Distributed Snapshots

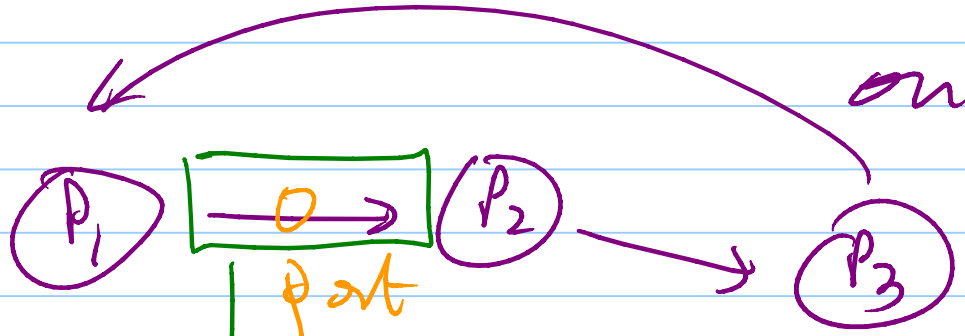
- Chandy & Lamport 1985

↳ checkpoint or "record state" of distributed computations

System → processes, nodes, messages

channel ↓

every process on a diff node



↳ channel → FIFO

msg are delivered in order sent

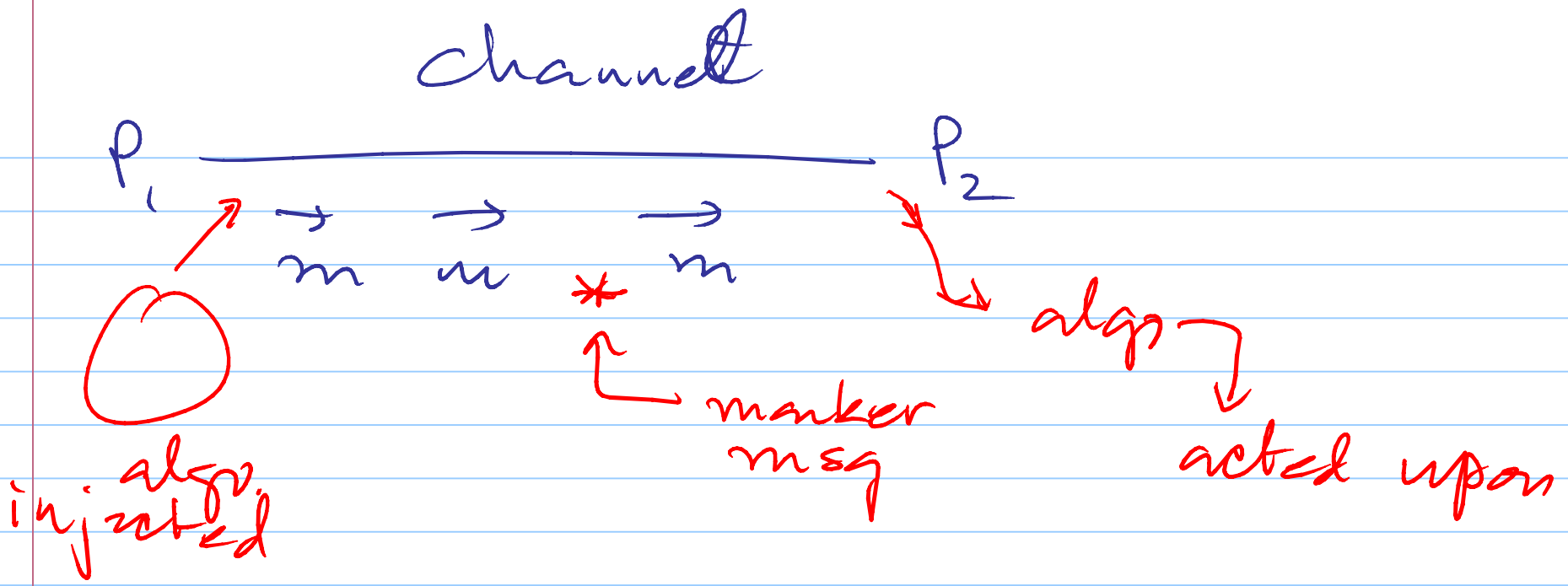
"State recording" → checkpoint
→ snapshot

