Uncoordinated Checkpointing

The Global State Recording Algorithm

Cristian Solano

Virginia Tech

Introduction

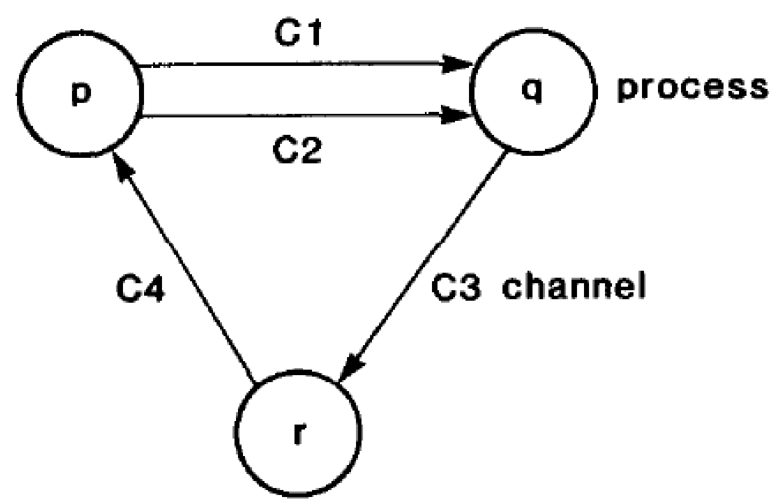
- We need an algorithm that allows a process in a Distributed System (DS) to determine a global state of the system during a computation.
- Processes in a DS communicate by sending and receiving messages.
- A process can record its own state and the messages it sends and receives; it can record nothing else.
- To determine a global system state, a process *p* must get the cooperation of other processes that record their own local states and send to *p*.

The Problem

- The problem is to devise algorithms by which processes record their own states and the states of communication channels so that the set of process and channel states recorded form a global system state.
- Many problems in Distributed Systems can be cast in terms of the problem of detecting global states.
- The GSD algorithm helps to solve an important class of problems: stable property detection.
- A property is “stable” if, once it holds in a state, it holds in all subsequent states.
- Examples of stable properties are: “Computation has terminated”, “System is deadlocked”, etc

