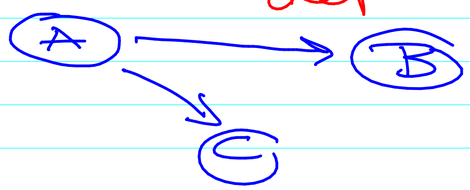
CONCURRENT ACTIVITY GRAPH

$G = (N, A)$ — concurrent activities; dependencies

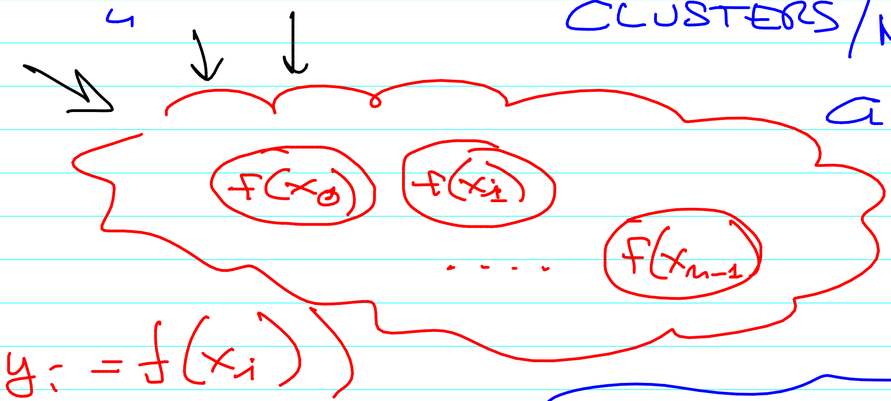


Coordination of conc. activities

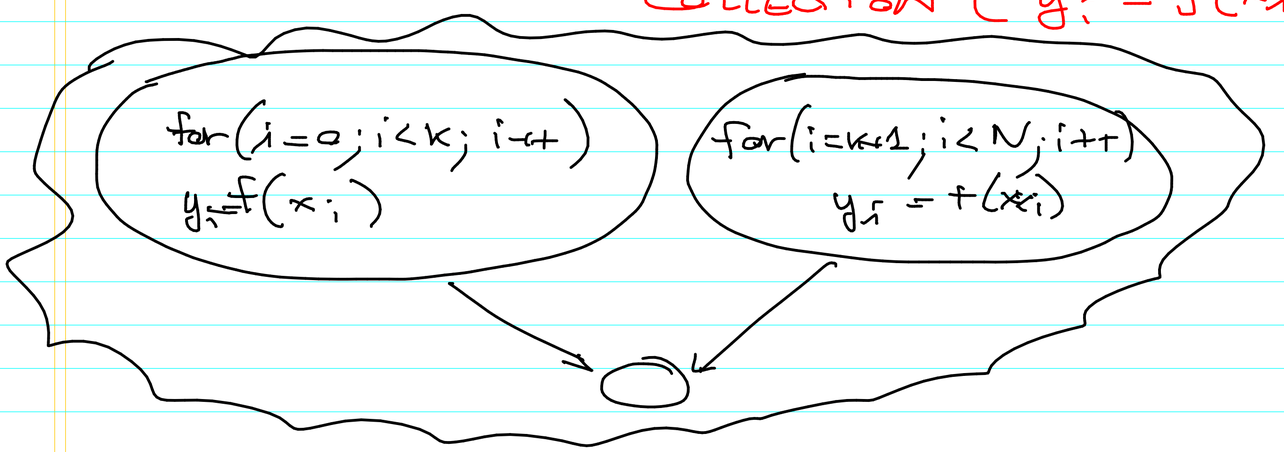
Implementation of " "

CONCURRENT ACTIVITIES

E.g. COLLECTION (x_i)



COLLECTION $(y_i = f(x_i))$



UPPER PART REASONING : FIND THE FINER GRAIN GRAPH (only depends on the application)

LOWER PART REASONING : FIND THE MORE CONVENIENT GRAPH (depends on target architecture, on network features, ...)

