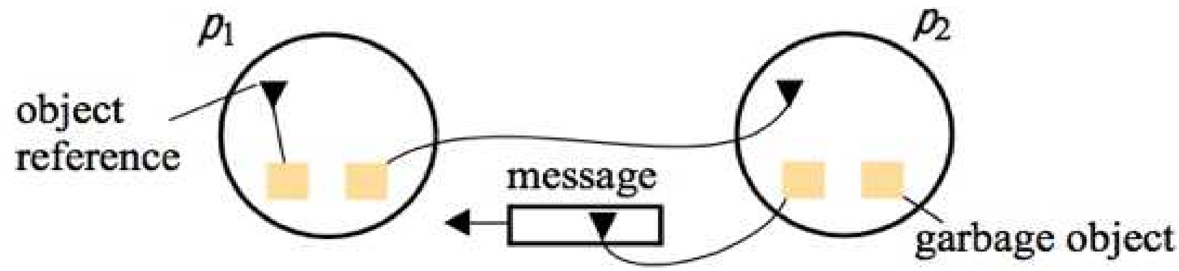
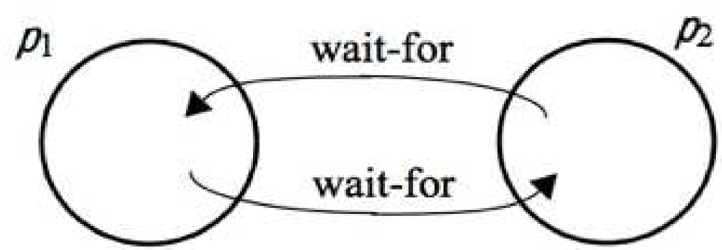
Global States

George Coulouris, Jean Dollimore and Tim Kindberg, "Distributed Systems Concepts and Design", Fifth Edition, Pearson Education, 2012

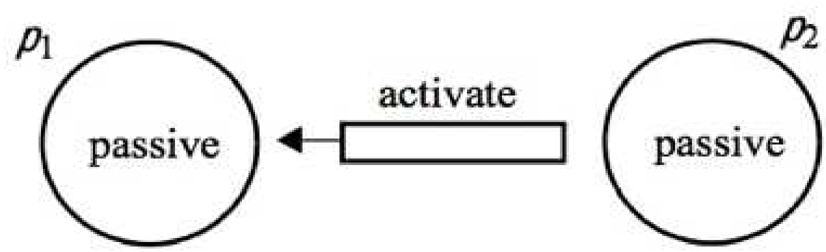
Detecting global properties



(a) Garbage collection



(b) Deadlock



(c) Termination

Local State

- **Local state LSi** of a site (process) Si is an assignment of values to variables of Si
- sending **send(mij)** and receiving **rec(mij)** of message **mij** from Si to Sj may influence LSi

we denote

- time(send(mij) or rec(mij)) is the time (physical or point in the computation) at which the send or receive occurs
- **time(LSi)** time the local state of Si was recorded .
- To aid the reasoning we consider the messages sent/received by the site as belonging to local state.
  - $send(m_{ij}) \in LS_i$ iff $time(send(m_{ij})) < time(LS_i)$
  - $rec(m_{ij}) \in LS_j$ iff $time(rec(m_{ij})) < time(LS_j)$.

Local State and Global State

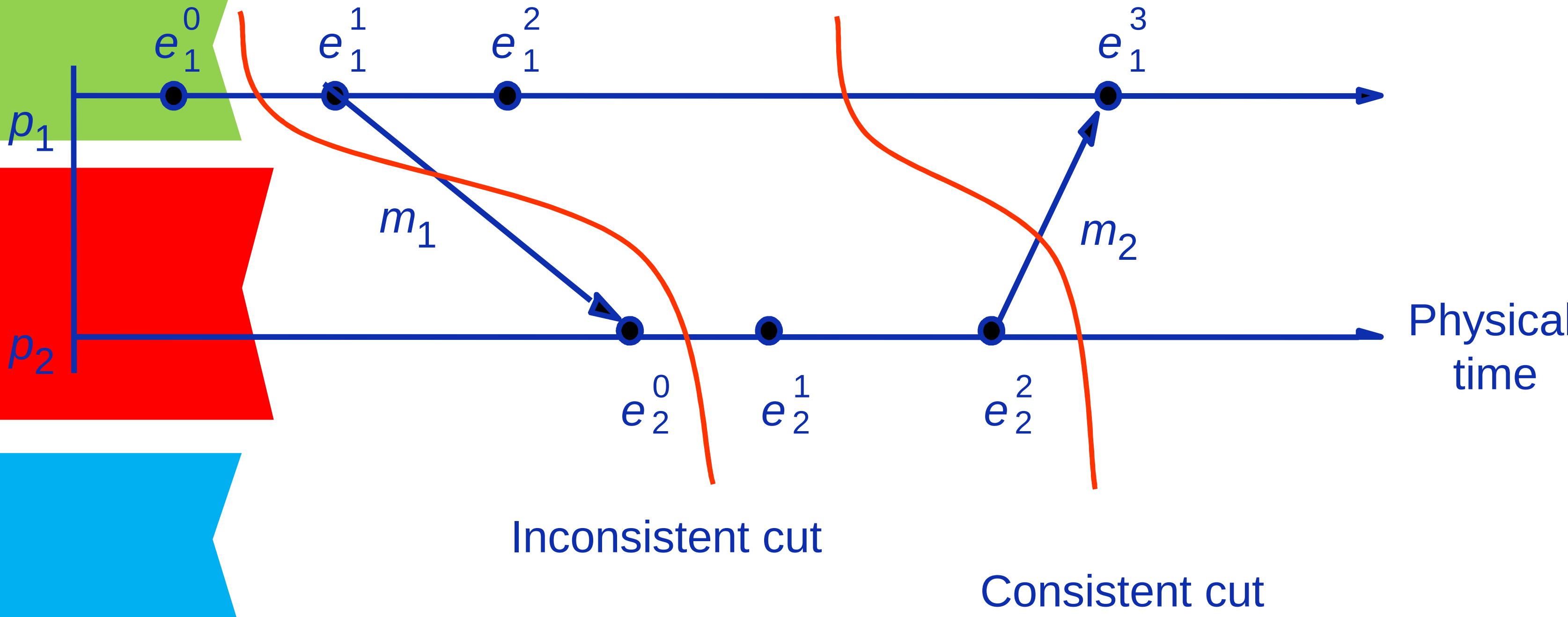
- The message is in **transit if it was sent but not received**
- The message is **inconsistent if it was received but never sent**

