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 agreement is guaranteed.

Note that a node may vote to commit a ballot and later vote to abort the same ballot if it knows that commit cannot possibly reach quorum threshold. This can happen when a validator votes to commit a ballot, but then accepts to abort it. Moreover, it is necessary for liveness to allow this.

Also note that we do not model *Byzantine* behavior explicitly. Instead, whenever a node checks that a set (a quorum or a blocking set) voted or accepted a statement, it only checks that the non-Byzantine members of the set did so. This soundly models what could happen under *Byzantine* behavior because *Byzantine* nodes, being unconstrained, could have voted or accepted whatever is needed to make the check pass.

EXTENDS *DomainModel*

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

$TypeOK \triangleq$
$\land\ voteToAbort \in [N \to \text{SUBSET } Ballot]$
$\land\ acceptedAborted \in [N \to \text{SUBSET } Ballot]$
$\land\ voteToCommit \in [N \to \text{SUBSET } Ballot]$
$\land\ acceptedCommitted \in [N \to \text{SUBSET } Ballot]$
$\land\ externalized \in [N \to \text{SUBSET } Ballot]$
$\land\ byz \in \text{SUBSET } N$

$Init \triangleq$
$\land\ voteToAbort = [n \in N \mapsto \{\}]$
$\land\ acceptedAborted = [n \in N \mapsto \{\}]$
$\land\ voteToCommit = [n \in N \mapsto \{\}]$
$\land\ acceptedCommitted = [n \in N \mapsto \{\}]$
$\land\ externalized = [n \in N \mapsto \{\}]$
$\land\ byz \in FailProneSet$

The second disjunct allows voting to commit all higher ballot numbers once a ballot is accepted committed (see meaning of COMMIT message in the IETF draft):

$IsPrepared(n,\ b1) \triangleq$
$\lor\ \forall\, b2 \in Ballot : LessThanAndIncompatible(b2,\ b1) \Rightarrow$

$\exists\, Q \in Quorum : \forall\, n2 \in Q \setminus byz : b2 \in acceptedAborted[n2]$

$\lor \exists\, cnt \in BallotNumber :$

$\land\ [counter \mapsto cnt,\ value \mapsto b1.value] \in acceptedCommitted[n]$

$\land\ \ cnt < b1.counter$ really necessary? yes

$Step(n) \triangleq$

$\land$ UNCHANGED $\langle byz \rangle$

NOTE for *TLC*, we must update *acceptedAborted* before *voteToAbort*,
because updating *voteToAbort* depends on *acceptedAborted′*:

$\land\ \exists\, B \in$ SUBSET $Ballot :$

$\land\ \forall\, b \in B :$

$\land\ \ \lor \exists\, Q \in Quorum : \forall\, n2 \in Q \setminus byz : b \in voteToAbort[n2] \cup acceptedAborted[n2]$

$\lor \exists\, Bl \in BlockingSet : \forall\, n2 \in Bl \setminus byz : b \in acceptedAborted[n2]$

$\land\ \ acceptedAborted' = [acceptedAborted$ EXCEPT $![n] = @ \cup B]$

$\land \exists\, B \in$ SUBSET $Ballot :$

$\land\ \forall\, b \in B : b \notin voteToCommit[n] \lor b \in acceptedAborted'[n]$

$\land\ \ voteToAbort' = [voteToAbort$ EXCEPT $![n] = @ \cup B]$

NOTE for *TLC*, we must update *acceptedCommitted* before *voteToCommit*,
because updating *voteToCommit* depends on *acceptedCommitted′*:

$\land\ \exists\, B \in$ SUBSET $Ballot :$

$\land\ \forall\, b \in B :$

$\land\ \ \lor \exists\, Q \in Quorum : \forall\, n2 \in Q \setminus byz : b \in voteToCommit[n2] \cup acceptedCommitted[n2]$

$\lor \exists\, Bl \in BlockingSet : \forall\, n2 \in Bl \setminus byz : b \in acceptedCommitted[n2]$

$\land\ \ acceptedCommitted' = [acceptedCommitted$ EXCEPT $![n] = @ \cup B]$

$\land\ \exists\, B \in$ 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:

$\land\ \ IsPrepared(n,\ b)'$

$\land\ \ voteToCommit' = [voteToCommit$ EXCEPT $![n] = @ \cup B]$

we vote to commit at most one value per ballot number:

$\land\ \forall\, b1,\ b2 \in voteToCommit'[n] : b1.counter = b2.counter \Rightarrow b1.value = b2.value$

$\land\ \exists\, B \in$ SUBSET $Ballot :$

$\land\ \forall\, b \in B : \exists\, Q \in Quorum :$

$\forall\, n2 \in Q \setminus byz : b \in acceptedCommitted[n2]$

$\land\ \ externalized' = [externalized$ EXCEPT $![n] = @ \cup B]$

$Next \triangleq$

$\lor \exists\, n \in N \setminus byz : Step(n)$

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

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

$$
\begin{array}{l}
\mathit{Agreement} \triangleq \\
\quad \forall\, n1,\, n2 \in N \setminus \mathit{byz} : \forall\, b1,\, b2 \in \mathit{Ballot} : \\
\qquad b1 \in \mathit{externalized}[n1] \land b2 \in \mathit{externalized}[n2] \Rightarrow b1.\mathit{value} = b2.\mathit{value}
\end{array}
$$

Inductive invariant proving safety:

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