

A Solution for the Exercise 3

Michał Kapałka

EPFL, LPD

STiDC'06, 28.XI 2006

2-consensus on an initialized queue

procedure $cons_i(Q, R, val_i)$

$R[i] \leftarrow val_i$

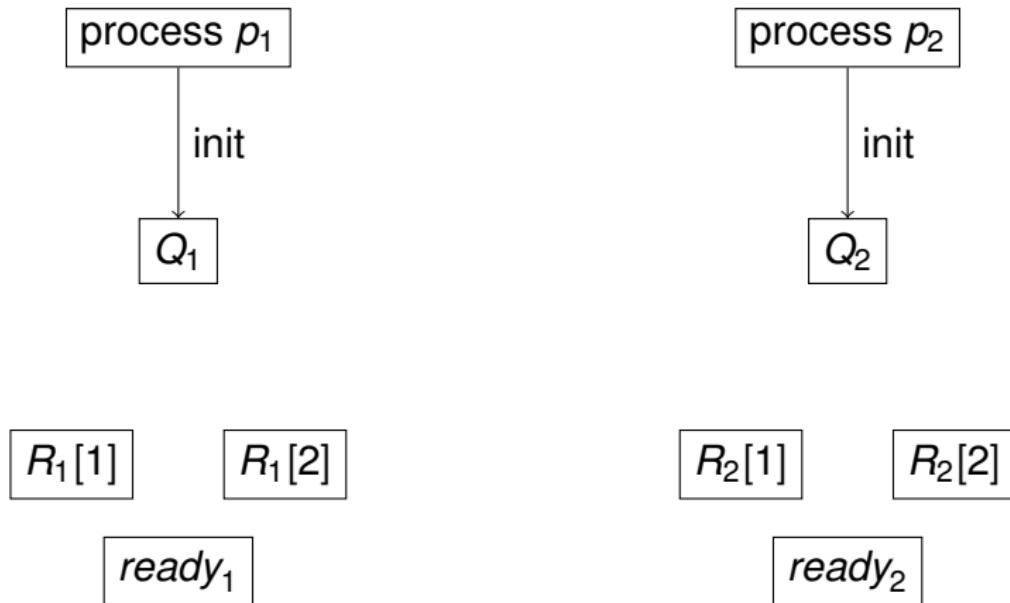
$q_i \leftarrow Q.deq()$

if $q_i = "winner"$ **then return** val_i

else return $R[1 - i]$

Q initialized to $\langle "winner", "loser" \rangle$.

2-consensus on uninitialized queues



2-consensus on uninitialized queues

procedure $propose_i(val_i)$

$Q_i.enq("winner")$

$Q_i.enq("loser")$

$ready_i \leftarrow true$

for $k \leftarrow 1, 2$ **do**

if $ready_k$ **then** $val_i \leftarrow cons_i(Q_k, R_k, val_i)$

return val_i

Uses: queues $Q_{1,2}$ (initially empty), registers $R_{1,2}[1, 2]$ and registers $ready_{1,2}$ (initialized to *false*).

n -consensus on uninitialized objects

procedure $propose_i(val_i)$

$Q_i.init(q)$

$ready_i \leftarrow true$

for $k \leftarrow 1, \dots, n$ **do**

if $ready_k$ **then** $val_i \leftarrow cons_i(Q_k, val_i)$

return val_i

Where: $Q_{1,\dots,n}$ are instances of Q initialized to \perp and $cons_i(Q_k, val_i)$ is an n -consensus algorithm (at process p_i) that uses object Q_k initialized to state q .