↳ of a process

↳ write the state to disk

state: ↑ Variables (registers, heap, stack)

disk: ↑ local


channel
P_1
P_2
m
m
m
*
algo injected
marker msg
algo acted upon

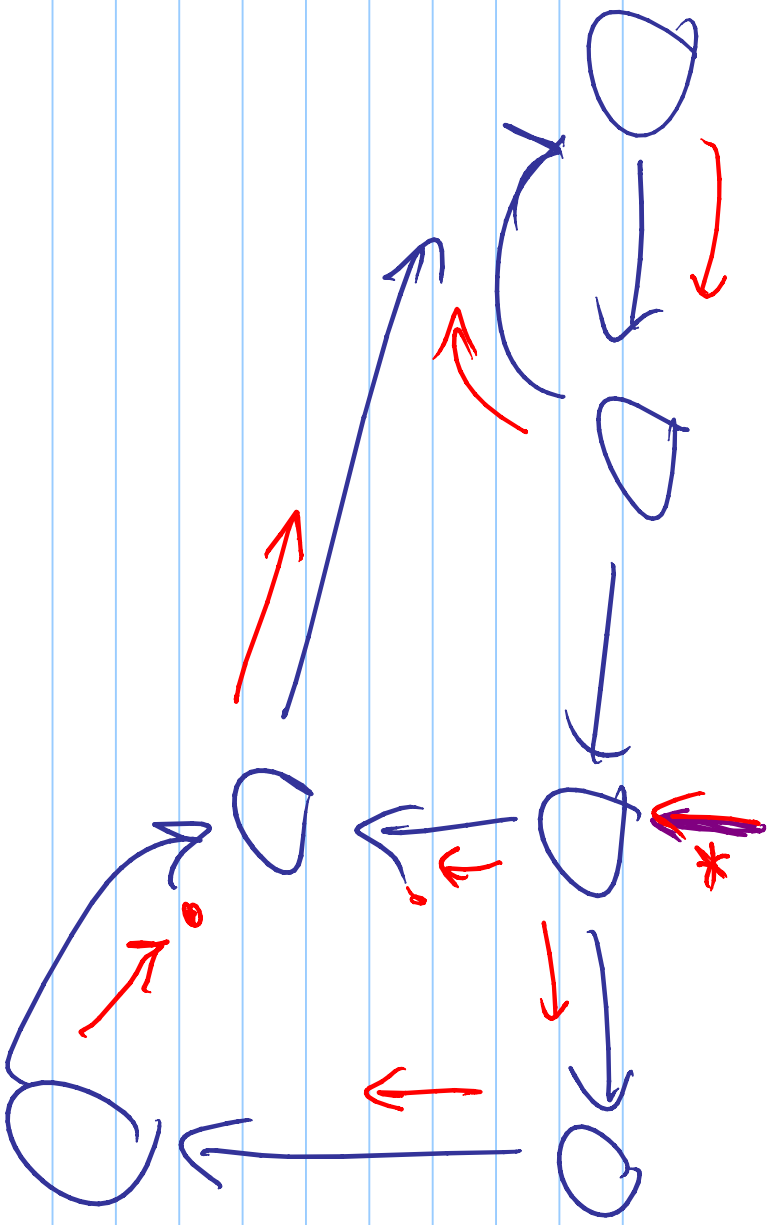
Marker msg algorithms

if process P_i receives a marker msg on channel C_j and this is the 1st time a marker was rcvd by P_i

→ record state of P_i

→ " " of C_j as null

→ send out marker msgs on all outgoing channels



if process P_i gets a marker on channel C_k → and this is not the 1st time for a marker to get to P_i

→ record the state of C_k as the sequence of msgs rcvd on C_k since the markers 1st visit to P_i

→ (no marker propagation)

