
CAUSAL-BROADCAST MEMORY

Amir Karniel, Ori Lahav
Tel Aviv University

OVERVIEW

- **CBM** - a new memory model capturing the **precise consistency guarantees** of a distributed memory implementation based on **broadcast** } Generalized causal to shared objects (beyond memory)



- **Stronger** than Causal Memory consistency
- **Testing** problem is solvable in polynomial time
- **Reachability** problem is undecidable

Time



Alice

Bob

Charlie

“All my work
got deleted!”

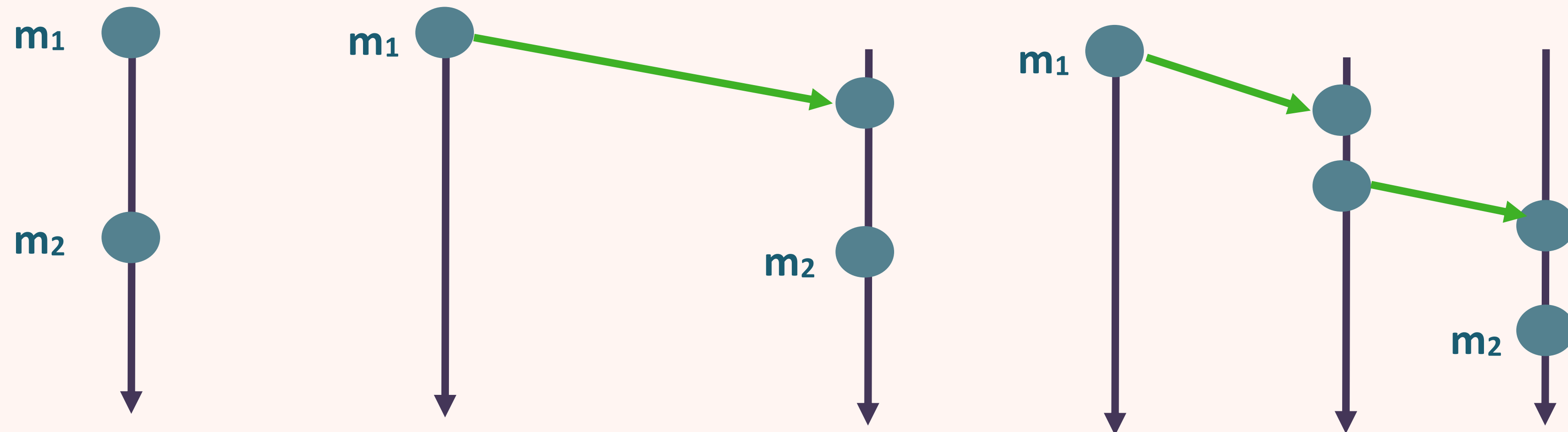
“Never mind,
it’s saved”

“Great, I’m glad
to hear”

