MODULE *TimeClock*

EXTENDS *Naturals*, *Sequences* Import *Naturals* and *Sequences* for *Nat* and *Append*, *Head*, *Tail*

CONSTANT *Proc*, $\_ \ll \_$, *NumofNats*

ASSUME $\forall\, p \in Proc :\ \land \neg p \ll p$
$\land \forall\, q \in Proc \setminus \{p\} : (p \ll q) \lor (q \ll p)$
$\land \forall\, q,\, r \in Proc : (p \ll q) \land (q \ll r) \Rightarrow (p \ll r)$

$a \prec b \triangleq \lor\, a.TS < b.TS$
$\lor\, (a.TS = b.TS) \land (a.proc \ll b.proc)$

VARIABLES *state*, *msgQ*, *reqSet*, *clock*, *lastTSent*, *lastTRcvd*

$vars \triangleq \langle state,\, msgQ,\, reqSet,\, clock,\, lastTSent,\, lastTRcvd \rangle$

$Init \triangleq \land\, state = [p \in Proc \mapsto \text{“idle”}]$
$\land\, msgQ = [p \in Proc \mapsto [q \in Proc \setminus \{p\} \mapsto \langle\rangle]]$
$\land\, reqSet = [p \in Proc \mapsto \{\}]$
$\land\, clock \in [Proc \to 0 \,..\, NumofNats]$
$\land\, lastTSent = [p \in Proc \mapsto [q \in Proc \setminus \{p\} \mapsto 0]]$
$\land\, lastTRcvd = [p \in Proc \mapsto [q \in Proc \setminus \{p\} \mapsto 0]]$

$Request(p) \triangleq$
$\land\, state[p] = \text{“idle”}$
$\land\, state' = [state \text{ EXCEPT } ![p] = \text{“waiting”}]$
$\land \exists\, n \in 0 \,..\, NumofNats :$
$\land\, clock' = [clock \text{ EXCEPT } ![p] = n]$
$\land\, n > clock[p]$
$\land$ LET $msg \triangleq [TS \mapsto n,\, proc \mapsto p,\ \ cmd \mapsto \text{“acquire”}]$
IN $\land\, msgQ' = [msgQ \text{ EXCEPT } ![p] =$
$[q \in Proc \setminus \{p\} \mapsto Append(@[q],\, msg)]]$
$\land\, reqSet' = [reqSet \text{ EXCEPT } ![p] = @ \cup \{msg\}]$
$\land\, lastTSent' = [lastTSent \text{ EXCEPT } ![p] = \ [q \in Proc \setminus \{p\} \mapsto n]]$
$\land$ UNCHANGED *lastTRcvd*

$Acquire(p) \triangleq$
LET $pReq \triangleq$ CHOOSE $req \in reqSet[p] : req.proc = p$
IN $\land\, state[p] \ = \text{“waiting”}$
$\land \forall\, req \in reqSet[p] \setminus \{pReq\} : pReq \prec req$

$$
\begin{array}{l}
\quad \land \forall\, q \in Proc \setminus \{p\} : pReq \prec [TS \mapsto lastTRcvd[p][q] + 1,\ proc \mapsto q] \\
\quad \land\ state' \quad\ = [state \text{ EXCEPT } ![p] = \text{“owner”}] \\
\quad \land\ reqSet' \quad = [reqSet \text{ EXCEPT } ![p] = @ \setminus \{pReq\}] \\
\quad \land \text{UNCHANGED } \langle msgQ,\ clock,\ lastTSent,\ lastTRcvd \rangle
\end{array}
$$

$$
\begin{array}{l}
Release(p) \triangleq \\
\quad \land\ state[p] = \text{“owner”} \\
\quad \land\ state' = [state \text{ EXCEPT } ![p] = \text{“idle”}] \\
\quad \land \text{LET } msg \triangleq [TS \mapsto clock[p],\ proc \mapsto p,\ cmd \mapsto \text{“release”}] \\
\qquad \text{IN} \quad msgQ' = [msgQ \text{ EXCEPT } ![p] = \\
\qquad\qquad\qquad\qquad [q \in Proc \setminus \{p\} \mapsto Append(@[q],\ msg)]] \\
\quad \land\ lastTSent' = [lastTSent \text{ EXCEPT } ![p] = \ [q \in Proc \setminus \{p\} \mapsto clock[p]]] \\
\quad \land \text{UNCHANGED } \langle clock,\ lastTRcvd,\ reqSet \rangle
\end{array}
$$