End result

→ set of process states

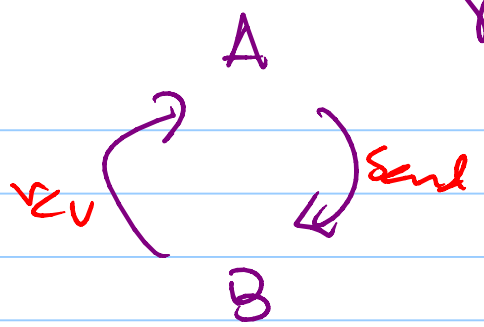
→ set of channel states

↳ of channels between processes in above

↳ property → recording happens @ 1 instance

2 processes P_1 P_2

State diagram



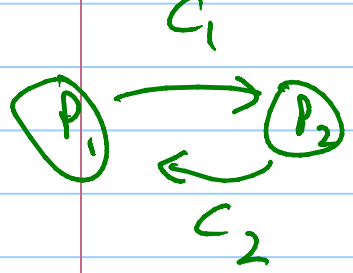
P_1 ∞ loop

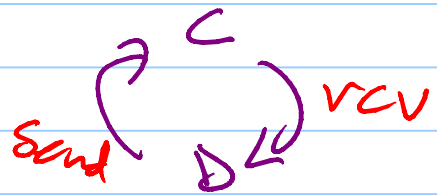
A

send (C_1, msg)

B

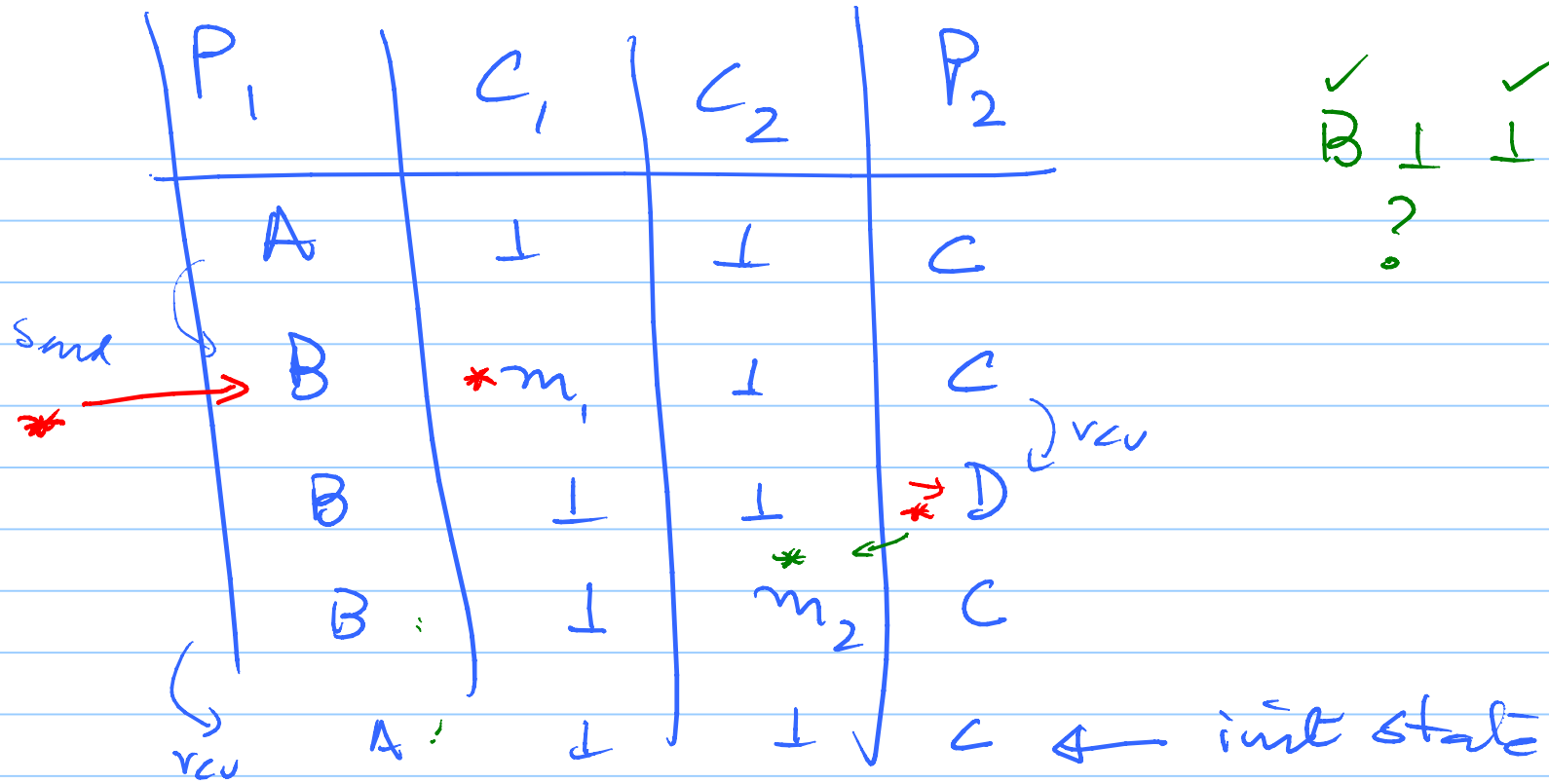
rcv (C_2, msg)





