MODULE *AbstractBalloting*

This is a formalization of SCP 's abstract balloting protocol described in Section 3.5 of the IETF draft at:

https://datatracker.ietf.org/doc/html/draft-mazieres-dinrg-scp-05#section-3.5

The goal if to then refine this specification to one that closely matches the concrete SCP protocol.

We provide an inductive invariant showing that, by following the 2 "restrictions on voting" described in Section 3.5 of the above document, safety is guaranteed.

Note that it is not true that a validator never votes to commit and abort the same ballot. This can happen when a validator votes to commit a ballot, but then accepts to abort it because a blocking set accepted to abort it. Moreover, it is necessary for liveness to allow this.

EXTENDS *DomainModel*

VARIABLES
voteToAbort
, *acceptedAborted*
, *voteToCommit*
, *acceptedCommitted*
, *externalized*
, *byz*

$$
\begin{aligned}
&\mathit{TypeOK} \triangleq \\
&\quad \land\ \mathit{voteToAbort} \in [N \to \text{SUBSET } \mathit{Ballot}] \\
&\quad \land\ \mathit{acceptedAborted} \in [N \to \text{SUBSET } \mathit{Ballot}] \\
&\quad \land\ \mathit{voteToCommit} \in [N \to \text{SUBSET } \mathit{Ballot}] \\
&\quad \land\ \mathit{acceptedCommitted} \in [N \to \text{SUBSET } \mathit{Ballot}] \\
&\quad \land\ \mathit{externalized} \in [N \to \text{SUBSET } \mathit{Ballot}] \\
&\quad \land\ \mathit{byz} \in \text{SUBSET } N
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{Init} \triangleq \\
&\quad \land\, \mathit{voteToAbort} = [n \in N \mapsto \{\}] \\
&\quad \land\, \mathit{acceptedAborted} = [n \in N \mapsto \{\}] \\
&\quad \land\, \mathit{voteToCommit} = [n \in N \mapsto \{\}] \\
&\quad \land\, \mathit{acceptedCommitted}\ = [n \in N \mapsto \{\}] \\
&\quad \land\, \mathit{externalized} = [n\ \ \in N \mapsto \{\}] \\
&\quad \land\, \mathit{byz} \in \mathit{FailProneSet}
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{IsPrepared}(n,\ b1) \triangleq \\
&\qquad \lor\ \forall\, b2 \in \mathit{Ballot} : \mathit{LessThanAndIncompatible}(b2,\ b1) \Rightarrow \\
&\qquad\qquad \exists\, Q \in \mathit{Quorum} : \forall\, n2 \in Q \setminus \mathit{byz} : b2 \in \mathit{acceptedAborted}[n2] \\
&\qquad \lor \exists\, \mathit{cnt} \in \mathit{BallotNumber} : \\
&\qquad\quad \land\, [\mathit{counter} \mapsto \mathit{cnt},\ \mathit{value} \mapsto b1.\mathit{value}] \in \mathit{acceptedCommitted}[n] \\
&\qquad\quad \land\ \ \mathit{cnt} < b1.\mathit{counter}\ \ \text{really necessary?}
\end{aligned}
$$

$$
\mathit{Step}(n) \triangleq
$$

$\land$ UNCHANGED $\langle byz \rangle$
NOTE for TLC, we must update $acceptedAborted$ before $voteToAbort$,
because updating $voteToAbort$ depends on $acceptedAborted'$:
$\land \exists\, B \in \text{SUBSET } Ballot :$
$\land \forall\, b \in B :$
$\land \lor \exists\, Q \in Quorum : \forall\, n2 \in Q \setminus byz : b \in voteToAbort[n2] \cup acceptedAborted[n2]$
$\lor \exists\, Bl \in BlockingSet : \forall\, n2 \in Bl \setminus byz : b \in acceptedAborted[n2]$
$\land\ acceptedAborted' = [acceptedAborted \text{ EXCEPT } ![n] = @ \cup B]$
$\land \exists\, B \in \text{SUBSET } Ballot :$
$\land \forall\, b \in B : b \notin voteToCommit[n] \lor b \in acceptedAborted'[n]$
$\land\ voteToAbort' = [voteToAbort \text{ EXCEPT } ![n] = @ \cup B]$
NOTE for TLC, we must update $acceptedCommitted$ before $voteToCommit$,
because updating $voteToCommit$ depends on $acceptedCommitted'$:
$\land \exists\, B \in \text{SUBSET } Ballot :$
$\land \forall\, b \in B :$
$\land \lor \exists\, Q \in Quorum : \forall\, n2 \in Q \setminus byz : b \in voteToCommit[n2] \cup acceptedCommitted[n2]$
$\lor \exists\, Bl \in BlockingSet : \forall\, n2 \in Bl \setminus byz : b \in acceptedCommitted[n2]$
$\land\ acceptedCommitted' = [acceptedCommitted \text{ EXCEPT } ![n] = @ \cup B]$
$\land \exists\, B \in \text{SUBSET } Ballot :$
$\land \forall\, b \in B :$
$\land\ b.counter > 0$ we start at ballot 1
if the ballot is already aborted, don't vote to commit
(using the primed version ensures we don't vote to commit and abort at the same time):
$\land\ b \notin voteToAbort'[n] \cup acceptedAborted'[n]$
the prime allows us to consider prepared something we accepted committed in this very step:
$\land\ IsPrepared(n,\ b)'$
$\land\ voteToCommit' = [voteToCommit \text{ EXCEPT } ![n] = @ \cup B]$
we vote to commit at most one value per ballot number:
$\land \forall\, b1,\ b2 \in voteToCommit'[n] : b1.counter = b2.counter \Rightarrow b1.value = b2.value$
$\land \exists\, B \in \text{SUBSET } Ballot :$
$\land \forall\, b \in B : \exists\, Q \in Quorum :$
$\forall\, n2 \in Q \setminus byz : b \in acceptedCommitted[n2]$
$\land\ externalized' = [externalized \text{ EXCEPT } ![n] = @ \cup B]$

