

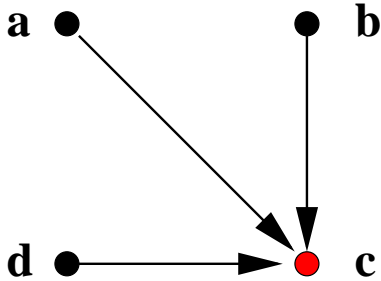
# 12. Routing Algorithm for Mobile Agent

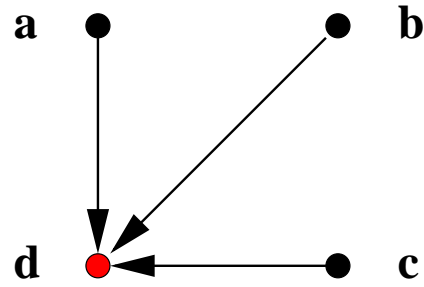
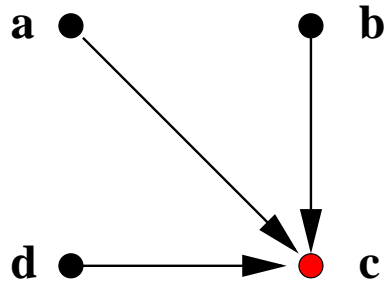
Jean-Raymond Abrial

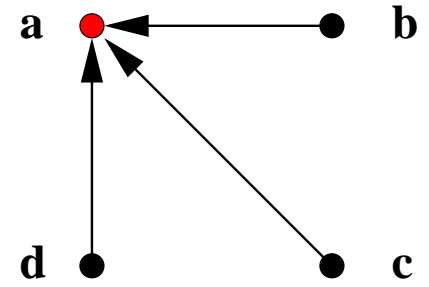
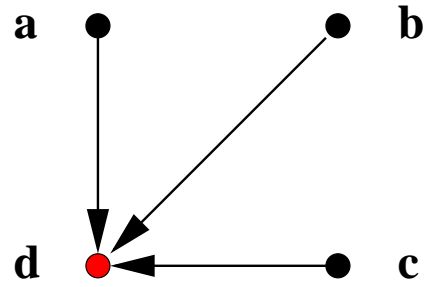
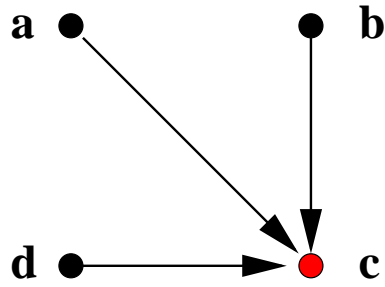
2009

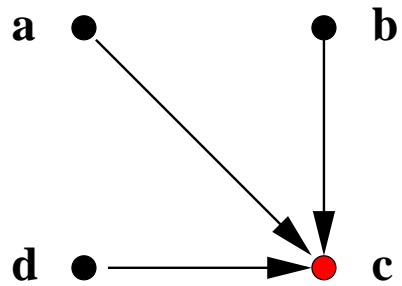
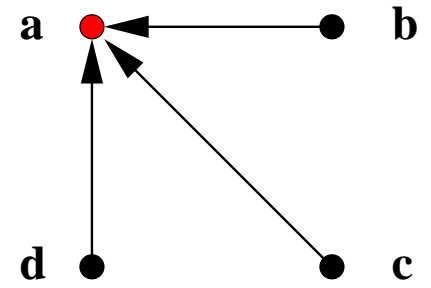
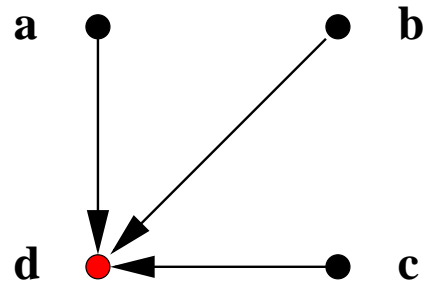
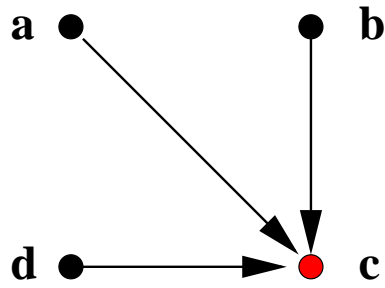
- 
- No more learning about refinement and abstraction (practicing)
  - No more learning about modeling conventions (practicing)
  - Re-using dynamically the small tree theory we already developed
  - Study a practical problem in distributed computing communication
  - The example comes from the following paper:  
L. Moreau. *Distributed Directory Service and Message Routing for Mobile Agent*. Science of Computer Programming 2001.

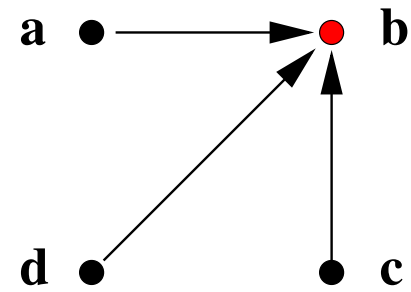
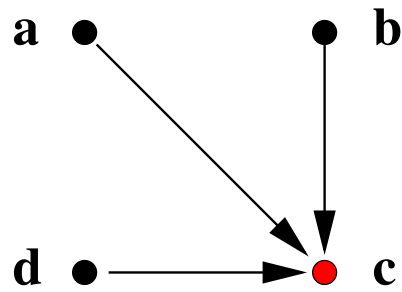
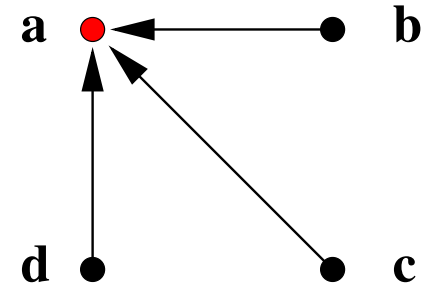
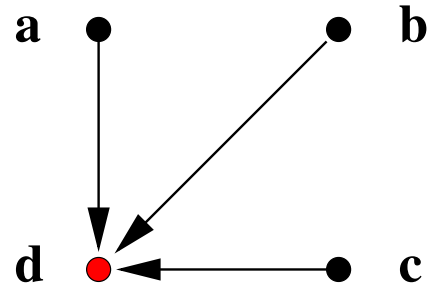
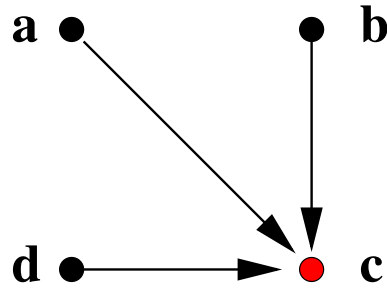
- 
- A **mobile agent**  $\mathcal{M}$  is supposed to travel between **sites**
  - Some **fixed agents** at sites want to **send messages** to  $\mathcal{M}$
  - In an **abstract world**:
    - the **moves** of  $\mathcal{M}$  are **instantaneous**
    - the **traveling** of messages between sites **takes no time**
    - the **knowledge** of the moves of  $\mathcal{M}$  is also **instantaneous**
  - Thus fixed agents **always send messages** where  $\mathcal{M}$  is





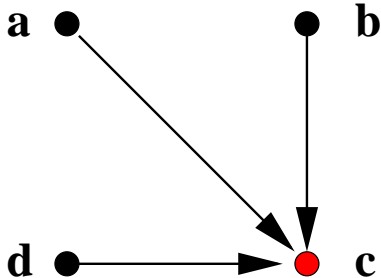


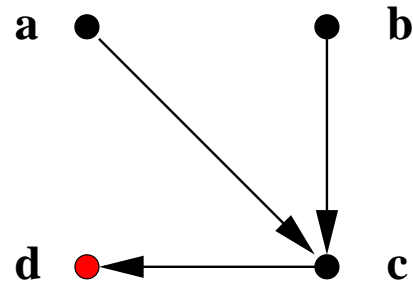
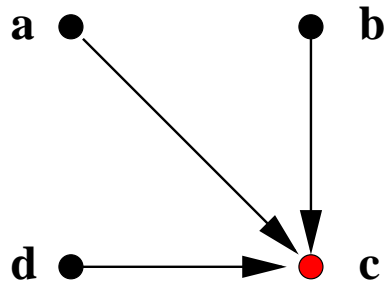


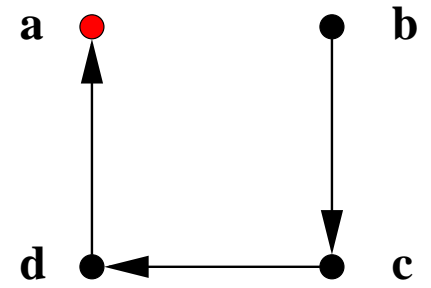
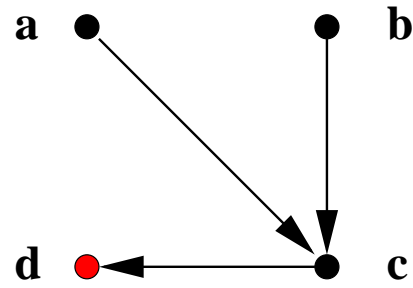
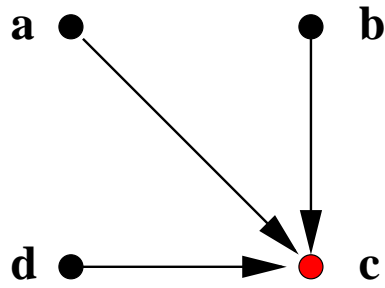