Transit: $transit(LS_i, LS_j) = \{m_{ij} \mid send(m_{ij}) \in LS_i \wedge rec(m_{ij}) \notin LS_j\}$

Inconsistent: $inconsistent(LS_i, LS_j) = \{m_{ij} \mid send(m_{ij}) \notin LS_i \wedge rec(m_{ij}) \in LS_j\}$

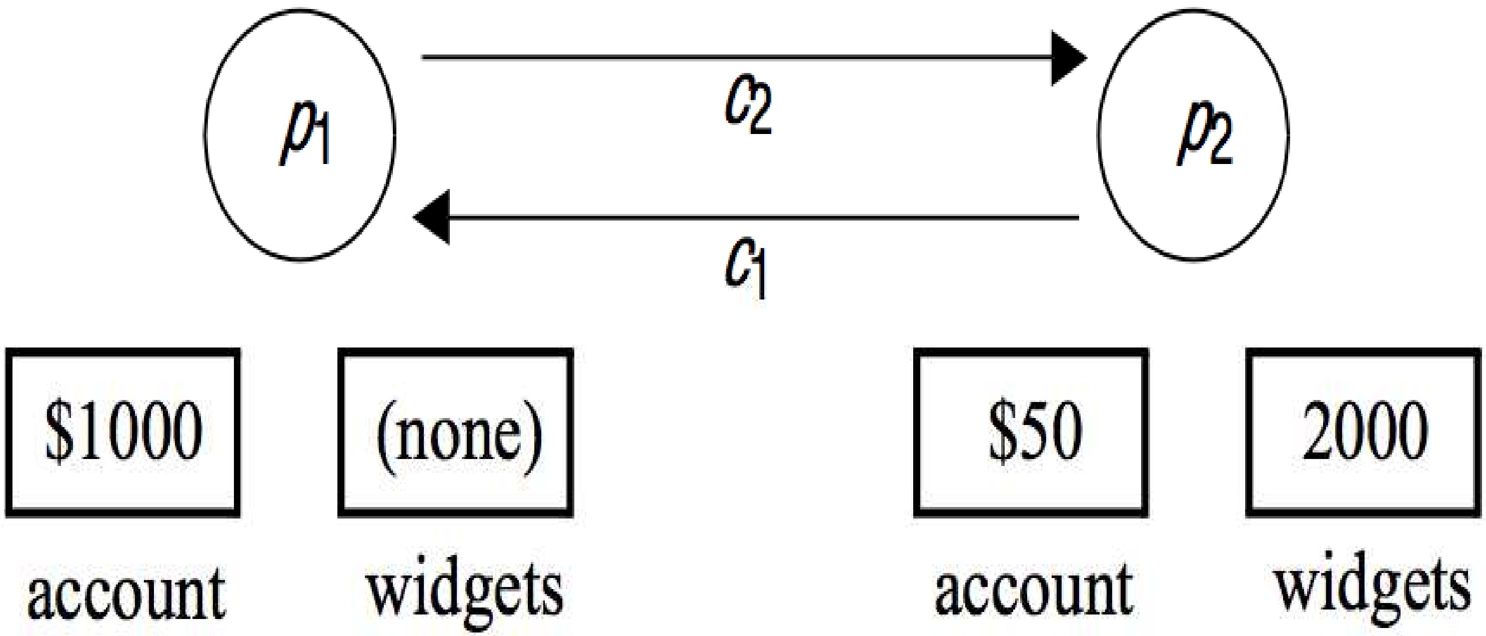
- **Global State**
  - **Global state is a collection of** local states of all sites and set of messages in the channels

