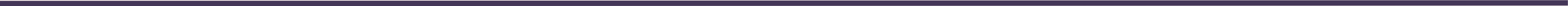

CAUSAL-BROADCAST MEMORY

Amir Karniel, Prof. Ori Lahav

OUTLINE

- Causal broadcast
 - Distributed shared memory
 - CBM consistency
 - Verification questions
 - Relation to other models
-

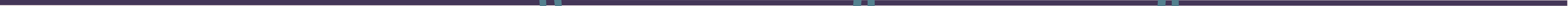


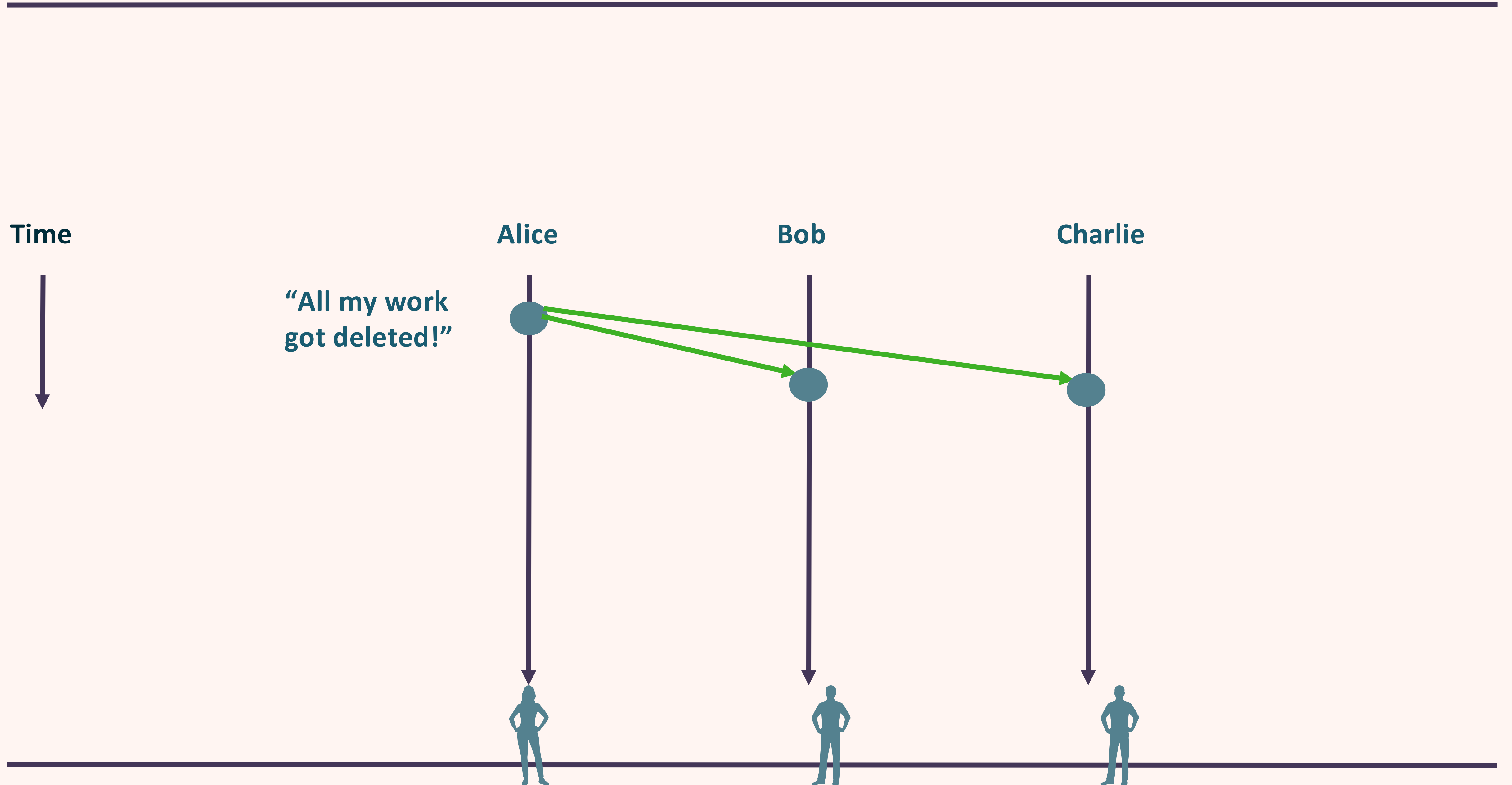
Time

Alice

Bob

Charlie





Time



Alice

Bob

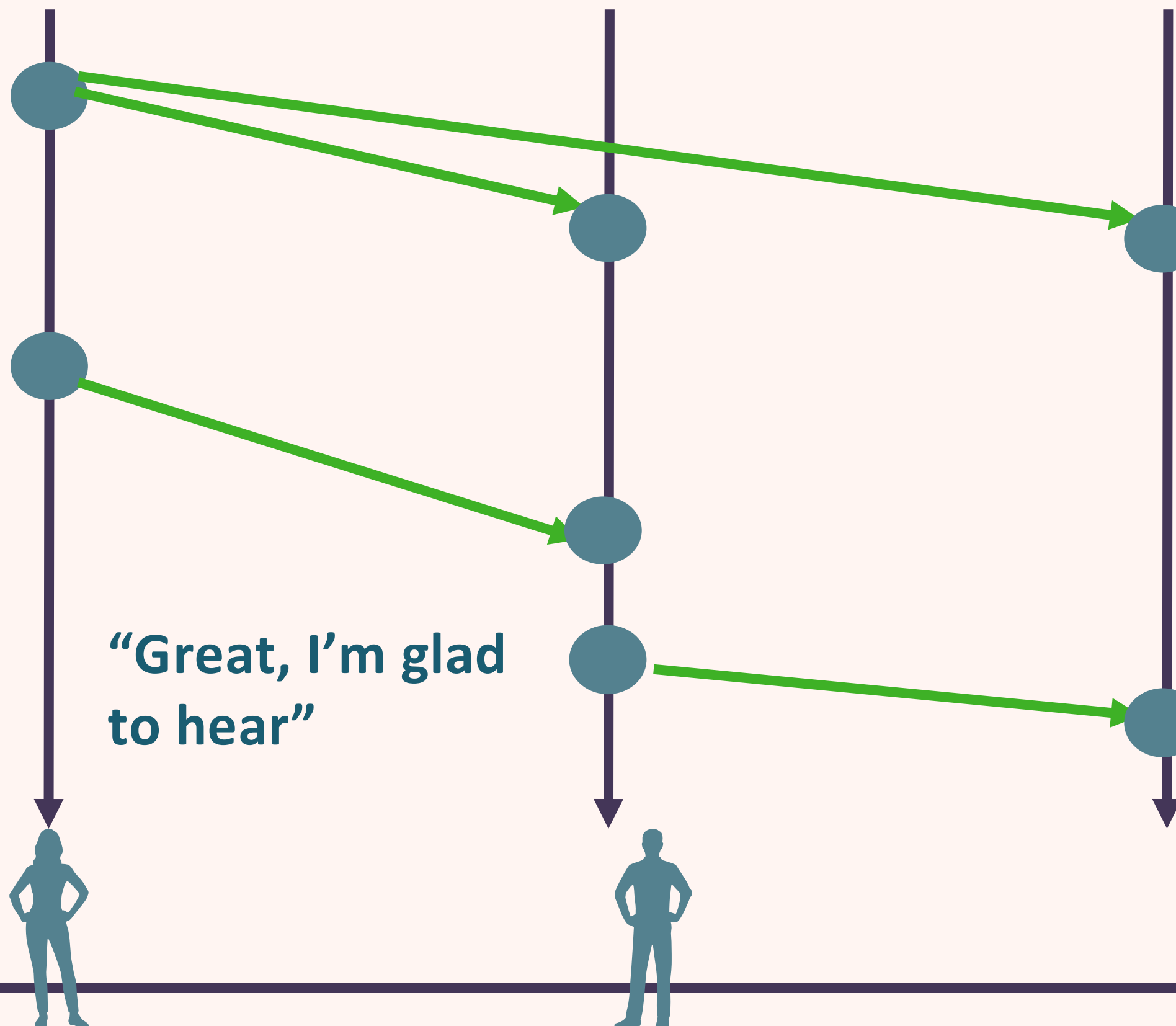
Charlie

"All my work
got deleted!"

"Never mind,
it's saved"

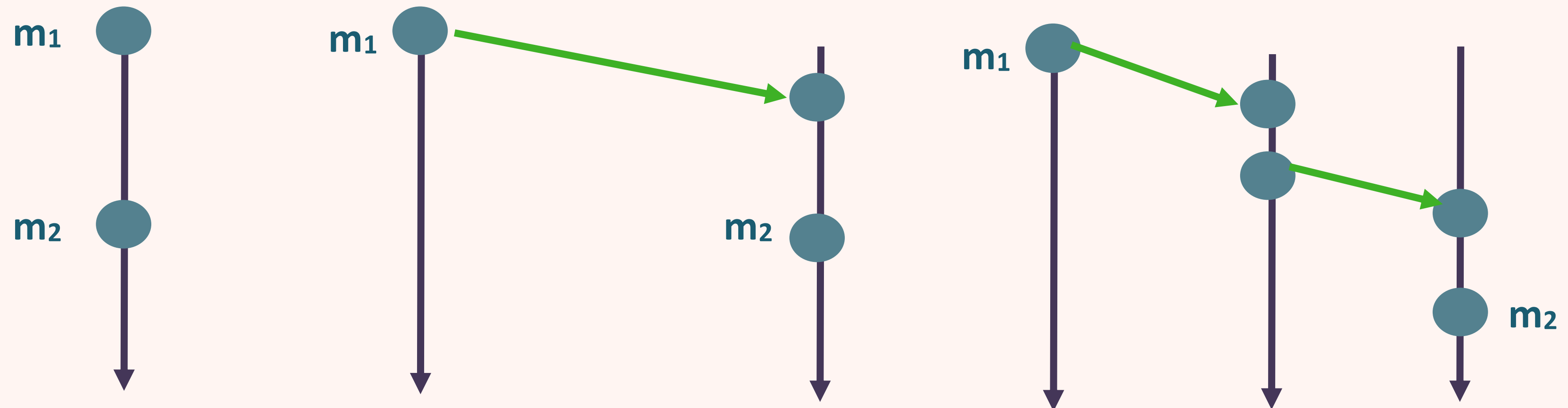
"Great, I'm glad
to hear"

Bob is rude...