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

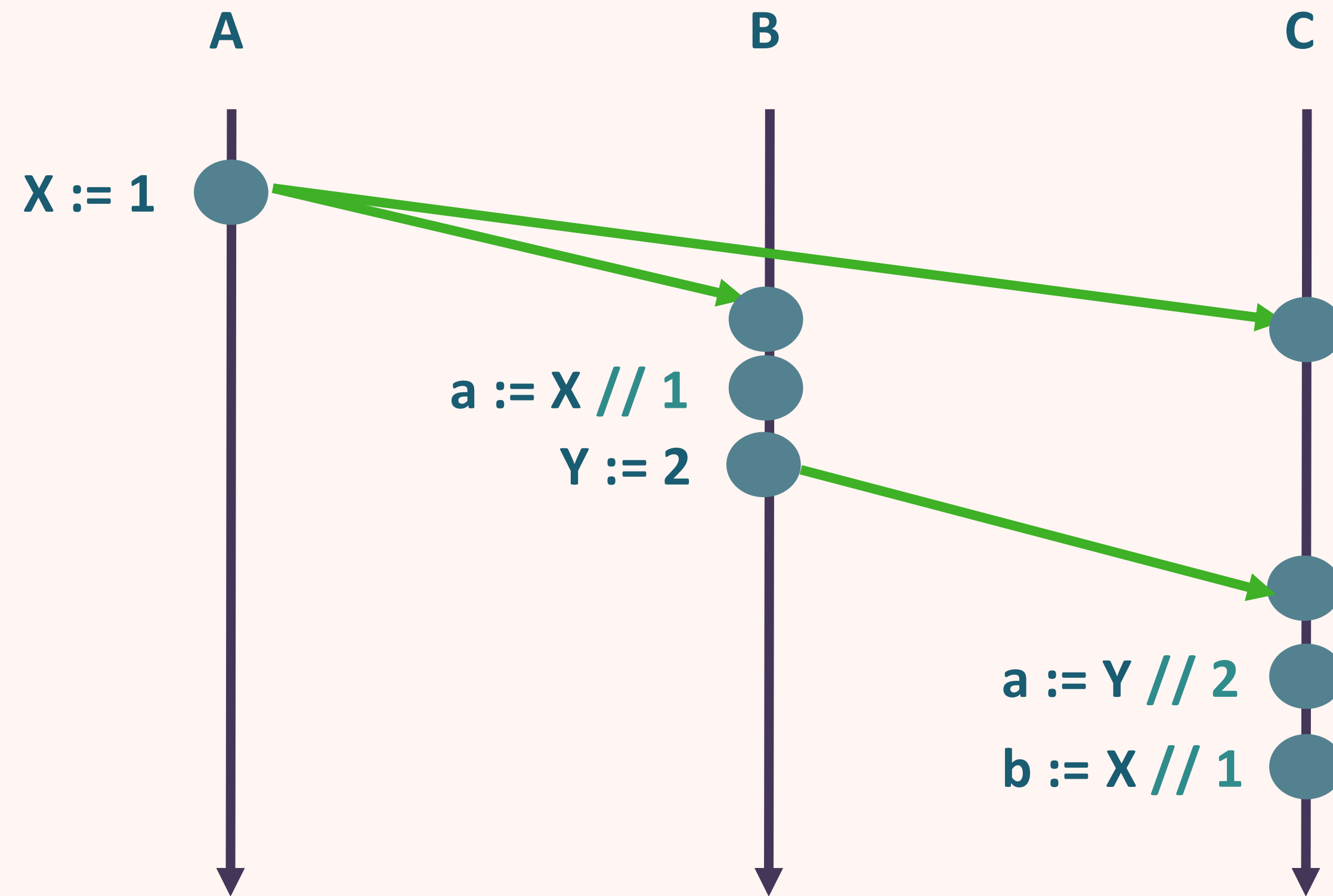
Every process p has a local mapping M_p

The code for a process p is:

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

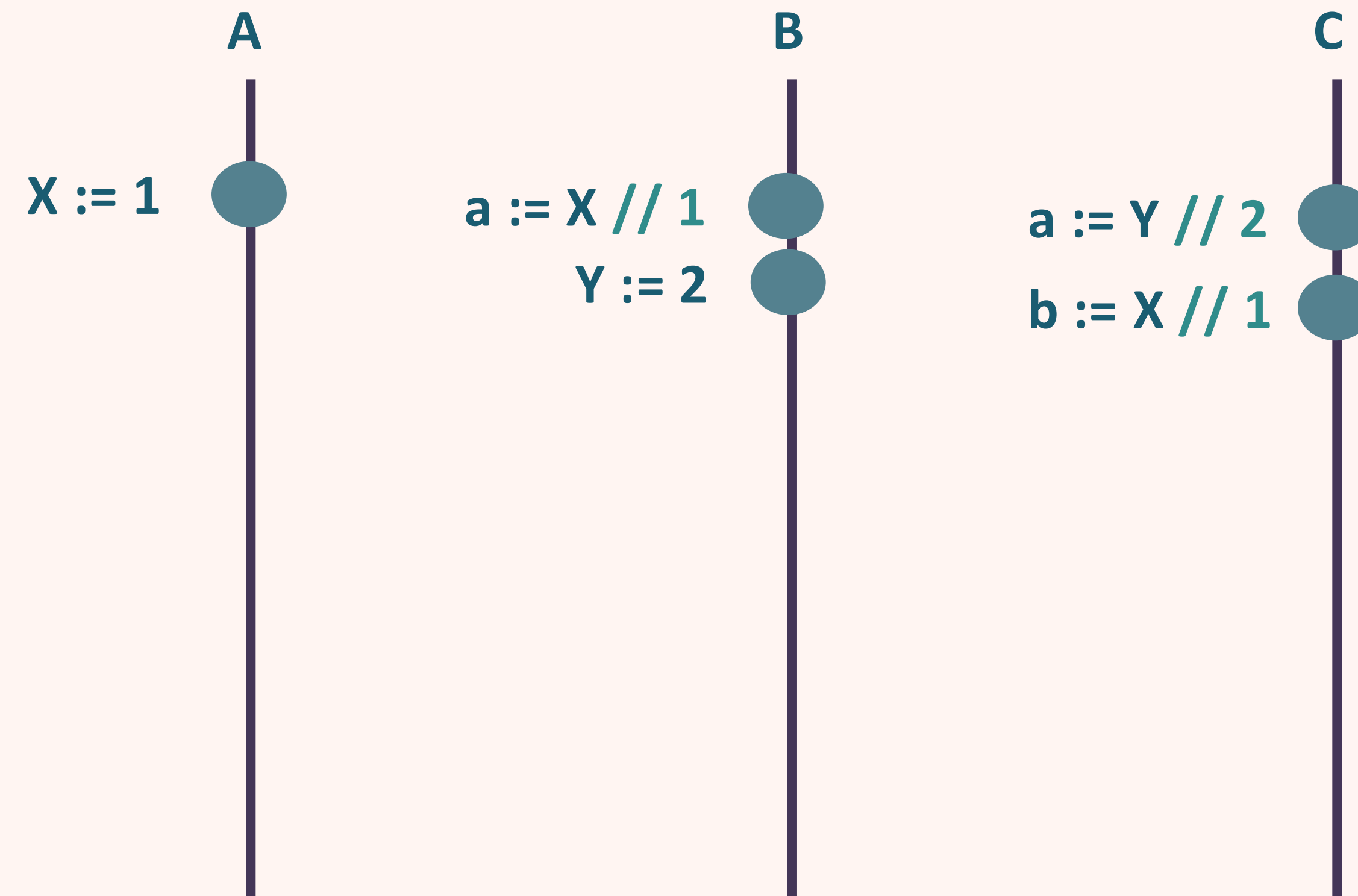
Ahamad, M., Neiger, G., Burns, J.E., Kohli, P., Hutto, P.W.:
Causal memory: Definitions, implementation, and programming.
Distributed Comput. 9(1), 37–49 (1995).

