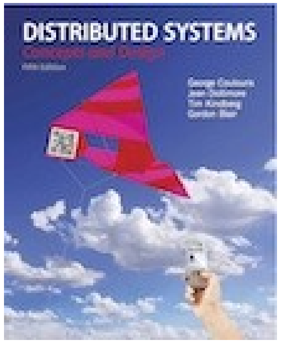
Slides for Chapter 14: Time and Global States



From **Coulouris, Dollimore, Kindberg and Blair**

Distributed Systems: Concepts and Design

Edition 5, © Addison-Wesley 2012

Figure 14.1
Skew between computer clocks in a distributed system

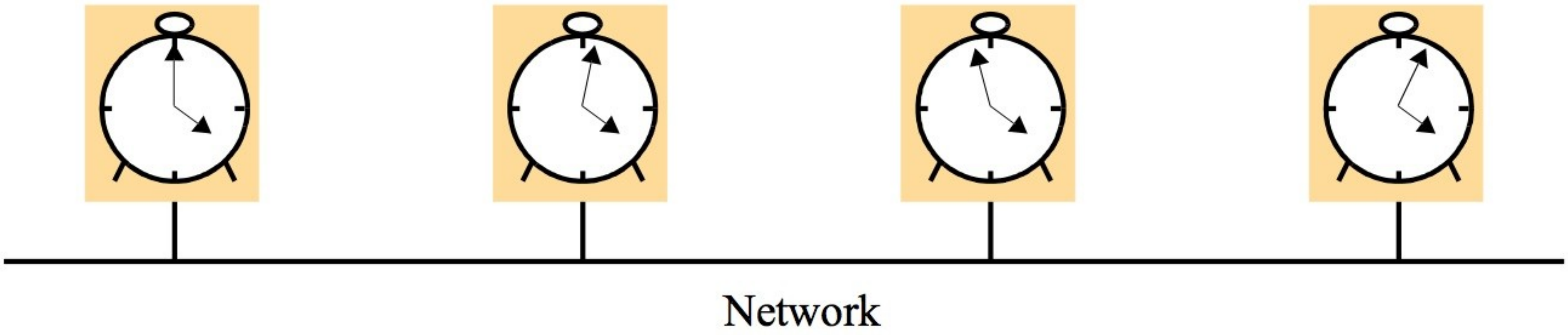


Figure 14.2
Clock synchronization using a time server

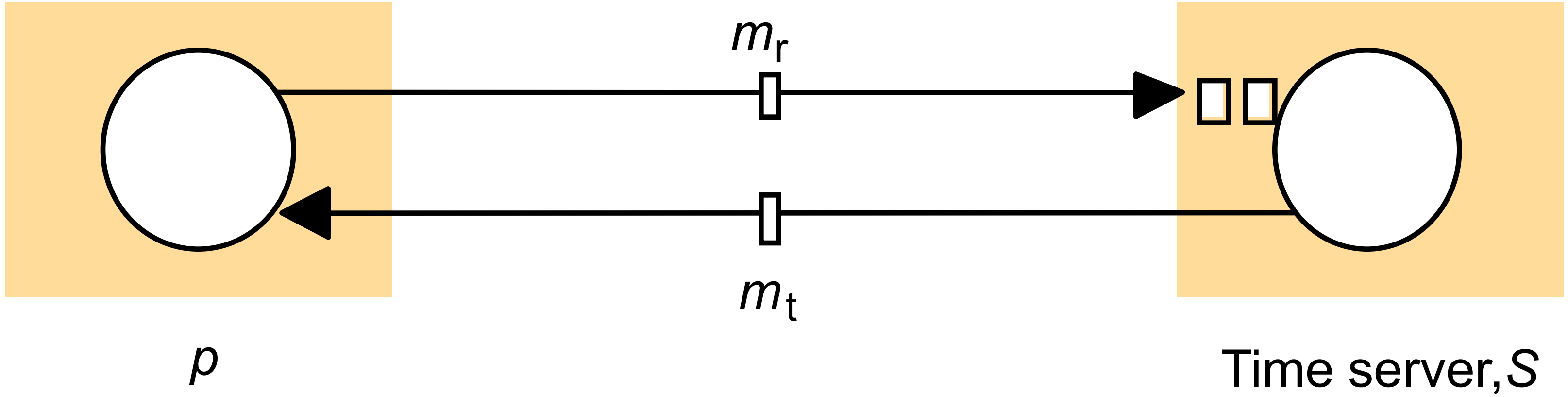
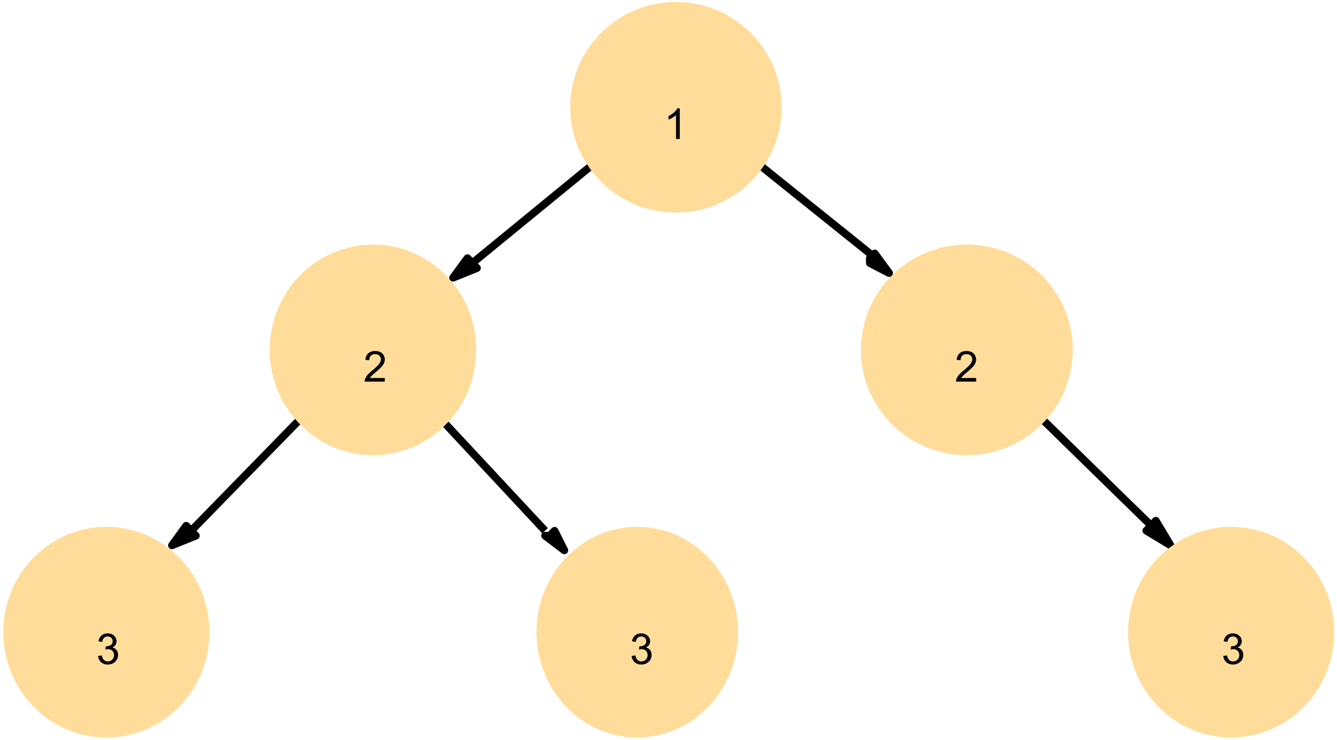