Causal Broadcast

If m_1 is a *potential cause* of m_2 then everyone must receive m_1 before receiving m_2



DistMem: Shared Memory Based on Causal Broadcast

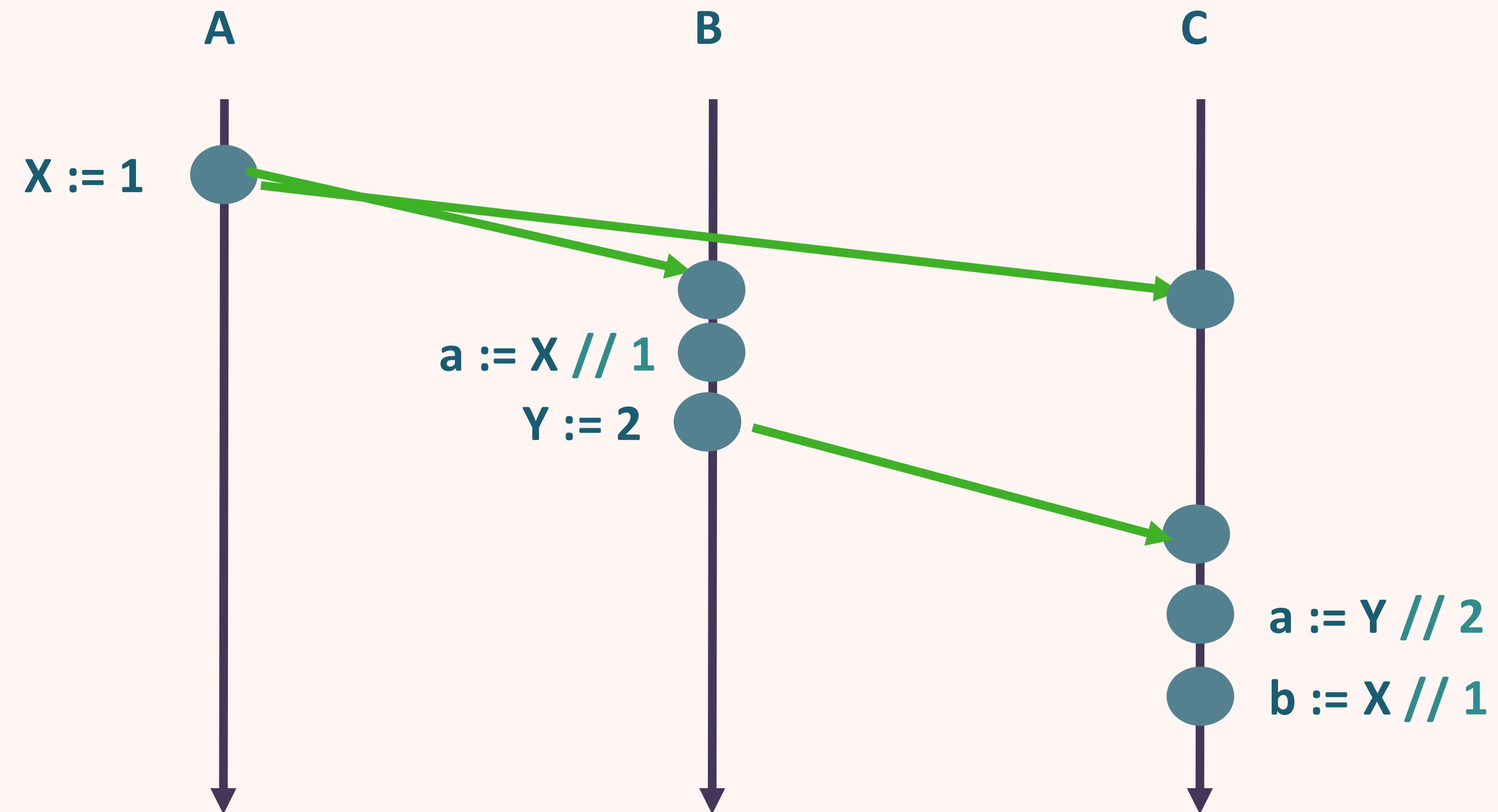
Every process p has a local mapping M_p

The code for a process p is:

method <code>write(x, v)</code>	method <code>read(x)</code>	when Wxv is received
$M_p := M_p[x \mapsto v]$	<code>return($M_p(x)$)</code>	from another process:
<code>causal_broadcast(Wxv)</code>		$M_p := M_p[x \mapsto v]$
<code>return()</code>		

