Reduction



Update Fairness - PCS Reduction



$\mathtt{wr}(x, m_1)$

$\mathtt{rd}(y, m_2)$

$\mathtt{wr}(x, m_2)$

$\mathtt{rd}(y, m_1)$

$\langle x, m_2 \rangle$

Memory:

$x = m_1$

$y = m_1$

...

$\mathtt{rd}(x, m_1)$

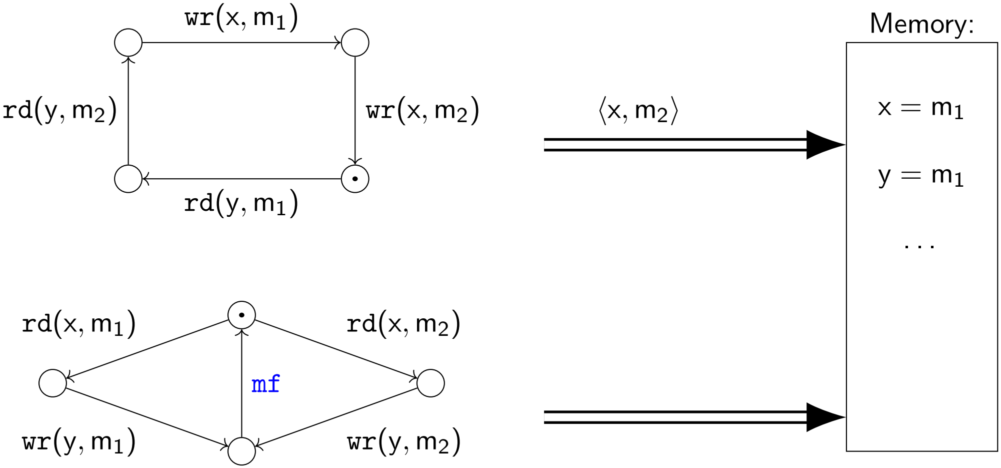
$\mathtt{rd}(x, m_2)$

$\mathtt{mf}$

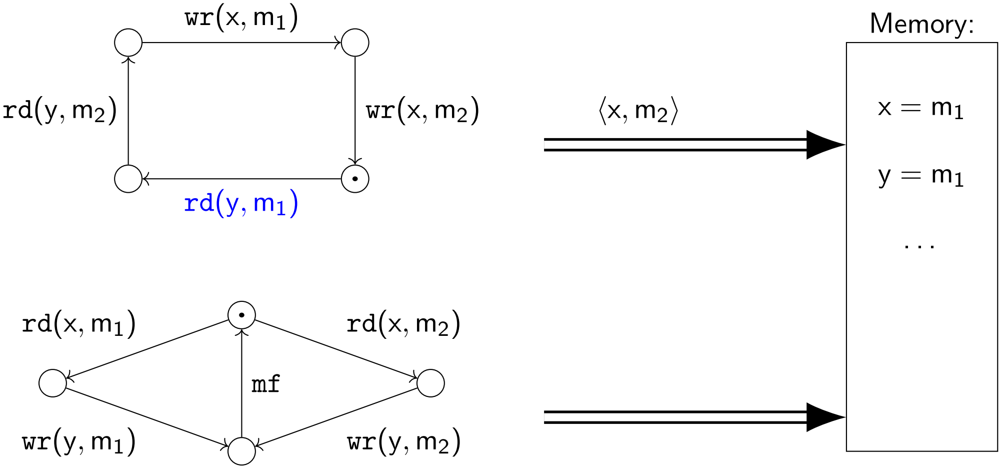
$\mathtt{wr}(y, m_1)$

$\mathtt{wr}(y, m_2)$

Update Fairness - PCS Reduction



Update Fairness - PCS Reduction



$\mathtt{wr}(x, m_1)$

$\mathtt{rd}(y, m_2)$

$\mathtt{wr}(x, m_2)$

$\mathtt{rd}(y, m_1)$

$\langle x, m_2 \rangle$

Memory:

