**Distributed algorithms TC1: Causal dependencies** Batna Oct 03 2004

**Exercice 1**

Consider 4 processes interconnected via channels and executing the following pseudo-code sequences:

|  |  |  |  |
| --- | --- | --- | --- |
| Processus : P1 | Processus : P2 | Processus : P3 | Processus : P4 |
| 1. x = 1 | 1. y = receive(P1) | 1. x = 4 | 1. z = 3 |
| 1. send(x, P2) | 1. y = y / 2 | 1. x = receive(P4) | 1. z = receive(P2) |
| 1. x = x \* 2 | 1. send(y, P4) | 1. x = 2 + x | 1. send(z, P3) |
| 1. x = receive(P2) | 1. y = receive(P3) | 1. send(x, P2) |  |
|  | 1. send(y, P1) |  |  |

**send (nb, Px)** sends the value of integer nb to process Px, **nb = receive(Px)** expects a message containing an integer from process Px. The integer received is placed in nb.

* Draw the chronogram corresponding to the parallel execution of the 4 processes.

**Exercice 2**

Consider the following chronogram (figure below) modeling the execution of a distributed calculation:

1. What is the causal past of ?
2. What is the causal future of ?
3. Is the path a causal path linking e1 to?
4. Add an event that depends causally on , and on .
5. Suggest two ways to introduce causal dependency between events and ?
6. Determine the causal history of the message .
7. Deduce whether the deliveries respect causality.
8. Give two other possible executions of the distributed calculation given above.

**Exercice**

In a distributed calculation, there are fourteen (14) events designated by . All the direct causal dependencies between these events are given as follows:



a) Draw a graph representing these dependencies.

b) Draw a chronogram representing a possible execution respecting these dependencies.