The Model of a Distributed System

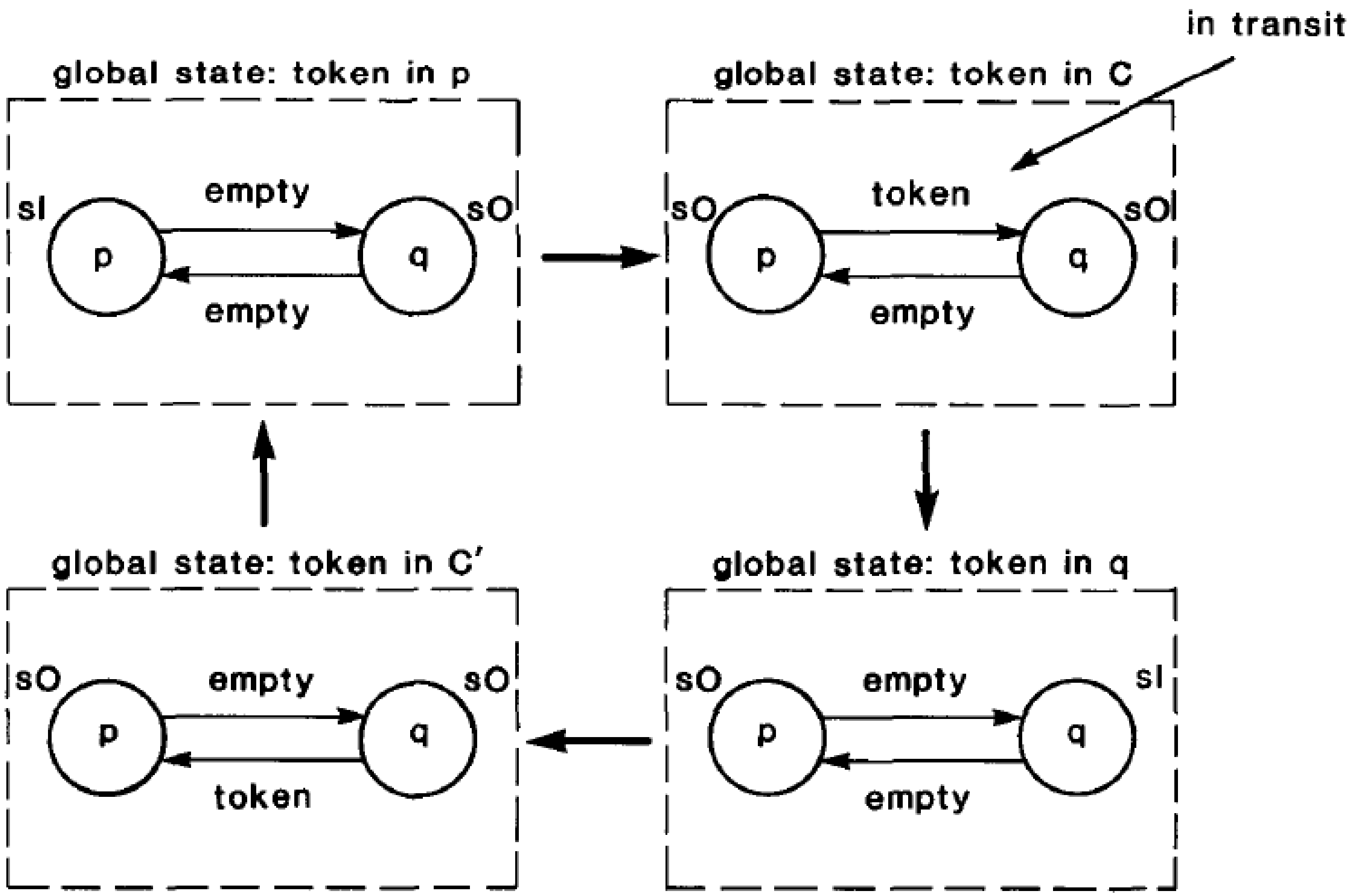
- A Distributed System consists of a finite set of processes and a finite set of channels.
- Nodes represent processes and the edges represent channels.
- Processes does not record their local states at the same time because we are assuming there is not share clocks or memory.
- Channels are assumed to have infinite buffers, to be error-free, unidirectional, loss free, FIFO and to deliver messages in the order sent.