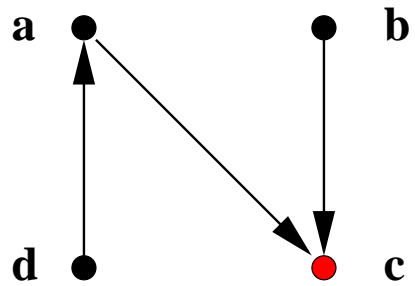
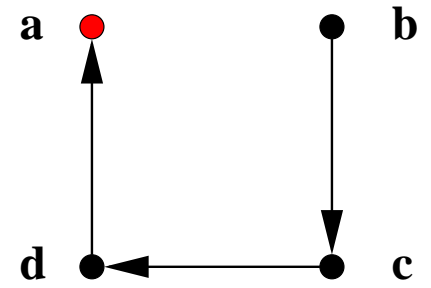
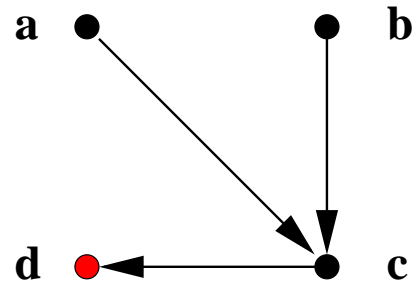
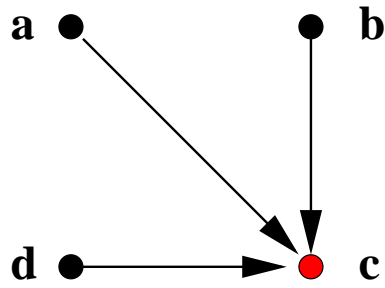
- The **moves** of  $\mathcal{M}$  are **still instantaneous**
- The **traveling** of messages between sites **still takes no time**
- The **knowledge** of the moves of  $\mathcal{M}$  is **not instantaneous** any more

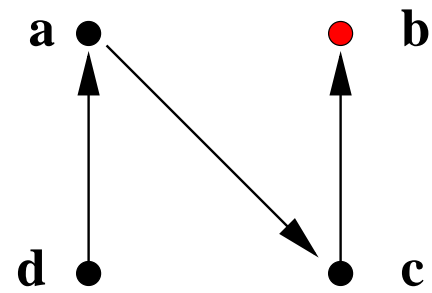
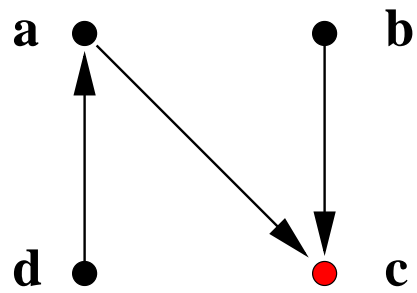
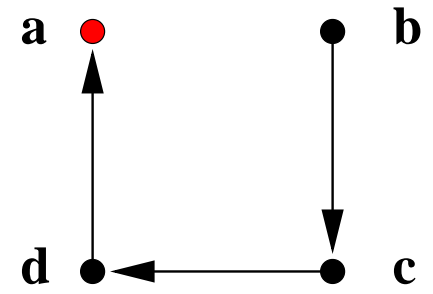
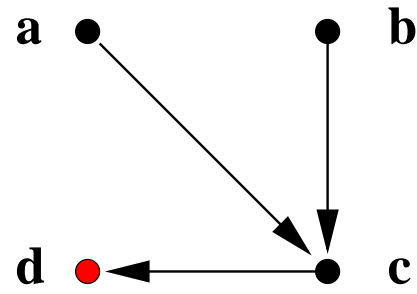
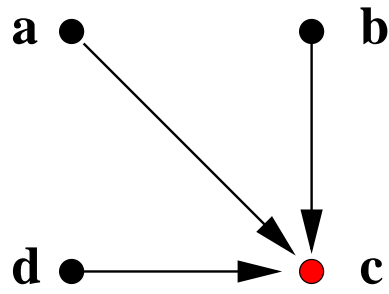
- When  $\mathcal{M}$  moves from site  $x$  to site  $y$  then
  - Agents of  $x$  and  $y$  **knows it immediately**
  - Agents of other sites are **not aware of the move**
  - They still sent their messages **where they believe  $\mathcal{M}$  is**
- A message arriving at a site which  $\mathcal{M}$  has left **can be forwarded**

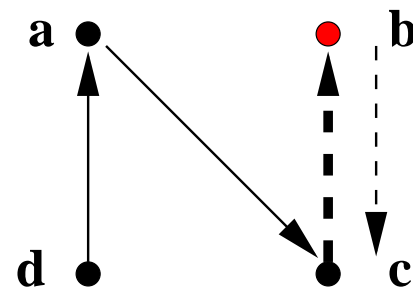
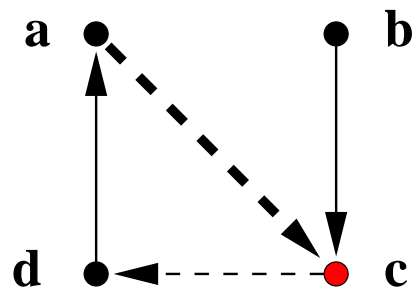
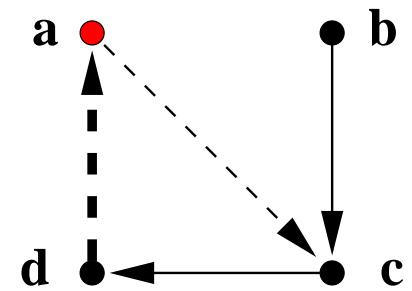
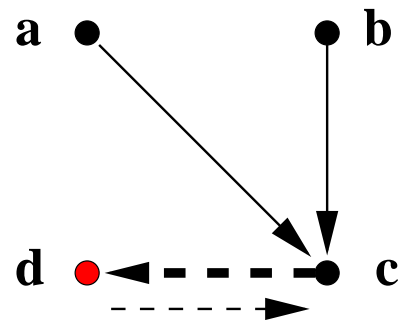
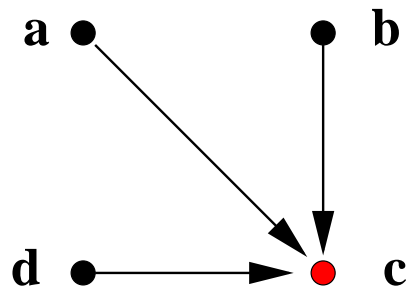


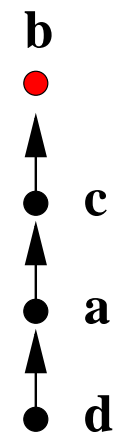
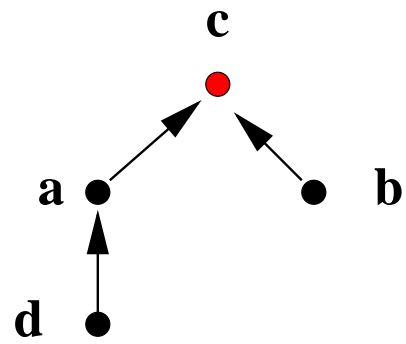
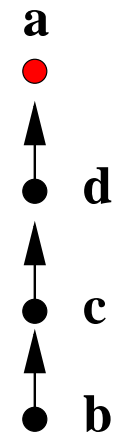
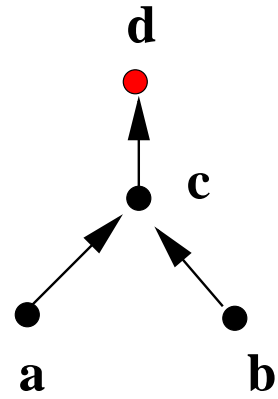
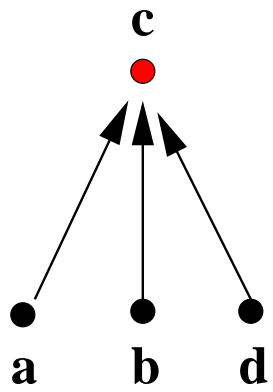