DistMem: Shared Memory Based on Causal Broadcast

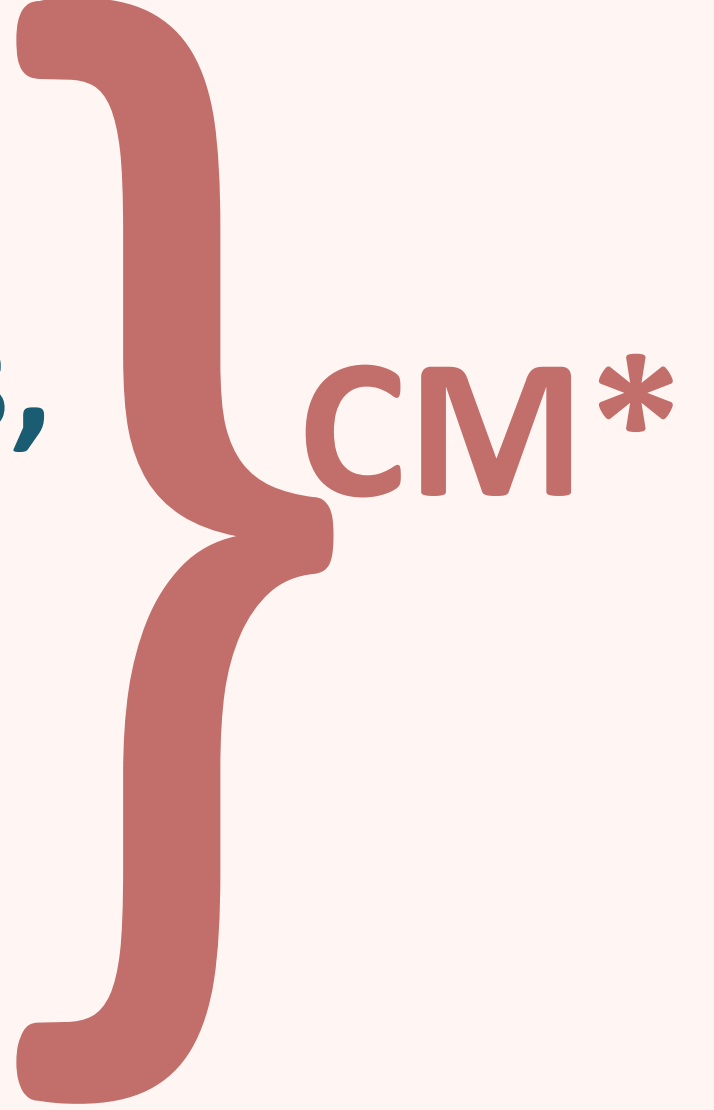


What if propagation is hidden?

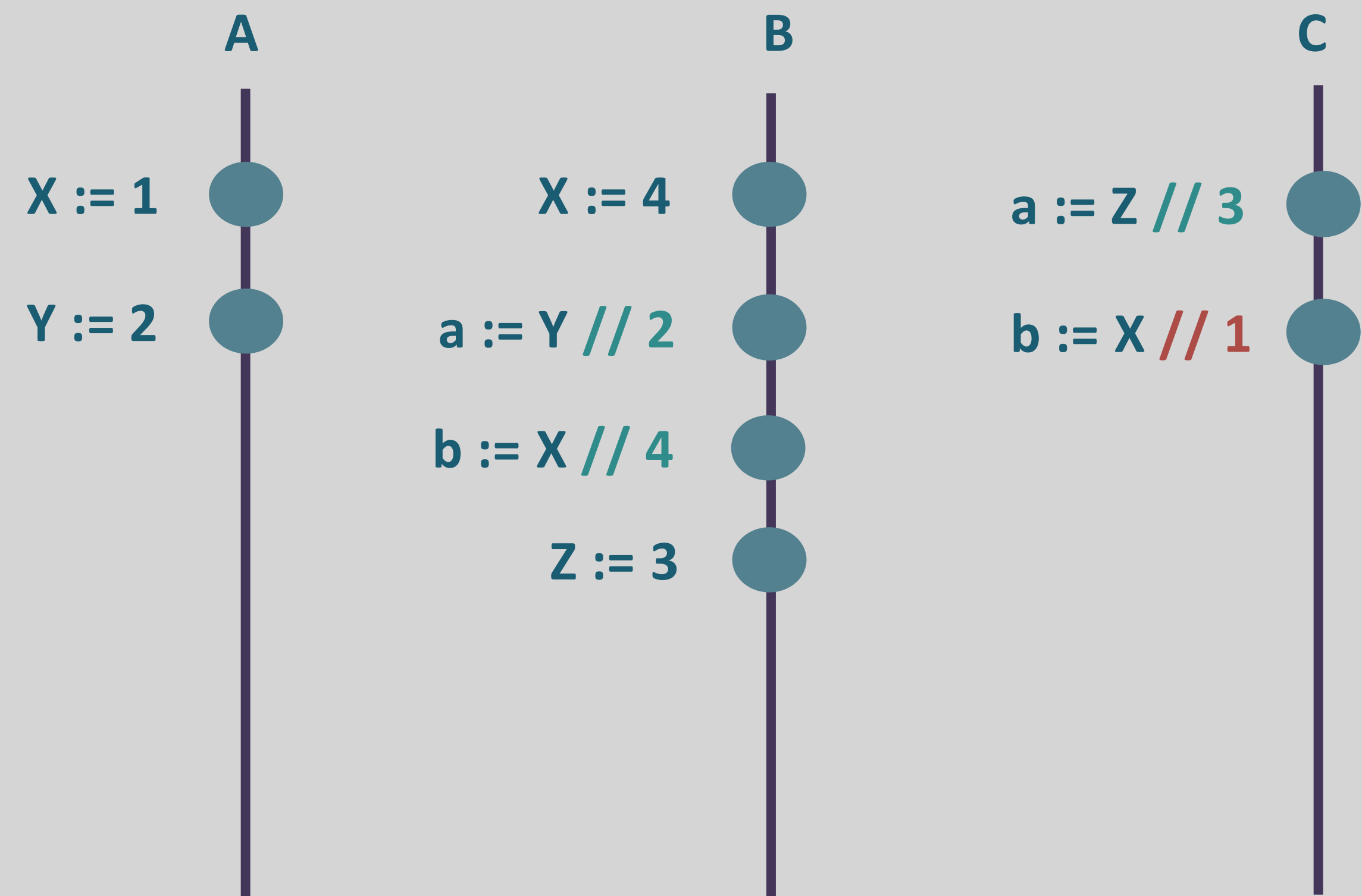
Goal: given an abstract execution, is it obtainable by DistMem?