$$
\begin{array}{l}
RcvMsg(p,\ q) \triangleq \\
\quad \text{LET } msg \quad\ \triangleq\ Head(msgQ[q][p]) \\
\qquad msgQTail \triangleq\ [msgQ \text{ EXCEPT } ![q][p] = Tail(@)] \\
\qquad ack \quad\ \triangleq\ [TS \mapsto clock'[p],\ proc \mapsto p,\ cmd \mapsto \text{“ack”}] \\
\quad \text{IN} \quad \land\ msgQ[q][p] \neq \langle\rangle \\
\qquad\quad \land\ clock' = [clock \text{ EXCEPT } ![p] = \text{IF } msg.TS > @ \text{ THEN } msg.TS \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{ELSE } @] \\
\qquad\quad \land \text{IF } \land\ msg.cmd = \text{“acquire”} \\
\qquad\qquad\qquad\ \land\ [TS \mapsto lastTSent[p][q] + 1,\ proc \mapsto p] \prec msg \\
\qquad\qquad \text{THEN } \land\ msgQ' = [msgQTail \text{ EXCEPT } ![p][q] = \ Append(@,\ ack)] \\
\qquad\qquad\qquad\quad\ \land\ lastTSent' = [lastTSent \text{ EXCEPT } ![p][q] = clock'[p]] \\
\qquad\qquad \text{ELSE } \ \land\ msgQ' = msgQTail \\
\qquad\qquad\qquad\quad\ \land \text{UNCHANGED } lastTSent \\
\qquad\quad \land\ lastTRcvd' = [lastTRcvd \text{ EXCEPT } ![p][q] = msg.TS] \\
\qquad\quad \land\ reqSet' = [reqSet \text{ EXCEPT } ![p] = \\
\qquad\qquad\qquad\qquad \text{CASE } msg.cmd = \text{“acquire”} \to @ \cup \{msg\} \\
\qquad\qquad\qquad\qquad\quad \square \quad msg.cmd = \text{“release”} \to \{m \in @ : m.proc \neq q\} \\
\qquad\qquad\qquad\qquad\quad \square \quad msg.cmd = \text{“ack”} \quad\ \to @] \\
\qquad\quad \land \text{UNCHANGED } state
\end{array}
$$

$$
\begin{array}{l}
Tick(p) \triangleq \ \land \exists\, n \in 0\,..\,NumofNats : \ \land\ n > clock[p] \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \land\ clock' = [clock \text{ EXCEPT } ![p] = n] \\
\qquad\qquad\quad \land \text{UNCHANGED } \langle state,\ msgQ,\ reqSet,\ lastTSent,\ lastTRcvd \rangle
\end{array}
$$

$$
\begin{array}{l}
Next \triangleq \exists\, p \in Proc : \ \lor\ Request(p) \lor Acquire(p) \lor Release(p) \\
\qquad\qquad\qquad\qquad\quad \lor \exists\, q \in Proc \setminus \{p\} : RcvMsg(p,\ q)
\end{array}
$$

$$\lor\ Tick(p)$$

$$Liveness \triangleq \forall\, p \in Proc : \land \mathrm{WF}_{vars}(Acquire(p)) \\ \land \forall\, q \in Proc \setminus \{p\} : \mathrm{WF}_{vars}(RcvMsg(p,\, q))$$

$$Constraint \triangleq \forall\, p \in Proc : clock[p] < NumofNats$$

$$Mutex \triangleq \forall\, p,\, q \in Proc : p \neq q \Rightarrow (\{state[p],\, state[q]\} \neq \{\text{“owner”}\})$$

$$EventuallyAcquires \triangleq \forall\, p \in Proc : (state[p] = \text{“waiting”}) \leadsto (state[p] = \text{“owner”})$$

$$AlwaysReleases \triangleq \forall\, p \in Proc : \mathrm{WF}_{state}(Release(p))$$

$$Spec \triangleq Init \land \Box[Next]_{vars} \land Liveness$$

$$LIVE \triangleq Spec \land AlwaysReleases$$

* Modification History
* Last modified Sun *Oct* 07 01:26:14 *EDT* 2018 by *mehuljain*
* Created *Fri Oct* 05 14:54:06 *EDT* 2018 by *mehuljain*