$x = m_1$

$y = m_1$

$\dots$

$\mathtt{rd}(x, m_1)$

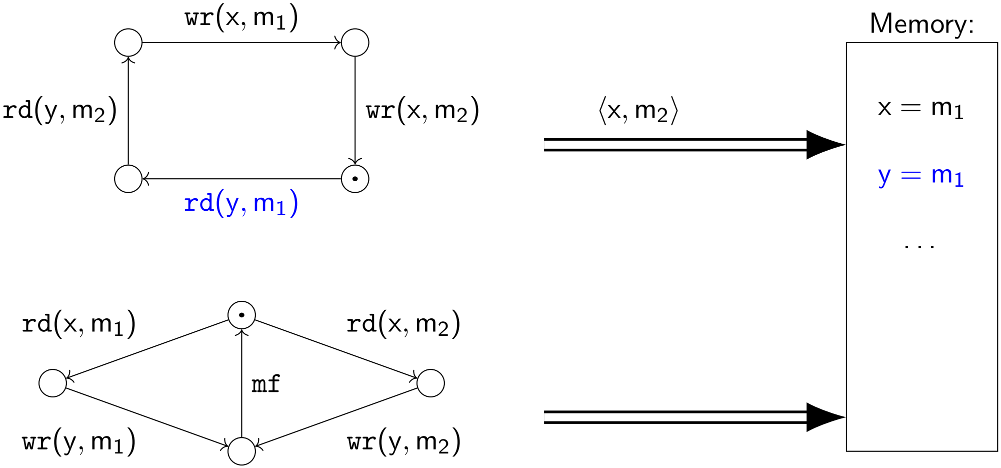
$\mathtt{rd}(x, m_2)$

$\mathtt{mf}$

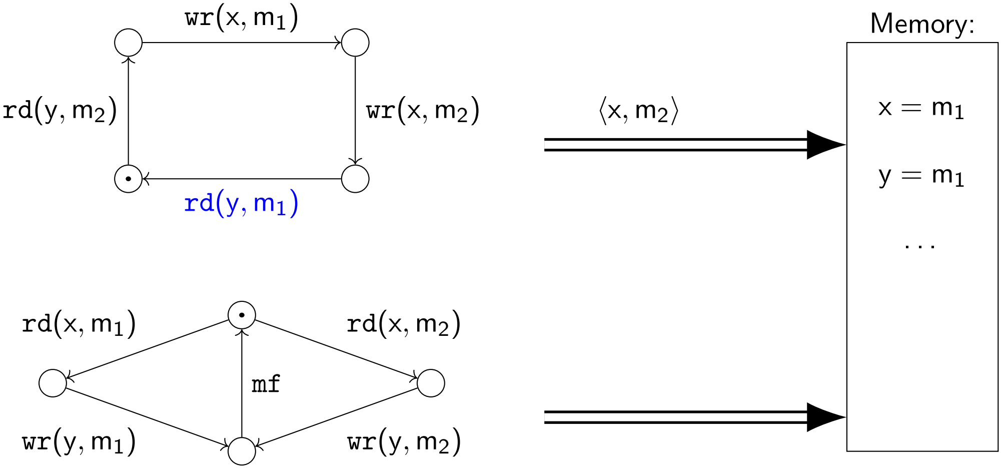
$\mathtt{wr}(y, m_1)$

$\mathtt{wr}(y, m_2)$

Update Fairness - PCS Reduction



Update Fairness - PCS Reduction



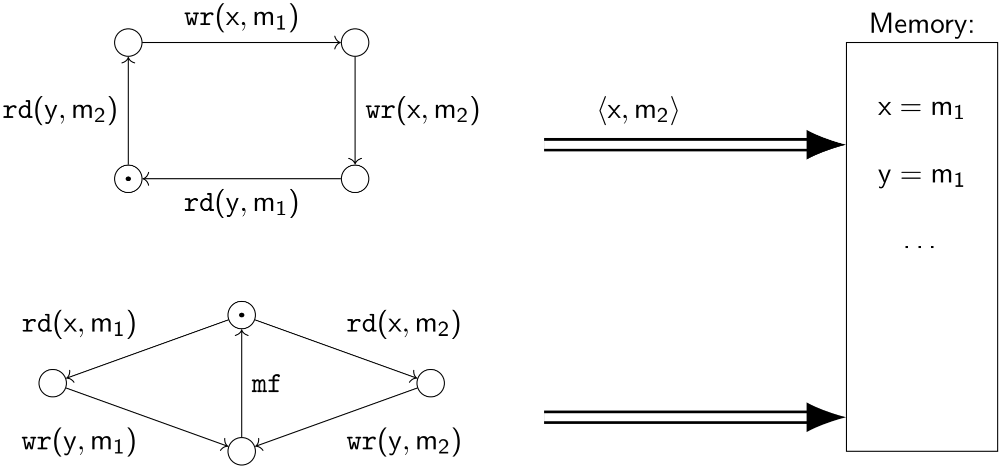
Memory:

$x = m_1$

$y = m_1$

$\dots$

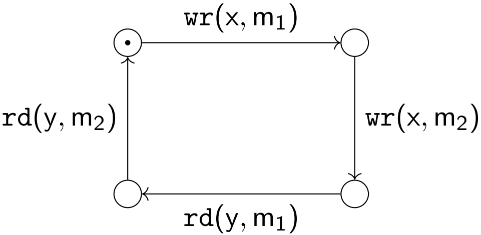
Update Fairness - PCS Reduction



Update Fairness

Eventually, every buffer message must be updated to the memory.

- use TSO buffer to simulate PCS channel
- reduce PCS reachability (undecidable) to TSO reachability game



Theorem

The reachability problem under TSO semantics with update fairness is undecidable.

Process Fairness

Every enabled process must be executed infinitely often.

Process Fairness

Every enabled process must be executed infinitely often.

reachability games?

Process Fairness

Every enabled process must be executed infinitely often.

