2 MODULE *BPConProof*

This module specifies a *Byzantine Paxos* algorithm–a version of *Paxos* in which failed acceptors and leaders can be malicious. It is an abstraction and generalization of the Castro-Liskov algorithm in

author = "Miguel *Castro* and *Barbara Liskov*", title = "Practical byzantine fault tolerance and proactive
recovery",
journal = *ACM* Transactions on Computer Systems,
volume = 20,
number = 4, year = 2002, pages = "398–461"

18 EXTENDS *Integers*, *FiniteSets*, *TLAPS*

19

We need the following trivial axioms and theorem about finite sets.

23 AXIOM $EmptySetFinite \triangleq IsFiniteSet(\{\})$

25 AXIOM $SingletonSetFinite \triangleq \forall\, e : IsFiniteSet(\{e\})$

27 AXIOM $ImageOfFiniteSetFinite \triangleq$
28 $\forall\, S, f : IsFiniteSet(S) \Rightarrow IsFiniteSet(\{f[x] : x \in S\})$

30 AXIOM $SubsetOfFiniteSetFinite \triangleq$
31 $\forall\, S,\ T : IsFiniteSet(T) \land (S \subseteq T) \Rightarrow IsFiniteSet(S)$

33 AXIOM $UnionOfFiniteSetsFinite \triangleq$
34 $\forall\, S,\ T : IsFiniteSet(T) \land IsFiniteSet(S) \Rightarrow IsFiniteSet(S \cup T)$

36 THEOREM $OnePlusFinite \triangleq \forall\, S,\ e : IsFiniteSet(S) \Rightarrow IsFiniteSet(S \cup \{e\})$
37 BY *SingletonSetFinite*, *UnionOfFiniteSetsFinite*

Testing that the following formula is true provides a check for typos in the axioms above.

43 $TestAxioms \triangleq$
44 *SingletonSetFinite*
45 $\land \forall\, e \in 1 \,..\, 3 : IsFiniteSet(\{e\})$

47 *ImageOfFiniteSetFinite*
48 $\land \forall\, S,\ T \in \text{SUBSET } (1 \,..\, 4) : \forall f \in [S \to T] :$
49 $IsFiniteSet(S) \Rightarrow IsFiniteSet(\{f[x] : x \in S\})$

51 *SubsetOfFiniteSetFinite*
52 $\land \forall\, S,\ T \in \text{SUBSET } (1 \,..\, 4) :$
53 $IsFiniteSet(T) \land (S \subseteq T) \Rightarrow IsFiniteSet(S)$

55 *UnionOfFiniteSetsFinite*
56 $\land \forall\, S,\ T \in \text{SUBSET } (1 \,..\, 4) :$
57 $IsFiniteSet(T) \land IsFiniteSet(S) \Rightarrow IsFiniteSet(S \cup T)$

58

The sets *Value* and *Ballot* are the same as in the *Voting* and *PaxosConsensus* specs.

63 CONSTANT $Value$

65 $Ballot \triangleq Nat$

As in module *PConProof*, we define *None* to be an unspecified value that is not an element of *Value*.

71 $None \triangleq \text{CHOOSE } v : v \notin Value$

72 ----

We pretend that which acceptors are good and which are malicious is specified in advance. Of course, the algorithm executed by the good acceptors makes no use of which acceptors are which. Hence, we can think of the sets of good and malicious acceptors as "prophecy constants" that are used only for showing that the algorithm implements the *AbstratPaxosConsensus* spec.

We can assume that a maximal set of acceptors are bad, since a bad acceptor is allowed to do anything–including ating like a good one.

The basic idea is that the good acceptors try to execute the *Paxos* consensus algorithm, while the bad acceptors may try to prevent them.

We do not distinguish between faulty and non-faulty leaders. Safety must be preserved even if all leaders are malicious, so we allow any leader to send any syntactically correct message at any time. (In an implementation, syntactically incorrect messages are simply ignored by non-faulty acceptors and have no effect.) Assumptions about leader behavior are required only for liveness.

94 CONSTANTS $Acceptor$, The set of good (non-faulty) acceptors.

95 $FakeAcceptor$, The set of possibly malicious (faulty) acceptors.

96 $ByzQuorum$,

A *Byzantine* quorum is set of acceptors that includes a quorum of good ones. In the case that there are $2f + 1$ good acceptors and f bad ones, a *Byzantine* quorum is any set of $2f + 1$ acceptors.

103 $WeakQuorum$

A weak quorum is a set of acceptors that includes at least one good one. If there are f bad acceptors, then a weak quorum is any set of $f + 1$ acceptors.

We define *ByzAcceptor* to be the set of all real or fake acceptors.

113 $ByzAcceptor \triangleq Acceptor \cup FakeAcceptor$

As in the *Paxos* consensus algorithm, we assume that the set of ballot numbers and -1 is disjoint from the set of all (real and fake) acceptors.

