Atomic Commitment Protocol with Simple Broadcast primitive (ACP-SB)

From:
Sape Mullender, editor. Distributed Systems.

Synchronous communication has been replaced with (implicit) asynchronous communication. Failures are detected “magically” instead of relying on timeouts.

This version of the protocol uses a “simple broadcast” where a broadcast is simply a series of messages sent, possibly interrupted by a failure. Consequently, this algorithm is “non terminating” and property AC5 does not hold.

CONSTANTS
participants, yes, no, undecided, commit, abort, waiting, notsent

VARIABLES
participant, participants (N)
coordinator, coordinator (1)

TypeInvParticipant \[= \text{participant} \in \]
participants \[\rightarrow \]
vote : \{yes, no\},
alive : BOOLEAN,
decision : \{undecided, commit, abort\},
faulty : BOOLEAN,
voteSent : BOOLEAN

TypeInvCoordinator \[= \text{coordinator} \in \]
request : \{participants \rightarrow BOOLEAN\},
vote : \{participants \rightarrow \{waiting, yes, no\}\},
broadcast : \{participants \rightarrow \{waiting, yes, no, undecided\}\},
decision : \{commit, abort, notsent\},
alive : BOOLEAN,
faulty : BOOLEAN

TypeInv \[= \text{TypeInvParticipant} \land \text{TypeInvCoordinator}\]

Initially:
All the participants:
have a yes/no vote
are alive and not faulty
have not sent in their votes yet
are undecided about final decision

The coordinator:
has not sent vote requests yet
has not received votes from any participant
is alive and not faulty
has not sent broadcast messages to any participant
is undecided about final decision

1
\[
\begin{align*}
\text{InitParticipant} & \triangleq \text{participant } \in \left[ \text{participants } \rightarrow \left[ \begin{array}{l}
\text{vote } : \{\text{yes, no}\}, \\
\text{alive } : \{\text{true}\}, \\
\text{decision } : \{\text{undecided}\}, \\
\text{faulty } : \{\text{false}\}, \\
\text{voteSent } : \{\text{false}\}
\end{array} \right] \right] \\
\text{InitCoordinator} & \triangleq \text{coordinator } \in \left[ \begin{array}{l}
\text{request } : \{\text{participants } \rightarrow \{\text{false}\}\}, \\
\text{vote } : \{\text{participants } \rightarrow \{\text{waiting}\}\}, \\
\text{alive } : \{\text{true}\}, \\
\text{broadcast } : \{\text{participants } \rightarrow \{\text{notsent}\}\}, \\
\text{decision } : \{\text{undecided}\}, \\
\text{faulty } : \{\text{false}\}
\end{array} \right] \\
\text{Init} & \triangleq \text{InitParticipant } \land \text{InitCoordinator}
\end{align*}
\]

**COORDINATOR STATEMENTS**

\[\text{request}(i) : \]
\[\text{IF} \]
\[\text{coordinator is alive} \]
\[\text{request for vote has not been sent to participant } i \]
\[\text{THEN} \]
\[\text{request for vote is sent to participant } i \]

\[\text{request}(i) \triangleq \land \text{coordinator.alive} \]
\[\land \neg \text{coordinator.request}[i] \]
\[\land \text{coordinator}' = [\text{coordinator} \text{ \text{EXCEPT} } !.\text{request} = \]
\[[@ \text{\text{EXCEPT} } ![i] = \text{true}]\]
\[\land \text{UNCHANGED } \langle \text{participant} \rangle \]

\[\text{getVote}(i) : \]
\[\text{IF} \]
\[\text{coordinator is alive} \]
\[\text{coordinator is still undecided} \]
\[\text{coordinator has sent request for votes to all participants} \]
\[\text{coordinator is waiting to receive a vote from participant } i \]
\[\text{participant } i \text{ has sent the vote message} \]
\[\text{THEN} \]
\[\text{the coordinator can record the vote of participant } i \]

\[\text{getVote}(i) \triangleq \land \text{coordinator.alive} \]
\[\land \text{coordinator.decision } = \text{undecided} \]
\[\land \forall j \in \text{participants} : \text{coordinator.request}[j] \]
\[\land \text{coordinator.vote}[i] = \text{waiting} \]
\[\land \text{participant}[i].\text{voteSent} \]
\[\land \text{coordinator}' = [\text{coordinator} \text{ \text{EXCEPT} } !.\text{vote} = \]
\[[@ \text{\text{EXCEPT} } ![i] = \text{participant}[i].\text{vote}]\]
\[\land \text{UNCHANGED } \langle \text{participant} \rangle \]

\[\text{detectFault}(i) : \]
\[\text{IF} \]
\[\text{coordinator is alive} \]
coordinator is still undecided
coordinator has sent request for votes to all participants
coordinator is waiting for vote from participant \(i\)

```
THEN
```
coordinator times out on participant \(i\) and decides to abort