CONCURRENT

A concurrent B

A may happen in parallel B

PARALLEL

A parallel B

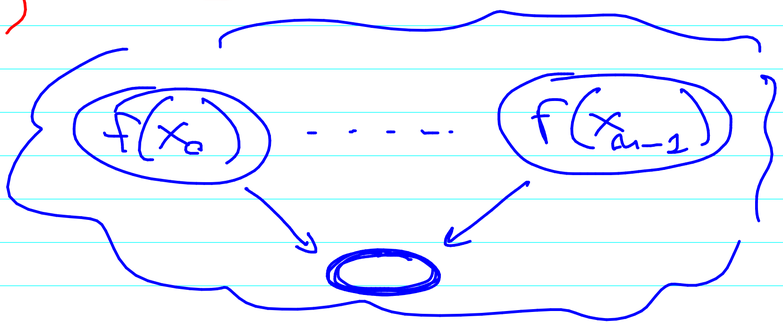
A happens while B is happening

DISTRIBUTED VS PARALLEL

CLUSTERS/NETWORKS VS. MULTI CORES

GRAIN

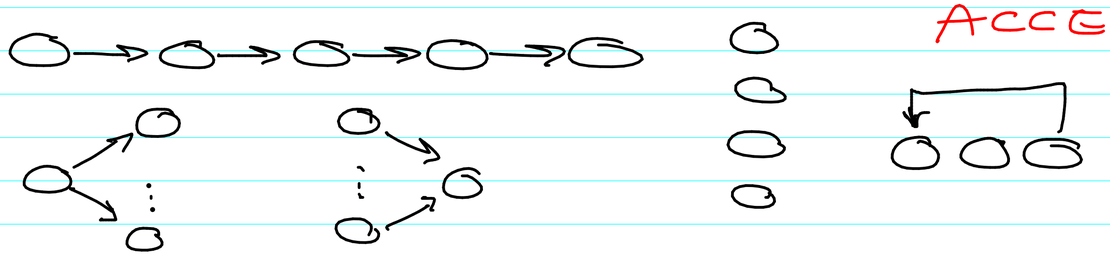
$\frac{\text{time for computing}}{\text{time for comms}}$

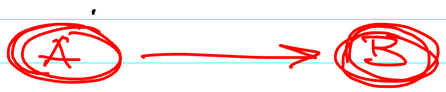


"FEW USEFULL PATTERNS USED TO COORDINATE CONCURRENT ACTIVITIES"

