MODULE $EWD998$

TLA+ specification of an algorithm for distributed termination detection on a ring, due to *Shmuel Safra*, published as *EWD* 998: *Shmuel Safra*'s version of termination detection. https:=*www:cs:utexas:edu*/users/*EWD*/*ewd*09*xx*/*EWD*998:*PDF*

EXTENDS *Integers*, *FiniteSets*, *Functions*, *SequencesExt*, *Randomization*

CONSTANT
@type: *Int*;
N

ASSUME $NAssumption \triangleq N \in Nat \setminus \{0\}$ At least one node.

$Node \triangleq 0 \,..\, N-1$
$Color \triangleq \{\text{“white”}, \text{“black”}\}$
$Token \triangleq [pos : Node,\ q : Int,\ color : Color]$

VARIABLES
@type: $Int \to Bool$;
$active$, activation status of nodes
@type: $Int \to Str$;
$color$, color of nodes
@type: $Int \to Int$;
$counter$, nb of sent messages $-$ nb of rcvd messages per node
@type: $Int \to Int$;
$pending$, nb of messages in transit to node
@type: [*pos*: *Int*, *q*: *Int*, color: *Str*];
$token$ token structure

$vars \triangleq \langle active,\ color,\ counter,\ pending,\ token \rangle$

$TypeOK \triangleq$
$\land\ active \in [Node \to \text{BOOLEAN}\ ]$
$\land\ color \in [Node \to Color]$
$\land\ counter \in [Node \to Int]$
$\land\ pending \in [Node \to Nat]$
$\land\ token \in Token$

$Init \triangleq$
*EWD*840 but nodes
$\land\ active \in [Node \to \text{BOOLEAN}\ ]$
$\land\ color \in [Node \to Color]$
Rule 0
$\land\ counter = [i \in Node \mapsto 0]$ c properly initialized
$\land\ pending = [i \in Node \mapsto 0]$
$\land\ token \in [pos : \{0\},\ q : \{0\},\ color : \{\text{“black”}\}]$

$InitiateProbe \triangleq$

Rules $1 + 5 + 6$
$\land\ token.pos = 0$
$\land$ previous round not conclusive if:
$\quad \lor\ token.color = \text{“black”}$
$\quad \lor\ color[0] = \text{“black”}$
$\quad \lor\ counter[0] + token.q > 0$
$\land\ token^{\theta} = [pos \mapsto N - 1,\ q \mapsto 0,\ color \mapsto \text{“white”}]$
$\land\ color^{\theta} = [color \text{ EXCEPT } ![0] = \text{“white”}]$
The state of the nodes remains unchanged by token-related actions.
$\land$ UNCHANGED $\langle active,\ counter,\ pending \rangle$

$PassToken(i) \triangleq$
Rules $2 + 4 + 7$
$\land\ \neg active[i]$ If machine i is active, keep the token.
$\land\ token.pos = i$
$\land\ token^{\theta} = [pos \mapsto token.pos - 1,$
$\qquad q \mapsto token.q + counter[i],$
$\qquad color \mapsto \text{IF } color[i] = \text{“black”} \text{ THEN “black” ELSE } token.color]$
$\qquad color \mapsto color[i]\ ]$
$\land\ color^{\theta} = [color \text{ EXCEPT } ![i] = \text{“white”}]$
The state of the nodes remains unchanged by token-related actions.
$\land$ UNCHANGED $\langle active,\ counter,\ pending \rangle$

$System \triangleq \lor\ InitiateProbe$
$\qquad\qquad \lor \exists\, i \in Node \setminus \{0\} : PassToken(i)$

$SendMsg(i) \triangleq$
Only allowed to send msgs if node i is active.
$\land\ active[i]$
Rule 0
$\land\ counter^{\theta} = [counter \text{ EXCEPT } ![i] = @ + 1]$
Non-deterministically choose a receiver node.
$\land \exists\, j \in Node \setminus \{i\} : pending^{\theta} = [pending \text{ EXCEPT } ![j] = @ + 1]$
Note that we don't blacken node i as in $EWD840$ if node i
sends a message to node j with $j > i$
$\land$ UNCHANGED $\langle active,\ color,\ token \rangle$

$RecvMsg(i) \triangleq$
$\land\ pending[i] > 0$
$\land\ pending^{\theta} = [pending \text{ EXCEPT } ![i] = @ - 1]$
Rule 0
$\land\ counter^{\theta} = [counter \text{ EXCEPT } ![i] = @ - 1]$
Rule 3
$\land\ color^{\theta} = [color \text{ EXCEPT } ![i] = \text{“black”}]$

Receipt of a message activates i.