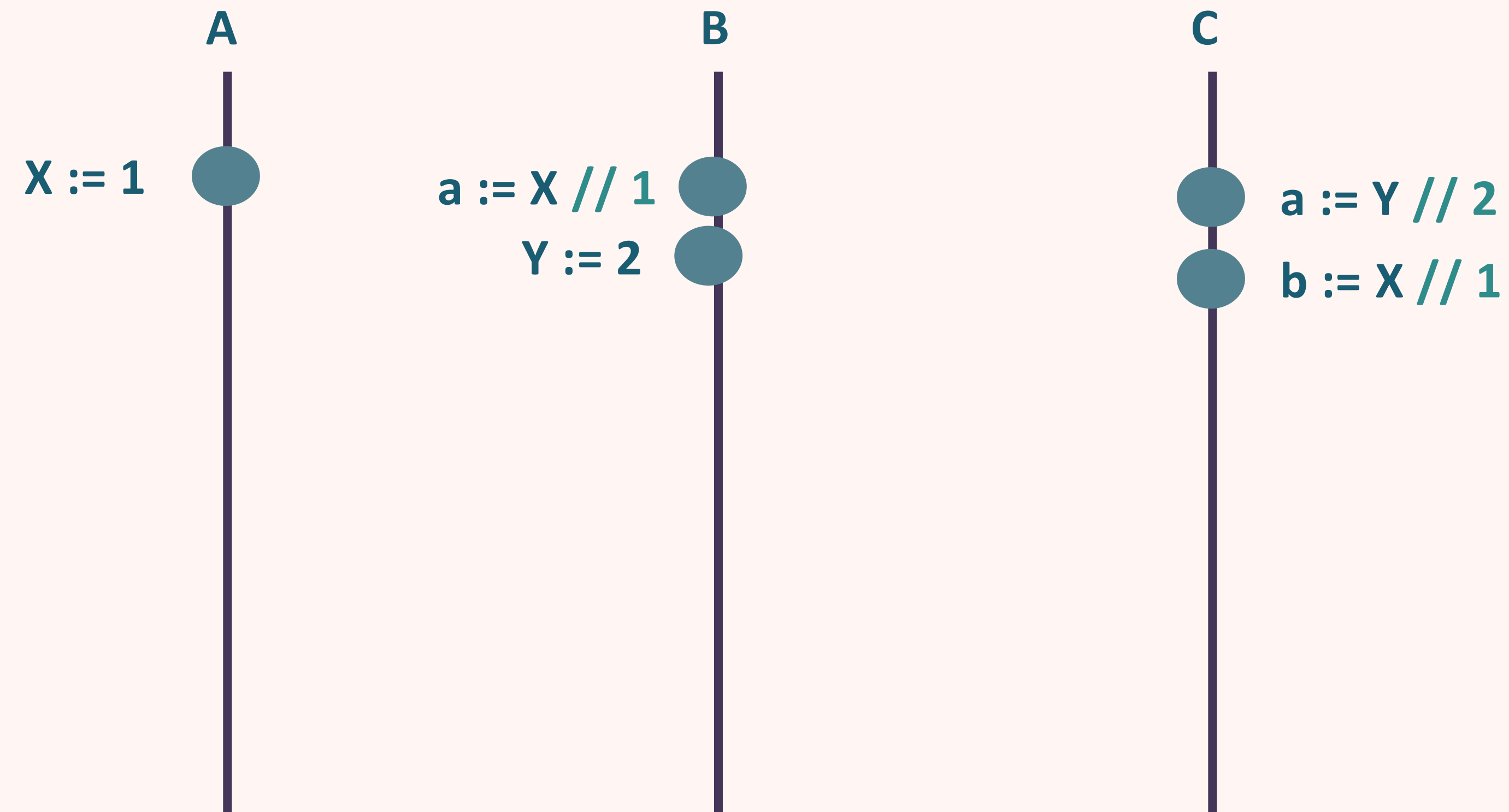
Ahamad, M., Neiger, G., Burns, J.E., Kohli, P., Hutto, P.W.:
Causal memory: Definitions, implementation, and programming.
Distributed Comput. 9(1), 37–49 (1995).

DistMem: Shared Memory Based on Causal Broadcast



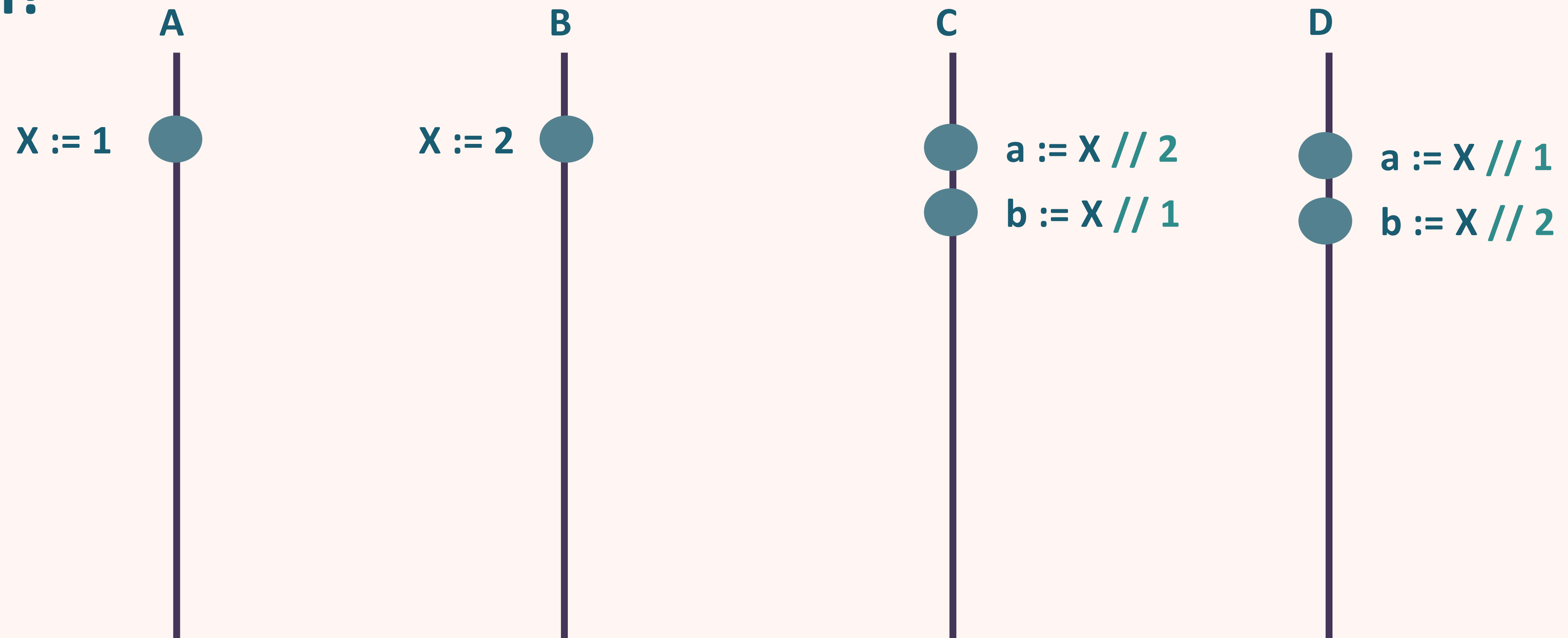
What if propagation is hidden?

