Example

- A distributed program P consists of a finite set $P_1, \ldots, P_n$ of processes running on different processors and communicating via messages.
- Each process P_i generates a set E_i of events (including sending or receiving messages) that are executed according to a given ordering $\rightarrow_i$

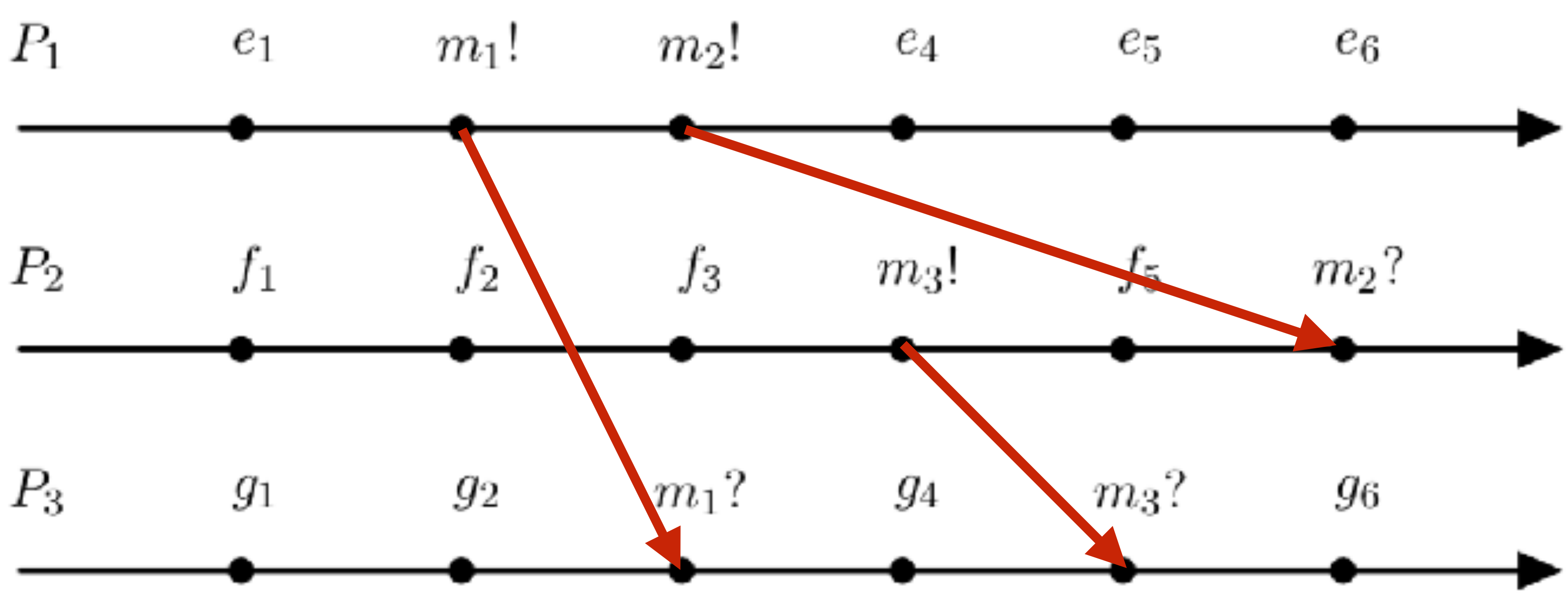
$$e_{i,0} \rightarrow_i e_{i,1} \rightarrow_i \ldots e_{i,0} \rightarrow_i e_{i,m}$$

- What is the ordering associated with execution of P knowing that a message cannot be received before it is sent?

José Fiadeiro – 2017/18

Example

- We define the execution ordering $\rightarrow$ of P as the transitive closure of:
  - For every e, f $\in$ E_i, if $e \rightarrow_i f$ then $e \rightarrow f$
  - If *m!* is the event of sending the message *m* and *m?* is the event of receiving the message *m*, then *m!*$\rightarrow$ *m?*



Example

- A *global state* of P consists of a set of events.
- A global state E is *consistent* if for every $e \in E$,
  if $f \rightarrow e$ then $f \in E$

  That is, E is a set of events that contains all the events that precede them.

José Fiadeiro – 2017/18

Lower and upper sets

- Let $\leq$ be a partial ordering on a set A and $B \subseteq A$
  - B is said to be a *lower set* if, for every $b \in B$ and $a \in A$, if $a \leq b$ then $a \in B$
  - B is said to be an *upper set* if, for every $b \in B$ and $a \in A$, if $a \geq b$ then $a \in B$
- Therefore, a global consistent state of a distributed program is a lower set of events.