Figure 14.3
An example synchronization subnet in an NTP implementation



Note: Arrows denote synchronization control, numbers denote strata.

Figure 14.4
Messages exchanged between a pair of NTP peers

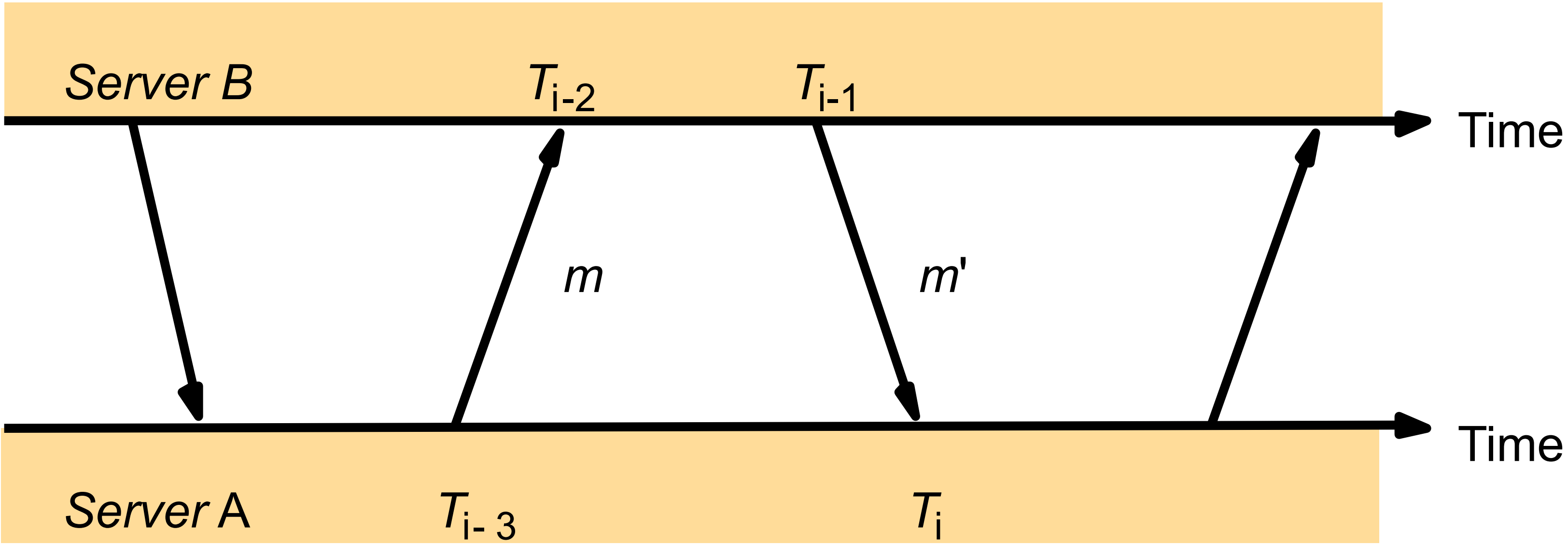


Figure 14.5
Events occurring at three processes

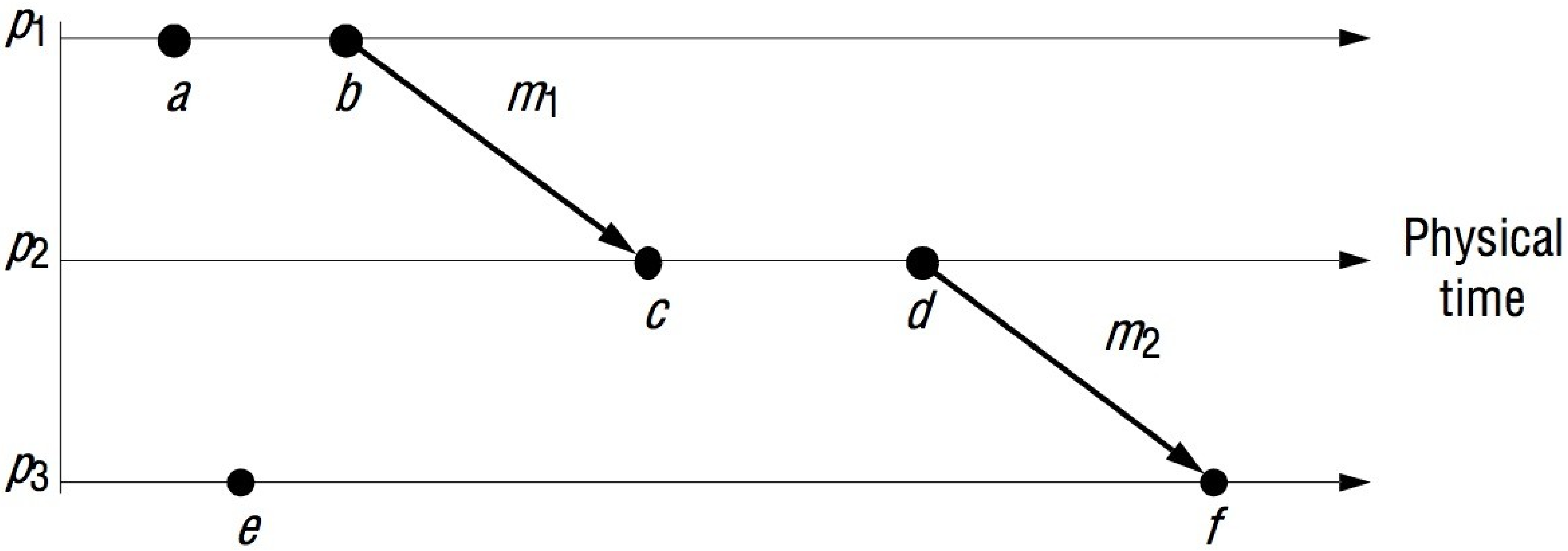


Figure 14.6
Lamport timestamps for the events shown in Figure 14.5