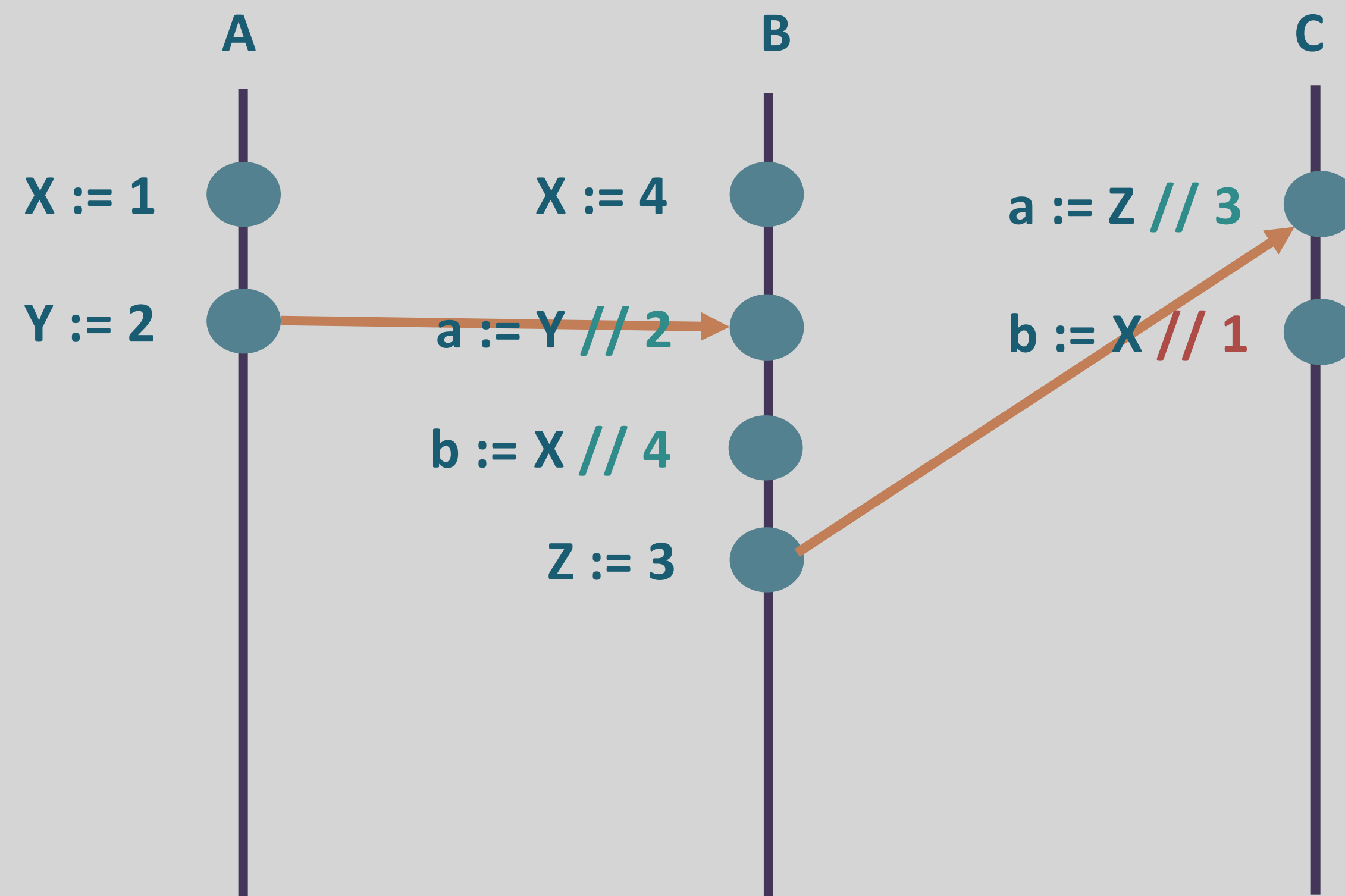


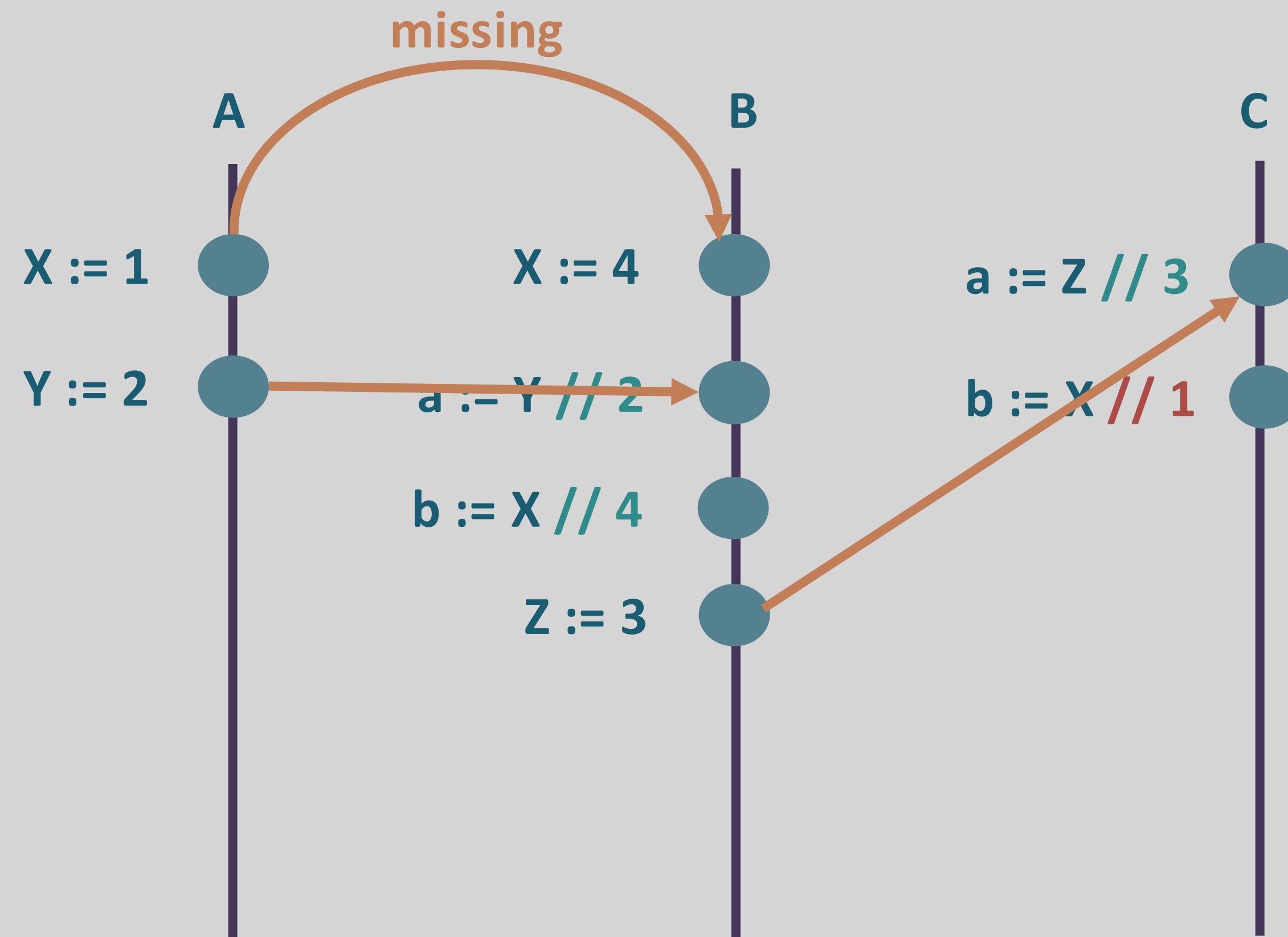
Causal-Broadcast Memory Model

- There exists a (partial) *causal order* *co*
 - Every process *p* should have a *serialization* S_p of all events, justifying its own read events by their *co*-past
 - Each serialization has to agree with *co*
- 
- CM*

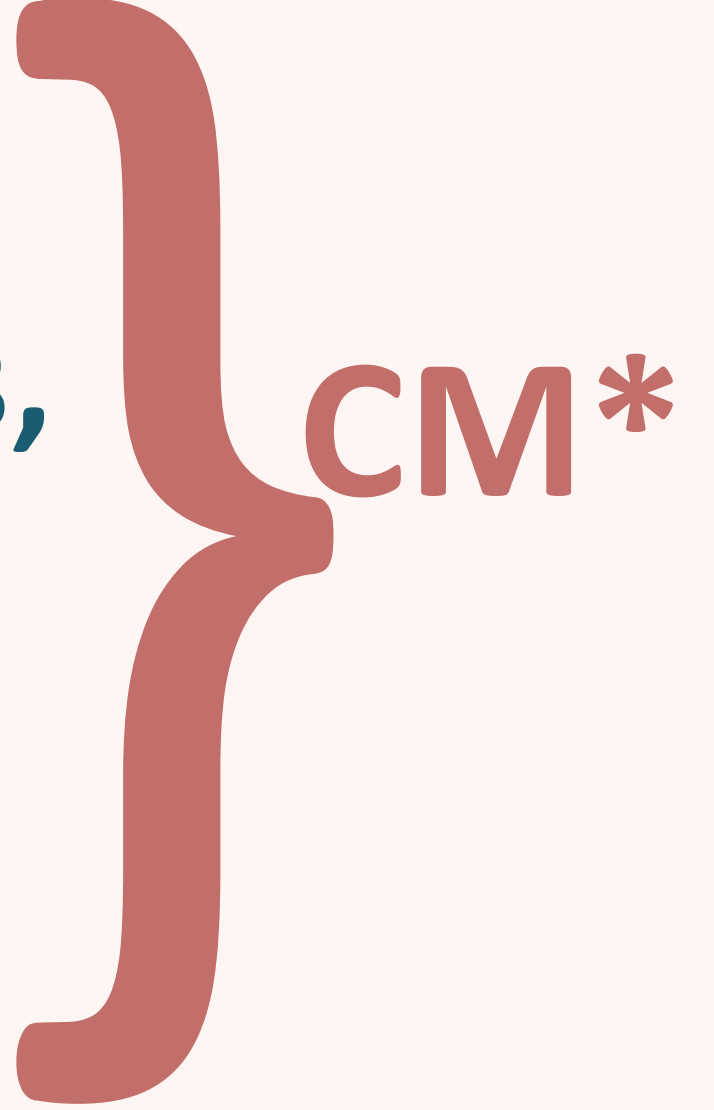
Is it enough?