$\land\ active^{\theta} = [active \text{ EXCEPT } ![i] = \text{TRUE}]$
$\land\ \text{UNCHANGED } \langle token \rangle$

$Deactivate(i) \triangleq$
$\land\ active[i]$
$\land\ active^{\theta} = [active \text{ EXCEPT } ![i] = \text{FALSE}]$
$\land\ \text{UNCHANGED } \langle color, counter, pending, token \rangle$

$Environment \triangleq \exists\, i \in Node : SendMsg(i) \lor RecvMsg(i) \lor Deactivate(i)$

$Next \triangleq$
$System \lor Environment$

$Spec \triangleq Init \land \mathcal{2}[Next]_{vars} \land \text{WF}_{vars}(System)$

Bound the otherwise infinite state space that TLC has to check.

$StateConstraint \triangleq$
$\land\ \forall\, i \in Node : counter[i] \leq 3 \land pending[i] \leq 3$
$\land\ token.q \leq 9$

Main safety property: if there is a white token at node 0 and there are no in-flight messages then every node is inactive.

$terminationDetected \triangleq$
$\land\ token.pos = 0$
$\land\ token.color = \text{“white”}$
$\land\ token.q + counter[0] = 0$
$\land\ color[0] = \text{“white”}$
$\land\ \neg active[0]$
$\land\ pending[0] = 0$

Sum of the values $f[x]$, for $x \in S \subseteq \text{DOMAIN } f$.

$Sum(f, S) \triangleq FoldFunctionOnSet(+, 0, f, S)$

The number of messages on their way. “in-flight”

$B \triangleq Sum(pending, Node)$

The system has terminated if no node is active and there are no in-flight messages.

$Termination \triangleq$
$\land\ \forall\, i \in Node : \neg active[i]$
$\land\ B = 0$

$TerminationDetection \triangleq$
$terminationDetected \Rightarrow Termination$

Interval of nodes between a and b: this is just $a \mathinner{..} b$, but the following definition helps *Apalache* to construct a bounded set.

$Rng(a, b) \triangleq \{i \in Node : a \leq i \land i \leq b\}$

Safra's inductive invariant

$Inv \triangleq$
$\land\ P0{::}\ B = Sum(counter, Node)$
(Ai: $t < i < N$: machine $nr{:}i$ is passive) $\land$
(Si: $t < i < N$: $ci{:}i$) $= q$
$\land\ \lor\ P1{::}\ \land \forall\, i \in Rng(token.pos + 1,\ N - 1) : active[i] = \text{FALSE}$ machine $nr{:}i$ is passive
$\land$ IF $token.pos = N - 1$
THEN $token.q = 0$
ELSE $token.q = Sum(counter,\ Rng(token.pos + 1,\ N - 1))$
(Si: $0 \leq i \leq t$: $c{:}i) + q > 0$.
$\lor\ P2{::}\ Sum(counter,\ Rng(0,\ token.pos)) + token.q > 0$
Ei: $0 \leq i \leq t$: machine $nr{:}i$ is black.
$\lor\ P3{::}\ \exists\, i \in Rng(0,\ token.pos) : color[i] =$ "black"
The token is black.
$\lor\ P4{::}\ token.color =$ "black"

The inductive invariant combined with the type invariant

$TypedInv \triangleq$
$\land\ TypeOK$
$\land\ Inv$

Liveness property: termination is eventually detected.

$Liveness \triangleq$
$Termination\ \ ;\ \ terminationDetected$

The algorithm implements the specification of termination detection in a ring with asynchronous communication. The parameters of module *AsyncTerminationDetection* are instantiated by the symbols of the same name of the present module.

$TD \triangleq$ INSTANCE $AsyncTerminationDetection$

$TDSpec \triangleq TD!Spec$

THEOREM $Spec \Rightarrow TDSpec$

Checked with *TLC* in 01/2021 with two cores on a fairly modern desktop and the given state constraint *StateConstraint* above:

N	Diameter	Distinct States	States	Time

| 3 | 60 | 1:3m | 10:1m | 42 s |
| 4 | 105 | 219m | 2:3b | 50 m |