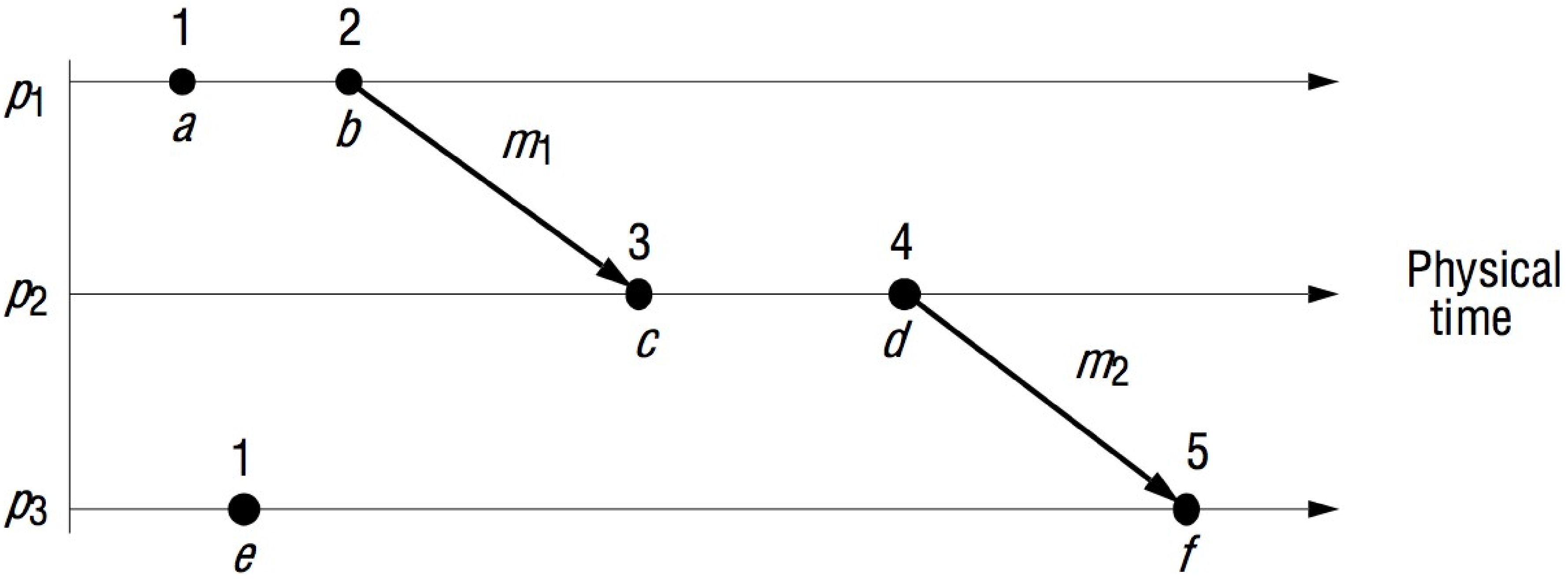


Figure 14.7
Vector timestamps for the events shown in Figure 14.5

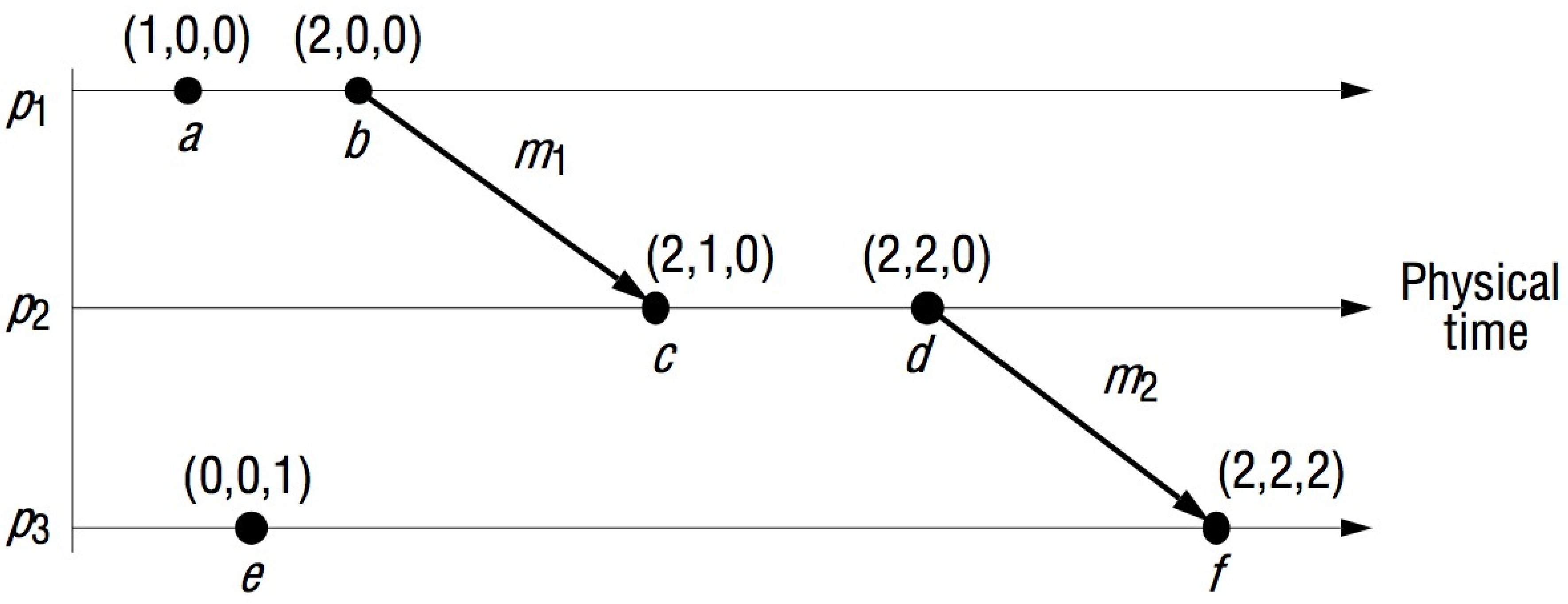
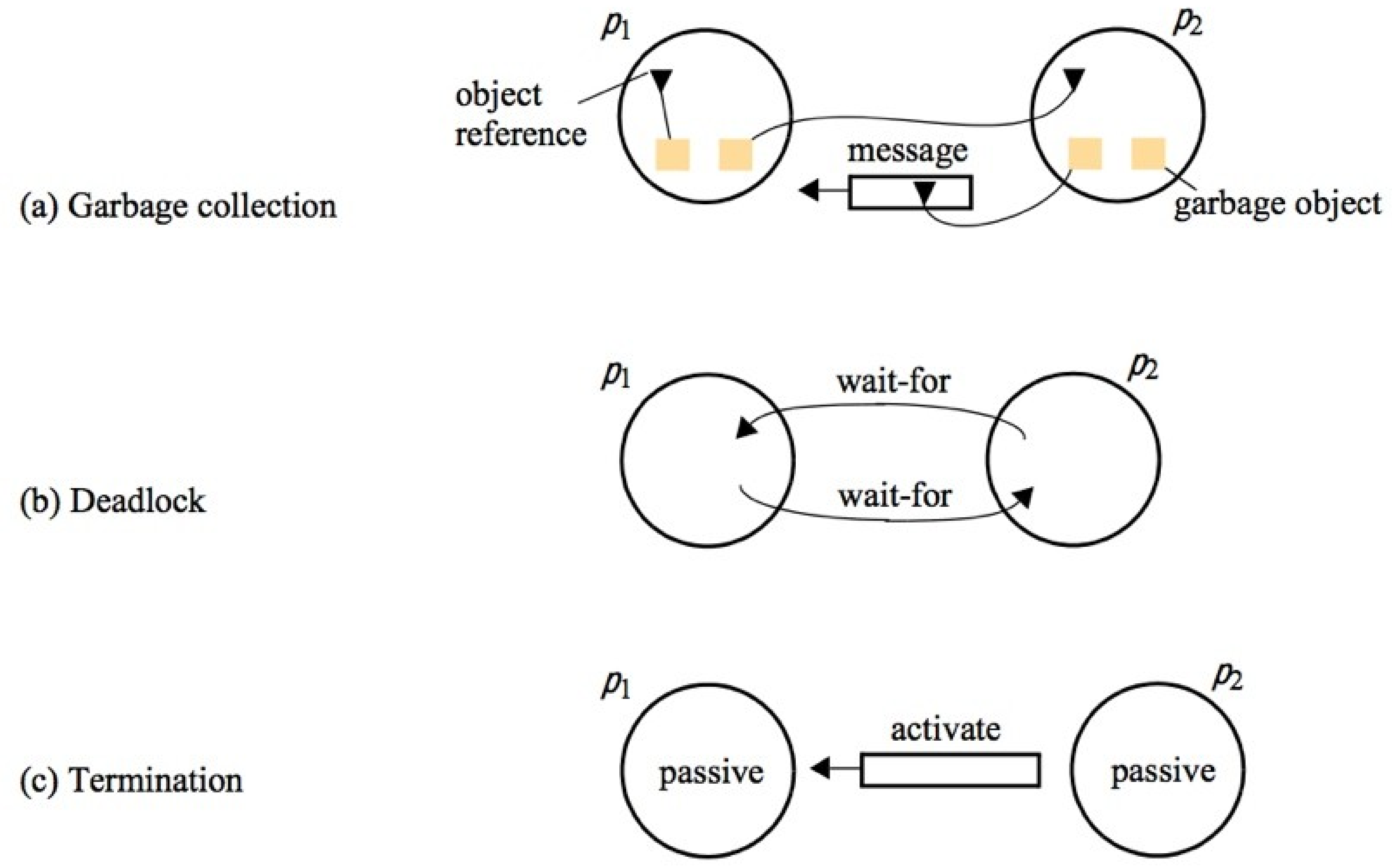


