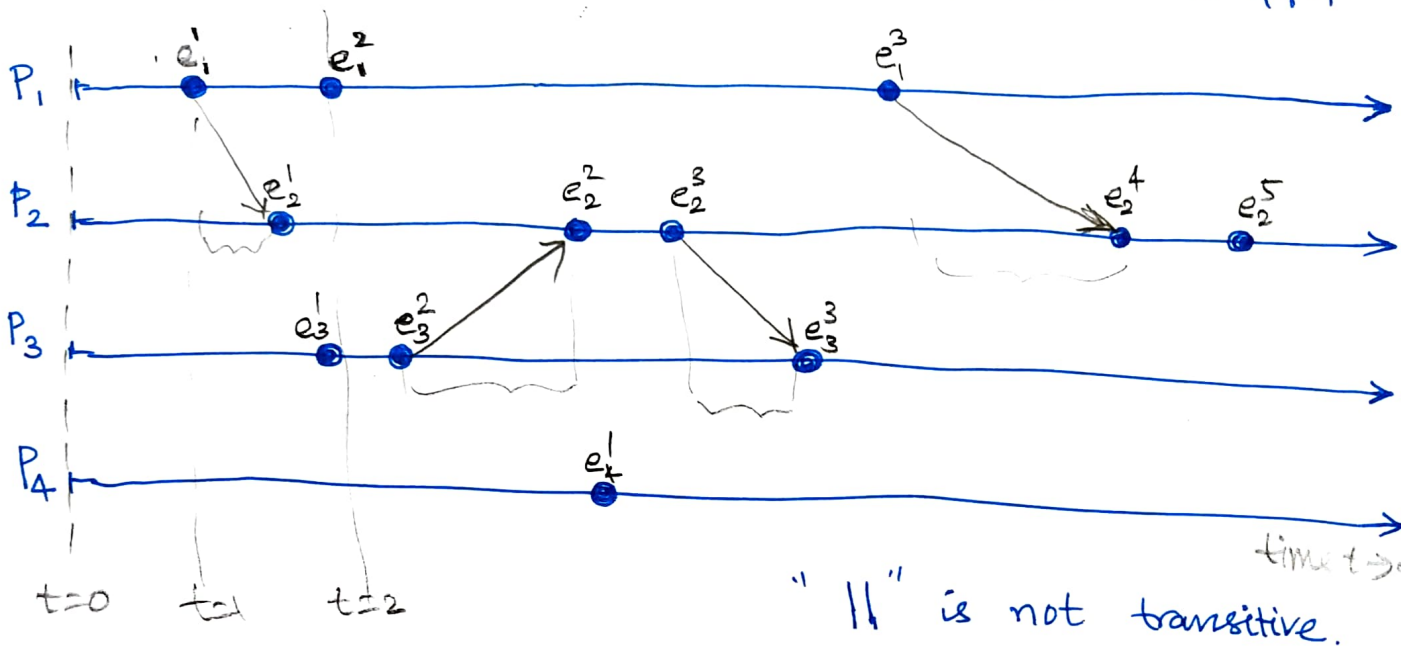
$n = 4$ Processes

$e_i \rightarrow e_j$ (OR) $e_j \rightarrow e_i \Rightarrow e_i \| e_j$



" || " is not transitive.

Causal Events:

e_i → "happened before" → e_j

e_i "happened before" e_j → on the (Same) / (across) / (Combination of P_i & P_j & P_k)

$e_i \not\rightarrow e_j$

↳ does not causally affect

Concurrent Events:

Two events: e_i & e_j

if $e_i \not\rightarrow e_j$ and $e_j \not\rightarrow e_i$

Then e_i & e_j are said to be Concurrent)

$(e_i \| e_j)$

↳ Concurrent Events.

Causal Events:

① $e_1^1 \rightarrow e_1^2$

② $e_1^1 \rightarrow e_2^2$

③ $e_3^1 \rightarrow e_2^3$

④ $e_1^2 \rightarrow e_2^4$

Concurrent Events:

① $e_1^1 \parallel e_3^1$

② $e_3^1 \not\rightarrow e_4^1$ & $e_4^1 \not\rightarrow e_3^1$
$\Rightarrow e_3^1 \parallel e_4^1$

③ $e_3^3 \parallel e_2^5$

④ $e_1^2 \parallel e_3^1$
e_4^1

Logical vs Physical Concurrency

Send $\xrightarrow{msg}$ Receive

FIFO:

$\rightarrow \ldots msg_2, msg_1, \longrightarrow$ FIFo

Non-FIFO:

