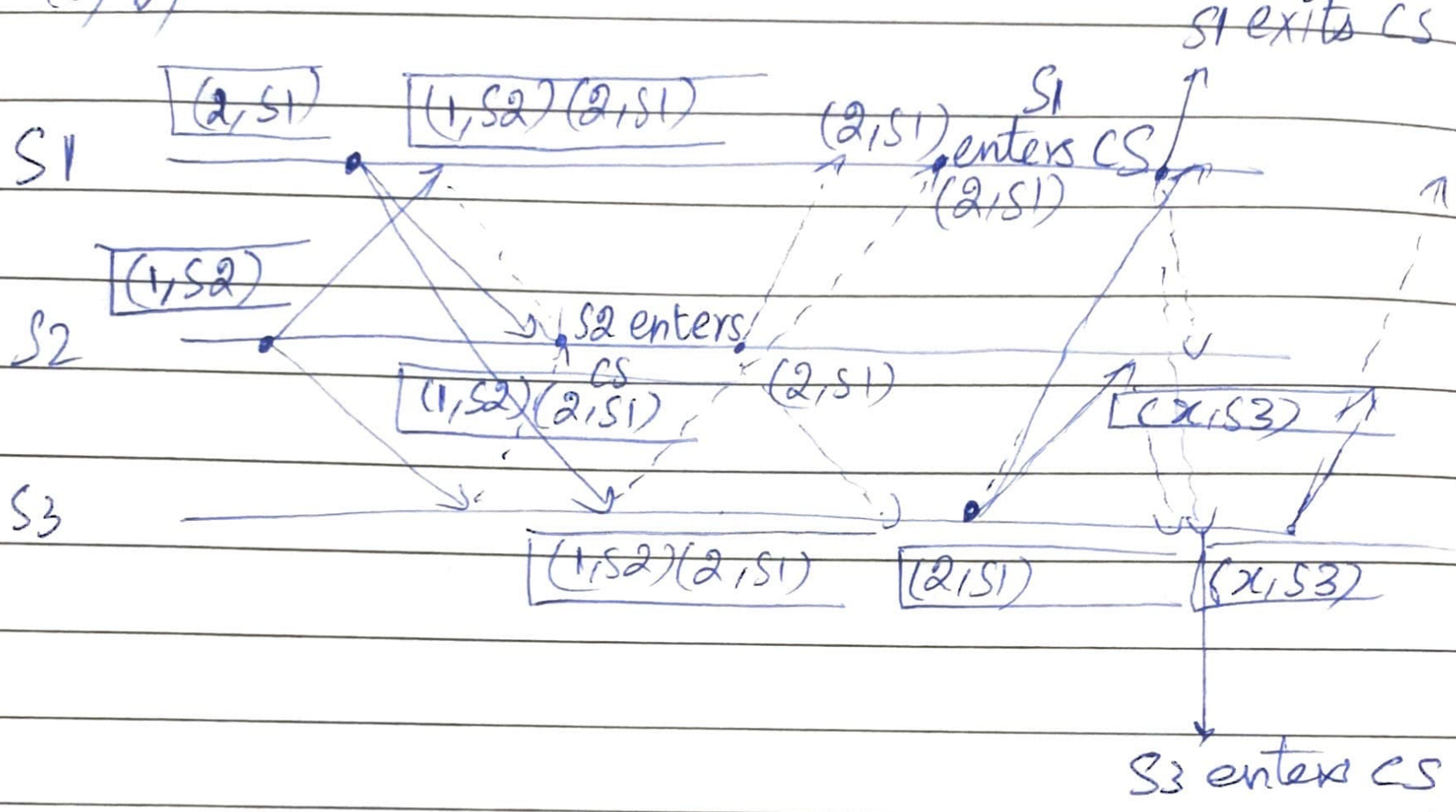
10) b)


S1 exits CS
(2,S1)
(1,S2)(2,S1)
S1
(2,S1)
S1 enters CS
(2,S1)
(1,S2)
S2
S2 enters CS
(2,S1)
(1,S2)(2,S1)
(2,S1)
(x,S3)
S3
(1,S2)(2,S1)
(2,S1)
(x,S3)
S3 enters CS

How CS will be executed:

A site S_i will enter CS if

i) S_i receives request with timestamps larger than itself.

ii) S_i's request is at the top of request-queue.

So

S_2 will execute when S_1's request timestamp is greater. It already satisfies L2 because S_1, S_2 are the first ones wanting to execute the CS.

⇒ Once S_2 is done it will send release msg to other two sites.

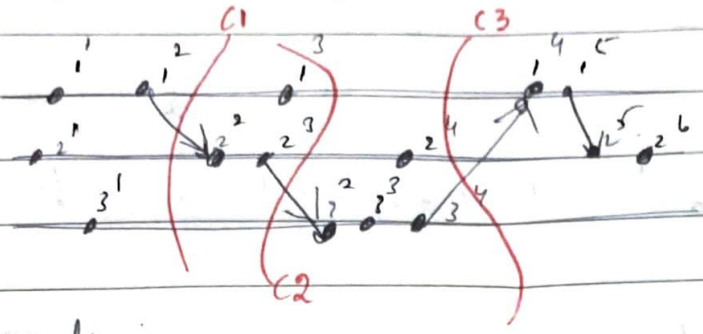
Even in this case, S_1 has requested CS before S_3, so timestamp of S_1 is smaller ⇒ S_2 is executed

Finally after S_1 is done, it sends RELEASE msg. S_3 gets to enter the CS.

Order of execution

S_2, S_1, S_3

6. (ii)



a) Causal dependencies

$e_1^1 \rightarrow e_3^3$; $e_3^3 \rightarrow e_2^6$, $e_3^1 \rightarrow e_2^6$

$e_1^3 \rightarrow e_2^6$, $e_1^1 \rightarrow e_2^4$

Causal precedence:

$\forall e_i^x, \forall e_j^y \in H,$

$e_i^x \rightarrow e_j^y \Leftrightarrow$

$e_i^x \rightarrow_i e_j^y$ (es) $(i=j) \wedge (x<y)$

$e_i^x \rightarrow_{msg} e_j^y$

$\exists e_k^z \in H ; e_i^x \rightarrow e_k^z \wedge e_k^z \rightarrow e_j^y$

Causal precedence:

all 5 satisfies, all 4 messages, events within same process

Concurrent relations

$e_1^3 \parallel e_3^3$, $e_2^4 \parallel e_3^1$, $e_2^4 \parallel e_3^3$

$e_2^4 \parallel e_1^5$, $e_2^4 \parallel e_1^4$, $e_1^1 \parallel e_2^1$,

$e_1^1 \parallel e_3^1$, $e_2^1 \parallel e_3^1$, $e_1^3 \parallel e_2^3$

b) Consistent global state

$\{ e_1^2 \; e_2^3 \; e_3^3 \text{ or } e_3^2 \}$ — Send ✓ Recv ✓

Sample global state

$\{ LS_1^2, LS_2^3, LS_3^2 \}$

$\{ LS_1^2, LS_2^3, LS_3^3 \}$

$\{ LS_1^5, LS_2^6, LS_3^4 \}$ $\{ LS_1^5, LS_2^5, LS_3^4 \}$

So for causal precedence, all 5 causal dependencies satisfy third condition.

By first condition: all events within a process

By second condition: all events $\longrightarrow$ msg

satisfy causal precedence.