Minimal and maximal elements

- Let ≤ be a partial ordering on a set A and $a \in A$
  - a is said to be a *minimal element* of A if there is no $b \in A$ such that $b < a$.
  - a is said to be a *maximal element* of A if there is no $b \in A$ such that $a < b$.

Theorem

- Every finite nonempty subset of a poset has minimal elements and maximal elements.
- If a poset is finite and non-empty then it has minimal elements and maximal elements.

José Fiadeiro – 2017/18

Proof

- Let $B=\{b_0, \ldots, b_{n-1}\}$, $n \geq 1$, and
    - $m_0=b_0$,
    - $m_k=b_k$ if $b_k < m_{k-1}$, otherwise $m_k=m_{k-1}$
- We have $m_{n-1} \leq m_{n-2} \leq \ldots \leq m_0$. We prove by contradiction that m_{n-1} is a minimal element.
    - Suppose that $b_i < m_{n-1}$ for some i; because $m_0=b_0$ it follows that $i>0$.
    - Because $b_i < m_{n-1} \leq m_i \leq m_{i-1}$ it follows that $m_i=b_i$ which would imply $b_i < m_{n-1} \leq b_i$

Greatest and least elements

- Let $\leq$ be a partial ordering on a set A and $a \in A$
  - a is said to be a *least element* of A if for all $b \in A$ $a \leq b$.
  - a is said to be a *greatest element* of A if for all $b \in A$ $b \leq a$.
- If they exist, least and greatest elements are unique.

Well-founded orderings

- We say that a partial ordering $\leq$ on a set A is *well-founded* if and only if every non-empty subset $B \subseteq A$ has a minimal element.
- If $\leq$ is a total order (a chain), we say that the poset is *well-ordered*.
- Essentially, $(A,\leq)$ is well-founded if there is no infinite descending sequence

$$\ldots < a_3 < a_2 < a_1 < a_0$$

Examples

- $(\mathcal{N},\leq)$ is well-ordered.
- $(\mathcal{Z},\leq)$ is not well-ordered.
- Every finite poset is well-founded
- The lexicographic extension of a well-ordered chain is also well-ordered.

Question

- We are given a bag containing red, yellow, and blue chips. Over that bag, we execute the following procedure:
  - If only one chip remains, we remove it and terminate.
  - If there are two or more, we remove two at random and:
    1. If one of the two is red, we keep them and execute the procedure again.
    2. If both are yellow, we put one yellow and five blue counters in the bag, and execute the procedure again.
    3. If one of the two is blue and the other is not red, we put ten red chips in the bag and execute the procedure again.
- Does this process terminate?

José Fiadeiro – 2017/18

Answer

- We represent the state of the game/process – the contents of the bag – as a triple (y,b,r) of natural numbers where y is the number of yellow chips, b the number of blue chips, and r the number of red chips.
- Each step changes the state (y,b,r) to another state (y',b',r').
- We choose the lexicographic ordering over $\mathcal{N}^3$ to order the states.
- If we can prove that $(y',b',r')\leq(y,b,r)$ then, given that the lexicographic ordering is well-founded, we conclude that execution terminates because there is no infinite descending chain.

José Fiadeiro – 2017/18

Answer

- Consider then a state (y,b,r) such that $y+b+r > 1$, i.e. the execution hasn't terminated.
  - We remove two chips at random and:

    1. *If one of the two is red, we keep them.*

       In this case we move to the state $(y,b,r-2) < (y,b,r)$

    2. *If both are yellow, we put one yellow and five blue counters in the bag.*

       In this case we move to the state $(y-1,b+5,r) < (y,b,r)$

Answer

- Consider then a state (y,b,r) such that $y+b+r > 1$, i.e. the execution hasn't terminated.
  - We remove two chips at random and:

    1. *If one of the two is red, we keep them.*

    There are three possibilities

    - The other chip is yellow. In this case, we move to the state $(y-1,b,r-1) < (y,b,r)$
    - The other chip is blue. In this case, we move to the state $(y,b-1,r-1) < (y,b,r)$
    - The other chip is red. In this case, we move to the state $(y,b,r-2) < (y,b,r)$

José Fiadeiro – 2017/18

Answer

2. *If both are yellow, we put one yellow and five blue counters in the bag.*

   In this case we move to the state $(y-1,b+5,r) < (y,b,r)$

3. *If one of the two is blue and the other is not red, we put ten red chips in the bag.*

   There are two possibilities

   - The two chips are blue. In this case, we move to the state $(y,b-2,r+10) < (y,b,r)$
   - One of the chips is blue and the other is yellow. In this case, we move to the state $(y-1,b-1,r+10) < (y,b,r)$