Causal-Broadcast Memory Model

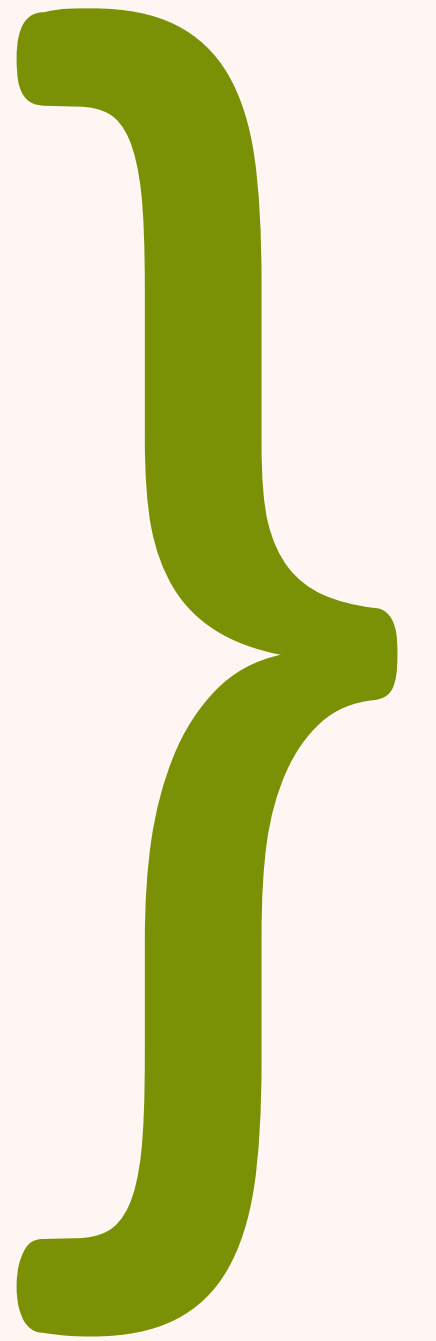
- There exists a (partial) *causal order* *co*
 - Every process *p* should have a *serialization* S_p of all events, justifying its own read events by their *co*-past
 - Each serialization has to agree with *co*
- 
- CM*

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 agree with *co*
- If S_p places an event before an event of *p*, there must be *co* between them