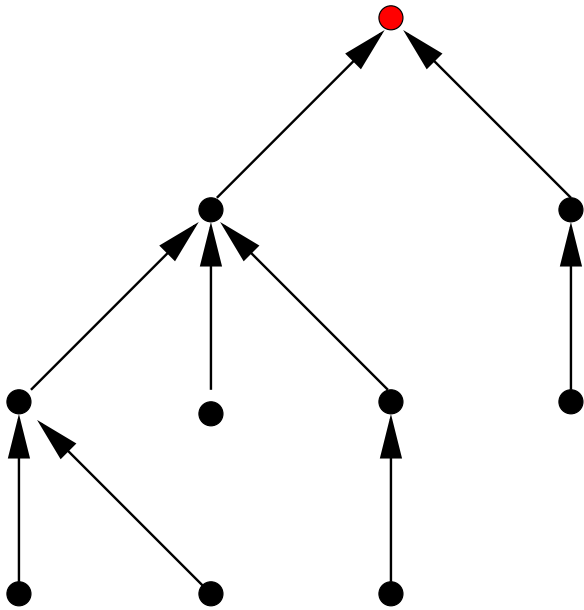




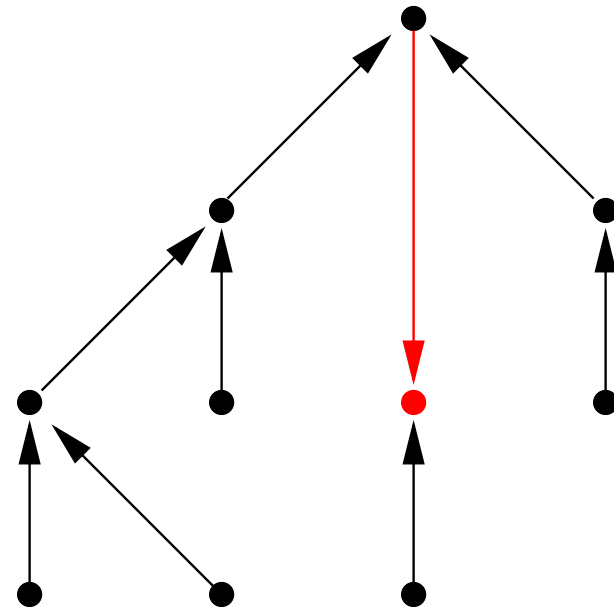
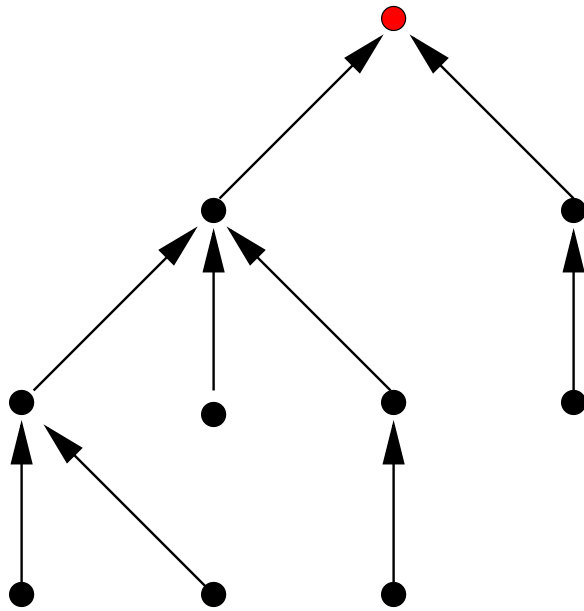


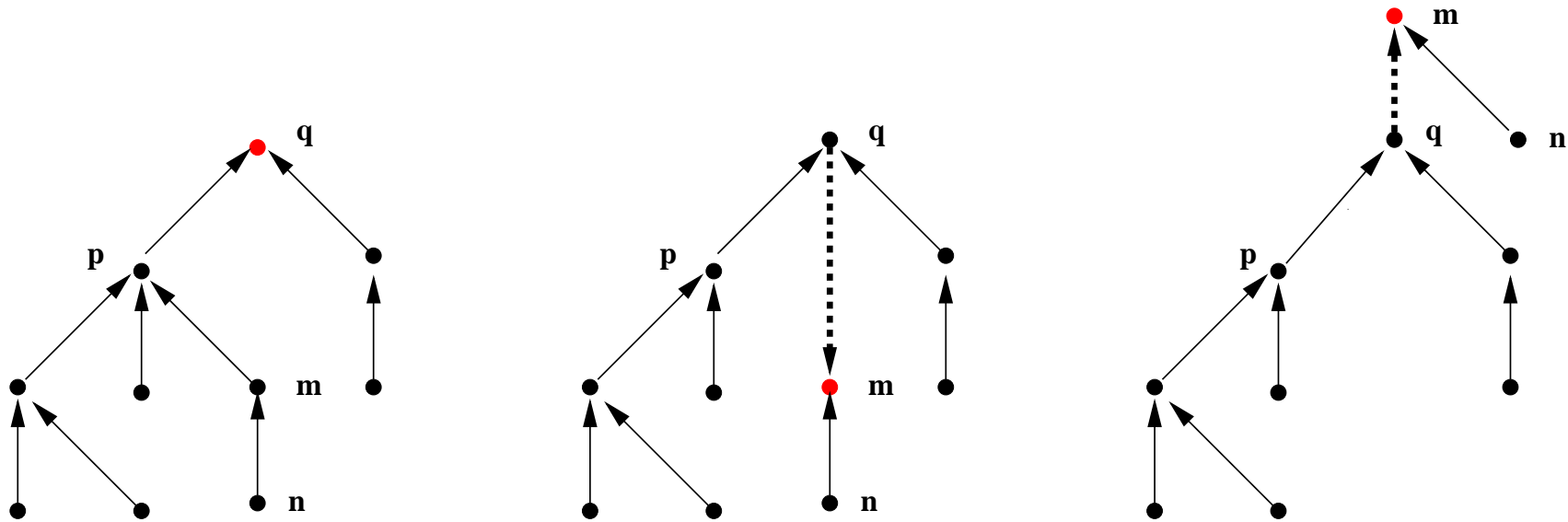






- The mobile  $\mathcal{M}$  is at the root of a tree





The mobile  $\mathcal{M}$  remains at the **root of a tree** (to be **proved** however)

- $S$  denotes the set of **sites**
- $M$  denotes the set of **messages**

**carrier sets:**  $S$   
 $M$

**constant:**  $il$

**axm0\_1:**  $il \in S$

**axm0\_2:**  $\text{finite}(S)$

- Constant  $il$  denotes the **initial location** of the mobile

**variables:**  $l$   
 $c$   
 $p$

**inv0\_1:**  $l \in S$

**inv0\_2:**  $c \in S \setminus \{l\} \rightarrow S$

**inv0\_3:**  $p \in M \leftrightarrow S$

- Variable  $l$  denotes the **actual location** of the mobile
- Variable  $c$  denotes the **dynamic channel** structure
- Variable  $p$  denotes the **position of each message**

This invariant states that the **channel structure is a tree** with:

- **root:  $l$**
- **parent function:  $c$**

$$\mathbf{inv0\_4:} \quad \forall T \cdot T \subseteq S \wedge T \subseteq c^{-1}[T] \Rightarrow T = \emptyset$$

init

$l := il$

$c := (S \setminus \{il\}) \times \{il\}$

$p := \emptyset$

rcv\_agt

**any**  $s$  **where**

$s \neq l$

**then**

$l := s$

$c := (\{s\} \triangleleft c) \cup \{l \mapsto s\}$

**end**

- This event describes the move of the mobile **from  $l$  to  $s$**
- The move of the mobile from  $l$  to  $s$  is supposed to be **instantaneous**

- Node  $s$  sends a message to the Mobile
- This message is stored locally

```
snd_msg
  any  $s, m$  where
     $s \in S$ 
     $m \in M \setminus \text{dom}(p)$ 
  then
     $p(m) := s$ 
  end
```

- Messages are either **delivered** or **forwarded**

```
dlv_msg
  any m where
     $m \in \text{dom}(p)$ 
     $p(m) = l$ 
  then
     $p := \{m\} \triangleleft p$ 
  end
```

```
fwd_msg
  any m where
     $m \in \text{dom}(p)$ 
     $p(m) \neq l$ 
  then
     $p(m) := c(p(m))$ 
  end
```

- When delivered, a message is **removed**

```

rcv_agt
  any  $s$  where
     $s \in S \setminus \{l\}$ 
  then
     $l := s$ 
     $c := (\{s\} \triangleleft c) \cup \{l \mapsto s\}$ 
  end

```

...

Invariant **inv0\_4**

Guard of **rcv\_agt**  
 $\vdash$

Modified Invariant **inv0\_4**

...

$$\forall T . \left( \begin{array}{l} T \subseteq S \\ T \subseteq c^{-1}[T] \\ \Rightarrow \\ T = \emptyset \end{array} \right)$$

$s \in S \setminus \{l\}$   
 $\vdash$

$$\forall T . \left( \begin{array}{l} T \subseteq S \\ T \subseteq (\{s\} \triangleleft c) \cup \{l \mapsto s\})^{-1}[T] \\ \Rightarrow \\ T = \emptyset \end{array} \right)$$

$$\dots$$

$$\forall T. \left( \begin{array}{l} T \subseteq S \\ T \subseteq c^{-1}[T] \\ \Rightarrow \\ T = \emptyset \end{array} \right)$$

$$s \in S \setminus \{l\}$$

$$T \subseteq S$$

$$T \subseteq (\{s\} \triangleleft c) \cup \{l \mapsto s\}^{-1}[T]$$

$$\vdash T = \emptyset$$

ALL\_L

$$\dots$$

$$\left( \begin{array}{l} T \subseteq S \\ T \subseteq c^{-1}[T] \\ \Rightarrow \\ T = \emptyset \end{array} \right)$$

$$s \in S \setminus \{l\}$$

$$T \subseteq S$$

$$T \subseteq (\{s\} \triangleleft c) \cup \{l \mapsto s\}^{-1}[T]$$

$$\vdash T = \emptyset$$

SET ...

