Hand in 2

$$(\mathsf{P}_j, m_j) \| (\mathsf{P}_i, m_i) \ .$$

Exercise 4.3 (Causal Past) Recall the series of events we considered in Example 4.3

- P_1 sends message a. (send-event (P_1, a) happens).
- P_3 sends message b.
- Message (P_1, a) arrives at P_2.
- P_2 sends message c.
- Message (P_3, b) arrives at P_2.
- Message (P_2, c) arrives at P_3.
- P_3 sends message d, which arrives at P_2. 2.

Assume players use the above protocol based on vector clocks to communicate. For each send-event, specify the vector clock that accompanies the message that is sent. There are 4 messages sent, and hence 6 pairs of messages. For each such pair, use the vector clocks associated to the messages to determine if the messages in the pair are concurrent, or the pair is in the causal past relation.

Example 4.3 Let us look at a small concrete example. Suppose the following events happen:

- P_1 sends message a (send-event (P_1, a) happens at P_1).
- P_3 sends message b (send-event (P_3, b) happens at P_3.
- Message (P_1, a) arrives at P_2. (receive-event (P_1, a) happens at P_2)
- P_2 sends message c to P_3.
- Message (P_3, b) arrives at P_2 .
- Message (P_2, c) arrives at P_3.
- P_3 sends message d, which arrives at P_2.

See Fig. 4.10. In this example, we have 4 send-events. First, (P_1, a) and (P_3, b) are independent, the two involved parties have not talked to each other yet, so there is no way a could depend on b or vice versa. Second, we have $(\mathsf{P}_1, a) \hookrightarrow (\mathsf{P}_2, c)$ since P_2 gets a before it sends c. And finally, (P_3, b) and (P_2, c) are independent, because P_2 produces c before it receives b. △

definition, as we only have access to the communicat

03

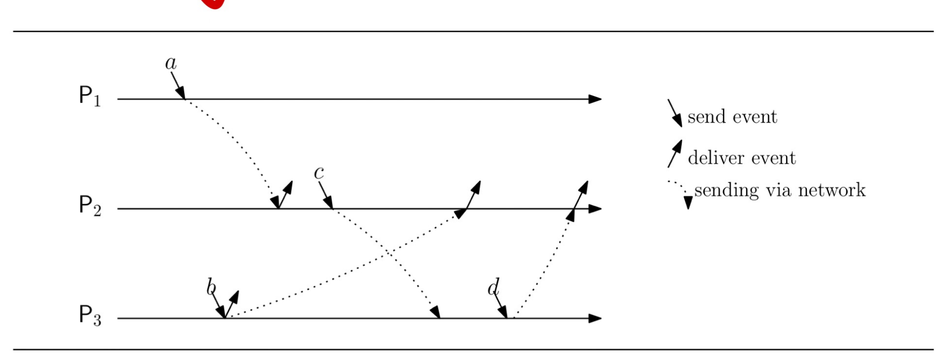

Figure 4.10 Causal Past Example

concurrent = C
casual past = CP

		a (1,0,0)	b (0,0,1)	c (1,1,0)	d (1,1,2)
(1,0,0)	a		C	CP	CP
(0,0,1)	b			C	CP
(1,1,0)	c				CP
(1,1,2)	d