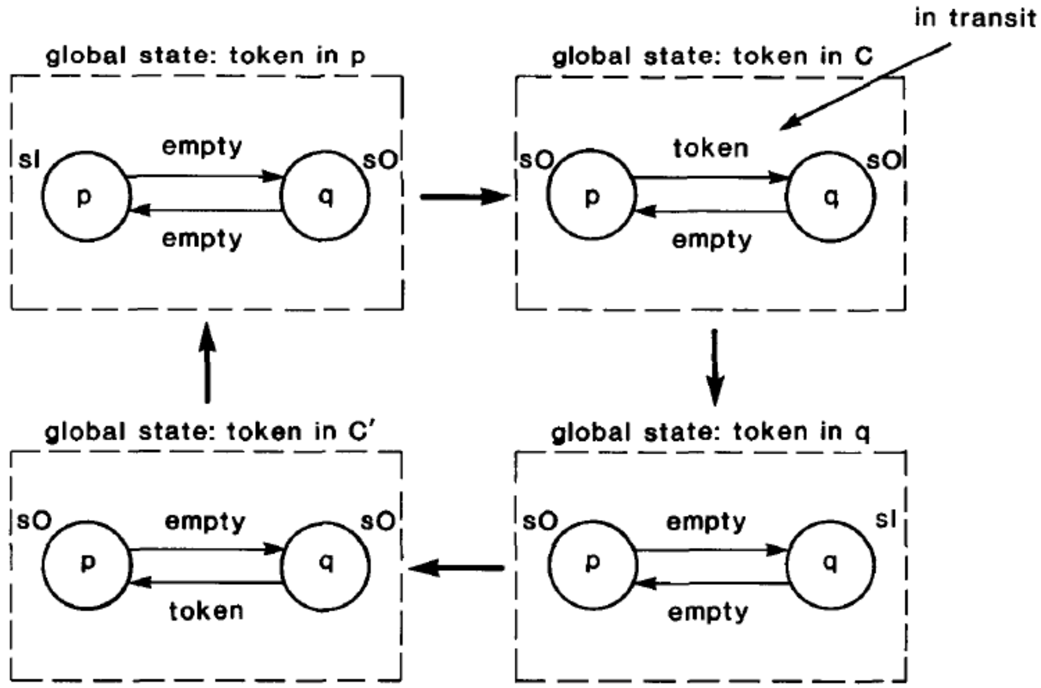
The Model of a Distributed System

- The state of a channel is a set of messages sent along the channel.
- A process is defined by a set of states, an initial state (from this set), and a set of events.
- An event *e* in a process *p* is an atomic action that may change the state of *p* itself and the state of at most one channel *c* incident on *p*.
- A global state of a distributed system is a set of component process and channel states: the initial global state is one in which the state of each process is its initial state and the state of each channel is the empty sequence.

The Model of a Distributed System



Motivation for the Algorithm



- Inconsistencies:
  - State of *p* is recorded before *p* sent a msg and the state of *c* is recorded after *p* sent the msg.
  - C is recorded before p sends a msg and the state of p is recorded after p sends a msg.

The Global State Recording Algorithm

- How it works?
  - Each process records its own state, and the two processes that a channel is incident on cooperate in recording the channel state.

- States of all processes and channels are not recorded at the same time because there is no global clock.

- The Global State Detection Algorithm is to be superimposed on the underlying computation: it must run concurrently with, but not alter, this underlying computation.

- Process p sends a special message, called market, after the nth message is sent along c. (and before sending further messages along c). The marker has no effect in the underlying computation.

The Global State Recording Algorithm

Marker-Sending Rule for a Process p:

for (each channel c, incident on, and directed away from p)

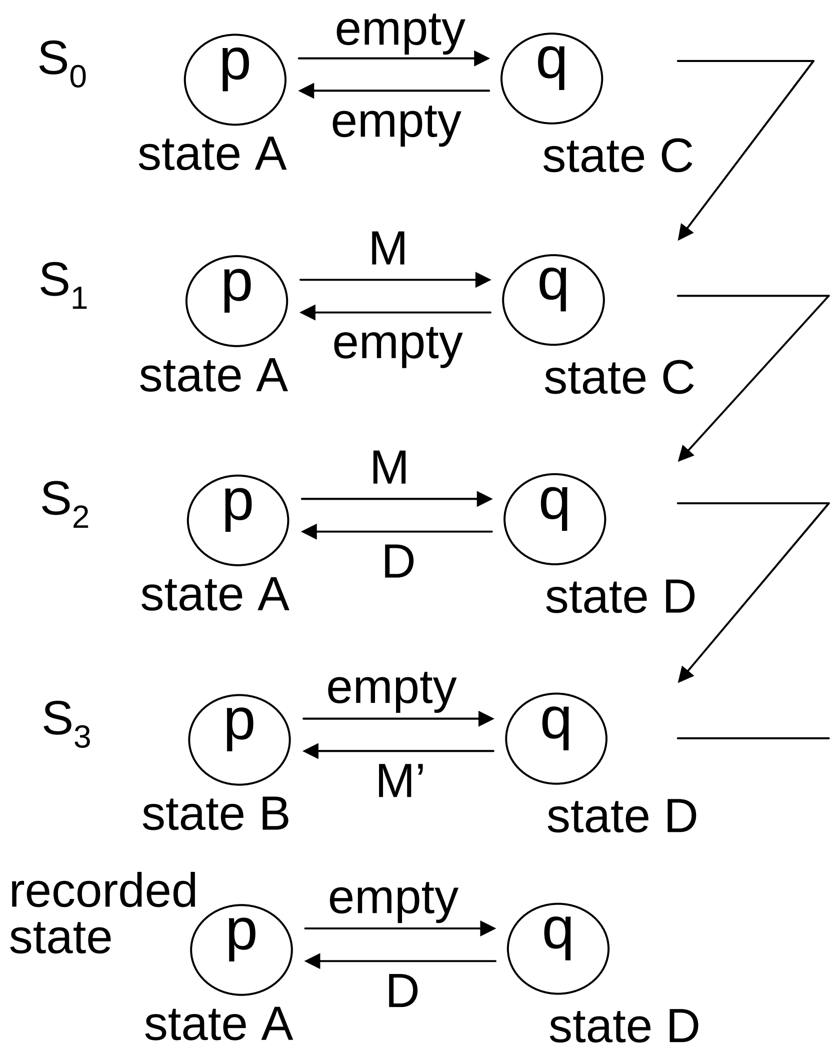
{ p sends one marker along c after p records its state and before p sends further messages along c; }