~~reachability games?~~ $\longrightarrow$ safety games!

Process Fairness

Every enabled process must be executed infinitely often.

~~reachability games?~~ ⟶ safety games!

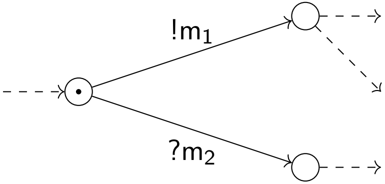
Idea: update player simulates PCS run, process player is passive

Process Fairness

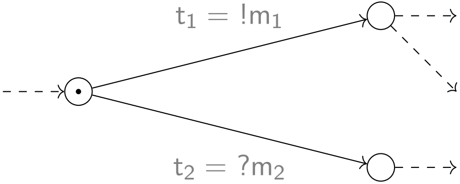
Every enabled process must be executed infinitely often.

~~reachability games?~~ ⟶ safety games!

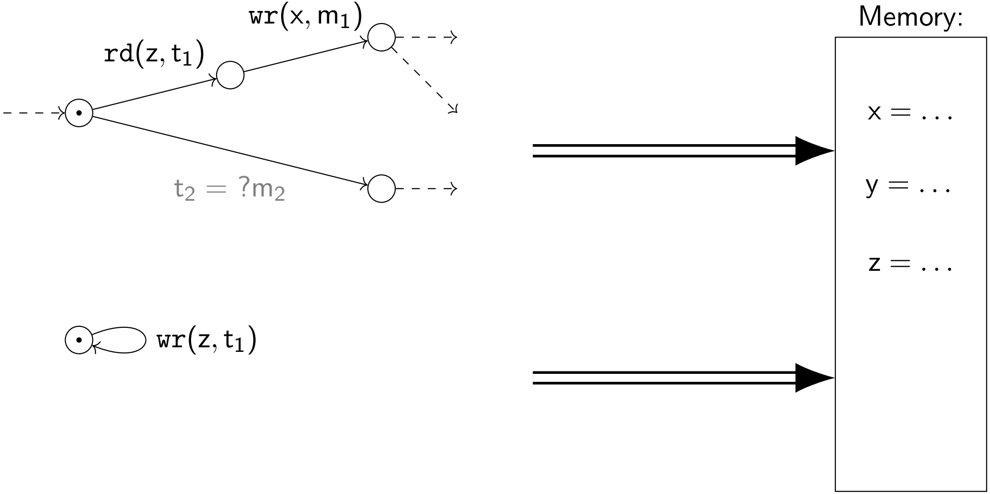
Idea: update player simulates PCS run, process player is passive



