
MODULE *AbsJupiter*

Abstract *Jupiter*, inspired by the COT algorithm proposed by Sun and Sun; see *TPDS'2009*.
EXTENDS *JupiterSerial*, *SetStateSpace*

VARIABLES
copss *copss*[*r*]: the state space (i.e., a set) of *Cop* maintained at replia *r* ∈ *Replica*
vars \triangleq $\langle \textit{intVars}, \textit{ctxVars}, \textit{serialVars}, \textit{copss} \rangle$

TypeOK \triangleq
 \wedge *TypeOKInt*
 \wedge *TypeOKCtx*
 \wedge *TypeOKSerial*
 \wedge *copss* ∈ [*Replica* → SUBSET *Cop*]

Init \triangleq
 \wedge *InitInt*
 \wedge *InitCtx*
 \wedge *InitSerial*
 \wedge *copss* = [*r* ∈ *Replica* ↦ {}]

NextCop(*r*, *cop*, *ss*, *ctx*) \triangleq Return the next *fcop* ∈ *Cop* against which *cop* is to be transformed.
LET *foid* \triangleq CHOOSE *oid* ∈ *ctx* : the first *oid* in *ctx* according to *serial*[*r*]
 $\forall id \in \textit{ctx} \setminus \{oid\} : tb(oid, id, serial[r])$
IN CHOOSE *fcop* ∈ *ss* : THEOREM : Existence of *fcop*
fcop.oid = *foid* \wedge *fcop.ctx* = *cop.ctx*

Perform(*r*, *cop*) \triangleq
LET *xform* \triangleq *xForm*(*NextCop*, *r*, *cop*, *copss*[*r*]) [*xcop*, *xss*]
IN \wedge *copss*' = [*copss* EXCEPT ![*r*] = *xform.xss*]
 \wedge *SetNewAop*(*r*, *xform.xcop.op*)

ClientPerform(*c*, *cop*) \triangleq *Perform*(*c*, *cop*)

ServerPerform(*cop*) \triangleq
 \wedge *Perform*(*Server*, *cop*)
 \wedge *Comm*!SSendSame(*ClientOf*(*cop*), *cop*)

DoOp(*c*, *op*) \triangleq
LET *cop* \triangleq [*op* ↦ *op*, *oid* ↦ [*c* ↦ *c*, *seq* ↦ *cseq*[*c*]], *ctx* ↦ *ds*[*c*]]
IN \wedge *ClientPerform*(*c*, *cop*)
 \wedge *Comm*!CSend(*cop*)

Do(*c*) \triangleq
 \wedge *DoInt*(*DoOp*, *c*)
 \wedge *DoCtx*(*c*)

$$\begin{aligned}
& \wedge DoSerial(c) \\
Rev(c) & \triangleq \\
& \wedge RevInt(ClientPerform, c) \\
& \wedge RevCtx(c) \\
& \wedge RevSerial(c) \\
SRev & \triangleq \\
& \wedge SRevInt(ServerPerform) \\
& \wedge SRevCtx \\
& \wedge SRevSerial
\end{aligned}$$

$$\begin{aligned}
Next & \triangleq \\
& \vee \exists c \in Client : Do(c) \vee Rev(c) \\
& \vee SRev \\
Fairness & \triangleq \\
& WF_{vars}(SRev \vee \exists c \in Client : Rev(c)) \\
Spec & \triangleq Init \wedge \Box[Next]_{vars} \wedge Fairness
\end{aligned}$$

$$\begin{aligned}
QC & \triangleq \text{Quiescent Consistency} \\
& Comm!EmptyChannel \Rightarrow Cardinality(Range(state)) = 1 \\
THEOREM & Spec \Rightarrow \Box QC \\
SEC & \triangleq \text{Strong Eventual Consistency} \\
& \forall r1, r2 \in Replica : \\
& \quad ds[r1] = ds[r2] \Rightarrow state[r1] = state[r2] \\
THEOREM & Spec \Rightarrow \Box SEC \\
Compactness & \triangleq \text{Compactness of state space} \\
& Comm!EmptyChannel \Rightarrow Cardinality(Range(copss)) = 1 \\
THEOREM & Spec \Rightarrow \Box Compactness
\end{aligned}$$

\ * Modification History
\ * Last modified Tue Jan 29 10:11:54 CST 2019 by anonymous
\ * Created Wed Dec 05 19:55:52 CST 2018 by anonymous