Goals of the lecture

- Consistent State
- Algorithm
- Correctness
- Stable Properties

Reference: Chandy and Lamport

Case of the dubious dollars

picture taken here ($400)

Send \$100 *from A to B*

picture taken here ($400)

The total amount becomes $800

Problem

To determine global system state

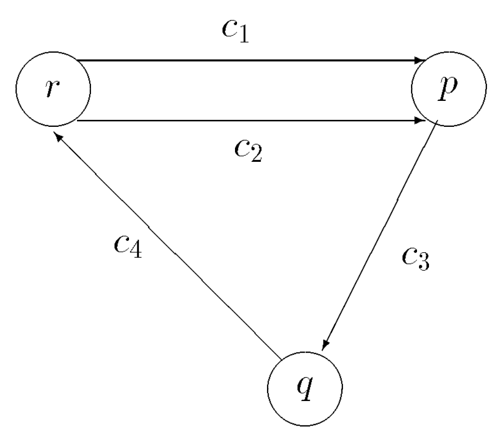
Each process can record its own state and messages it sends and receives.

No shared clock or memory

Analogy: group of photographers

Model of a Distributed System

- Finite set of processes
- Finite set of channels



Channel := FIFO, error free, and infinite buffer

Definition of a process

$$\text{Process } P = (S, I, E)$$

Set of States   Initial State   Set of Events

An event can change the state of P and at most one channel.

$$e = \langle\, p,\ s,\ s',\ m,\ c \,\rangle$$

process   pre-state   post-state   message   channel

Global state and global sequence

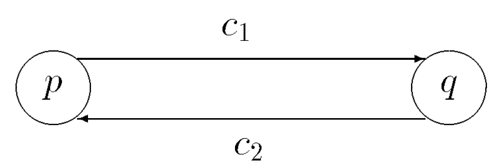
$$\text{state}(D) = \times_i\ \text{state}(p_i)\ \times\ \times_i\ \text{state}(c_j).$$

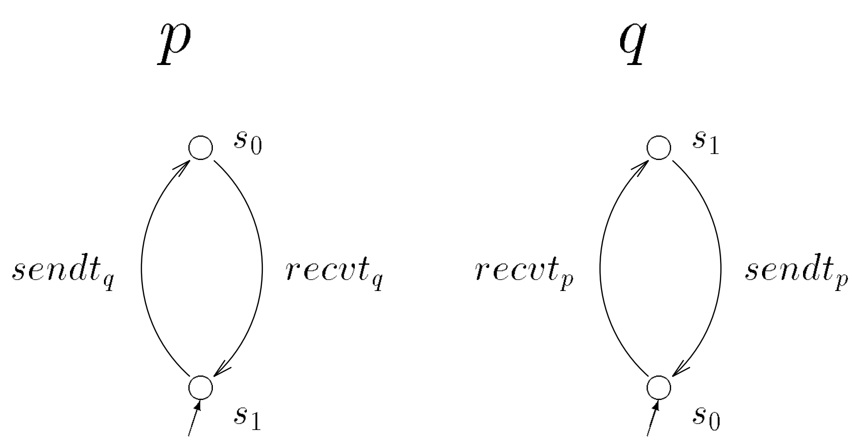
$$\text{next}(s, e) = \text{ global state immediately after } e$$

$$\text{seq } = (e_i\ :\ 0 \le i \le n) \text{ is a computation of } D \text{ iff}$$

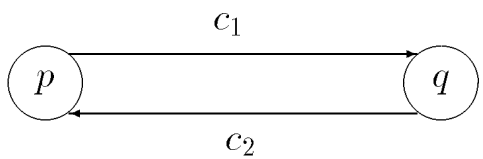
$$\begin{aligned} s_0 &= \text{ initial global state} \\ s_{i+1} &= \text{ next}(s_i, e_i) \quad 0 \le i \le n \end{aligned}$$