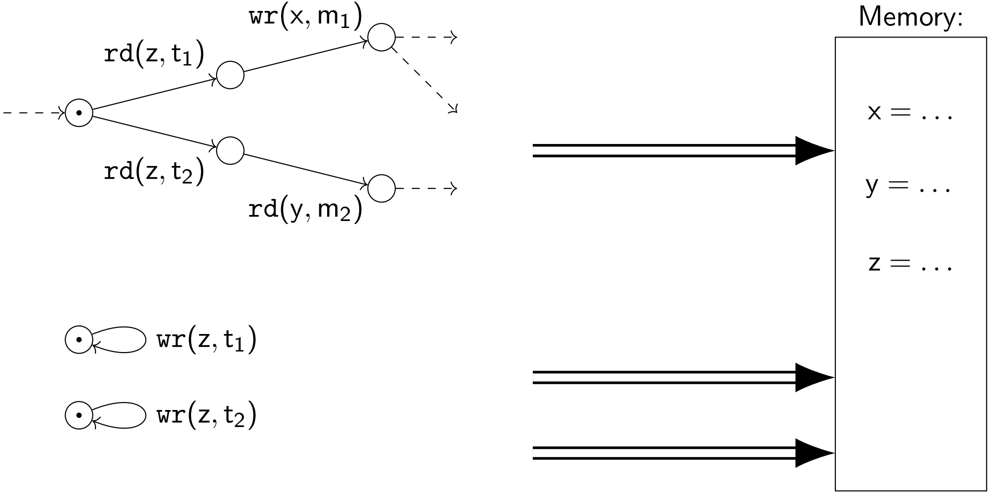
Process Fairness



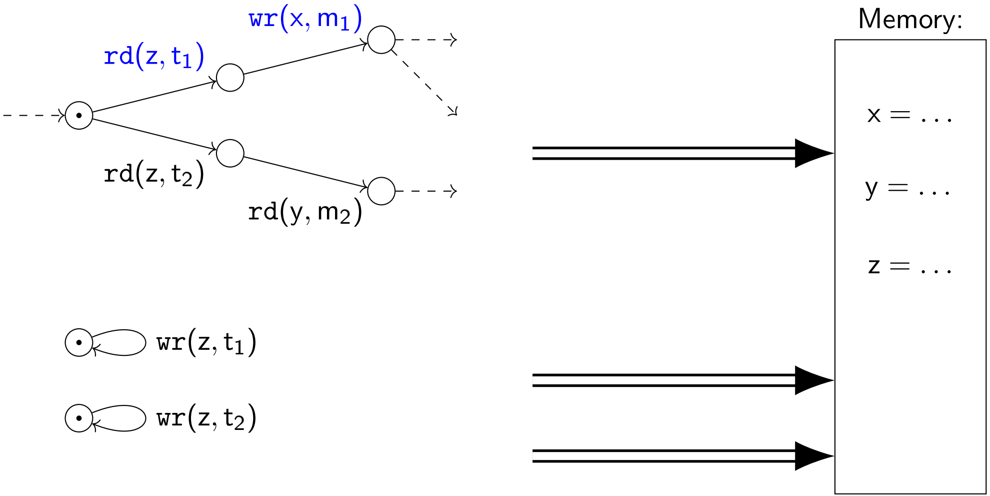
Process Fairness



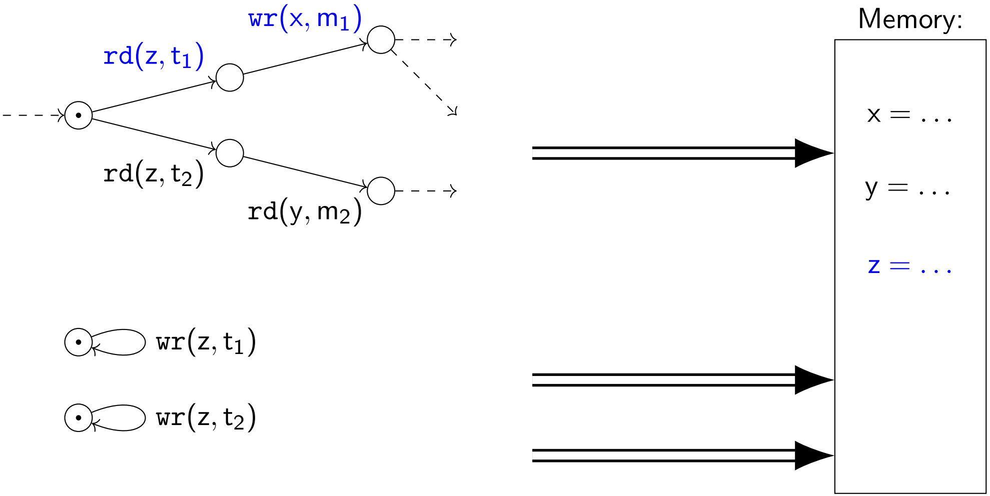
Process Fairness



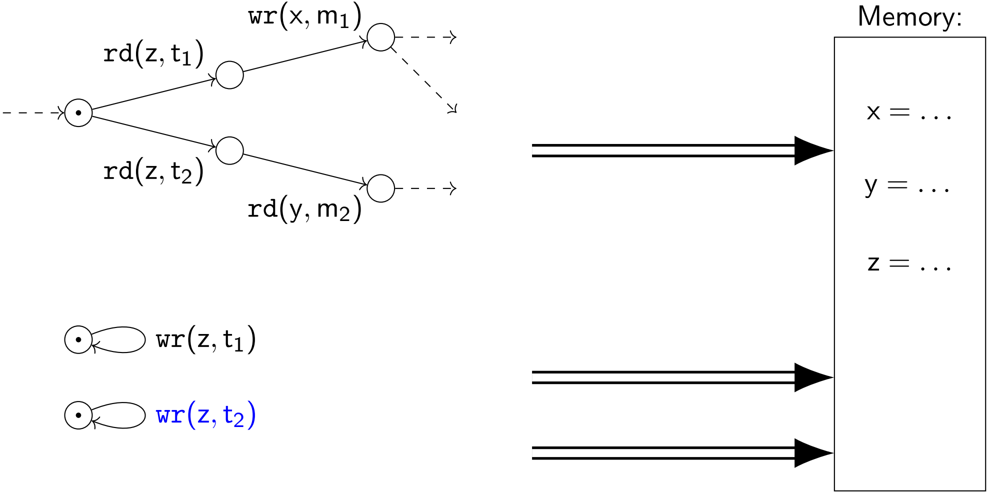