$ByzantineHavoc \triangleq$
$\land \exists\, x \in [byz \to \text{SUBSET } Ballot] :$
$voteToAbort' = [n \in N \mapsto \text{IF } n \in byz \text{ THEN } x[n] \text{ ELSE } voteToAbort[n]]$
$\land \exists\, x \in [byz \to \text{SUBSET } Ballot] :$
$acceptedAborted' = [n \in N \mapsto \text{IF } n \in byz \text{ THEN } x[n] \text{ ELSE } acceptedAborted[n]]$
$\land \exists\, x \in [byz \to \text{SUBSET } Ballot] :$
$voteToCommit' = [n \in N \mapsto \text{IF } n \in byz \text{ THEN } x[n] \text{ ELSE } voteToCommit[n]]$
$\land \exists\, x \in [byz \to \text{SUBSET } Ballot] :$
$acceptedCommitted' = [n \in N \mapsto \text{IF } n \in byz \text{ THEN } x[n] \text{ ELSE } acceptedCommitted[n]]$
$\land$ UNCHANGED $\langle externalized,\ byz \rangle$

$Next \triangleq$
$\quad \lor \exists\, n \in N : Step(n)$
$\quad \lor \ ByzantineHavoc$

$vars \triangleq \langle voteToAbort,\ acceptedAborted,\ voteToCommit,\ acceptedCommitted,\ externalized,\ byz \rangle$

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

$Agreement \triangleq$
$\quad \forall\, n1,\ n2 \in N \setminus byz : \forall\, b1,\ b2 \in Ballot :$
$\qquad b1 \in externalized[n1] \land b2 \in externalized[n2] \Rightarrow b1.value = b2.value$

Inductive invariant proving safety:
$InductiveInvariant \triangleq$
$\quad \land \ TypeOK$
$\quad \land \ byz \in FailProneSet$
$\quad \land \ \forall\, n \in N \setminus byz :$
$\qquad \land \ \forall\, b \in Ballot :$
$\qquad\quad \land \ b \in voteToCommit[n] \Rightarrow b \notin voteToAbort[n] \lor b \in acceptedAborted[n]$
$\qquad\quad \land \ b \in voteToCommit[n] \cup acceptedCommitted[n] \cup externalized[n] \Rightarrow b.counter > 0$
$\qquad\quad \land \ \forall\, b2 \in Ballot :$
$\qquad\qquad b \in voteToCommit[n] \land b2 \in voteToCommit[n] \land b \neq b2 \Rightarrow b.counter \neq b2.counter$
$\qquad\quad \land \ b \in acceptedAborted[n] \Rightarrow \exists\, Q \in Quorum :$
$\qquad\qquad \forall\, n2 \in Q \setminus byz : b \in voteToAbort[n2]$
$\qquad\quad \land \ b \in acceptedCommitted[n] \Rightarrow \exists\, Q \in Quorum :$
$\qquad\qquad \forall\, n2 \in Q \setminus byz : b \in voteToCommit[n2]$
$\qquad\quad \land \ b \in externalized[n] \Rightarrow \exists\, Q \in Quorum :$
$\qquad\qquad \forall\, n2 \in Q \setminus byz : b \in acceptedCommitted[n2]$
$\qquad\quad \land \ b \in voteToCommit[n] \Rightarrow$
$\qquad\qquad \lor \ b.counter = 1$
$\qquad\qquad \lor \ \forall\, b2 \in Ballot : LessThanAndIncompatible(b2,\ b) \Rightarrow$
$\qquad\qquad\qquad \exists\, Q \in Quorum : \forall\, n2 \in Q \setminus byz : b2 \in acceptedAborted[n2]$
$\qquad\qquad \lor \ \exists\, cnt \in BallotNumber :$
$\qquad\qquad\qquad \land \ cnt < b.counter$
$\qquad\qquad\qquad \land \ [counter \mapsto cnt,\ value \mapsto b.value] \in acceptedCommitted[n]$
$\qquad\quad \land \ b \in acceptedAborted[n] \Rightarrow \forall\, Q \in Quorum : \exists\, n2 \in Q \setminus byz : b \notin voteToCommit[n2]$
$\quad \land \ Agreement$