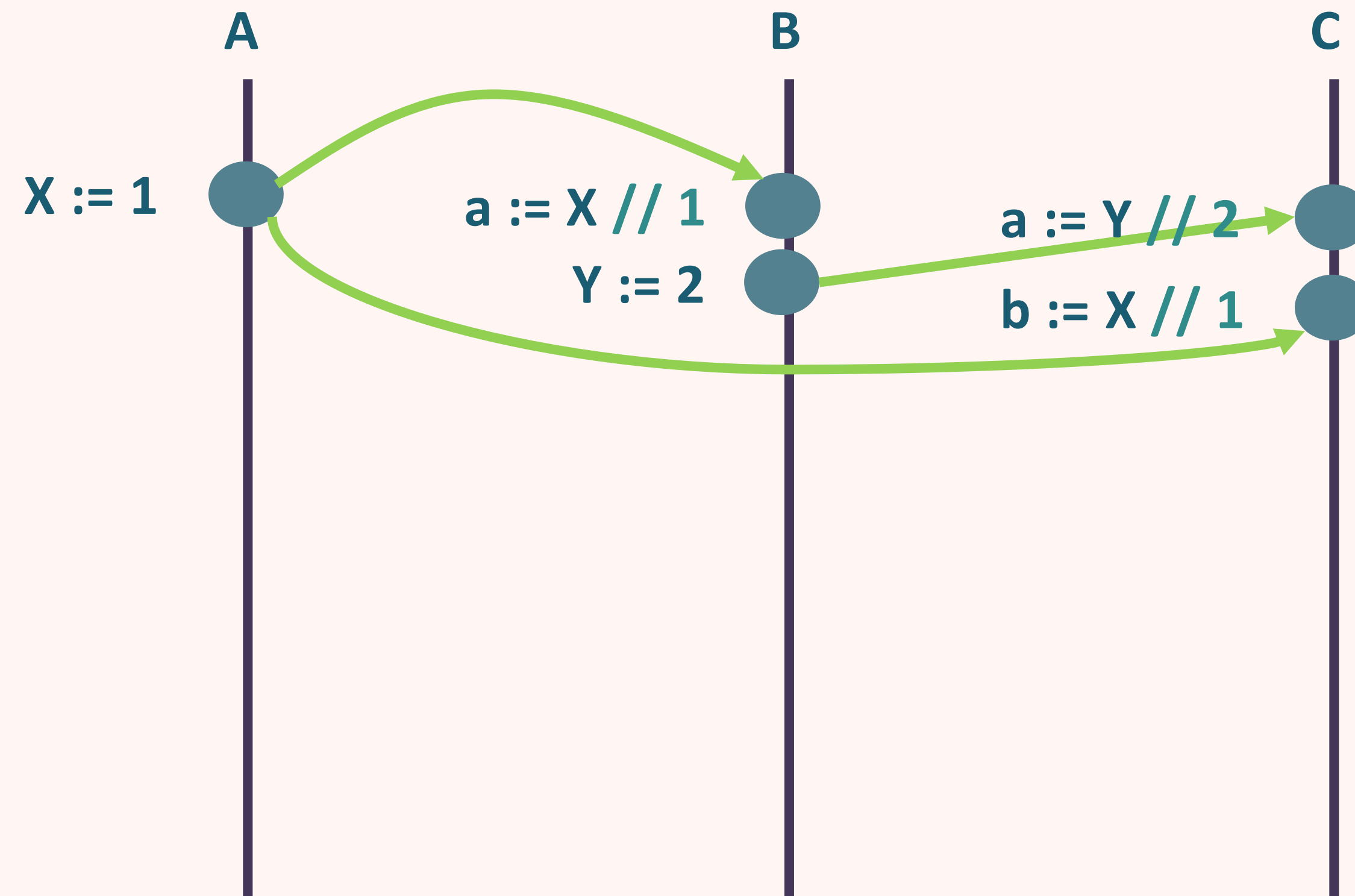
New: CBM

DistMem $\Rightarrow$ CM CM $\nRightarrow$ DistMem

CBM $\Leftrightarrow$ DistMem

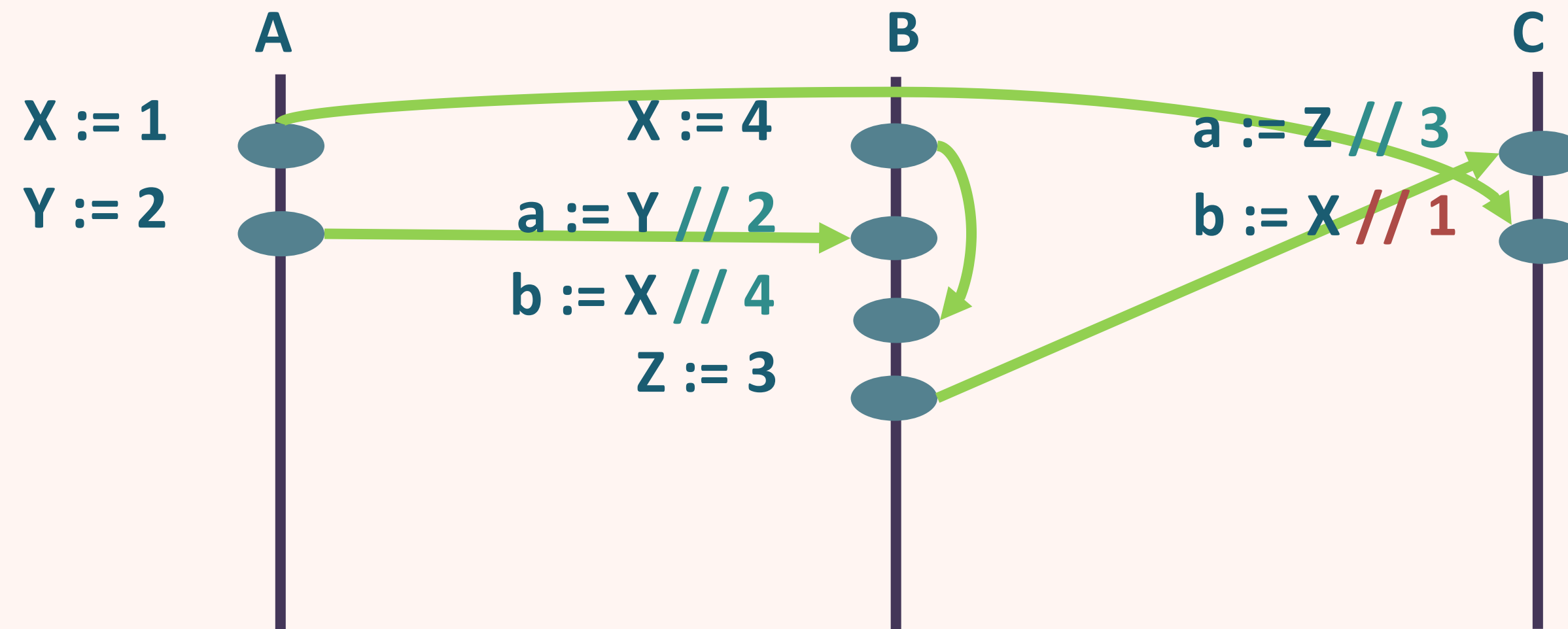
More Formally: Execution Graphs

Given a **reads-from** relation **rf**, the serializations must comply with it



Verification Challenges

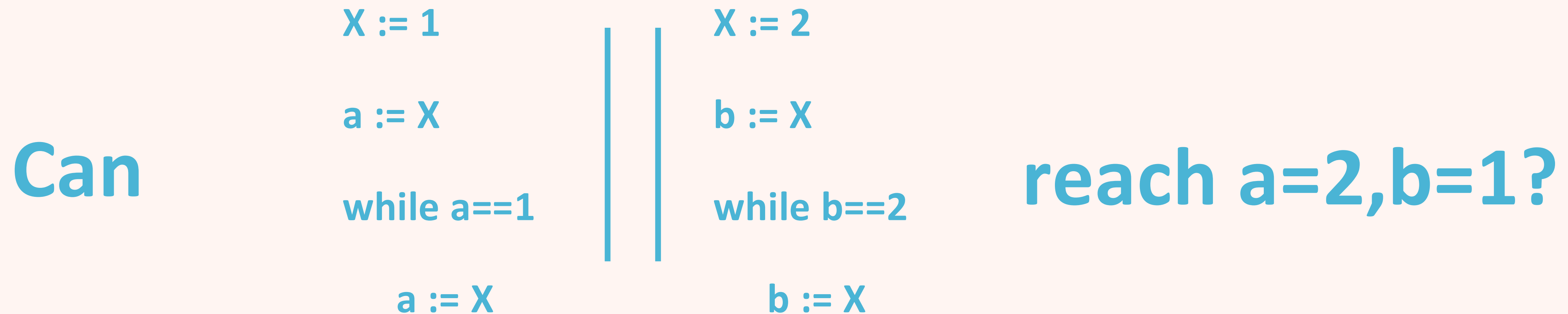
- Checking consistency of an execution graph