Process Fairness



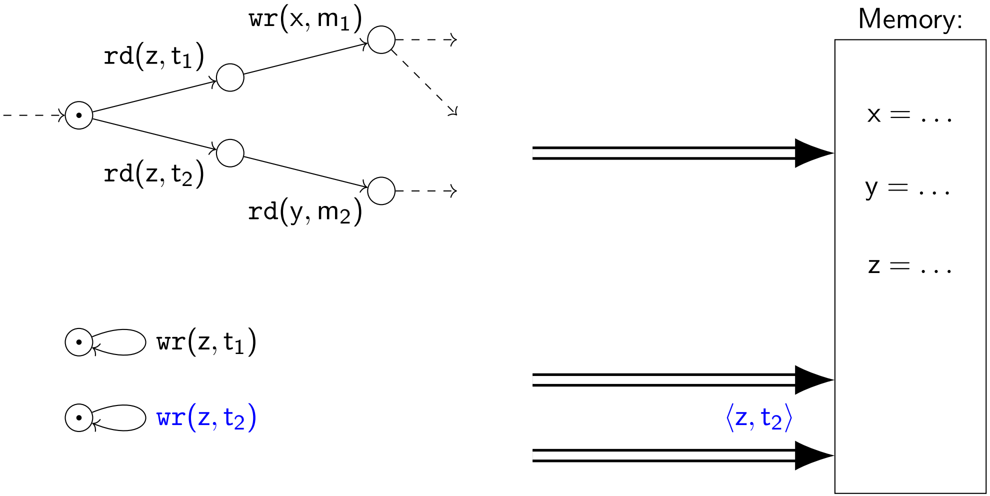
Process Fairness



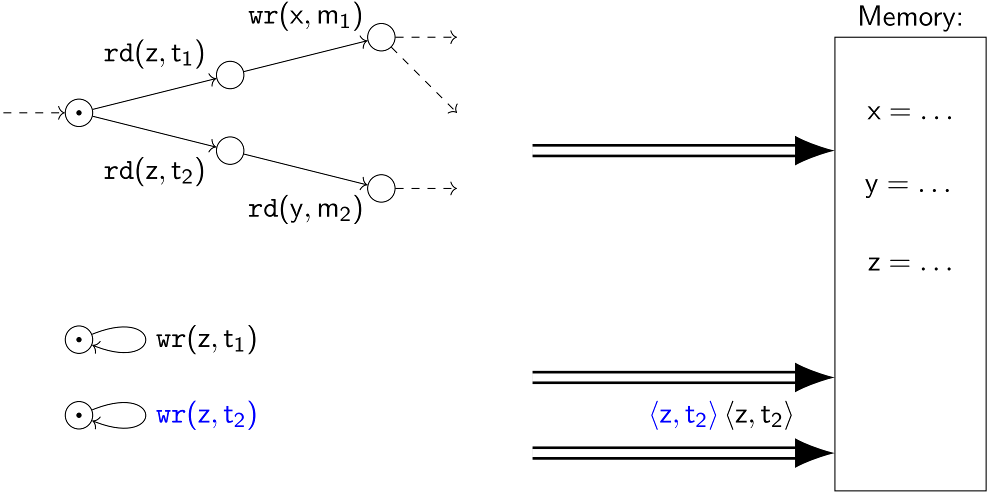
Process Fairness



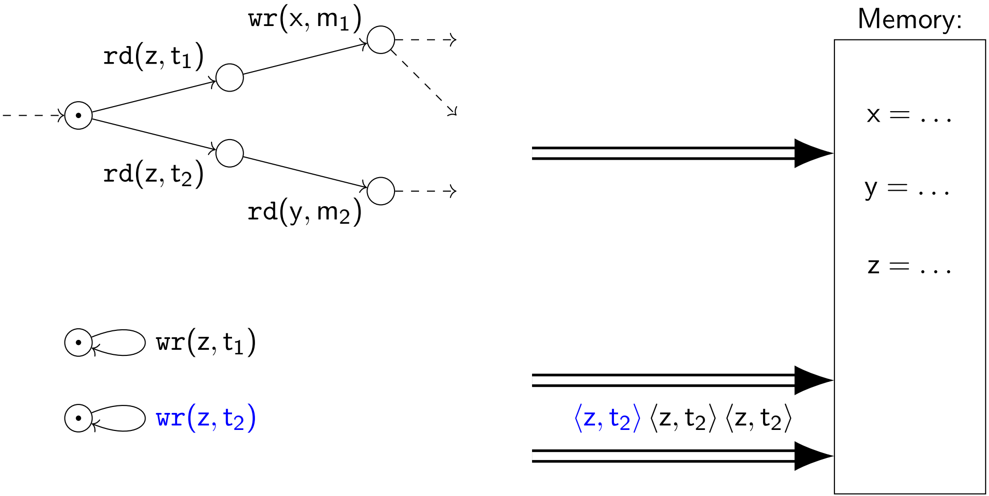
Process Fairness



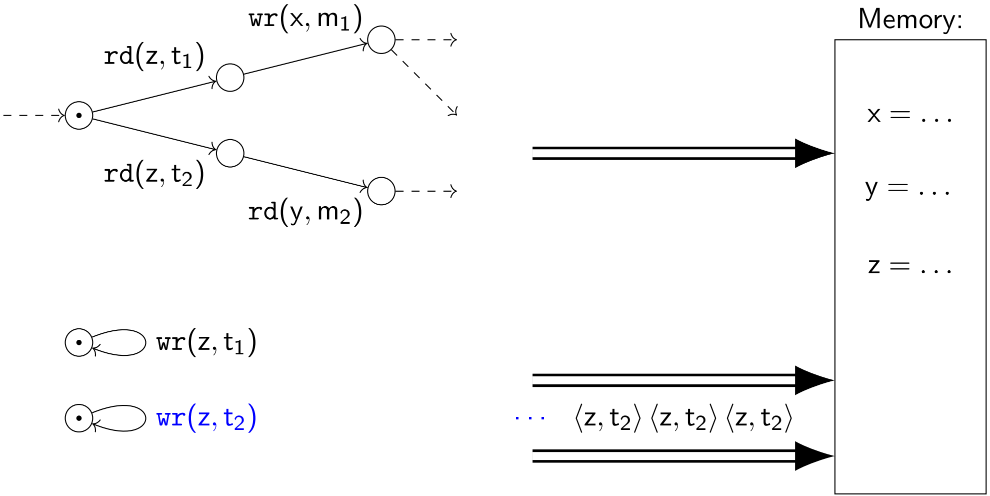