Figure 14.8
Detecting global properties



(a) Garbage collection

(b) Deadlock

(c) Termination

Figure 14.9
Cuts

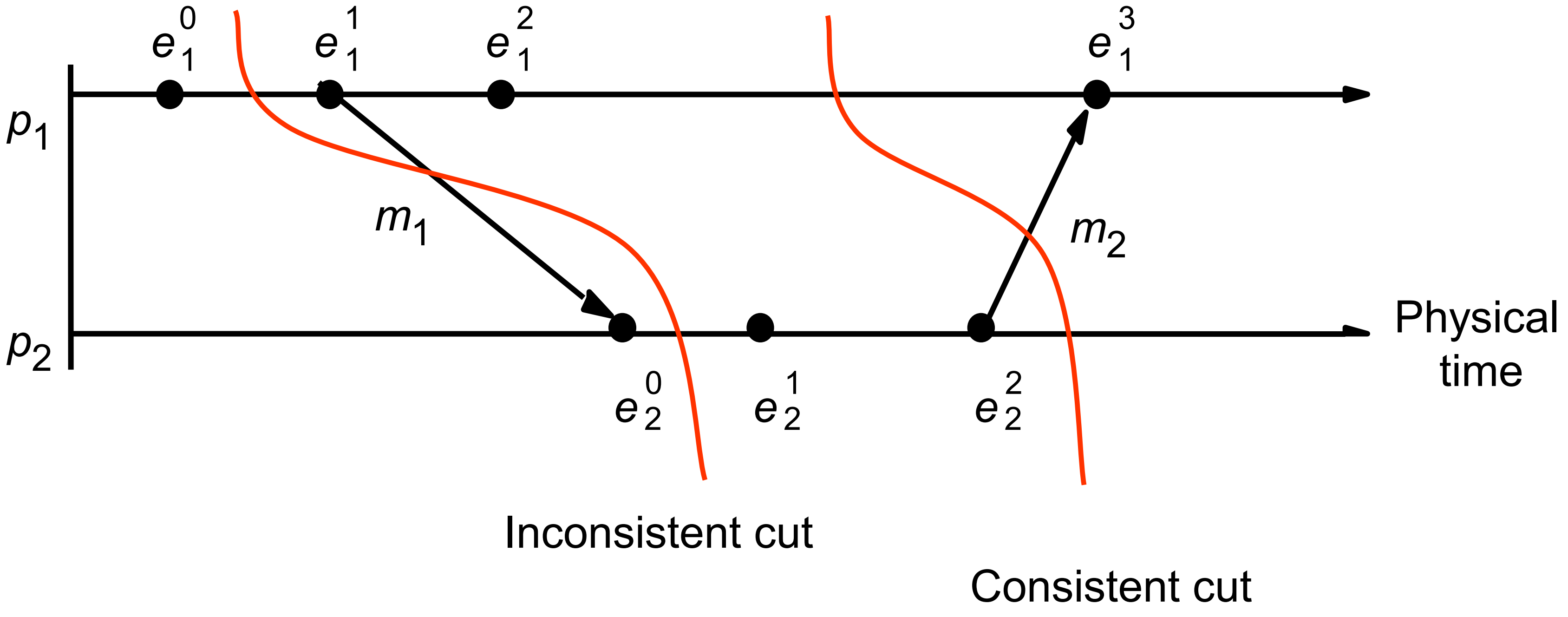


Figure 14.10
Chandy and Lamport’s ‘snapshot’ algorithm

```
Marker receiving rule for process $p_i$
On $p_i$’s receipt of a marker message over channel $c$:
    if ($p_i$ has not yet recorded its state) it
        records its process state now;
        records the state of $c$ as the empty set;
        turns on recording of messages arriving over other incoming channels;
    else
        $p_i$ records the state of $c$ as the set of messages it has received over $c$
        since it saved its state.
    end if
Marker sending rule for process $p_i$
After $p_i$ has recorded its state, for each outgoing channel $c$:
    $p_i$ sends one marker message over $c$
    (before it sends any other message over $c$).
```