COORDINATION of CONCURRENT ACTIVITIES

- SHARED DATA ACCESS
- SYNCHRONIZATION

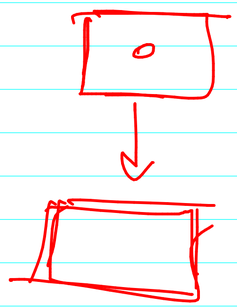




control dependencies
data dependencies

only data dep ⟹ DATA FLOW GRAPH

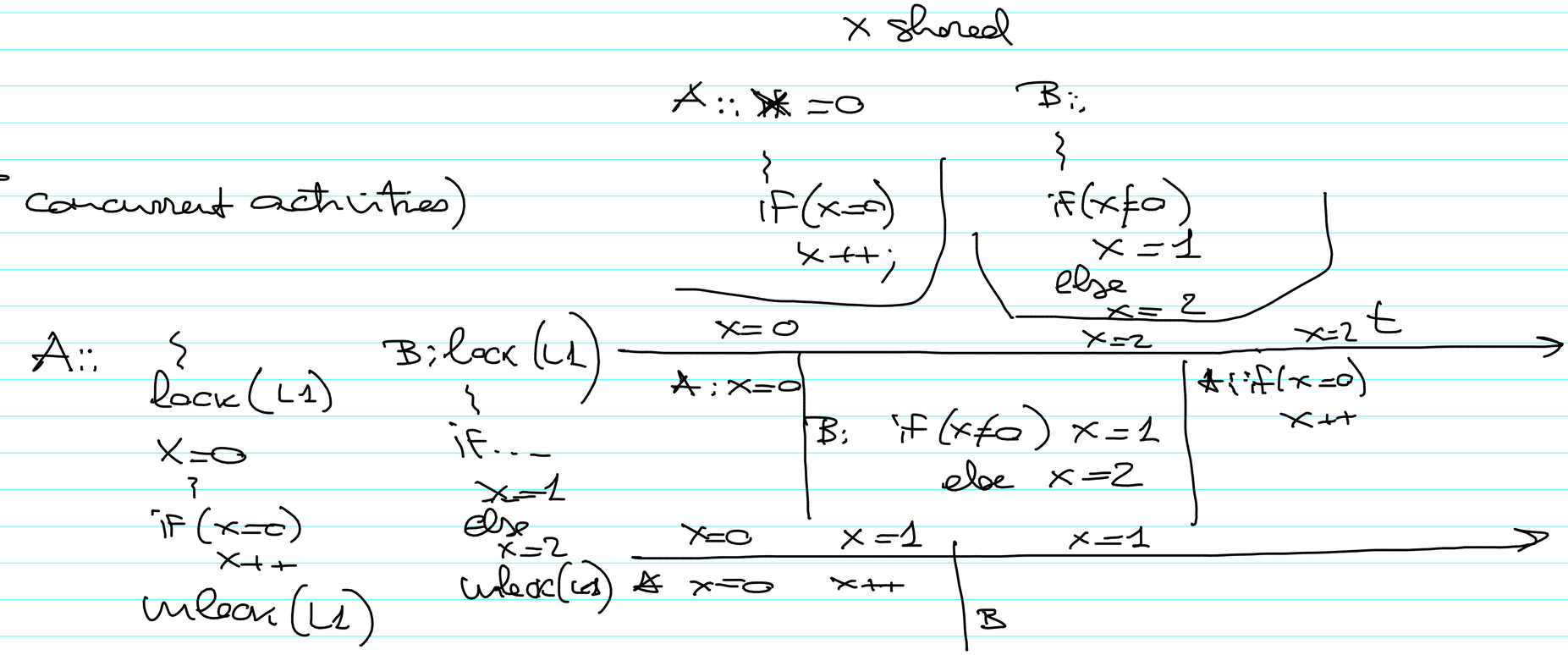
only control dep ⟹ WORKFLOWS



List of PROBLEMS to be solved

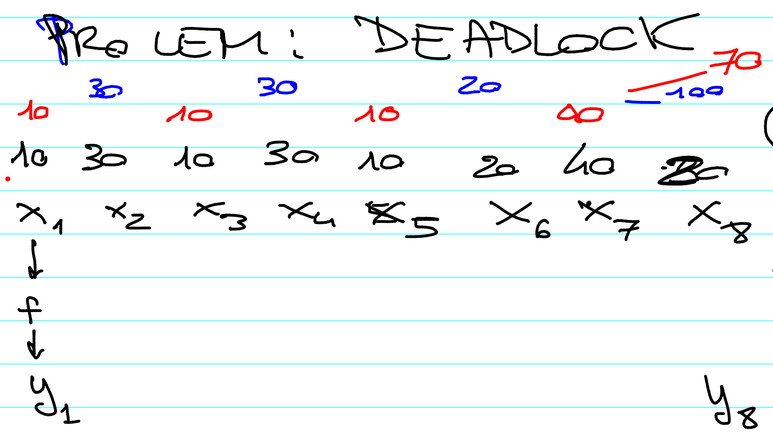
RACE CONDITIONS
(problems relative to relative speed of concurrent activities)

MUTEXES
LOCKS



Sequential consistency

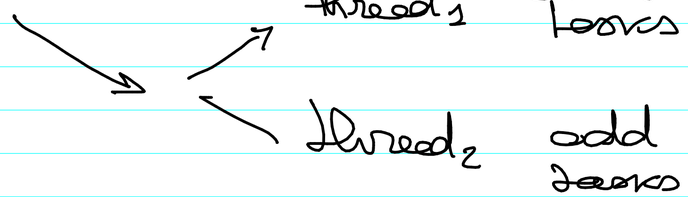
PROBLEM: DEADLOCK

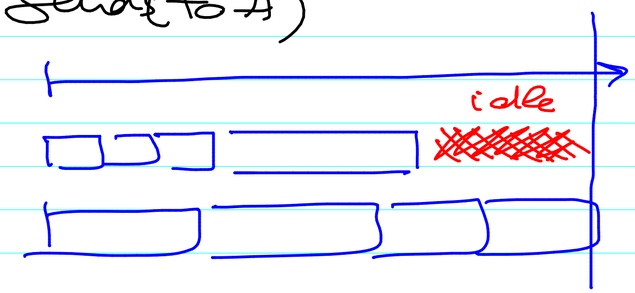


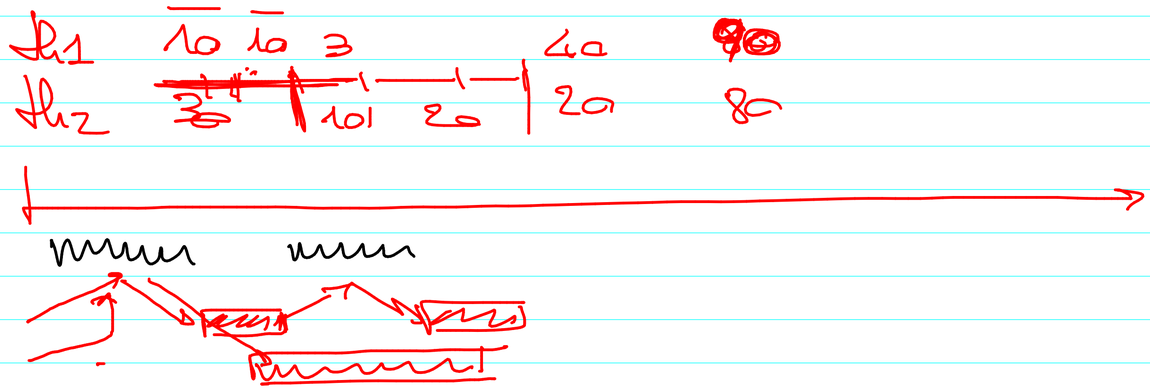
A: Receive(from B)
Send(to B)

B: Receive(from A)
Send(to A)

$170/2 = 85$







LOAD BALANCING