$$\dots$$

$$\left( \begin{array}{l} T \subseteq S \\ T \subseteq c^{-1}[T] \\ \Rightarrow \\ T = \emptyset \end{array} \right)$$

$$s \in S \setminus \{l\}$$

$$T \subseteq S$$

$$T \subseteq (\{s\} \triangleleft c) \cup \{l \mapsto s\}^{-1}[T]$$

$$T \subseteq c^{-1}[T]$$

$$\vdash T = \emptyset$$

IMP\_L

$$\dots$$

$$T = \emptyset$$

$$s \in S \setminus \{l\}$$

$$T \subseteq S$$

$$T \subseteq (\{s\} \triangleleft c) \cup \{l \mapsto s\}^{-1}[T]$$

$$T \subseteq c^{-1}[T]$$

$$\vdash T = \emptyset$$

HYP

- The key to this proof is the following lemma:

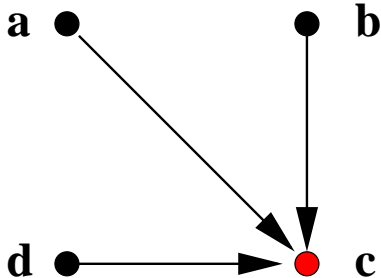
$$\begin{array}{l} \dots \\ s \in S \setminus \{l\} \\ T \subseteq (\{s\} \triangleleft c) \cup \{l \mapsto s\})^{-1}[T] \\ \vdash \\ T \subseteq c^{-1}[T] \end{array}$$

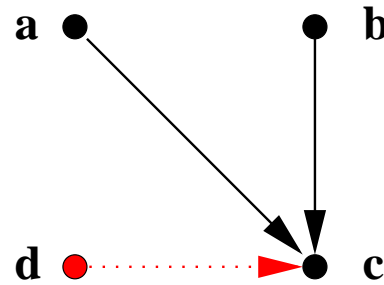
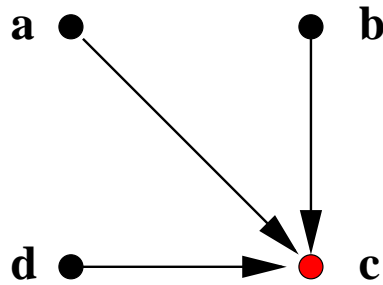
Hint: Consider two cases successively,  $s \in T$  and  $s \notin T$ .

- The **moves** of  $\mathcal{M}$  are **not completely instantaneous** any more
- The **traveling** of messages between sites **still takes no time**
- The **knowledge** of the moves of  $\mathcal{M}$  is **not instantaneous** any more

- Agents of  $l$  do not know where  $\mathcal{M}$  is going
- Agents of other sites are not aware of the move
- Messages at  $l$  cannot be forwarded until  $l$  knows where  $\mathcal{M}$  is
- Messages at other sites can be forwarded (in general)

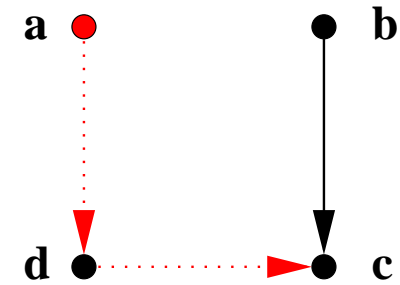
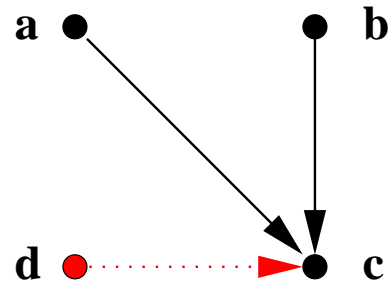
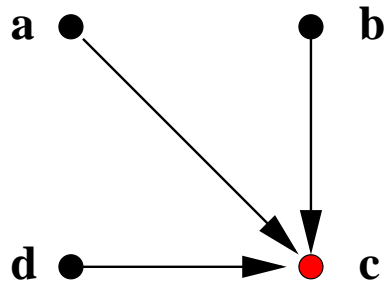
- It sends a “service message” to  $l$  to inform it about its new position
- Once  $l$  has received the “service message” it can forward again communication messages which were pending
- From now on, we have to distinguish:
  - communication messages (still instantaneous)
  - service messages (which take some time)





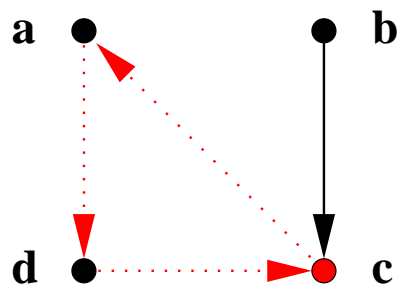
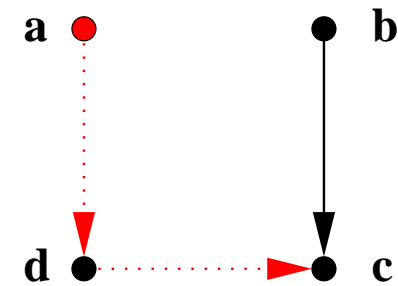
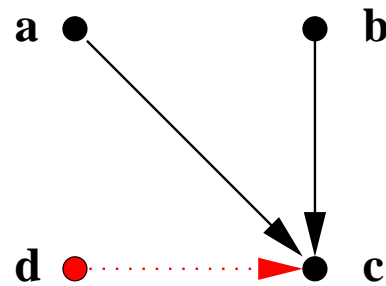
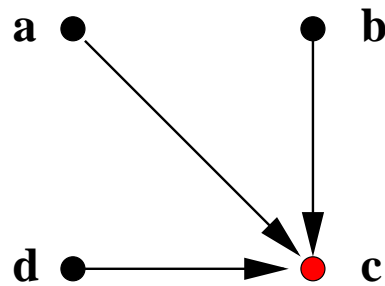
$\mathcal{M}$  sends a service message to  $c$ : "I am now in d"

Site  $c$  suspend sending com. msg. until it knows where  $\mathcal{M}$  is



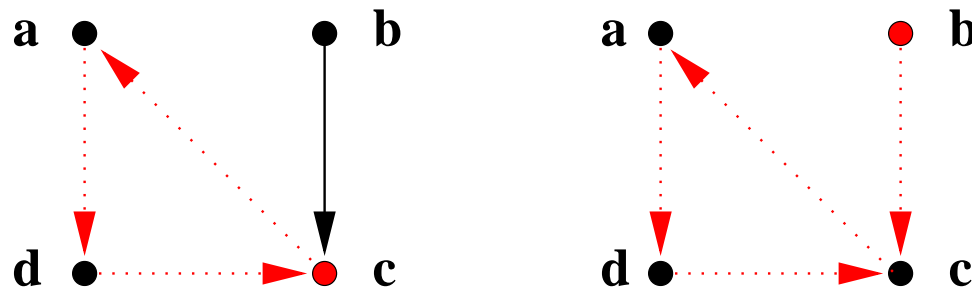
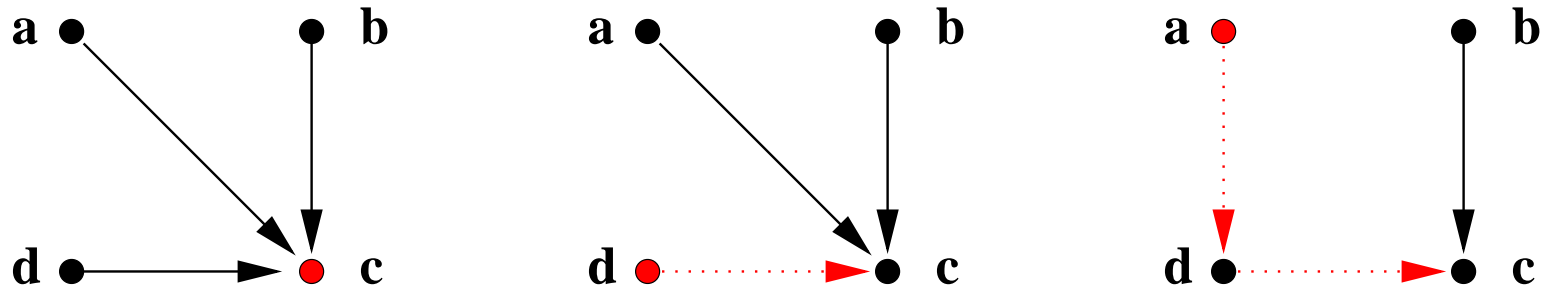
$\mathcal{M}$  sends a service message to  $d$ : "I am now in a"

Site  $d$  suspend sending com. msg. until it knows where  $\mathcal{M}$  is



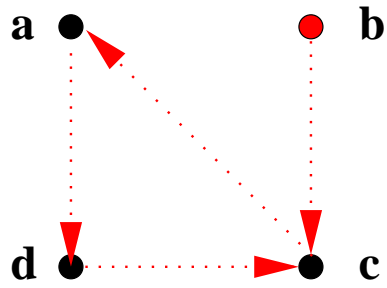
$\mathcal{M}$  sends a service message to  $a$ : "I am now in c"

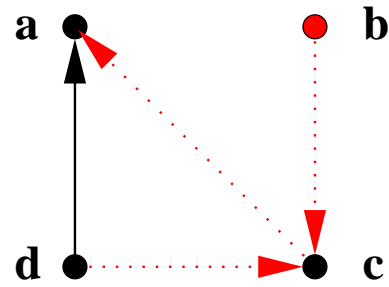
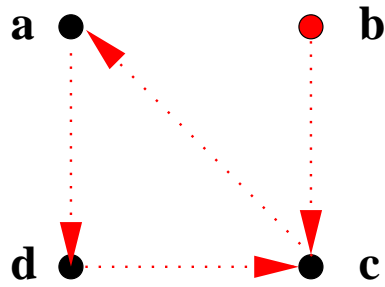
Site  $a$  suspend sending com. msg. until it knows where  $\mathcal{M}$  is