detectFault\((i)\) \(\triangleq \)
\(\land \ \text{coordinator.alive} \)
\(\land \ \text{coordinator.decision} = \text{undecided} \)
\(\land \ \forall j \in \text{participants} : \ \text{coordinator.request}[j] \)
\(\land \ \text{coordinator.vote}[i] = \text{waiting} \)
\(\land \ \neg \ \text{participant}[i].\text{alive} \)
\(\land \ \neg \ \text{participant}[i].\text{voteSent} \)
\(\land \ \text{coordinator}' = [\ \text{coordinator} \ \text{EXCEPT} !.\text{decision} = \text{abort} ] \)
\(\land \ \text{UNCHANGED} \ \langle \text{participant} \rangle \)

```
makeDecision:
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
coordDie ≜ \land \text{coordinator.alive} \\
\land \text{coordinator'} = [\text{coordinator \ except !.alive = \text{false}, \!.faulty = \text{true]} \\
\land \text{UNCHANGED} \langle \text{participant} \rangle

PARTICIPANT STATEMENTS

sendVote(i):
\text{IF} \\
\text{participant is alive} \\
\text{participant has received a request for vote} \\
\text{THEN} \\
\text{participant sends vote}

sendVote(i) ≜ \land \text{participant[i].alive} \\
\land \text{coordinator.request[i]} \\
\land \text{participant'} = [\text{participant \ except ![i] =} \\
\quad [\text{@ \ except !.voteSent = \text{true]} \\
\quad ] \land \text{UNCHANGED} \langle \text{coordinator} \rangle

abortOnVote(i):
\text{IF} \\
\text{participant is alive} \\
\text{participant is undecided} \\
\text{participant has sent its vote to the coordinator} \\
\text{participant’s vote is no} \\
\text{THEN} \\
\text{participant decides (unilaterally) to abort}

abortOnVote(i) ≜ \land \text{participant[i].alive} \\
\land \text{participant[i].decision = undecided} \\
\land \text{participant[i].voteSent} \\
\land \text{participant[i].vote = no} \\
\land \text{participant'} = [\text{participant \ except ![i] =} \\
\quad [\text{@ \ except !.decision = abort]} \\
\quad ] \land \text{UNCHANGED} \langle \text{coordinator} \rangle

abortOnTimeoutRequest(i):
\text{IF} \\
\text{participant is alive} \\
\text{participant is still undecided} \\
\text{coordinator has died without sending request for vote} \\
\text{THEN} \\
\text{participant decides (unilaterally) to abort}

abortOnTimeoutRequest(i) ≜ \land \text{participant[i].alive} \\
\land \text{participant[i].decision = undecided} \\
\land \neg \text{coordinator.alive} \\
\land \neg \text{coordinator.request[i]} \\
\land \text{participant'} = [\text{participant \ except ![i] =} \\
\quad [\text{@ \ except !.decision = abort]} \\
\quad ] \land \text{UNCHANGED} \langle \text{coordinator} \rangle

decide(i):
\text{IF} \\
\text{participant is alive} \\
\text{participant is undecided
participant has received decision from the coordinator

THEN

participant decides according to decision from coordinator

decide(i) ≡ \( \land \text{participant}[i].\text{alive} \land \text{participant}[i].\text{decision} = \text{undecided} \land \text{coordinator}.\text{broadcast}[i] \neq \text{notsent} \land \text{participant}' = [\text{participant} \text{EXCEPT } ![i] = \] "∃ \text{EXCEPT } !.\text{decision} = \text{coordinator}.\text{broadcast}[i]"

\land \text{UNCHANGED } \langle \text{coordinator} \rangle

parDie(i):

IF

participant is alive

THEN

participant dies and is now faulty

parDie(i) ≡ \( \land \text{participant}[i].\text{alive} \land \text{participant}' = [\text{participant} \text{EXCEPT } ![i] = \] "∃ \text{EXCEPT } !.\text{alive} = \text{false}, ![.\text{faulty} = \text{true}]"

\land \text{UNCHANGED } \langle \text{coordinator} \rangle

FOR N PARTICIPANTS

parProg(i) ≡ \text{sendVote}(i) \lor \text{abortOnVote}(i) \lor \text{abortOnTimeoutRequest}(i) \lor \text{decide}(i)

parProgN ≡ \\exists i \in \text{participants} : \text{parDie}(i) \lor \text{parProg}(i)

coordProgA(i) ≡ \text{request}(i) \lor \text{getVote}(i) \lor \text{detectFault}(i) \lor \text{coordBroadcast}(i)

coordProgB ≡ \text{makeDecision} \lor \exists i \in \text{participants} : \text{coordProgA}(i)

coordProgN ≡ \text{coordDie} \lor \text{coordProgB}

progN ≡ \text{parProgN} \lor \text{coordProgN}

Death transitions are left outside of fairness

fairness ≡ \land \forall i \in \text{participants} : \text{WF}_{(\text{coordinator}, \text{participant})}(\text{parProg}(i)) \land \text{WF}_{(\text{coordinator}, \text{participant})}(\text{coordProgB})