P_2 ∞ loop → C rcv (C_1, msg)

D send (C_2, msg)



B 1 1 D
?

P_1

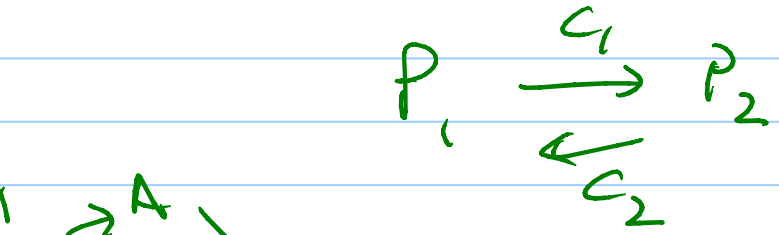
A: Send (c_1, m)

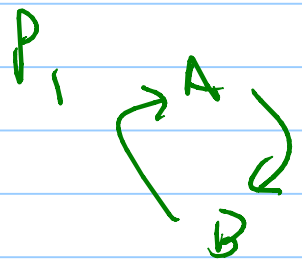
B: rcv (c_2, m)

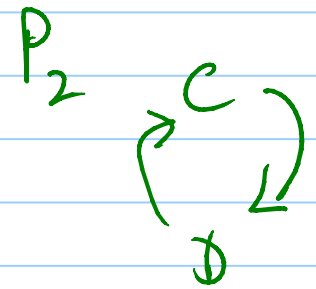
P_2