Cuts

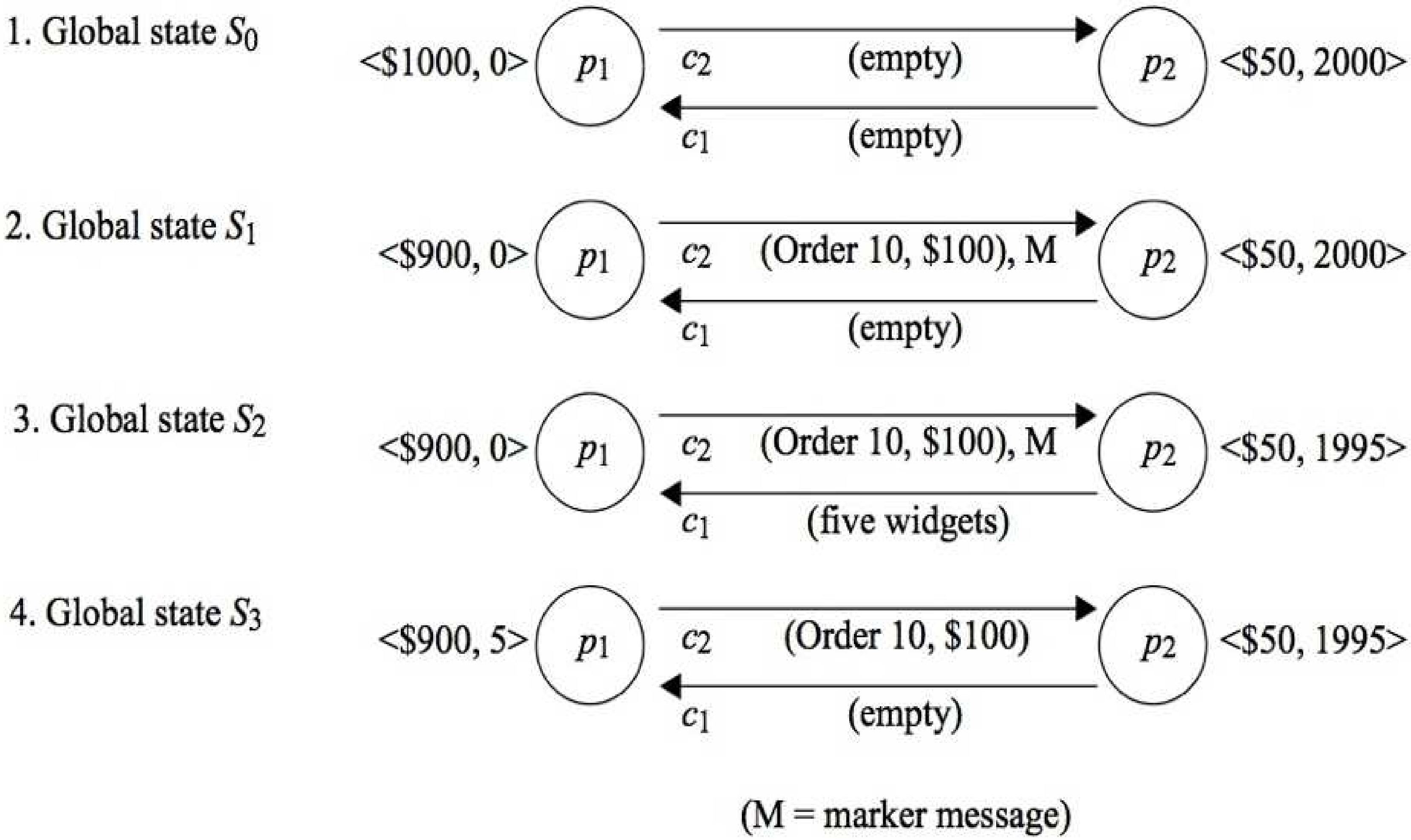


e_1^0
e_1^1
e_1^2
e_1^3
p_1
m_1
m_2
p_2
Physical time
e_2^0
e_2^1
e_2^2
Inconsistent cut
Consistent cut

Chandy and Lamport Snapshot example



Chandy and Lamport Snapshot example



Chandy and Lamport Snapshot example

- Process p1 records its state in the actual global state S0 , when the state of p1 is <$1000, 0>.
- Following the marker sending rule, process p1 then emits a marker message over its outgoing channel c2 before it sends the next application-level message: (Order 10, $100), over channel c2 . The system enters actual global state S1 .
- Before p2 receives the marker, it emits an application message (five widgets) over c1 in response to p1 's previous order, yielding a new actual global state S2 .
- Now process p1 receives p2 's message (five widgets), and p2 receives the marker.
- Following the marker receiving rule, p2 records its state as <$50, 1995> and that of channel c2 as the empty sequence. Following the marker sending rule, it sends a marker message over c1 .
- When process p1 receives p2 's marker message, it records the state of channel c1 as the single message (five widgets) that it received after it first recorded its state. The final actual global state is S3 .