Marker-Receiving Rule for a Process q:

```
if (q has not recorded its state) then
        { q records its state;
          q records the state of c as the empty sequence;
          q sends marker messages  on all of its outgoing channels.
        }
 else  { q records the state of c as the sequence of message
           received along c after q's state was recorded and before
           q received the marker along c.
       }
```

- When a process receive a mark, it knows that a snapshot is in process.
- An individual node knows that it is done when it records its own state and all the states in my incoming channels.

Snapshot/State Recording Example 1



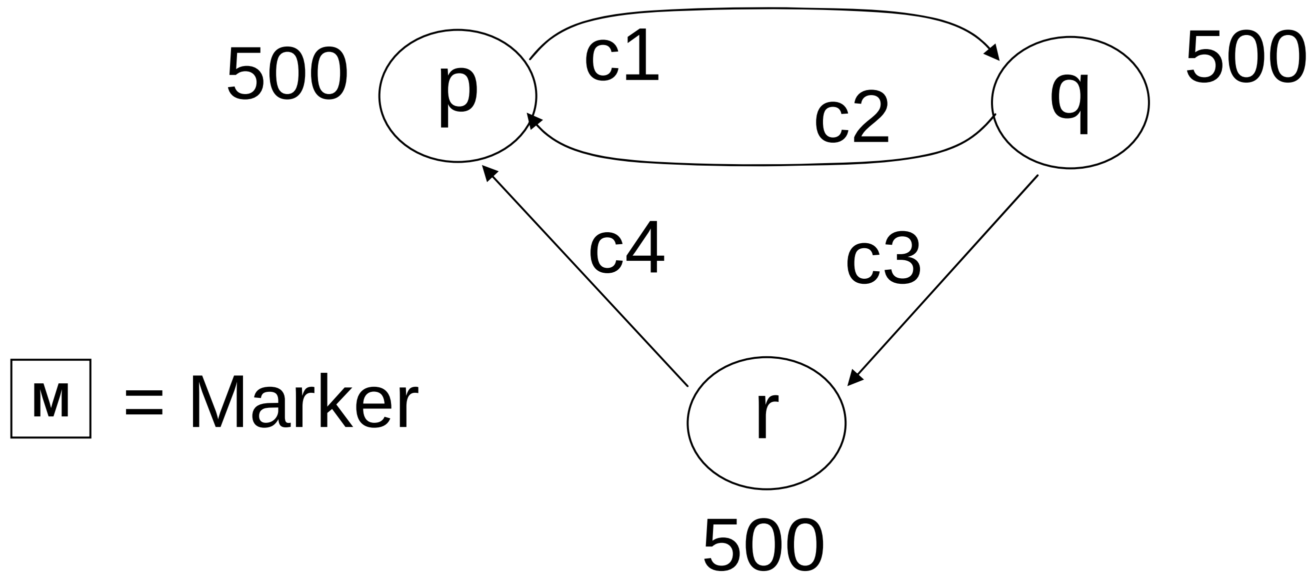
p records its state (A) and sends marker M on channel

before receiving the marker, q changes its state and sends message D.

q receives the marker and records its state (D) and the incoming channel as empty; q send marker M' on its outgoing channel.