Figure 14.11
Two processes and their initial states

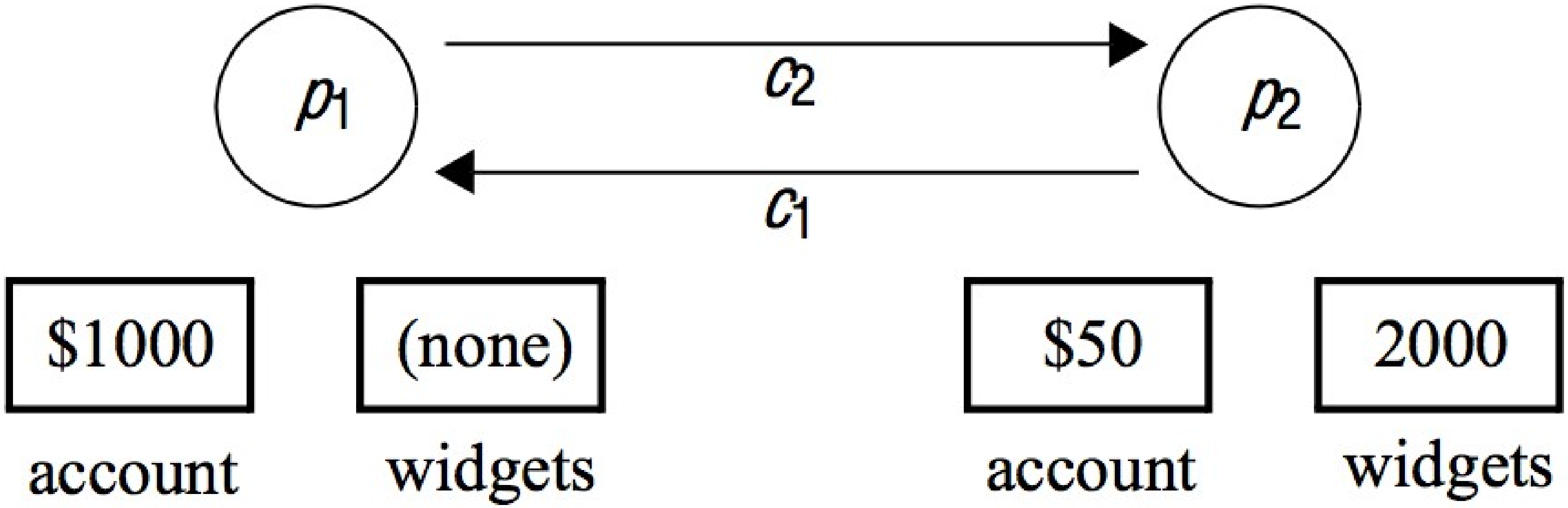
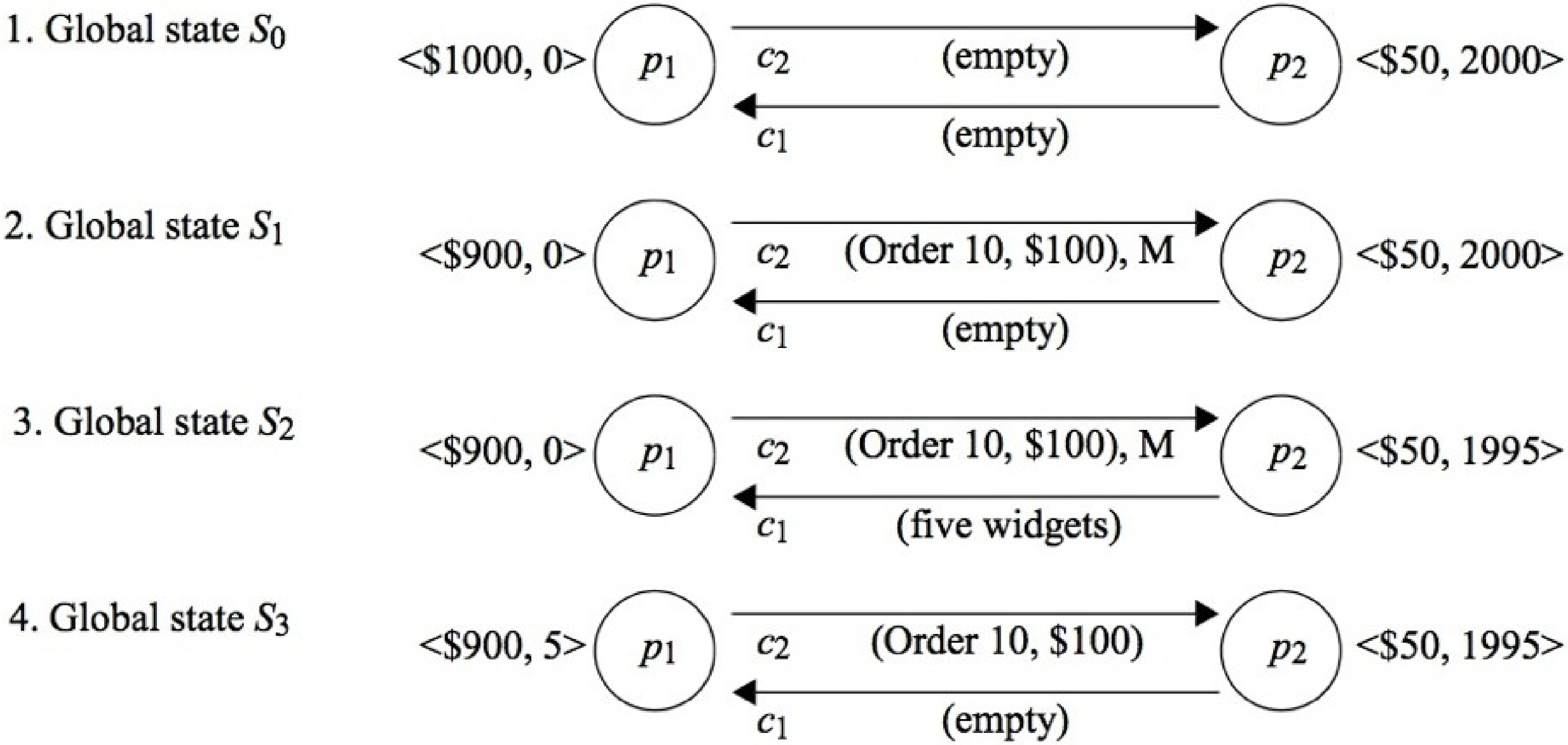


Figure 14.12
The execution of the processes in Figure 14.11



1. Global state S_0

<\$1000, 0> p_1 c_2 (empty) p_2 <\$50, 2000>

c_1 (empty)

2. Global state S_1

<\$900, 0> p_1 c_2 (Order 10, \$100), M p_2 <\$50, 2000>

c_1 (empty)

3. Global state S_2

<\$900, 0> p_1 c_2 (Order 10, \$100), M p_2 <\$50, 1995>

c_1 (five widgets)

4. Global state S_3

<\$900, 5> p_1 c_2 (Order 10, \$100) p_2 <\$50, 1995>

c_1 (empty)

(M = marker message)