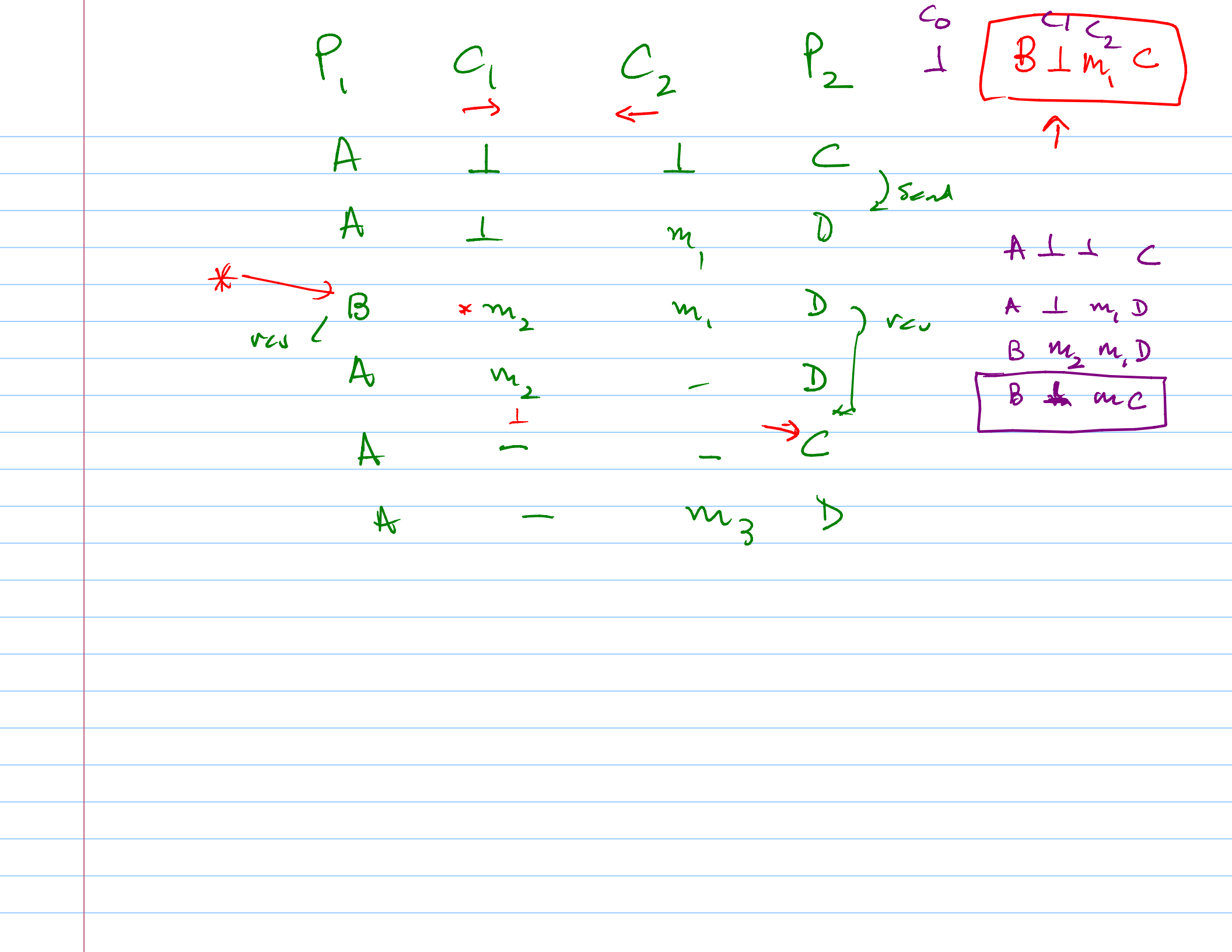
C: Send (c_2, m)

D: rcv (c_1, m)







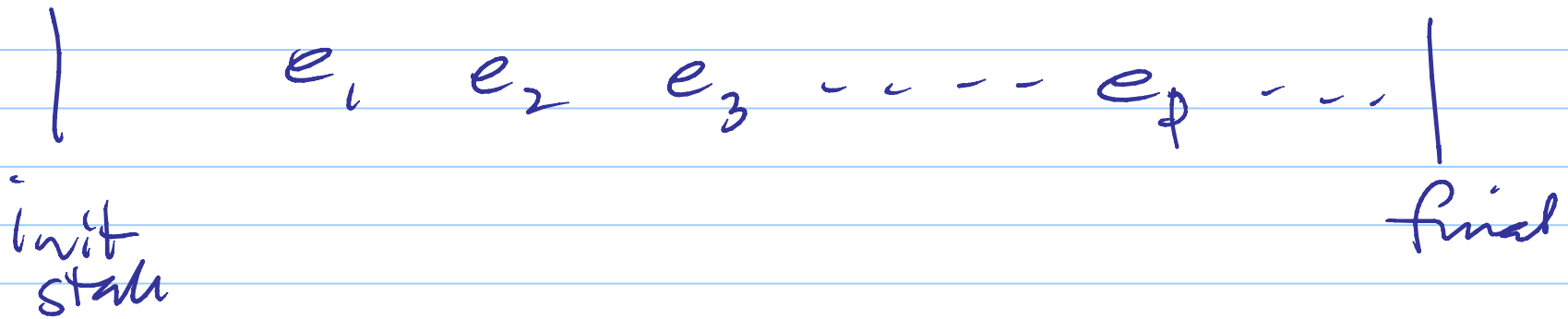

P1
C1
C2
P2
C0
1
C1 C2
B ⊥ m1 C
A ⊥ ⊥ C
send
A ⊥ m1 D
B m2 m1 D
B ⊥ m C
A ⊥ ⊥ D
B ⊥ m2 m1 D
rcv
A m2 ⊥ D
A ⊥ — C
A — m3 D
rcv