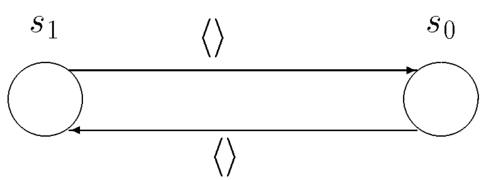
Example 1



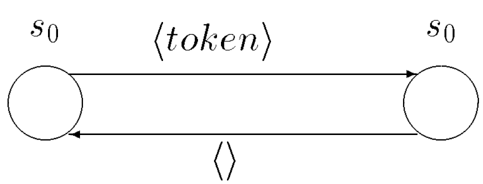


Example 1 [Contd.]





$$state(p) = s_1$$



$$state(c_1) = \langle token \rangle$$
$$state(c_2) = \langle\rangle$$
$$state(q) = s_0$$

Global State Detection Algorithm

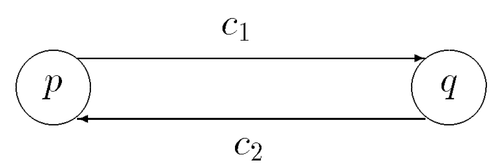
Sending Rule : For all channels c directed away from p, p sends one marker after p records its state and before it sends further messages along c.

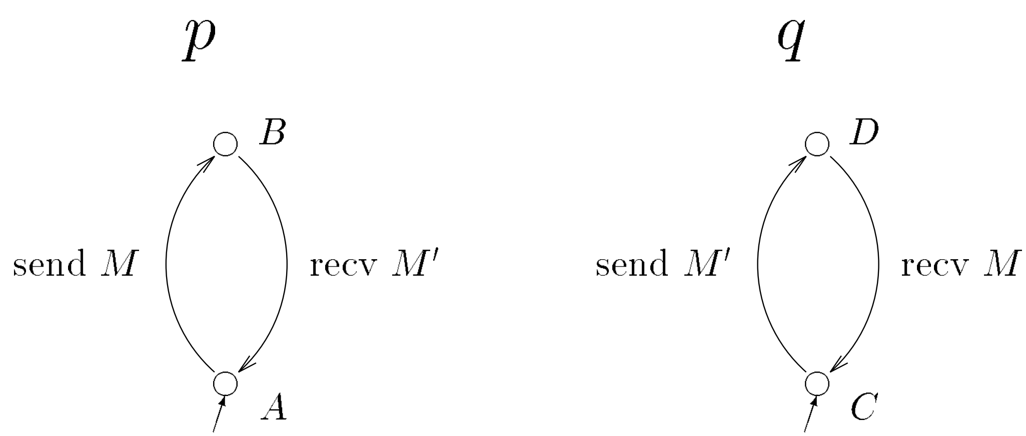
Receiving Rule : On receiving a marker along c if q has not recorded its state