Goal: given an abstract execution, is it obtainable by DistMem?

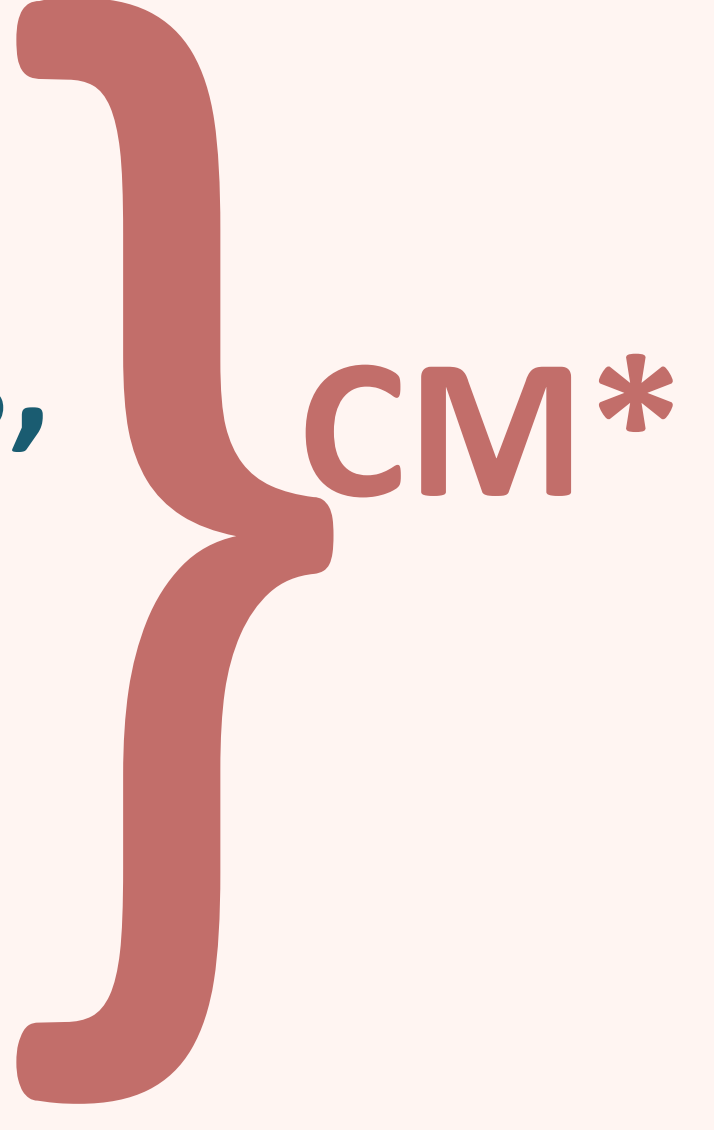


What if propagation is hidden?