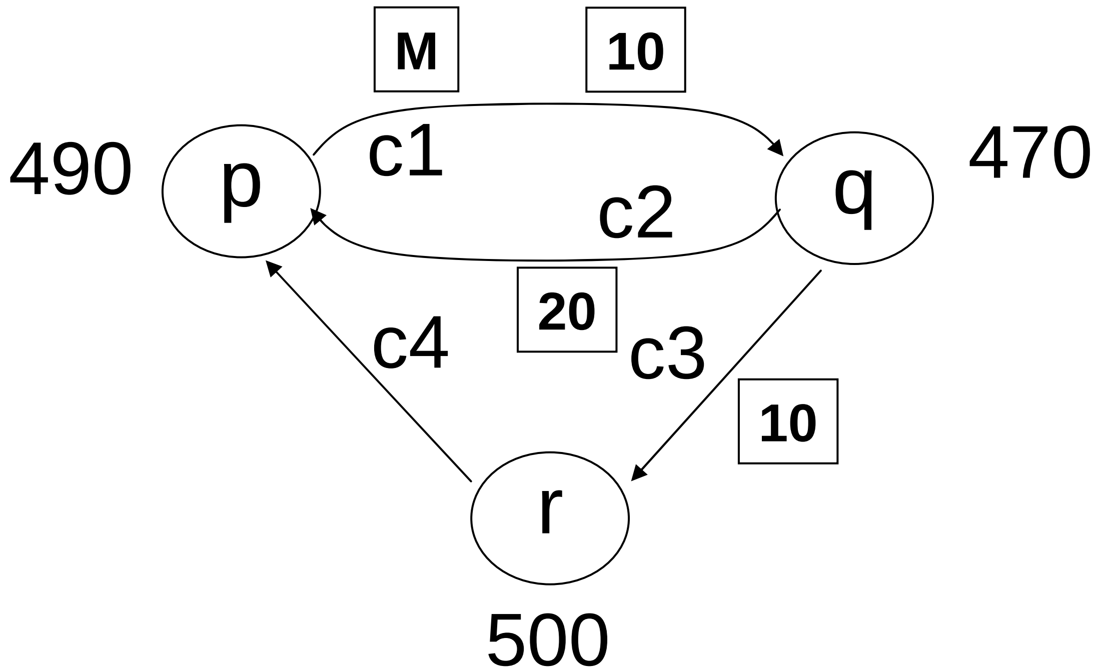
on receiving the marker, p records the channel as having message D

Snapshot/State Recording Example 2



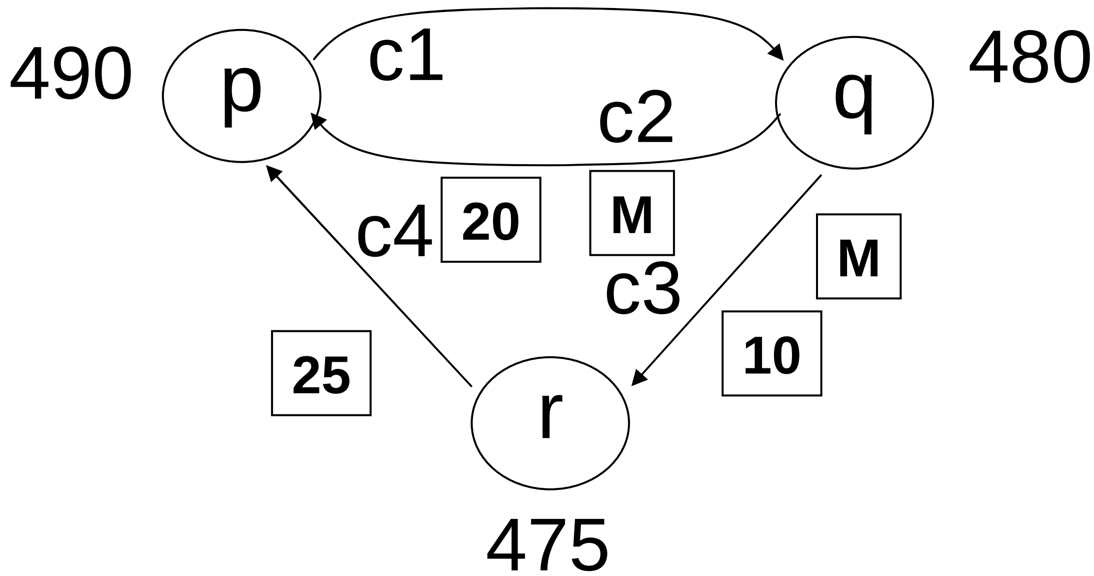
Node	Recorded state				
		c1	c2	c3	c4
p			{}		{}
q		{}			
r				{}	