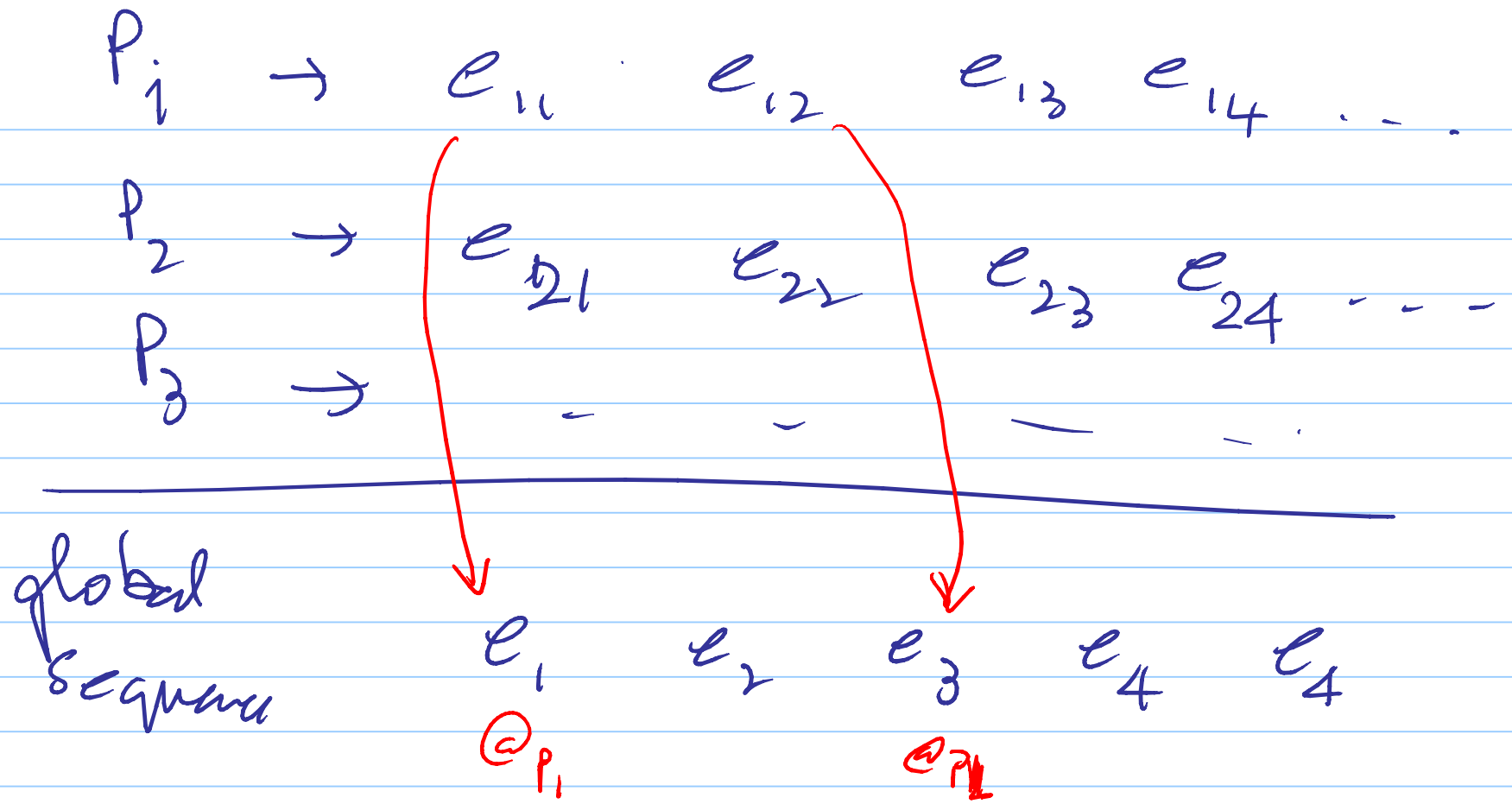
$P_1 \quad P_2 \quad \cdots \quad P_n$

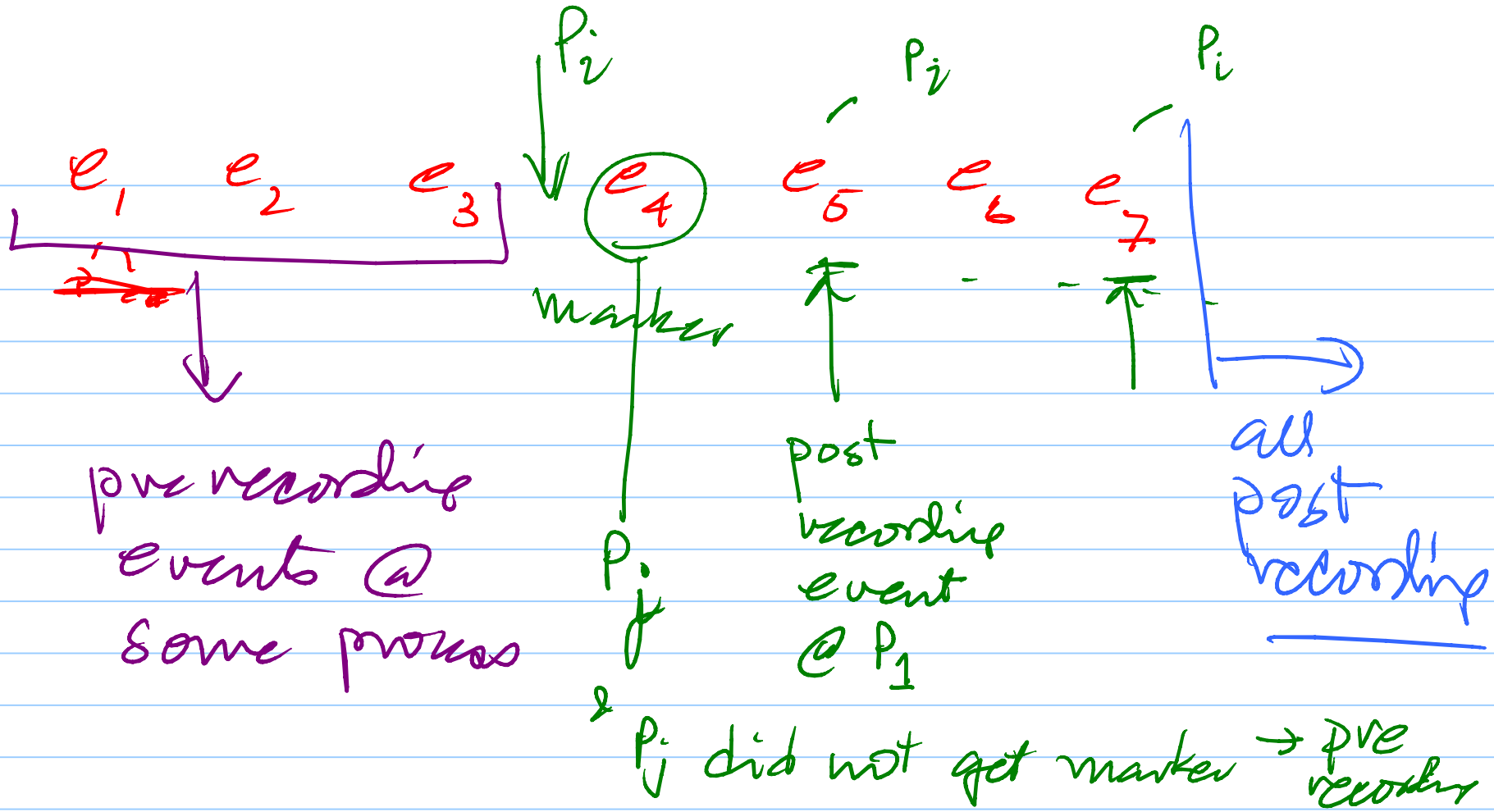
events happen

events cause state changes

events




$P_1 \rightarrow e_{11}\ e_{12}\ e_{13}\ e_{14}\ \ldots$
$P_2 \rightarrow e_{21}\ e_{22}\ e_{23}\ e_{24}\ \ldots$
$P_3 \rightarrow$
global sequence
$e_1\ e_2\ e_3\ e_4\ e_4$
@P_1
@P_2



& P_j did not get marker → pre recording