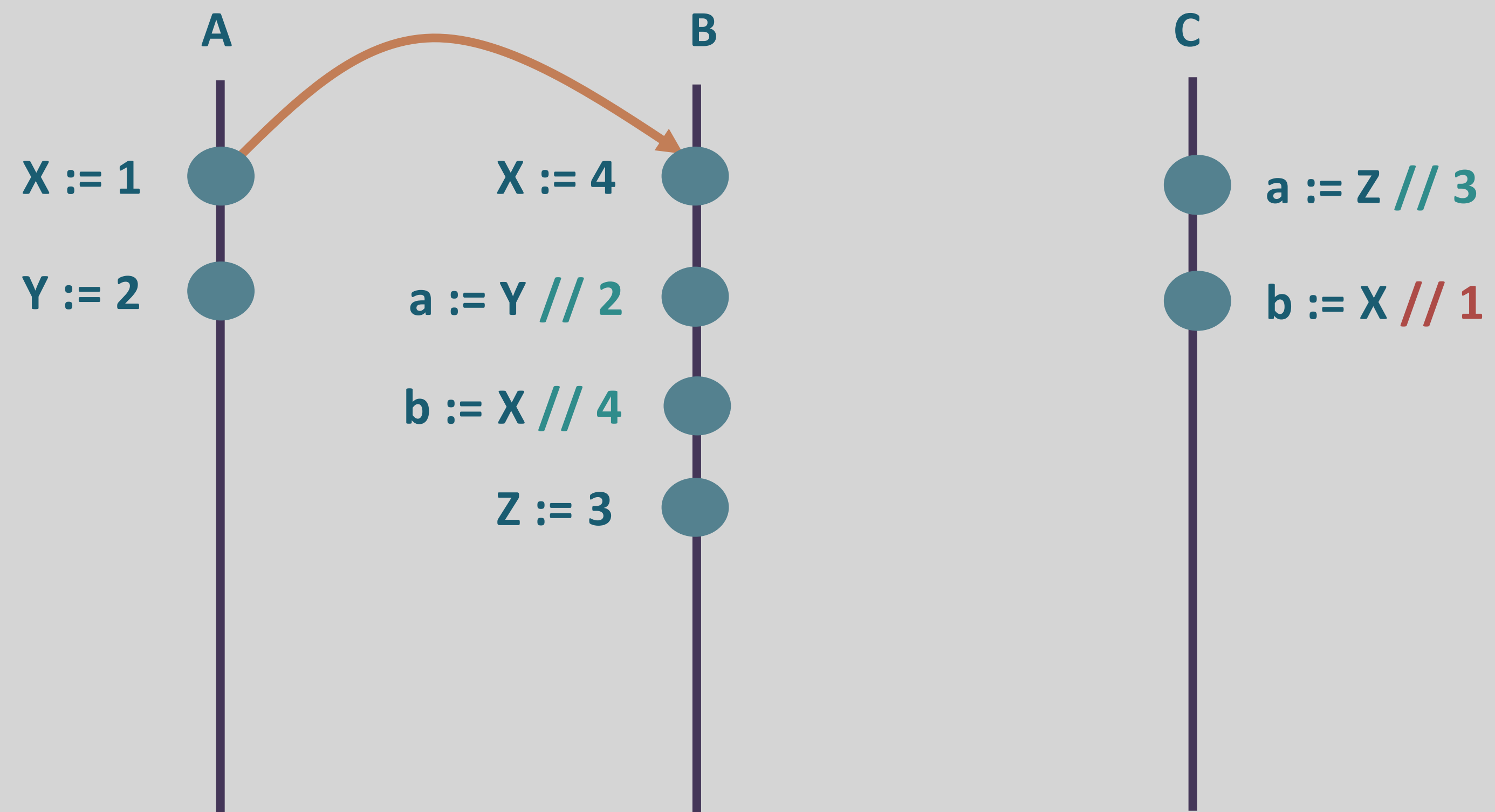
Goal: given an abstract execution, is it obtainable by DistMem?



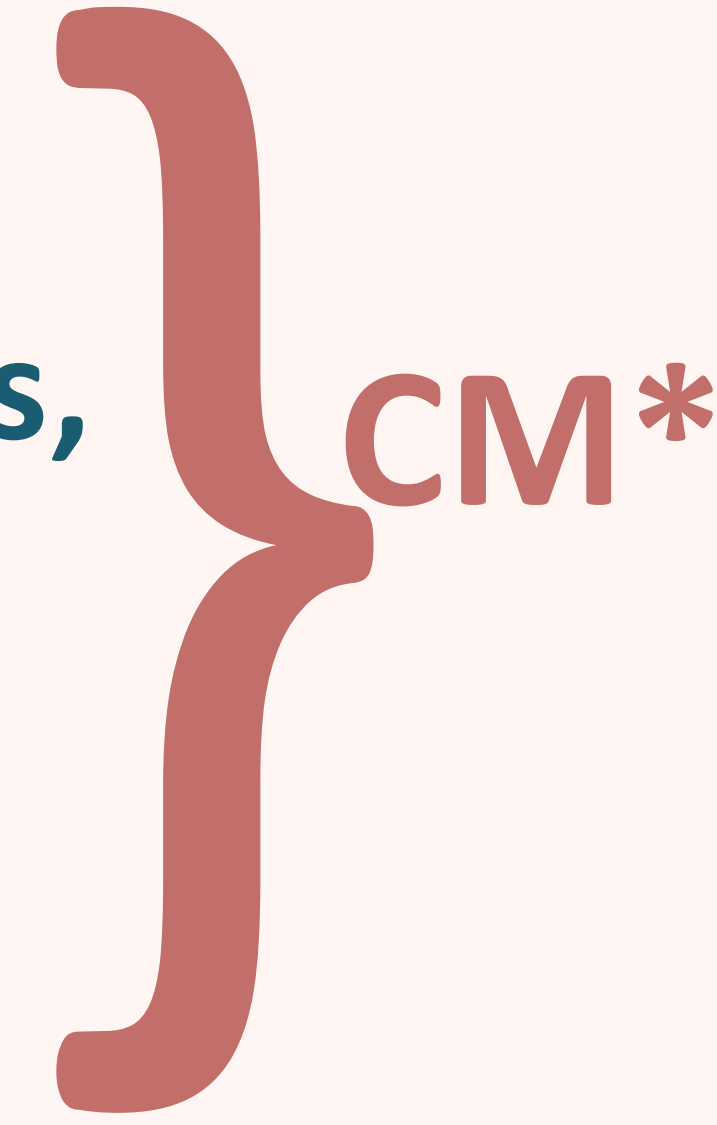
Causal-Broadcast Memory Model

- There exists a (partial) *causal order* *co*
 - Every process *P* should have a *serialization* S_P of all events, justifying its own read (load) events by their *co*-past
 - Each serialization has to respect *co*
- 

Is it enough?