Figure 14.13
Reachability between states in the snapshot algorithm

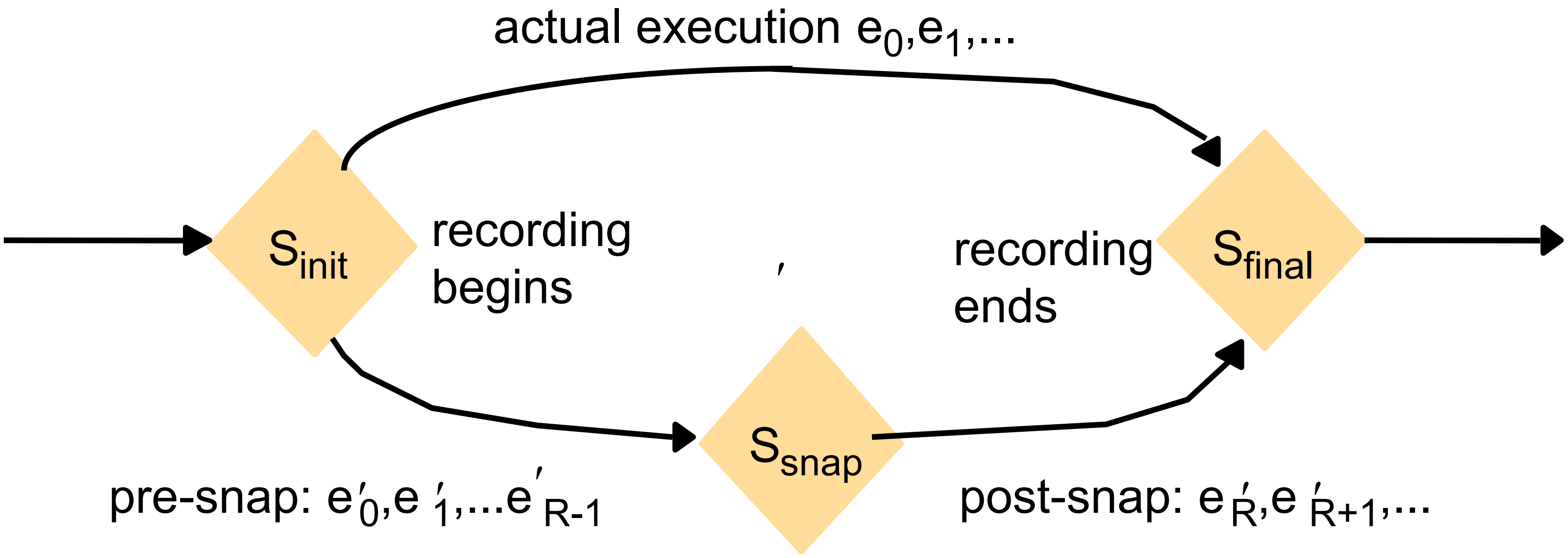

actual execution $e_0,e_1,...$
S_{init}
recording begins
recording ends
S_{final}
S_{snap}
pre-snap: $e'_0,e'_1,...e'_{R-1}$
post-snap: $e'_R,e'_{R+1},...$

Figure 14.14
Vector timestamps and variable values for the execution of Figure 14.9

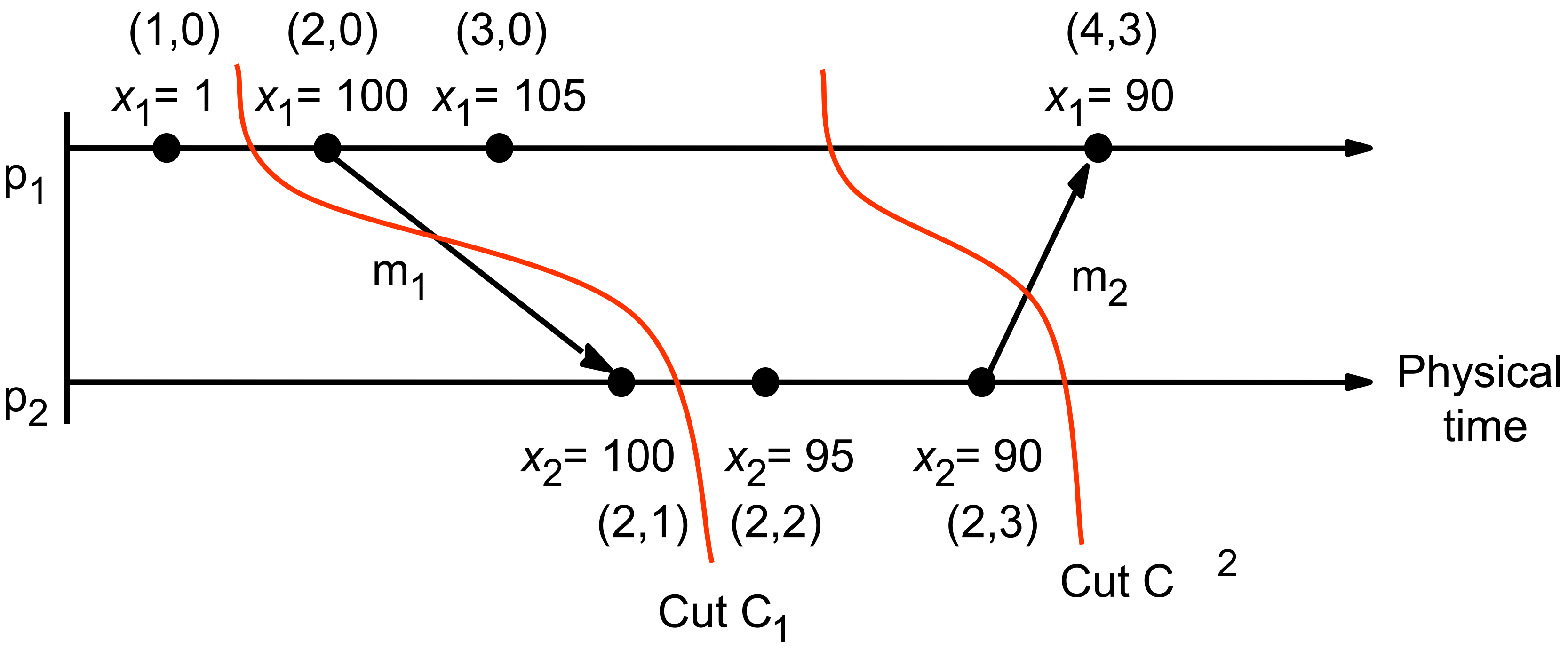
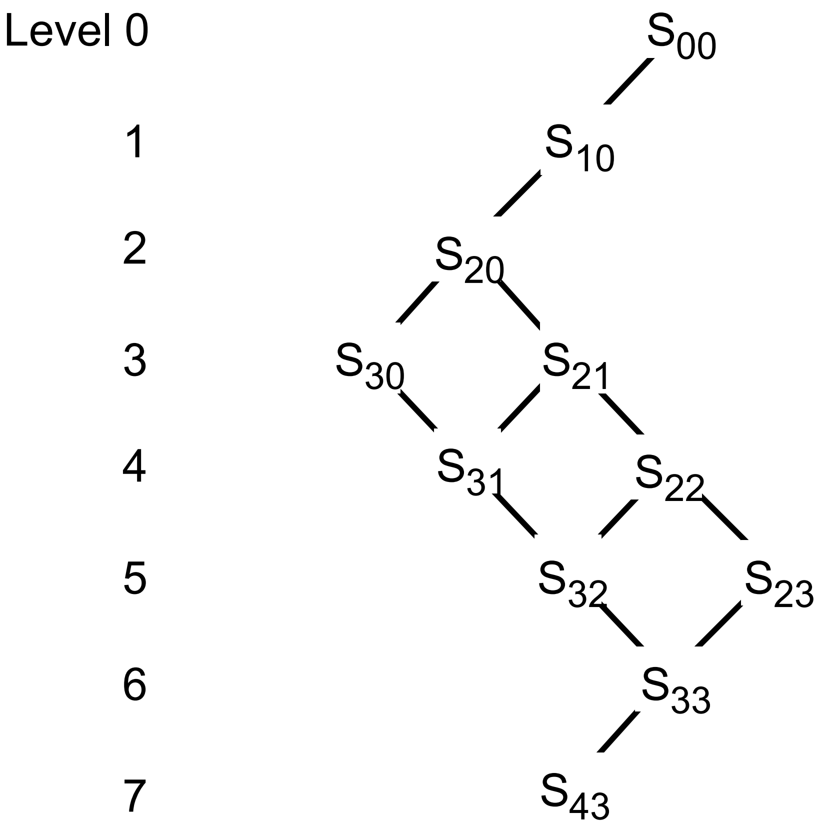