Process Fairness



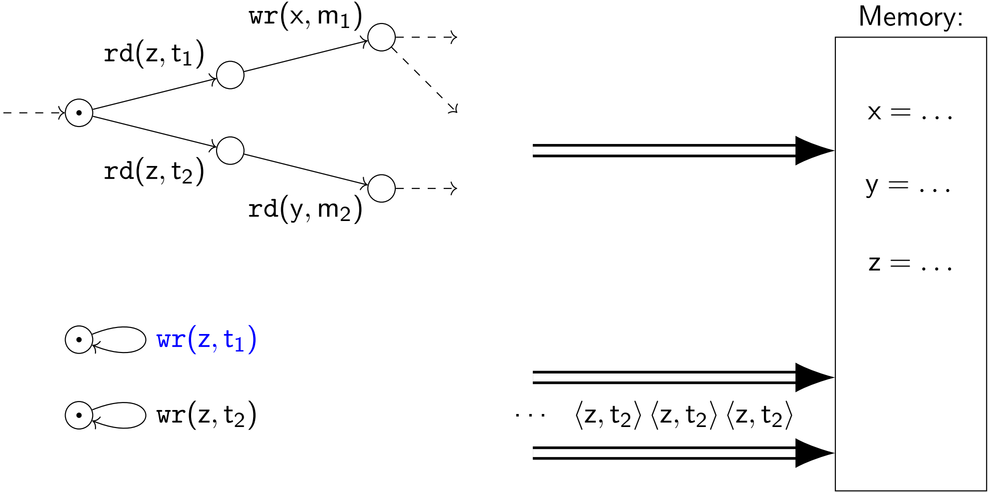
Process Fairness



Process Fairness



Process Fairness



$\mathtt{wr}(x, m_1)$

$\mathtt{rd}(z, t_1)$

$\mathtt{rd}(z, t_2)$

$\mathtt{rd}(y, m_2)$

Memory:

$x = \ldots$

$y = \ldots$

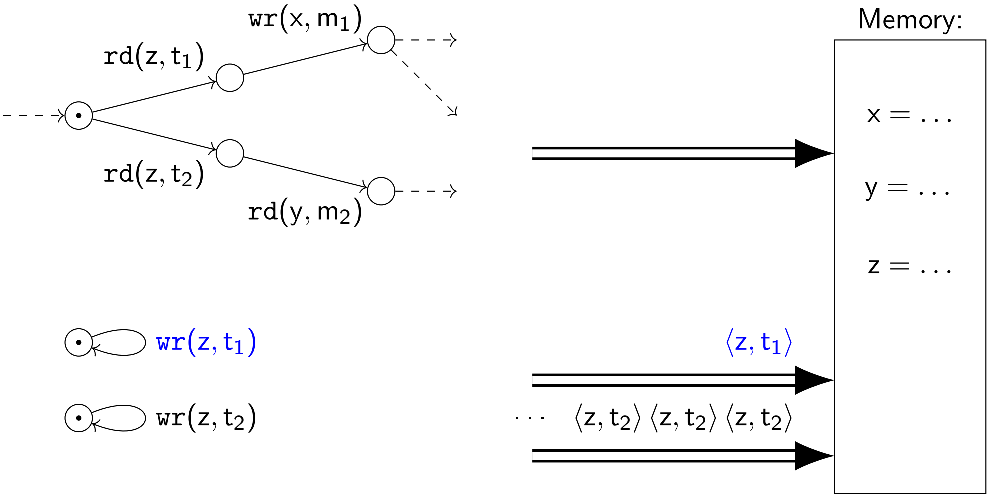
$z = \ldots$

$\mathtt{wr}(z, t_1)$

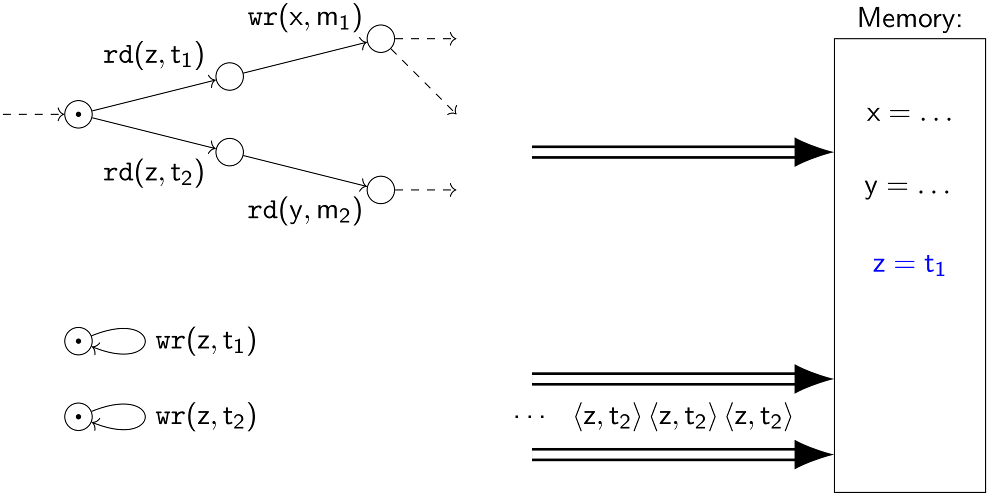
$\mathtt{wr}(z, t_2)$

$\cdots \quad \langle z, t_2 \rangle \, \langle z, t_2 \rangle \, \langle z, t_2 \rangle$