- Polynomial time (saturation based argument)

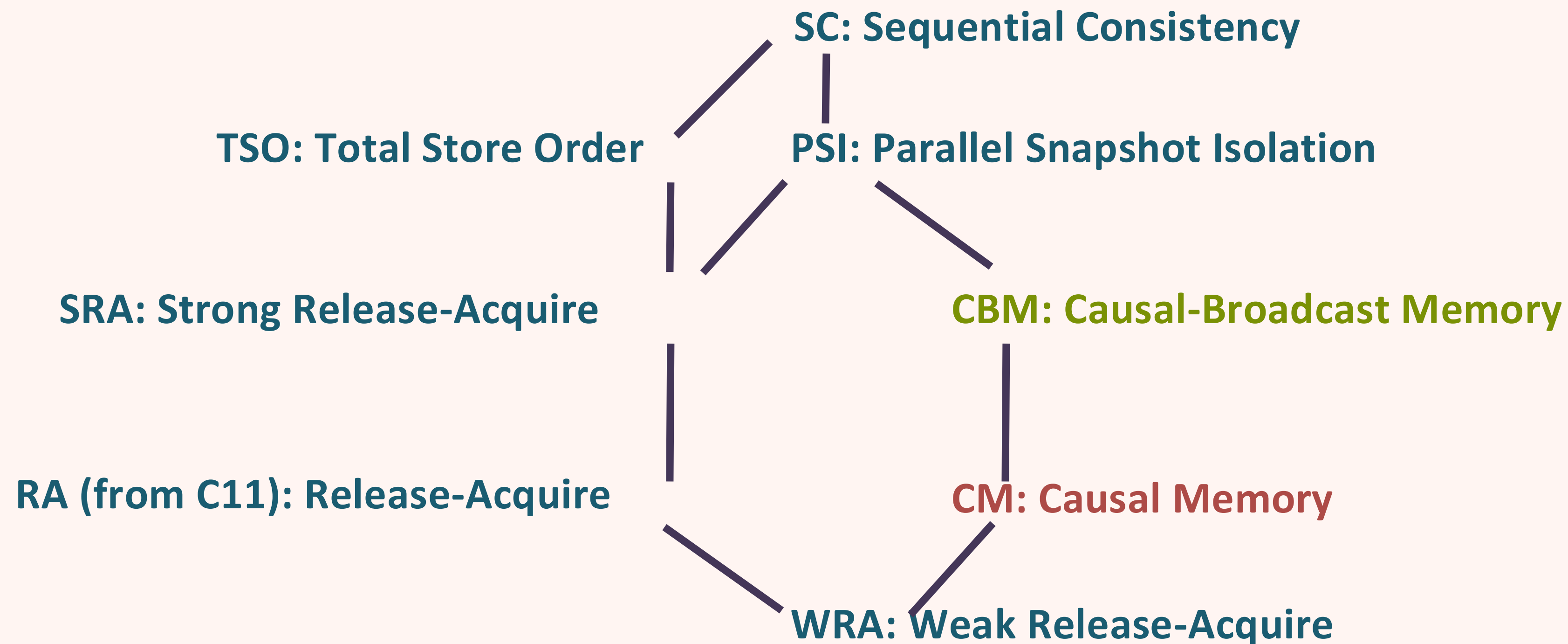
Verification Challenges

- Checking state reachability of client programs operating on top of DistMem (*safety* problem)



- Undecidable (reduction from reachability in communicating finite state machines)

Other weak memory models



- If a program has no write-write-races under PSI, the whole “hexagon” 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!