Snapshot/State Recording Example 2 - (Step 1)



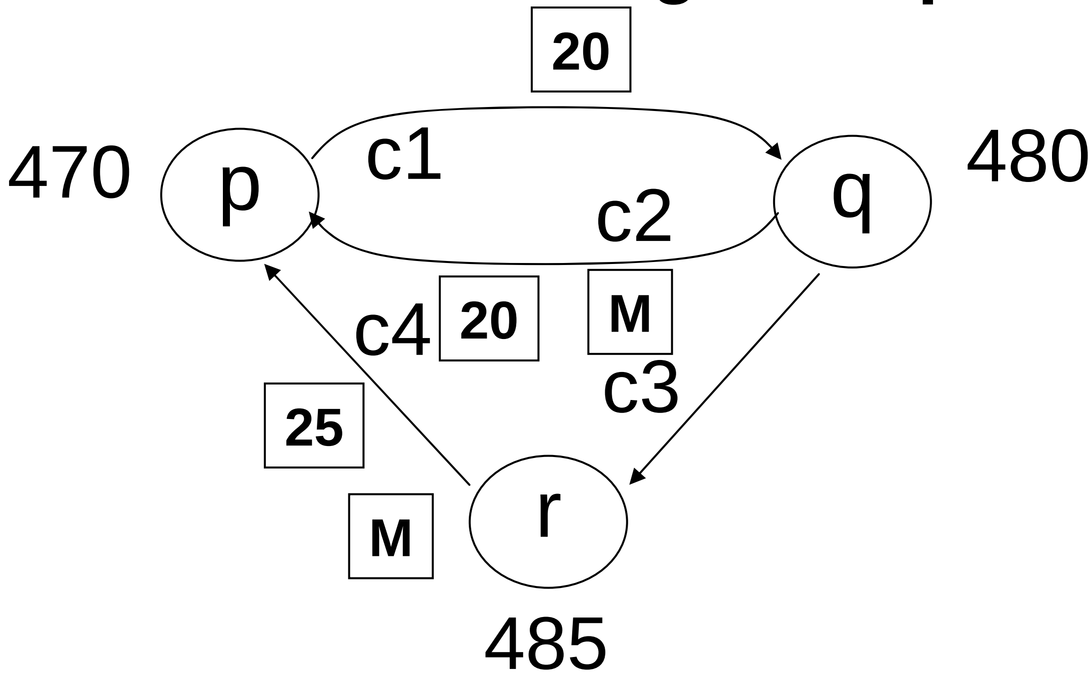
Node	Recorded state				
	state	c1	c2	c3	c4
p	490		{}		{}
q		{}			
r				{}	

Snapshot/State Recording Example 2 - (Step 2)