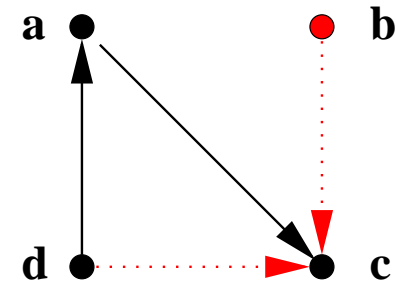
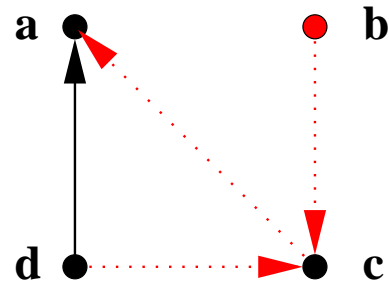
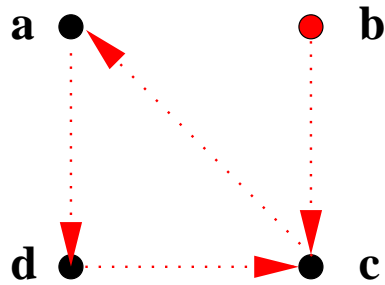
$\mathcal{M}$  sends a service message to  $c$ : "I am now in b"

Site  $c$  suspend sending com. msg. until it knows where  $\mathcal{M}$  is

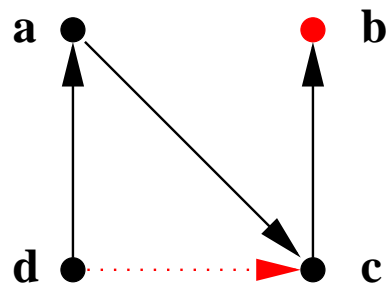
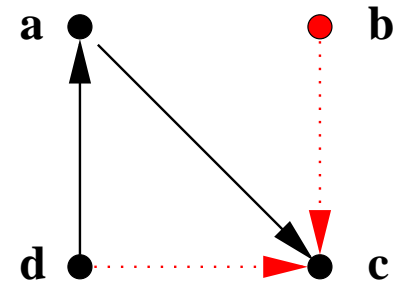
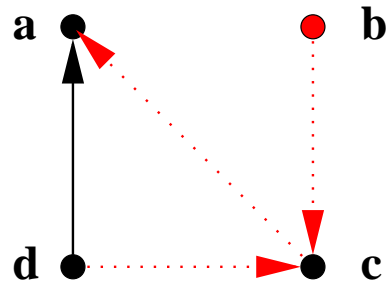
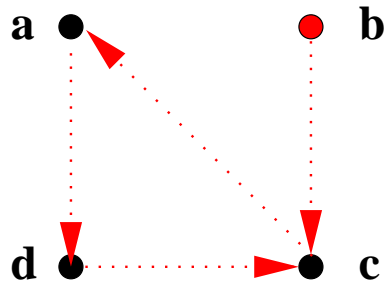




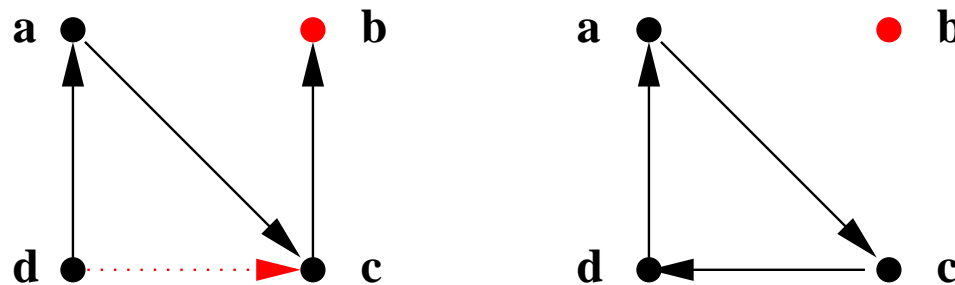
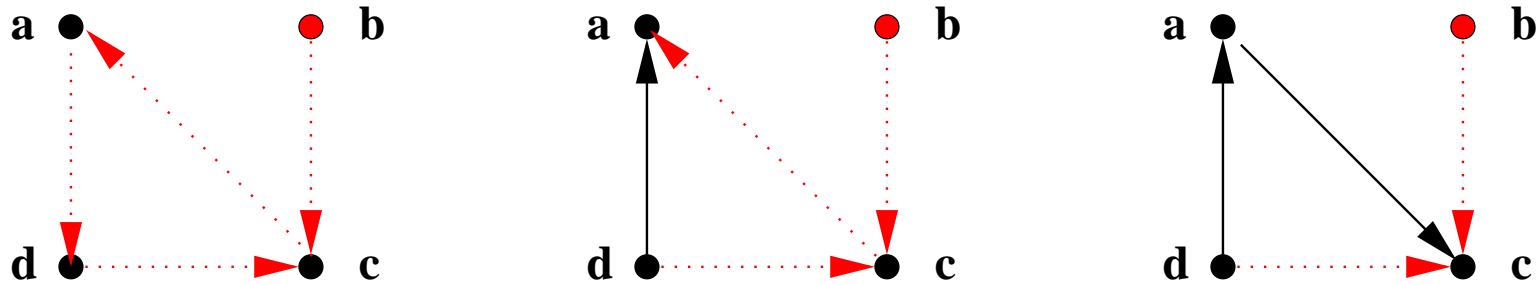
Site  $d$  believes  $\mathcal{M}$  is in  $a$ . It now forwards pending com. msg. to  $a$



Site  $a$  believes  $\mathcal{M}$  is in  $c$ . It now forwards pending com. msg. to  $c$

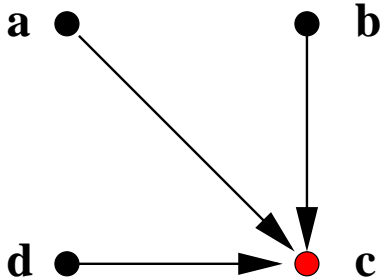


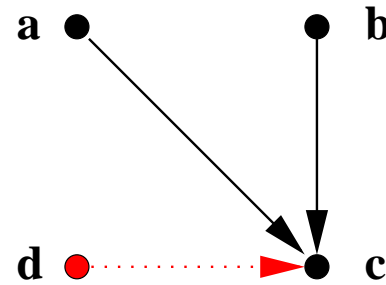
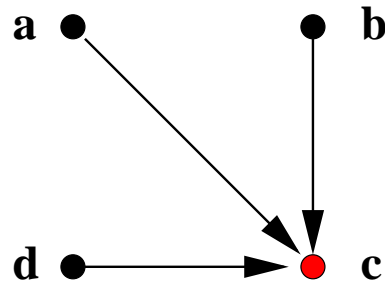
Site  $c$  believes  $\mathcal{M}$  is in  $b$ . It now forwards pending com. msg. to  $b$



Site  $c$  believes  $\mathcal{M}$  is in  $d$ . It now forwards pending com. msg. to  $d$   
 The tree structure is destroyed: we have a **CYCLE**.

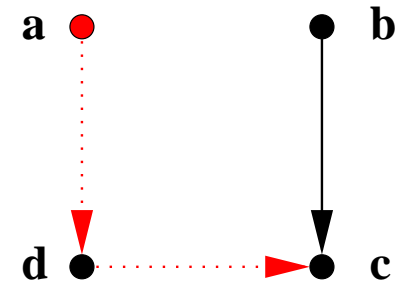
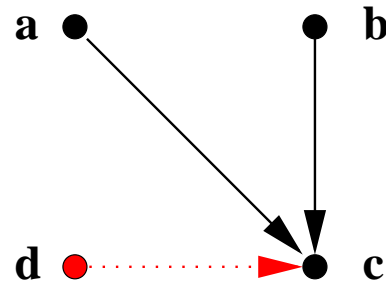
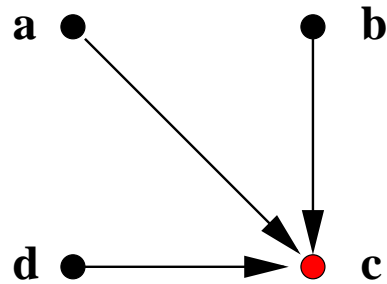
- The failure comes from the **two srv. msg.** arriving in the **same place**
- We must **preclude this to happen**
- We shall suppose that we have the following **“magic” behavior**
  - When  $\mathcal{M}$  **sends a service message** to site  $x$
  - It is able to **remove all other pending service messages**  
**whose destination is also  $x$**





