

## Exercise 10

### 1 Atomic Register using Total-Order Broadcast

Consider the following emulation of an  $(N, N)$ -atomic register, instance  $nnar$  (Module 4.3 [CGR11, p. 161]) using total-order broadcast, instance  $tob$ , also called *atomic broadcast* (Module 6.1 [CGR11, p. 283]). Every process stores a copy  $x$  of the value of the register.

- To handle a  $\langle nnar, Read \rangle$  event on the register, use total-order broadcast to send the request, that is, trigger an event  $\langle tob, Broadcast \mid [READ] \rangle$ .
- Upon receiving such a request, when an event  $\langle tob, Deliver \mid p, [READ] \rangle$  occurs, every process checks if  $p = self$ ; if this holds, it returns  $x$  in an event  $\langle nnar, ReadReturn \mid x \rangle$ .
- To handle a  $\langle nnar, Write \mid v \rangle$  event on the register with value  $v$ , broadcast the request by triggering an event  $\langle tob, Broadcast \mid [WRITE, v] \rangle$ .
- Upon receiving such a request, that is, when an event  $\langle tob, Deliver \mid p, [WRITE, v] \rangle$  occurs, every process sets  $x := v$ , checks if  $p = self$ , and triggers  $\langle nnar, WriteReturn \rangle$  if this holds.

Consider a modification of this algorithm such that read requests from the register are executed locally, that is, a read directly returns  $x$  without  $tob$ -broadcasting the request. This is often done in practice for increasing efficiency. Does the modified algorithm still implement an  $(N, N)$ -atomic register? Justify your answer.

### 2 ZooKeeper

Yahoo!'s ZooKeeper (<http://zookeeper.apache.org/>) is a prominent distributed coordination system for web-application services, built from atomic broadcast. Read the following paper:

P. Hunt, M. Konar, F. P. Junqueira, and B. Reed, "ZooKeeper: Wait-free coordination for internet-scale systems," in *Proc. USENIX Annual Technical Conference*, 2010.

[http://www.usenix.org/event/usenix10/tech/full\\_papers/Hunt.pdf](http://www.usenix.org/event/usenix10/tech/full_papers/Hunt.pdf)

Background information is given in a tutorial on ZooKeeper:

<https://cwiki.apache.org/confluence/display/ZOOKEEPER/EurosystTutorial>

Optionally, find here the details of the ZooKeeper Atomic Broadcast protocol (for very interested students):

F. Junqueira, B. Reed, and M. Serafini, “Zab: High-performance broadcast for primary-backup systems,” in *Proc. 41st International Conference on Dependable Systems and Networks (DSN-DCCS)*, 2011.

<http://doi.ieeecomputersociety.org/10.1109/DSN.2011.5958223>