
MODULE *AJupiterExtended*

AJupiter extended with *JupiterCtx*. This is used to show that *AJupiter* implements *XJupiter*.

EXTENDS *JupiterCtx*, *BufferStateSpace* *TODO*: To extend *AJupiter*

VARIABLES *cbuf*, *crc*, *sbuf*, *srec*, *cincomingXJ*, *sincomingXJ*
varsEx \triangleq $\langle \text{intVars}, \text{ctxVars}, \text{cbuf}, \text{crc}, \text{sbuf}, \text{srec}, \text{cincomingXJ}, \text{sincomingXJ} \rangle$

AJMsgEx \triangleq [*ack* : Nat, *cop* : Cop, *oid* : Oid]
commXJ \triangleq INSTANCE *CSComm* WITH *Msg* \leftarrow *Seq*(*Cop*),
cincoming \leftarrow *cincomingXJ*, *sincoming* \leftarrow *sincomingXJ*

TypeOKEx \triangleq
 \wedge *TypeOKInt*
 \wedge *TypeOKCtx*
 \wedge *commXJ*! *TypeOK*
 \wedge *crc* \in [*Client* \rightarrow Nat]
 \wedge *srec* \in [*Client* \rightarrow Nat]
 \wedge *cbuf* \in [*Client* \rightarrow *Seq*(*Cop*)]
 \wedge *sbuf* \in [*Client* \rightarrow *Seq*(*Cop*)]

InitEx \triangleq
 \wedge *InitInt*
 \wedge *InitCtx*
 \wedge *commXJ*! *Init*
 \wedge *crc* = [*c* \in *Client* \mapsto 0]
 \wedge *srec* = [*c* \in *Client* \mapsto 0]
 \wedge *cbuf* = [*c* \in *Client* \mapsto $\langle \rangle$]
 \wedge *sbuf* = [*c* \in *Client* \mapsto $\langle \rangle$]

DoOpEx(*c*, *op*) \triangleq
LET *cop* \triangleq [*op* \mapsto *op*, *oid* \mapsto [*c* \mapsto *c*, *seq* \mapsto *cseq*[*c*], *ctx* \mapsto *ds*[*c*]]
IN \wedge *crc'* = [*crc* EXCEPT ![*c*] = 0]
 \wedge *cbuf'* = [*cbuf* EXCEPT ![*c*] = *Append*(@, *cop*)]
 \wedge *SetNewAop*(*c*, *op*)
 \wedge *Comm*! *CSend*([*ack* \mapsto *crc*[*c*], *cop* \mapsto *cop*, *oid* \mapsto *cop.oid*])
 \wedge *commXJ*! *CSend*(*cop*)

ClientPerformEx(*c*, *m*) \triangleq
LET *xform* \triangleq *xFormShift*(*COT*, *m.cop*, *cbuf*[*c*], *m.ack*)
IN \wedge *cbuf'* = [*cbuf* EXCEPT ![*c*] = *xform.xops*]
 \wedge *crc'* = [*crc* EXCEPT ![*c*] = @ + 1]
 \wedge *SetNewAop*(*c*, *xform.xop.op*)

ServerPerformEx(*m*) \triangleq
LET *c* \triangleq *ClientOf*(*m.cop*)

$$\begin{aligned}
& xform \triangleq xFormShift(COT, m.cop, sbuf[c], m.ack) \\
& xcop \triangleq xform.xop \\
\text{IN} \quad & \wedge srec' = [cl \in Client \mapsto \text{IF } cl = c \text{ THEN } srec[cl] + 1 \text{ ELSE } 0] \\
& \wedge sbuf' = [cl \in Client \mapsto \text{IF } cl = c \text{ THEN } xform.xops \\
& \hspace{15em} \text{ELSE } Append(sbuf[cl], xcop)] \\
& \wedge SetNewAop(Server, xcop.op) \\
& \wedge Comm!SSend(c, [cl \in Client \mapsto \\
& \hspace{15em} [ack \mapsto srec[cl], cop \mapsto xcop, oid \mapsto xcop.oid]]) \\
& \wedge commXJ!SSendSame(c, xcop)
\end{aligned}$$

$$\begin{aligned}
DoEx(c) & \triangleq \\
& \wedge DoInt(DoOpEx, c) \\
& \wedge DoCtx(c) \\
& \wedge \text{UNCHANGED } \langle sbuf, srec \rangle
\end{aligned}$$

$$\begin{aligned}
RevEx(c) & \triangleq \\
& \wedge RevInt(ClientPerformEx, c) \\
& \wedge RevCtx(c) \\
& \wedge commXJ!CRev(c) \\
& \wedge \text{UNCHANGED } \langle sbuf, srec \rangle
\end{aligned}$$

$$\begin{aligned}
SRevEx & \triangleq \\
& \wedge SRevInt(ServerPerformEx) \\
& \wedge SRevCtx \\
& \wedge commXJ!SRev \\
& \wedge \text{UNCHANGED } \langle cbuf, crec \rangle
\end{aligned}$$

$$\begin{aligned}
NextEx & \triangleq \\
& \vee \exists c \in Client : DoEx(c) \vee RevEx(c) \\
& \vee SRevEx
\end{aligned}$$

$$\begin{aligned}
FairnessEx & \triangleq \\
& \text{WF}_{varsEx}(SRevEx \vee \exists c \in Client : RevEx(c))
\end{aligned}$$

$$\begin{aligned}
SpecEx & \triangleq InitEx \wedge \Box[NextEx]_{varsEx} \wedge FairnessEx
\end{aligned}$$

$$\begin{aligned}
QC & \triangleq \text{Quiescent Consistency} \\
& Comm!EmptyChannel \Rightarrow Cardinality(Range(state)) = 1
\end{aligned}$$

THEOREM $SpecEx \Rightarrow \Box QC$

\ * Modification History
\ * Last modified *Thu Jan 17 10:38:50 CST 2019* by anonymous
\ * Created *Thu Dec 27 21:15:09 CST 2018* by anonymous