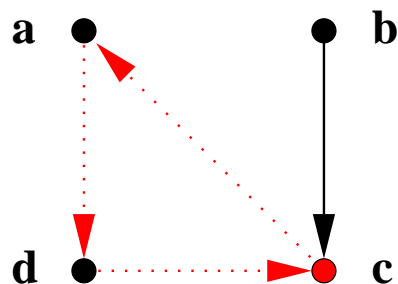
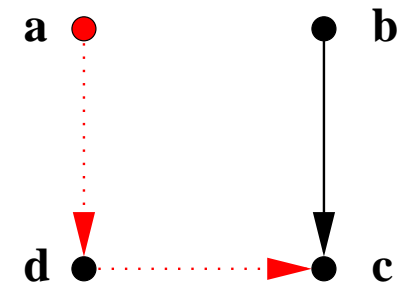
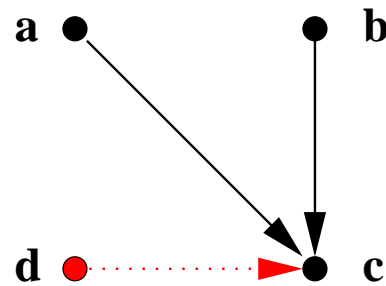
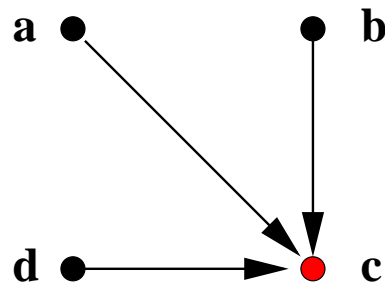
$\mathcal{M}$  sends a service message to  $c$ : "I am now in d"

Site  $c$  suspend forwarding com. msg. until it believes where  $\mathcal{M}$  is



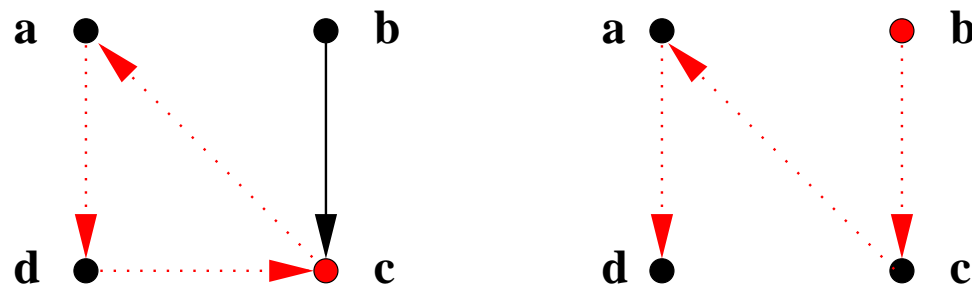
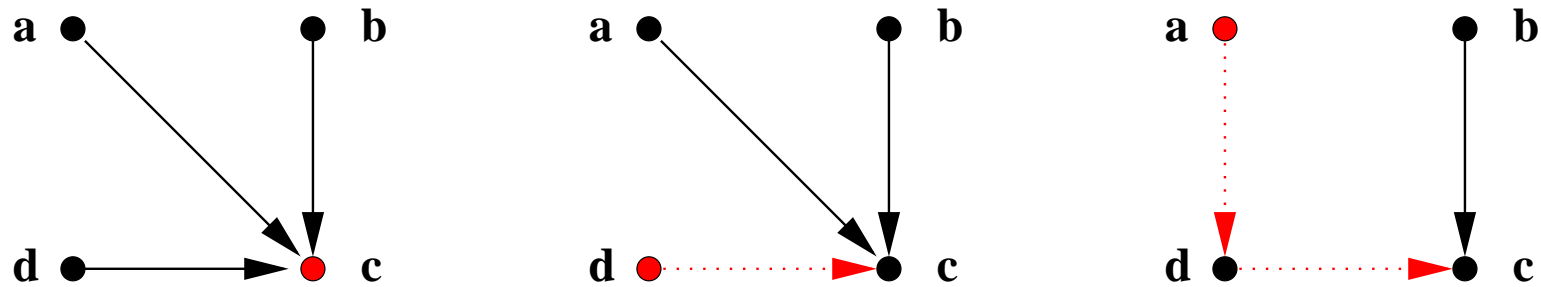
$\mathcal{M}$  sends a service message to  $d$ : "I am now in a"

Site  $d$  suspend forwarding com. msg. until it believes where  $\mathcal{M}$  is



$\mathcal{M}$  sends a service message to  $a$ : "I am now in c"

Site  $a$  suspend forwarding com. msg. until it believes where  $\mathcal{M}$  is



$\mathcal{M}$  sends a service message to  $c$ : "I am now in  $b$ "

$\mathcal{M}$  "magically" removes the other service message arriving to  $c$

**variables:**  $l, p, d, a, da$

**inv1\_1:**  $d \in S \setminus \{l\} \leftrightarrow S$

**inv1\_2:**  $a \in S \setminus \{l\} \leftrightarrow S$

**inv1\_3:**  $c = d \Leftarrow a$

- Variable  $d$  denotes the **new dynamic tree structure**
- Variable  $a$  denotes the **service message channel**.
- **inv1\_3** denotes the **link** between  $c$  and the concrete  $d$  and  $a$

$$\text{inv1\_2: } a \setminus \{l\} \in S \leftrightarrow S$$

- $s1 \mapsto s2$  in  $a$  means a message from  $s2$  (new site) to  $s1$  (old site)
- Notice that the new site cannot be  $l$
- **At most one service message** is in transit to site  $s1$  ( $a$  is a function)
- **This magic behavior is fundamental**

$$\text{inv1\_4: } da \subseteq S$$

$$\text{inv1\_5: } \text{dom}(a) = da \setminus \{l\}$$

- Variable *da* denotes the set of sites **expecting a service message**
- Such nodes **cannot forward a message**

dlv\_msg

**any**  $m$  **where**

$m \in \text{dom}(p)$

$p(m) \notin da$

$p(m) = l$

**then**

$p := \{m\} \triangleleft p$

**end**

fwd\_msg

**any**  $m$  **where**

$m \in \text{dom}(p)$

$p(m) \notin da$

$p(m) \neq l$

**then**

$p(m) := d(p(m))$

**end**

- The guards are now **local**
- We can later **data-refine**  $da$  with a local **boolean variable**

```

leave_agt
  when
     $l \notin da$ 
  then
     $da := da \cup \{l\}$ 
  end

```

```

rcv_agt
  any  $s$  where
     $s \in S \setminus \{l\}$ 
     $l \in da$ 
  then
     $l := s$ 
     $a := (s \triangleleft a) \triangleleft (l \mapsto s)$ 
     $d := \{s\} \triangleleft d$ 
     $da := da \setminus \{s\}$ 
  end

```

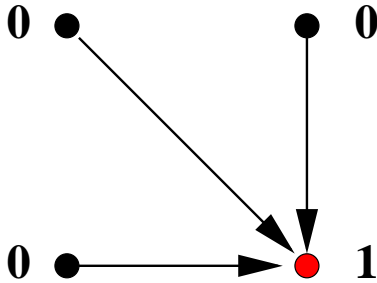
- Event **leave\_agt** is a **new event** where the set **da** is extended
- In event **rcv\_agt**, the new site location  $s$  is **removed from da**
- A previous service message to  $l$  is **removed**.

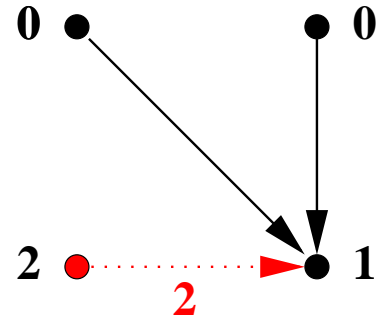
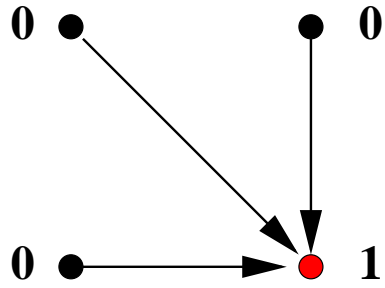
```
rcv_srv
  any s where
     $s \in \text{dom}(a)$ 
     $s \neq l$ 
  then
     $d(s) := a(s)$ 
     $a := \{s\} \triangleleft a$ 
     $da := da \setminus \{s\}$ 
  end
```

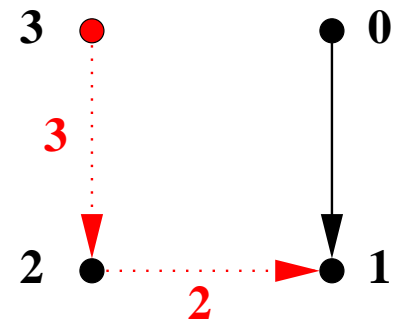
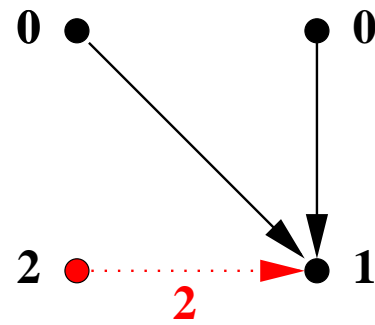
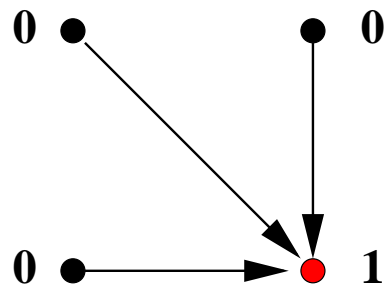
- This is a **new event**
- It corresponds to the **arrival of the service message**