c_{ij}

$m_4(20), m_3(1), m_2(10), m_1(5) \longrightarrow$

P_i ↑ P_j

(max/min) Priority Queue

m_4 m_3

CO: Any two messages: m_{ij} and m_{kj}

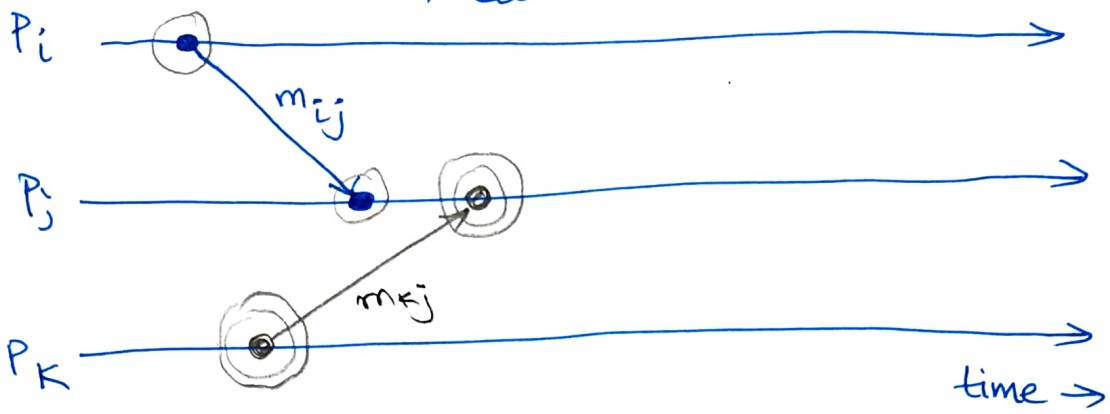
↓ "happens before"

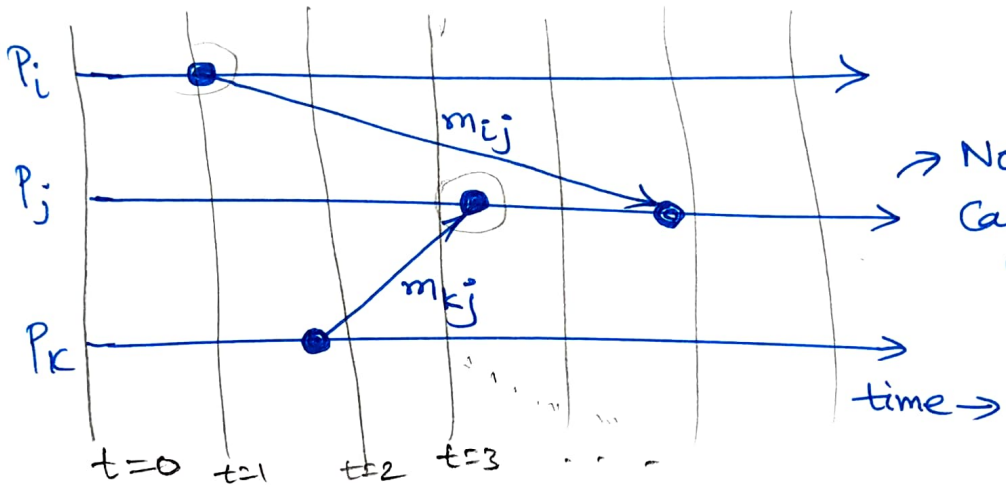
if send(m_{ij}) → send(m_{kj})

then Receive(m_{ij}) → receive(m_{kj})

$a \rightarrow g$, $k \rightarrow h$: $a \rightarrow k$ and $g \rightarrow h$
$\Rightarrow a \rightarrow g$ and $k \rightarrow h$ are CO.

↓"Causal ordering" (happened before)



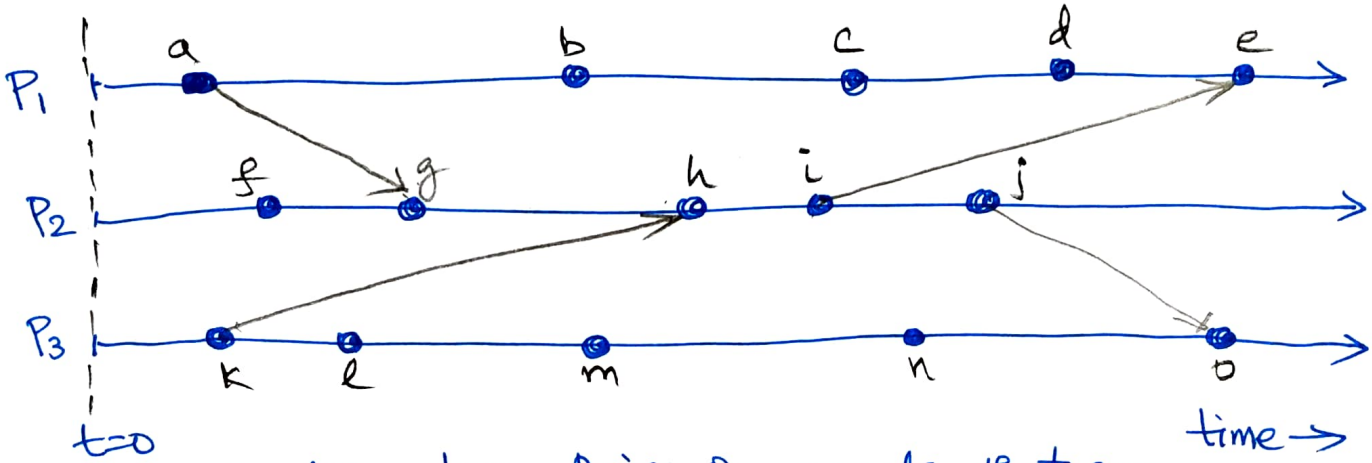


→ Not following Causal ordering

FIFO non-FIFO CO

CO ⊂ FIFO ⊂ non-FIFO

State-time diagram



Identify two pairs of events that are following CO / two non-CO ordering events.