Spec ≡ \text{Init} \land \Box[\text{progN}]_{(\text{coordinator}, \text{participant})} \land \text{fairness}

CORRECTNESS SPECIFICATION

This specification follows the original paper, except that AC3 is stronger: It forces participants to abort if one vote at least is NO (in the absence of failure).

The specification is split between safety and liveness.

SAFETY

All participants that decide reach the same decision

AC1 ≡ \Box \forall i, j \in \text{participants} :

\lor \text{participant}[i].\text{decision} \neq \text{commit}

\lor \text{participant}[j].\text{decision} \neq \text{abort}
If any participant decides commit, then all participants must have votes YES

\[ AC2 \triangleq \Box( (\exists i \in \text{participants} : \text{participant}[i].decision = \text{commit}) \Rightarrow (\forall j \in \text{participants} : \text{participant}[j].vote = \text{yes})) \]

If any participant decides abort, then:
- at least one participant voted NO, or
- at least one participant is faulty, or
- coordinator is faulty

\[ AC3_1 \triangleq \Box( (\exists i \in \text{participants} : \text{participant}[i].decision = \text{abort}) \Rightarrow \bigvee (\exists j \in \text{participants} : \text{participant}[j].vote = \text{no}) \lor (\exists j \in \text{participants} : \text{participant}[j].faulty) \lor \text{coordinator.faulty} ) \]

Each participant decides at most once

\[ AC4 \triangleq \Box \land (\forall i \in \text{participants} : \text{participant}[i].decision = \text{commit} \Rightarrow \Box(\text{participant}[i].decision = \text{commit})) \land (\forall j \in \text{participants} : \text{participant}[j].decision = \text{abort} \Rightarrow \Box(\text{participant}[j].decision = \text{abort})) \]

**LIVENESS**

(stronger for AC3 than in the original paper)

\[ AC3_2 \triangleq \Diamond \land \forall i \in \text{participants} : \text{participant}[i].decision \in \{\text{abort, commit}\} \land \exists j \in \text{participants} : \text{participant}[j].faulty \lor \text{coordinator.faulty} \]

**Some Intermediate Properties Used in Proofs**

\[ \text{FaultyStable} \triangleq \forall i \in \text{participants} : \Box(\text{participant}[i].faulty \Rightarrow \Box\text{coordinator.faulty}) \land \Box(\text{coordinator.faulty} \Rightarrow \Box\text{coordinator.faulty}) \]

\[ \text{VoteStable} \triangleq \forall i \in \text{participants} : \bigvee \Box(\text{participant}[i].vote = \text{yes}) \lor \Box(\text{participant}[i].vote = \text{no}) \bigvee \Box(\text{participant}[i].vote = \text{no}) \]

\[ \text{StrongerAC2} \triangleq \Box( (\exists i \in \text{participants} : \text{participant}[i].decision = \text{commit}) \Rightarrow \land (\forall j \in \text{participants} : \text{participant}[j].vote = \text{yes}) \land \text{coordinator.decision = commit} ) \]

\[ \text{StrongerAC3_1} \triangleq \Box( (\exists i \in \text{participants} : \text{participant}[i].decision = \text{abort}) \Rightarrow \land (\exists j \in \text{participants} : \text{participant}[j].vote = \text{no}) \lor (\exists j \in \text{participants} : \text{participant}[j].faulty) \land \text{coordinator.decision = abort} \lor \land \text{coordinator.decision = undecided} ) \]

(AC1 follows from StrongerAC2 \land StrongerAC3_1)

\[ \text{NoRecovery} \triangleq \Box \land \forall i \in \text{participants} : \text{participant}[i].alive \equiv \neg \text{participant}[i].faulty \land \text{coordinator.alive} \equiv \neg \text{coordinator.faulty} \]

**Some Invalid Properties**

\[ \text{DecisionReachedNoFault} \triangleq (\forall i \in \text{participants} : \text{participant}[i].alive) \lor (\forall k \in \text{participants} : \text{participant}[k].decision \neq \text{undecided}) \]

\[ \text{AbortImpliesNoVote} \triangleq \Box( (\exists i \in \text{participants} : \text{participant}[i].decision = \text{abort}) \Rightarrow (\exists j \in \text{participants} : \text{participant}[j].vote = \text{no})) \]

The following is the termination property that this SB algorithm doesn’t have
\[ AC5 \triangleq \forall i \in \text{participants} : \lor \text{participant}[i].\text{decision} \in \{\text{abort}, \text{commit}\} \]
\[ \lor \text{participant}[i].\text{faulty} \]