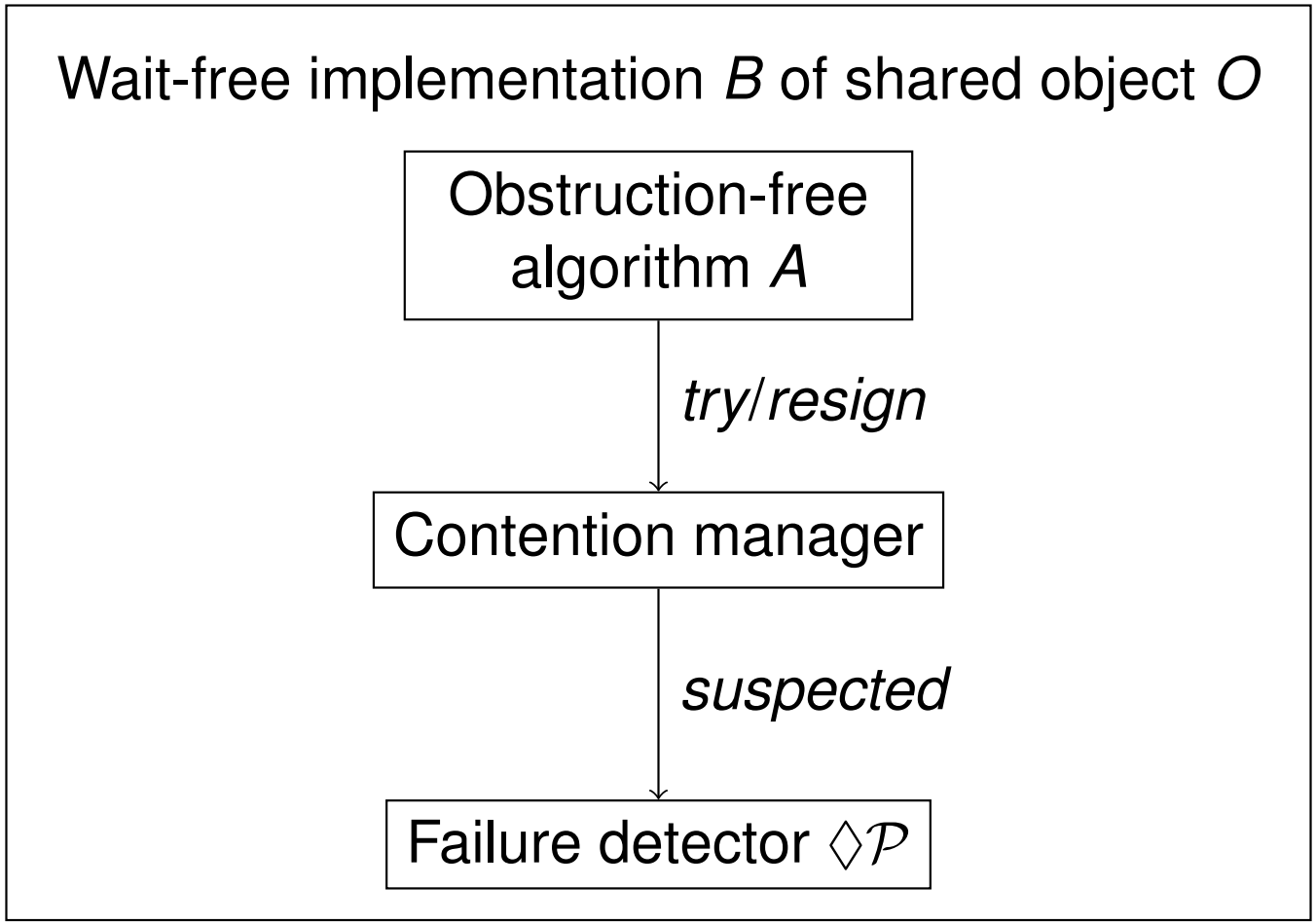
A Solution for Exercise 6

EPFL, LPD

Concurrent Algorithms 2010

The big picture



Assumptions

Algorithm A must communicate with a contention manager $\Rightarrow$calls *try* and *resign*:

- try_i is called always before an operation starts, and possibly many times within the operation,
- $resign_i$ is called *only* immediately before the operation returns,
- If a process p_i is correct but never returns from an operation then p_i calls try_i infinitely many times.

Failure detector $\Diamond\mathcal{P}$

An eventually perfect failure detector $\Diamond\mathcal{P}$ maintains, at every process p_i, a set $suspected_i$ of suspected processes. $\Diamond\mathcal{P}$ guarantees that eventually, after some unknown time, the following conditions are satisfied:

1. Every correct process permanently suspects every crashed process,
2. No correct process is ever suspected by any correct process.

A wait-free contention manager

uses: $T[1, \ldots, N]$—array of registers
initially: $T[1, \ldots, N] \leftarrow \perp$

upon try_i **do**
  if $T[i] = \perp$ **then** $T[i] \leftarrow$ `GetTimestamp()`
  repeat
    $sact_i \leftarrow \{ p_j \mid T[j] \neq \perp \wedge p_j \notin \Diamond\mathcal{P}.suspected_i \}$
    $leader_i \leftarrow$ the process in $sact_i$ with the lowest timestamp $T[leader_i]$
  until $leader_i = p_i$

upon $resign_i$ **do**
  $T[i] \leftarrow \perp$

Properties of `GetTimestamp()`

Timestamps:

- Have to be unique
- Should also be increasing

A solution: weak counter (from registers) | process id