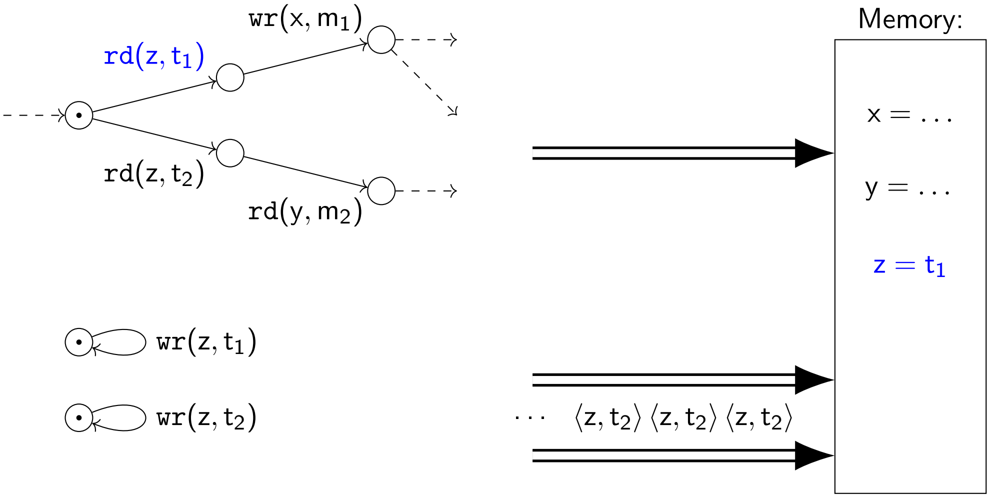
Process Fairness



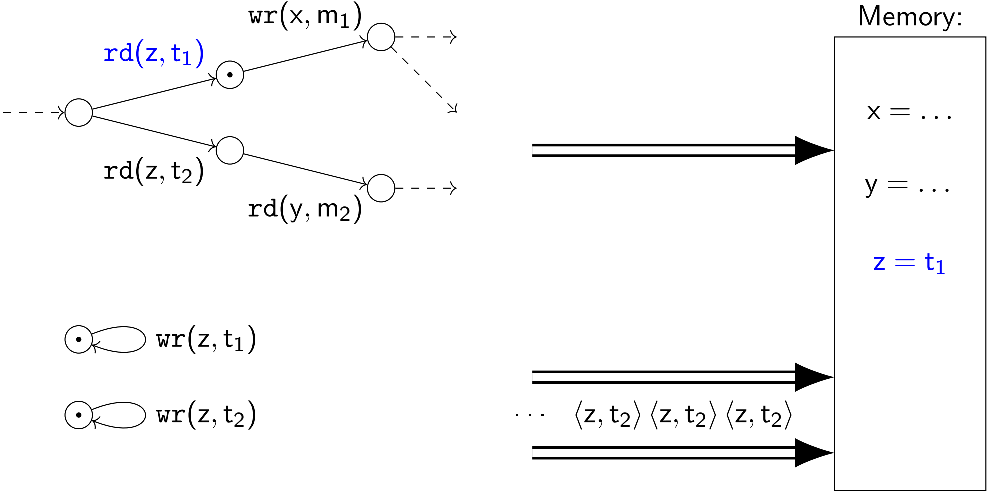
Process Fairness



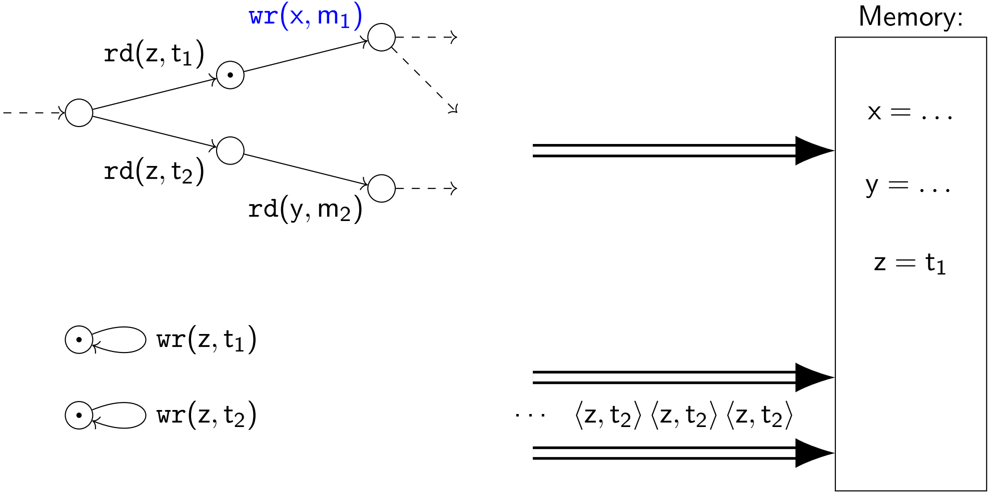
Process Fairness



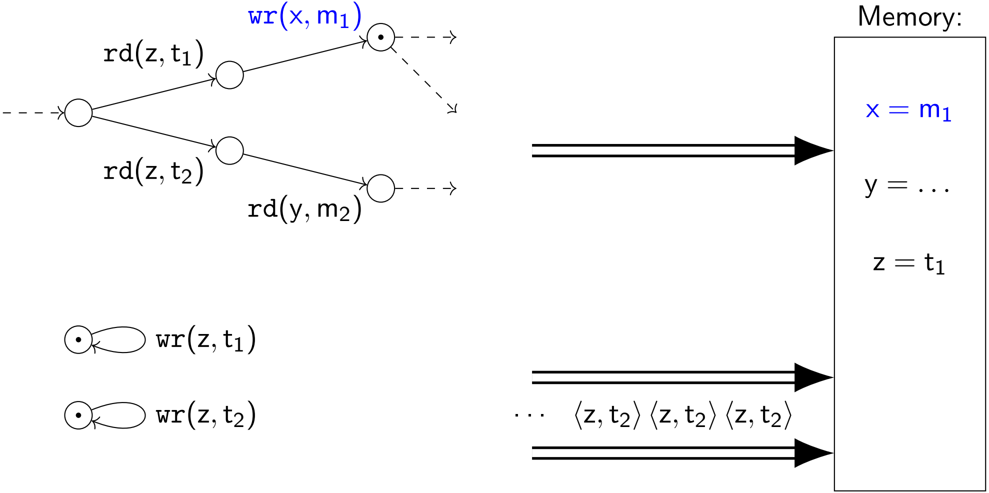
Process Fairness



Process Fairness



Process Fairness



$\mathtt{wr}(x, m_1)$

$\mathtt{rd}(z, t_1)$

$\mathtt{rd}(z, t_2)$

$\mathtt{rd}(y, m_2)$

$\mathtt{wr}(z, t_1)$

$\mathtt{wr}(z, t_2)$

$\cdots \quad \langle z, t_2 \rangle \, \langle z, t_2 \rangle \, \langle z, t_2 \rangle$

Memory:

$x = m_1$

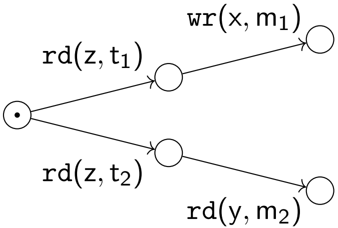
$y = \ldots$

$z = t_1$

Process Fairness

Every enabled process must be executed infinitely often.

- similar to reachability games
- update player simulates PCS run, process player is passive
- reduce PCS reachability (undecidable) to TSO safety game



Theorem

The safety problem under TSO semantics with process fairness is undecidable.

Conclusion

- reachability and safety *without* fairness
  - reduce to single-process programs
  - finite behaviour / PSPACE-complete

Conclusion

- reachability and safety *without* fairness
  - reduce to single-process programs
  - finite behaviour / PSPACE-complete

- reachability with update fairness and safety with process fairness
  - reduction from PCS reachability
  - undecidable

Conclusion

- reachability and safety *without* fairness
  - reduce to single-process programs
  - finite behaviour / PSPACE-complete
- reachability with update fairness and safety with process fairness
  - reduction from PCS reachability
  - undecidable
- further work could consider other
  - winning conditions
  - fairness conditions
  - weak memory models

Reachability and Safety Games under TSO Semantics

SCooL / GandALF 2024 in Reykjavik

Stephan Spengler

Uppsala University, Sweden

20 June 2024