$then$ records its state
marks c as empty

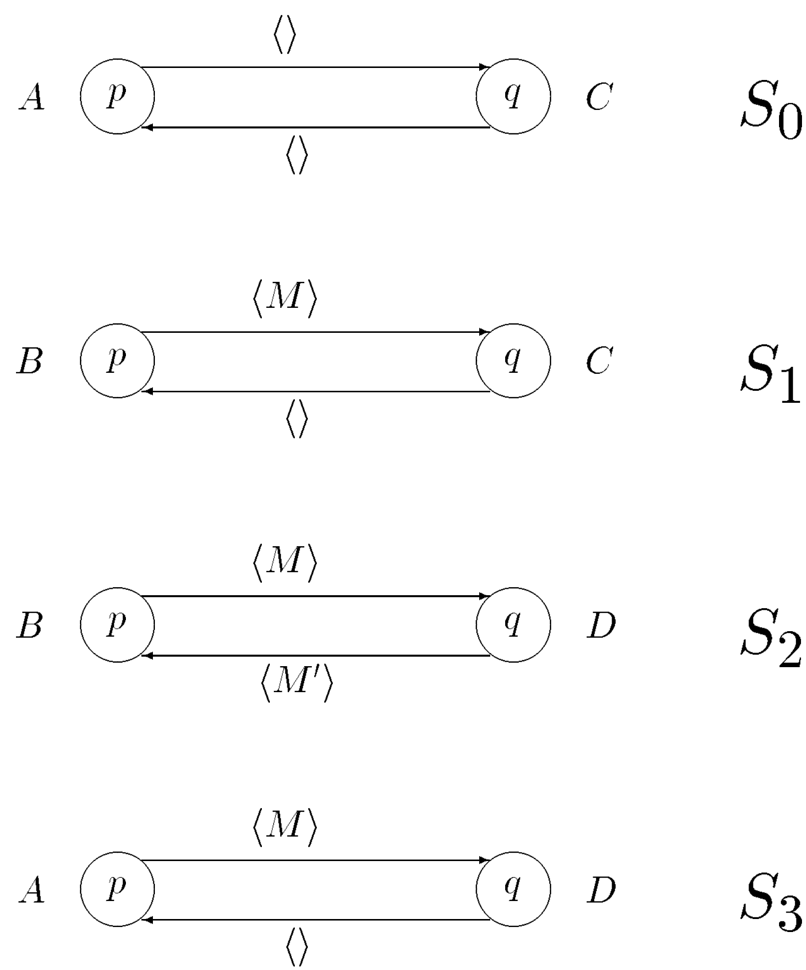
$else$ state(c) $= \langle$ seq of messages $\rangle$ received along c after the state was recorded and before marker is received.

Example 2

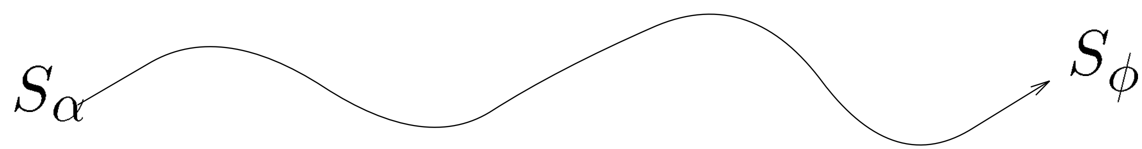




Example 2 [contd.]

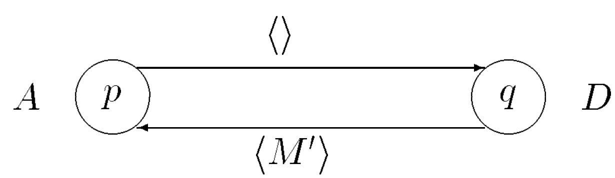


Property of the recorded global state



$S^* =$ snapshot

- S^* is reachable from S_α
- S_ϕ is reachable from S^*



Recorded global state (S^*)

Property of the recorded global state [Contd.]

Theorem 1 *There exists a computation* $seq' = (e'_i, 0 \leq i)$ *where*

1. *For all* i, *where* $i < \alpha$ *or* $i \geq \phi : e'_i = e_i$, *and*
2. *the subsequence* $(e'_i, \alpha \leq i < \phi)$ *is a permutation of the subsequence* $(e_i, \alpha \leq i < \phi)$, *and*
3. *for all* i *where* $i \leq \alpha$ *or* $i \geq \phi : S'_i = S_i$, *and*
4. *there exists some* $k, \alpha \leq k < \phi$, *such that* $S^* = S'_k$.

Colorful description (due to Dijkstra)

- Each machine, atomic action and message is either white or red

- S_0 ⇒ Snapshot (SS) ⇒ S_1
  white red

Color Assignment

Atomic Action : same color as the machine

Message : same color as the machine that sends it

Snapshot state SSS consists of

- state when it made the transition from white to red
- the sequence of white messages accepted by a red machine

Proof

Summary

- Beautiful paper
  Beautiful algorithm
- Example of generalization of a problem