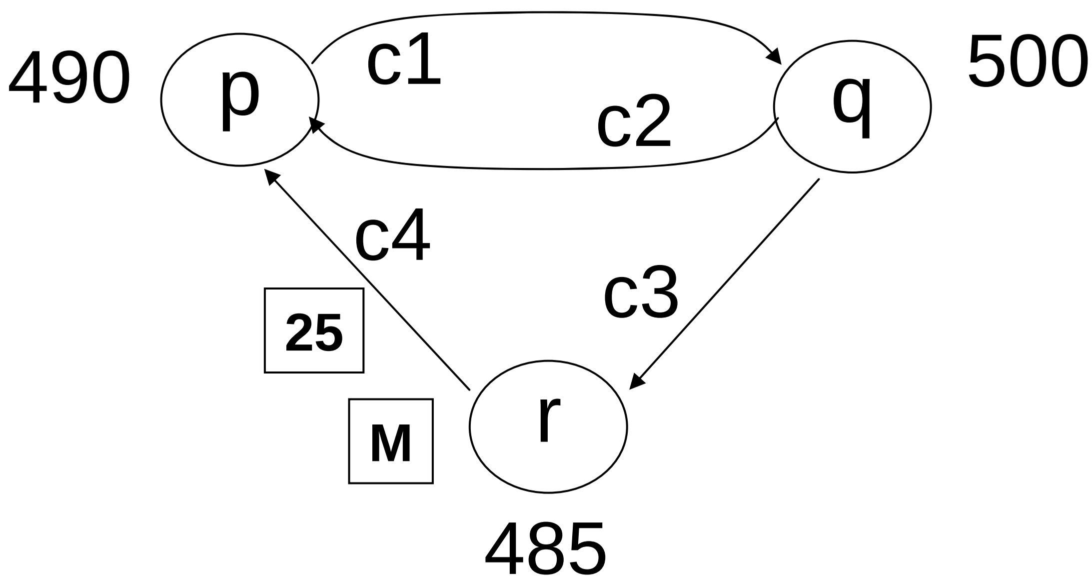
Node	Recorded state				
	state	c1	c2	c3	c4
p	490		{}		{}
q	480	{empty}			
r				{}	

Snapshot/State Recording Example 2 - (Step 3)



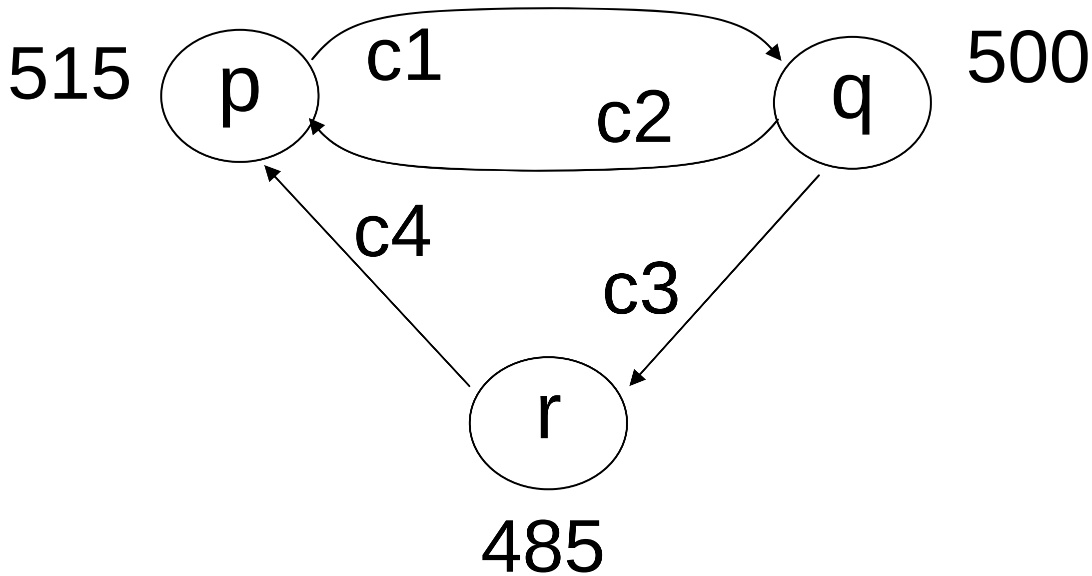
Node	Recorded state				
	state	c1	c2	c3	c4
p	490		{}		{}
q	480	{empty}			
r	485			{empty}	