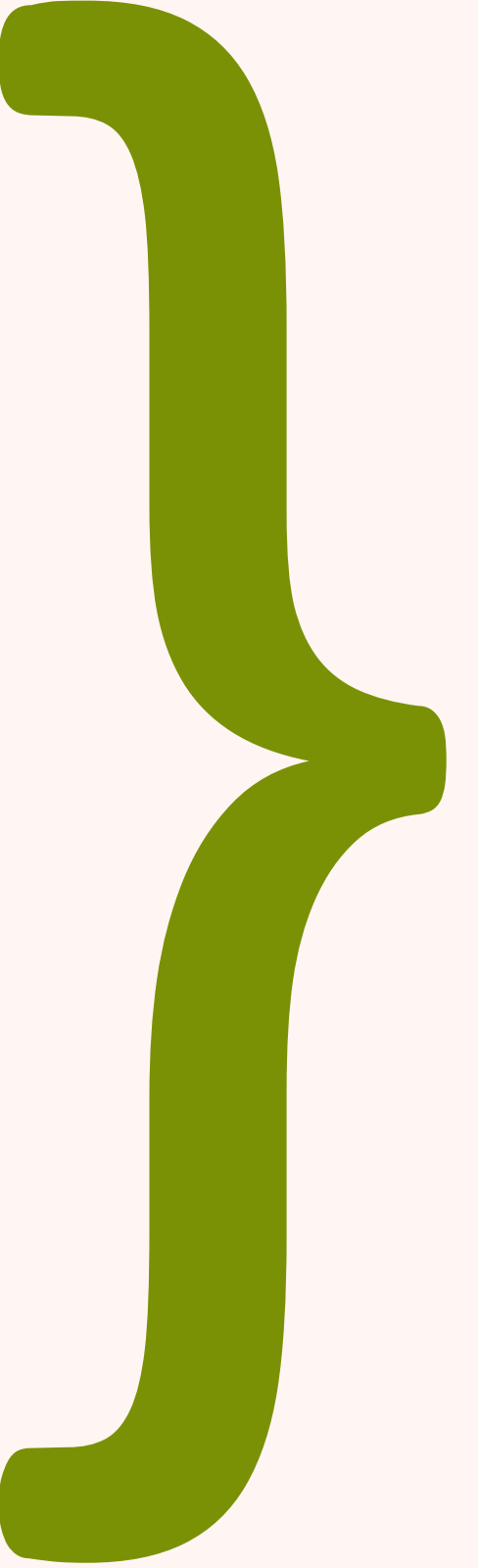
Causal-Broadcast Memory Model

- There exists a (partial) *causal order* *co*
 - Every process *P* should have a *serialization* S_P of all events, justifying its own read (load) events by their *co*-past
 - Each serialization has to respect *co*
- 
- CM*

Is it enough?

- If S_P places an event before an event of *P*, there must be *co* between them
-