Figure 14.15
The lattice of global states for the execution of Figure 14.14



Level 0 S_{00}

1 S_{10}

2 S_{20}

3 S_{30} S_{21}

4 S_{31} S_{22}

5 S_{32} S_{23}

6 S_{33}

7 S_{43}

Sij = global state after *i* events at process 1
and *j* events at process 2

Figure 14.16
Algorithms to evaluate *possibly*) and *definitely*)

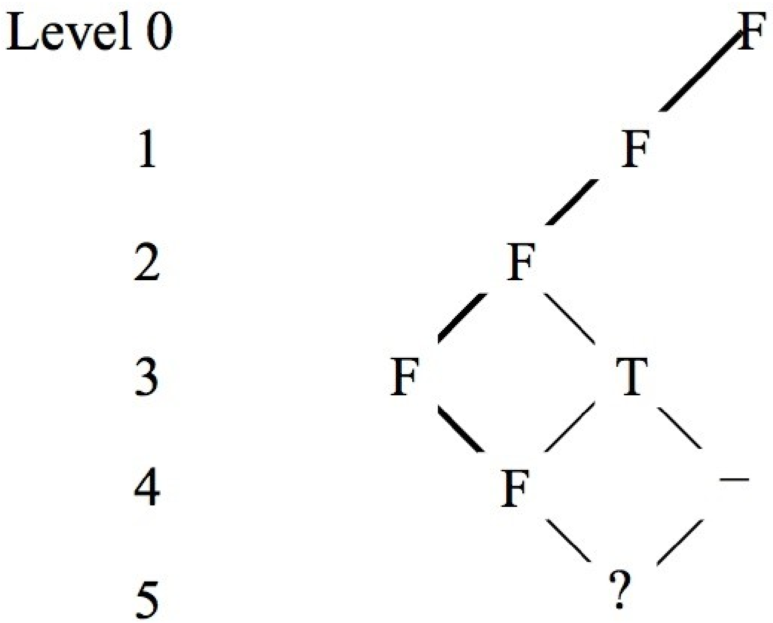
1. *Evaluating possibly* ϕ *for global history* H *of* N *processes*

```
$L := 0;$
$States := \{ (s_1^0, s_2^0, \ldots, s_N^0) \};$
while $(\phi(S) = False$ for all $S \in$ States$)$
        $L := L + 1;$
        $Reachable := \{ S' : S'$ reachable in $H$ from some $S \in$ States $\wedge$ level$(S') = $ L$\};$
        $States := Reachable$
end while
output "possibly $\phi$";
```

2. *Evaluating definitely* ϕ *for global history* H *of* N *processes*

```
$L := 0;$
if $(\phi(s_1^0, s_2^0, \ldots, s_N^0))$ then $States := \{\}$ else $States := \{ (s_1^0, s_2^0, \ldots, s_N^0) \};$
while $(States \neq \{\})$
        $L := L + 1;$
        $Reachable := \{ S' : S'$ reachable in $H$ from some $S \in$ States $\wedge$ level$(S') = $ L$\};$
        $States := \{ S \in Reachable : \phi(S) = False \}$
end while
output "definitely $\phi$";
```

Figure 14.17
Evaluating *definitely*)



F = ($\phi(S)$ = False); T = ($\phi(S)$ = True)