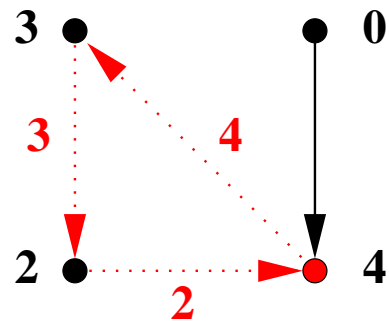
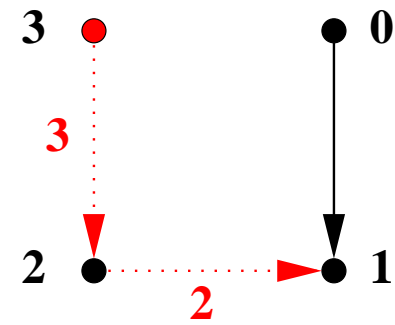
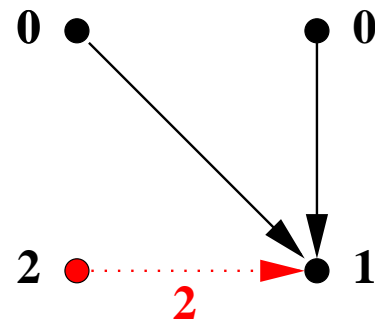
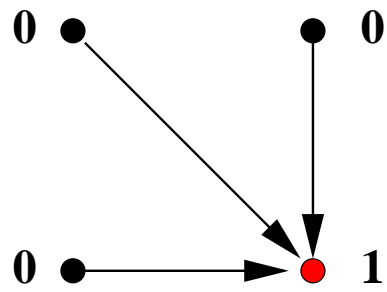
- Magic behavior when sending a new service message to  $x$ :
  - Pending service messages to  $x$  are removed
- The mobile  $\mathcal{M}$  travels with a logical clock
- Each site has a last time counter
- This counter records the “time” of the last visit of  $\mathcal{M}$

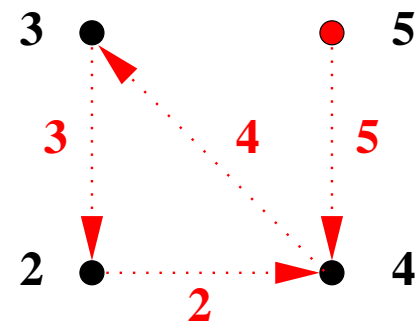
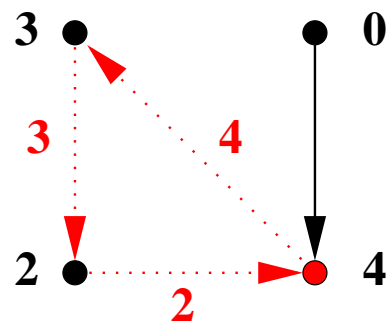
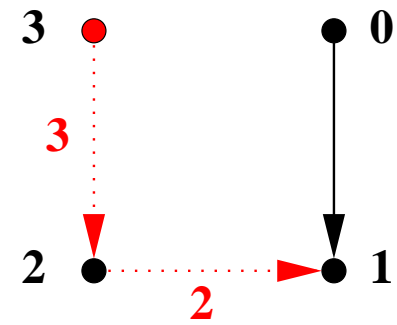
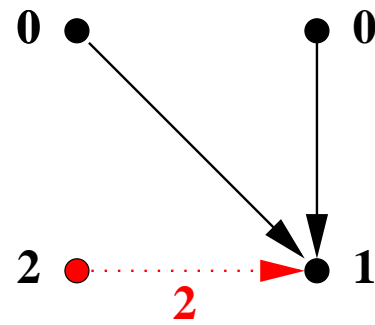
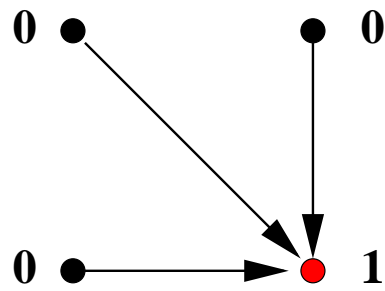
- 
- When  $\mathcal{M}$  arrives at a site  $y$ 
    - it **increments** its logical clock
    - it **stores** its incremented clock in the last time counter of  $y$
    - it **sends** a new service message to its previous location  $x$
  - The srv. msg. from  $y$  to  $x$  is **stamped** with the **new clock value**
  - When a **service message arrives at a site  $x$** , it is **accepted**
    - only if its **stamp value is greater than the time counter of  $x$**
    - the last time counter takes the **value of the stamp**

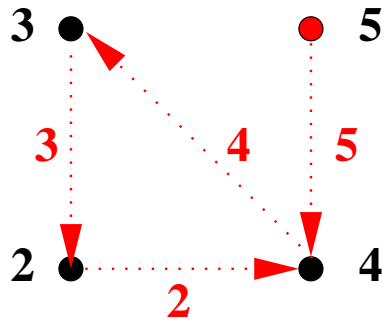


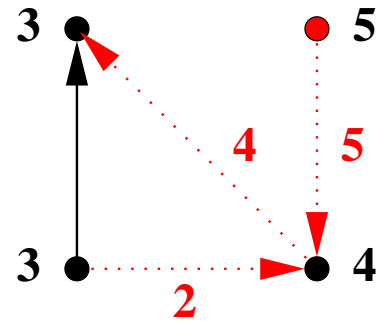
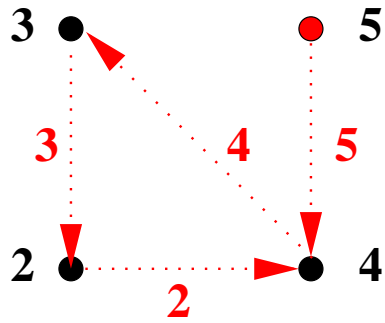




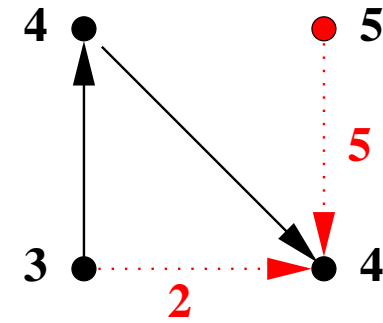
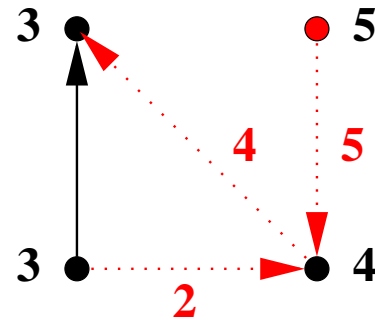
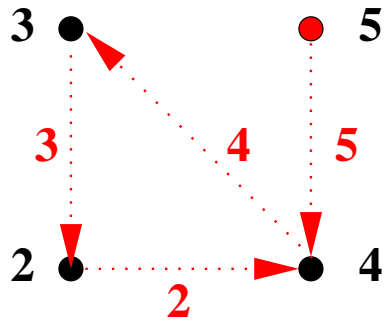




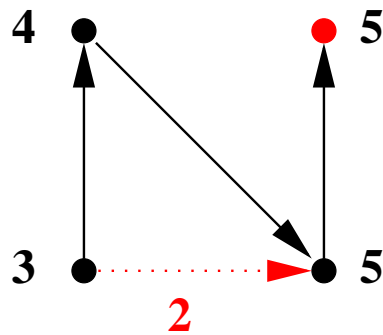
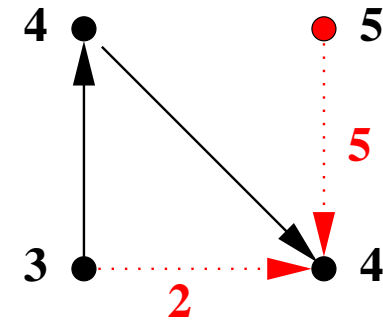
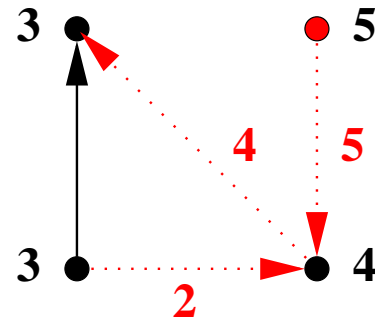
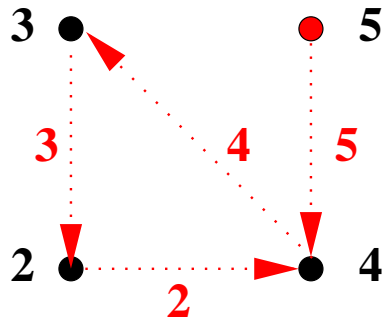