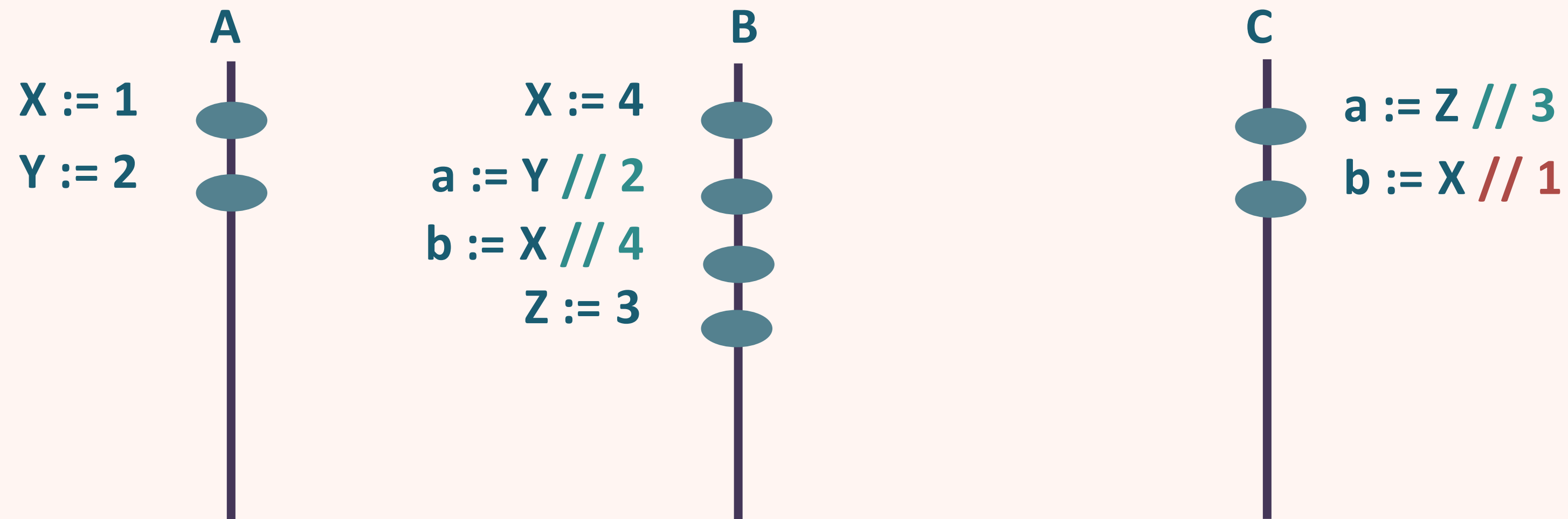
Causal-Broadcast Memory Model

- There exists a (partial) *causal order* *co*
 - Every process *P* should have a *serialization* S_P of all events, justifying its own read (load) events by their *co*-past
 - Each serialization has to respect *co*
 - If S_P places an event before an event of *P*, there must be *co* between them
- 

New: CBM

Verification Questions

➤ Checking consistency of an abstract execution



➤ Polynomial time

Verification Questions

- Checking state reachability of a program using DistMem (*safety* problem)

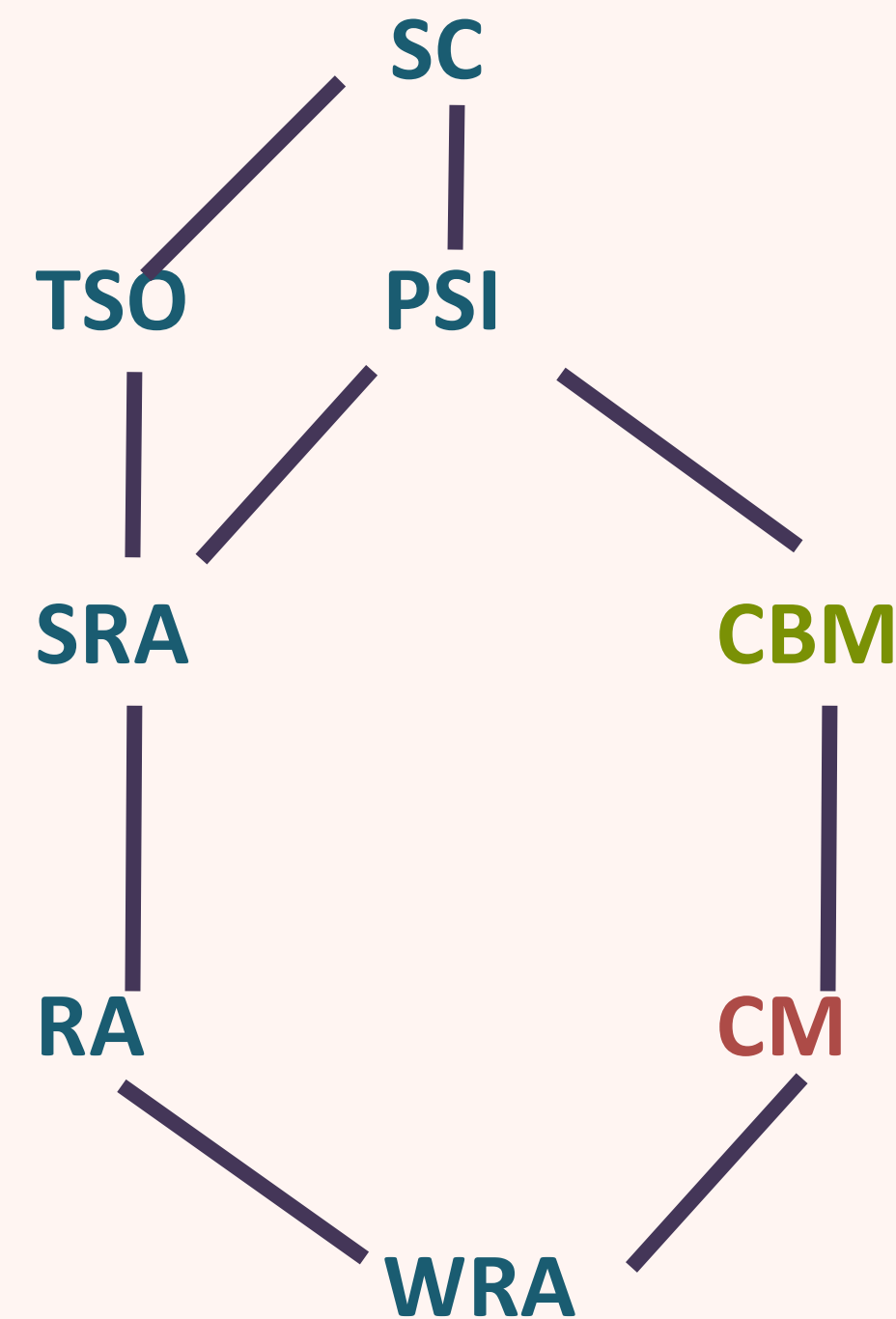
Can

$X := 1$		$Y := 1$
$a := Y$		$b := X$

reach $a=b=0$?

- Undecidable
-

Other weak memory models



- If a program has no write-write-races under PSI, the whole “diamond” collapses
-

Conclusion

- A new model CBM capturing DistMem precisely
 - Enhancing users' reasoning
- Answering verification questions regarding CBM
- Placing CBM among other models and formalizing it in similar ways
- Building a bridge between distributed systems and weak memory

Thank you!