Snapshot/State Recording Example 2 - (Step 4)



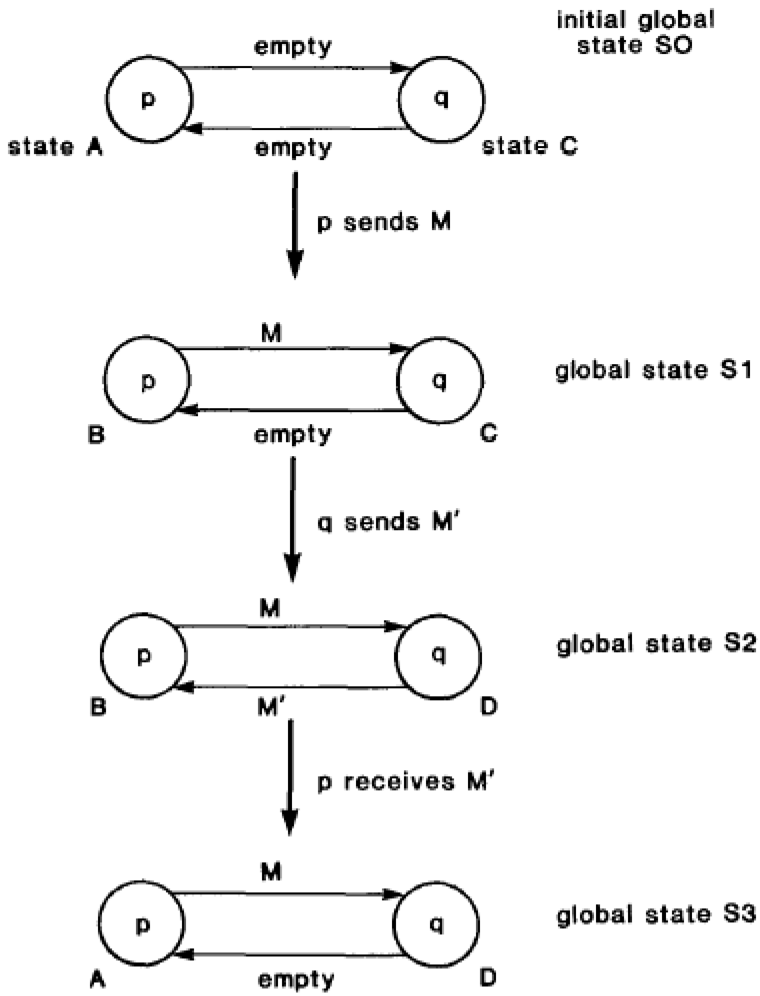
Node	Recorded state				
	state	c1	c2	c3	c4
p	490		{20}		{}
q	480	{empty}			
r	485			{empty}	

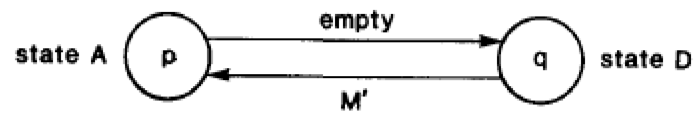
Snapshot/State Recording Example 2 - (Step 5)



Node	Recorded state				
	state	c1	c2	c3	c4
p	490		{20}		{25}
q	480	{empty}			
r	485			{empty}	

Properties of the Recorded Global State





- S* is reachable from Si
- Sf is reachable from S*

- Let Seq be a distributed computation:
  - Seq' is a permutation of seq, such that Si, S* and Sf occur as global states in seq'.

Conclusions

- The Global State Detection algorithm allows to determine a global state of the system during a computation.
- This algorithm runs concurrently with the system, but not alter, the underlying computation.
- Exist different permutations of states in a system from its initial state until its final state.

Thank You

Questions

?

Comments