Algorithm – Regular Register

Algorithm 1 Read-One Write-All

Implements:

(1, N)-RegularRegister, instance onrr.

Uses:

BestEffortBroadcast, instance beb.
PerfectPointToPointLinks, instance pp2p.
PerfectFailureDetector, instance P.

1: upon event (Init) do
2: \( \text{val} := \bot \)
3: \( \text{correct} := \Pi \)
4: \( \text{writeset} := \emptyset \)
5: upon event (P, Crash \mid p) do
6: \( \text{correct} := \text{correct} \cup \{p\} \)
7: upon event (onrr, Read) do
8: \( \text{trigger} \langle \text{onrr, ReadReturn} \mid \text{val} \rangle \)
9: upon event (onrr, Write \mid v) do
10: \( \text{trigger} \langle \text{beb, Broadcast} \mid \text{[Write, v]} \rangle \)
11: upon event (beb, Deliver \mid p, [Write, v]) do
12: \( \text{val} := v \)
13: \( \text{trigger} \langle \text{pp2p, Send} \mid p, \text{ACK} \rangle \)
14: upon event (pp2p, Deliver \mid p, ACK) do
15: \( \text{writeset} := \text{writeset} \cup \{p\} \)
16: upon event (correct \subseteq \text{writeset}) do
17: \( \text{writeset} := \emptyset \)
18: \( \text{trigger} \langle \text{onrr, WriteReturn} \mid \rangle \)
Algorithm 2 Majority Voting

Implements:

$(1, N)$-RegularRegister, instance $onrr$.

Uses:

BestEffortBroadcast, instance $beb$.
PerfectPointToPointLinks, instance $pp2p$.

1: upon event (Init) do
2: $(ts, val) := (0, \perp)$
3: $wts := 0$
4: $acks := 0$
5: rid := 0
6: $\forall p \in \Pi \; readlist[p] := \perp$
7: upon event (onrr, Read) do
8: rid := rid + 1
9: $\forall p \in \Pi \; readlist[p] := \perp$
10: trigger (beb, Broadcast | [Read, rid])
11: upon event (onrr, Write | v) do
12: $wts := wts + 1$
13: $acks := 0$
14: trigger (beb, Broadcast | [Write, wts, v])
15: upon event (beb, Deliver | p, [Read, r]) do
16: trigger (pp2p, Send | p, [Value, r, ts, val])
17: upon event (beb, Deliver | p, [Write, ts', v']) do
18: if $ts' > ts$ then
19: $(ts, val) := (ts', v')$
20: trigger (pp2p, Send | p, [Ack, ts'])
21: upon event (pp2p, Deliver | p, [Value, r, ts', v']) do
22: if $r = rid$ then
23: $readlist[p] := (ts', v')$
24: if $|readlist| > \frac{N}{2}$ then \hspace{1cm} \triangleright Where $N = |\Pi|$.
25: trigger (onrr, ReadReturn | HIGHESTVal(readlist))
26: $\forall q \in \Pi \; readlist[q] := \perp$
27: upon event (pp2p, Deliver | p, [Ack, ts']) do
28: if $ts' = wts$ then
29: $acks := acks + 1$
30: if $acks > \frac{N}{2}$ then
31: $acks := 0$
32: trigger (onrr, WriteReturn | )