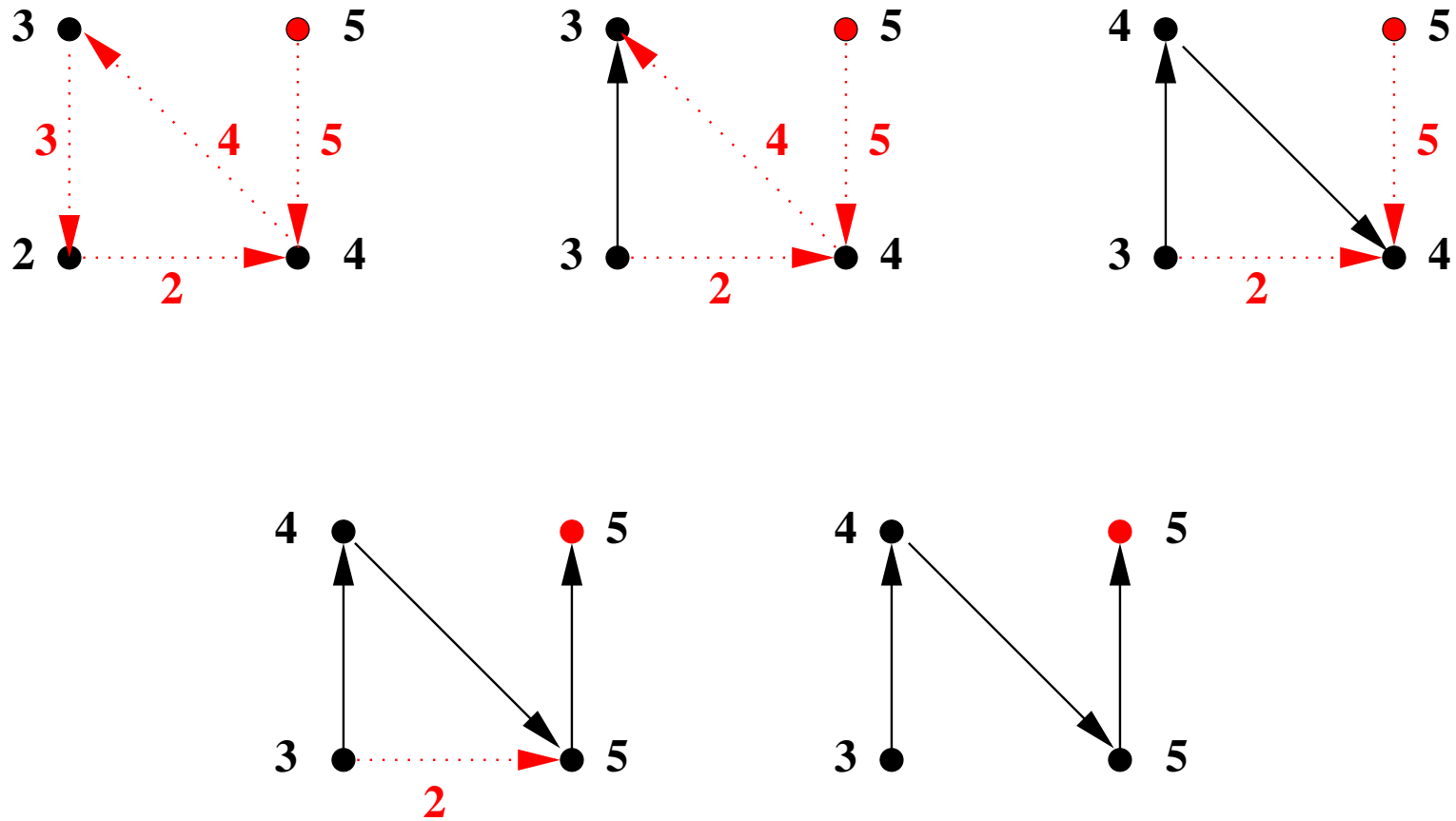
- It is **accepted**



- It is **accepted**



- It is **accepted**



- It is **rejected**

- Suppose:
  - $s1$  has emitted a service msg. to  $s$  at time  $3$
  - $s2$  has emitted a service msg. to  $s$  at time  $5$
  - $s3$  has emitted a service msg. to  $s$  at time  $9$
- This will be “recorded” in the **refined service channel** as follows:

$$s \mapsto \{3 \mapsto s1, 5 \mapsto s2, 9 \mapsto s3\}$$

- In the **abstract service channel** we had:  $s \mapsto s3$

**variables:**  $l, p, d, da,$   
 $k, t, b$

**inv2\_1:**  $k \in \mathbb{N}$

**inv2\_2:**  $t \in S \rightarrow \mathbb{N}$

**inv2\_3:**  $b \in S \rightarrow (\mathbb{N} \leftrightarrow S)$

- Variable  $k$  is the **clock** taken by the Mobile when it travels
- Variable  $t$  denotes the **time of the last visit** of the Mobile to a site
- Variable  $b$  is the new service channel, **it data-refines variable  $a$**

- An **abstract** service message is the **most recent concrete one**

$$\mathbf{inv2\_4:} \quad \forall s \cdot \left( \begin{array}{l} s \in \text{dom}(a) \\ \Rightarrow \\ \text{dom}(b(s)) \neq \emptyset \\ a(s) = b(s)(\max(\text{dom}(b(s)))) \end{array} \right)$$

$$\mathbf{inv2\_5:} \quad \forall s \cdot \left( \begin{array}{l} s \in S \\ \text{dom}(b(s)) \neq \emptyset \\ t(s) < \max(\text{dom}(b(s))) \\ \Rightarrow \\ s \in \text{dom}(a) \end{array} \right)$$

- This technical invariant will help us proving **guard strengthening** for event `rcv_srv`

**inv2\_6:**  $\forall s \cdot s \in S \Rightarrow \text{dom}(b(s)) \subseteq 0..k$

**inv2\_7:**  $t(l) = k$

**inv2\_8:**  $\forall s \cdot s \in S \setminus \{l\} \Rightarrow t(s) \leq k$

- The only **service message stamp** to a site  $s$  which is strictly **greater than the time of last visit** to that site  $s$  is the **maximum one**.

$$\text{inv2\_9: } \forall s, n \cdot \left( \begin{array}{l} s \in S \\ n \in \text{dom}(b(s)) \\ t(s) < n \\ \Rightarrow \\ n = \max(\text{dom}(b(s))) \end{array} \right)$$

- Sending the service message with the **time stamp  $k + 1$**

```
(abstract-)rcv_agt
any  $s$  where
   $s \in S \setminus \{l\}$ 
   $l \in da$ 
then
   $l := s$ 
   $a := (s \triangleleft a) \triangleleft (l \mapsto s)$ 
   $d := \{s\} \triangleleft d$ 
   $da := da \setminus \{s\}$ 
end
```

```
(concrete-)rcv_agt
any  $s$  where
   $s \in S \setminus \{l\}$ 
   $l \in da$ 
then
   $l := s$ 
   $t(s) := k + 1$ 
   $k := k + 1$ 
   $b(l) := b(l) \triangleleft \{k + 1 \mapsto s\}$ 
   $d := \{s\} \triangleleft d$ 
   $da := da \setminus \{s\}$ 
end
```

```
(abstract-)rcv_srv
  any s where
    s ∈ dom(a)
    s ≠ l
  then
    d(s) := a(s)
    a := {s} ◁ a
    da := da \ {s}
  end
```

```
(concrete-)rcv_srv
  any s, n where
    s ∈ S
    n ∈ dom(b(s))
    t(s) < n
  then
    d(s) := b(s)(n)
    t(s) := n
    da := da \ {s}
    b(s) := {n} ◁ b(s)
  end
```

$$\forall s \cdot \left( \begin{array}{l} s \in S \\ \text{dom}(b(s)) \neq \emptyset \\ t(s) < \max(\text{dom}(b(s))) \\ \Rightarrow \\ s \in \text{dom}(a) \end{array} \right)$$

$$\forall s, n \cdot \left( \begin{array}{l} s \in S \\ n \in \text{dom}(b(s)) \\ t(s) < n \\ \Rightarrow \\ n = \max(\text{dom}(b(s))) \end{array} \right)$$

**variables:**  $l, p, d, b,$   
 $dab, k, t$

**inv3\_1:**  $dab \in S \rightarrow \text{BOOL}$

**inv3\_3:**  $\forall x \cdot x \in S \Rightarrow (x \in da \Leftrightarrow dab(x) = \text{TRUE})$

```
init
   $l := il$ 
   $p := \emptyset$ 
   $d := (S \setminus \{il\}) \times \{il\}$ 
   $b := S \times \{\emptyset\}$ 
   $dab := S \times \{\text{FALSE}\}$ 
   $k := 1$ 
   $t := S \times \{0\} \triangleleft \{il \mapsto 1\}$ 
```

```
leave_agt
  when
     $dab(l) = \text{FALSE}$ 
  then
     $dab(l) := \text{TRUE}$ 
  end
```

```
rcv_agt
  any  $s$  where
     $s \in S \setminus \{l\}$ 
     $dab(l) = \text{TRUE}$ 
  then
     $l := s$ 
     $t(s) := k + 1$ 
     $k := k + 1$ 
     $b(l) := b(l) \triangleleft \{k + 1 \mapsto s\}$ 
     $d := \{s\} \triangleleft d$ 
     $dab(s) := \text{FALSE}$ 
  end
```

```
rcv_srv
  any  $s, n$  where
     $s \in S$ 
     $n \in \text{dom}(b(s))$ 
     $t(s) < n$ 
  then
     $d(s) := b(s)(n)$ 
     $t(s) := n$ 
     $dab(s) := \text{FALSE}$ 
  end
```

dlv\_msg

**any**  $m$  **where**

$m \in \text{dom}(p)$

$dab(p(m)) = \text{FALSE}$

$p(m) = l$

**then**

$p := \{m\} \triangleleft p$

**end**

fwd\_msg

**any**  $m$  **where**

$m \in \text{dom}(p)$

$dab(p(m)) = \text{FALSE}$

$p(m) \neq l$

**then**

$p(m) := d(p(m))$

**end**

Initial Model	11	0
1st Reft.	23	2
2nd Reft.	70	14
3rd Reft.	25	0
Total	129	16