120 ASSUME $BallotAssump \triangleq (Ballot \cup \{-1\}) \cap ByzAcceptor = \{\}$

The following are the assumptions about acceptors and quorums that are needed to ensure safety of our algorithm.

126 ASSUME $BQA \triangleq$

127 $\quad \land Acceptor \cap FakeAcceptor = \{\}$

128 $\quad \land \forall\, Q \in ByzQuorum : Q \subseteq ByzAcceptor$

129 $\quad \land \forall\, Q1,\, Q2 \in ByzQuorum : Q1 \cap Q2 \cap Acceptor \neq \{\}$

130 $\quad \land \forall\, Q \in WeakQuorum : \land\, Q \subseteq ByzAcceptor$

131 $\qquad\qquad\qquad\qquad\qquad\quad \land\, Q \cap Acceptor \neq \{\}$

The following assumption is not needed for safety, but it will be needed to ensure liveness.

ASSUME $BQLA \triangleq$
$\land \exists\, Q \in ByzQuorum : Q \subseteq Acceptor$
$\land \exists\, Q \in WeakQuorum : Q \subseteq Acceptor$

We now define the set *BMessage* of all possible messages.

$1aMessage \triangleq [type : \{\text{“1a”}\},\ \ bal : Ballot]$

Type $1a$ messages are the same as in module *PConProof*.

$1bMessage \triangleq$

A $1b$ message serves the same function as a $1b$ message in ordinary Paxos, where the *mbal* and *mval* components correspond to the *mbal* and *mval* components in the $1b$ messages of *PConProof*. The $m2av$ component is set containing all records with *val* and *bal* components equal to the corresponding of components of a $2av$ message that the acceptor has sent, except containing for each *val* only the record corresponding to the $2av$ message with the highest *bal* component.

$[type\ \ : \{\text{“1b”}\},\ bal : Ballot,$
$\ mbal\ \ : Ballot \cup \{-1\},\ mval : Value \cup \{None\},$
$\ m2av : \text{SUBSET}\ [val : Value,\ bal : Ballot],$
$\ acc : ByzAcceptor]$

$1cMessage \triangleq$

Type $1c$ messages are the same as in *PConProof*.

$[type : \{\text{“1c”}\},\ bal : Ballot,\ val : Value]$

$2avMessage \triangleq$

When an acceptor receives a $1c$ message, it relays that message's contents to the other acceptors in a $2av$ message. It does this only for the first $1c$ message it receives for that ballot; it can receive a second $1c$ message only if the leader is malicious, in which case it ignores that second $1c$ message.

$[type : \{\text{“2av”}\},\ bal : Ballot,\ val : Value,\ acc : ByzAcceptor]$

$2bMessage \triangleq [type : \{\text{“2b”}\},\ acc : ByzAcceptor,\ bal : Ballot,\ val : Value]$

$2b$ messages are the same as in ordinary *Paxos*.

$BMessage \triangleq$
$\ 1aMessage \cup 1bMessage \cup 1cMessage \cup 2avMessage \cup 2bMessage$

We will need the following simple fact about these sets of messages.

LEMMA $BMessageLemma \triangleq$
$\forall\, m \in BMessage :$
$\land\, (m \in 1aMessage) \equiv\ (m.type = \text{“1a”})$
$\land\, (m \in 1bMessage) \equiv\ (m.type = \text{“1b”})$
$\land\, (m \in 1cMessage) \equiv\ (m.type = \text{“1c”})$
$\land\, (m \in 2avMessage) \equiv\ (m.type = \text{“2av”})$
$\land\, (m \in 2bMessage) \equiv\ (m.type = \text{“2b”})$
$\langle 1 \rangle 1.\ \land \forall\, m \in 1aMessage : m.type = \text{“1a”}$
$\qquad \land \forall\, m \in 1bMessage : m.type = \text{“1b”}$

200 $\land \forall\, m \in 1cMessage : m.type = \text{“1c”}$
201 $\land \forall\, m \in 2avMessage : m.type = \text{“2av”}$
202 $\land \forall\, m \in 2bMessage : m.type = \text{“2b”}$
203 BY DEF $1aMessage$, $1bMessage$, $1cMessage$, $2avMessage$, $2bMessage$
204 $\langle 1 \rangle 2$. QED
205 BY $\langle 1 \rangle 1$ DEF $BMessage$
206 ⊢———⊣

****

We now give the algorithm. The basic idea is that the set *Acceptor* of real acceptors emulate an execution of the *PaxosConsensus* algorithm with *Acceptor* as its set of acceptors. Of course, they must do that without knowing which of the other processes in *ByzAcceptor* are real acceptors and which are fake acceptors. In addition, they don't know whether a leader is behaving according to the *PaxosConsensus* algorithm or if it is malicious.

The main idea of the algorithm is that, before performing an action of the *PaxosConsensus* algorithm, a good acceptor determines that this action is actually enabled in that algorithm. Since an action is enabled by the receipt of one or more messages, the acceptor has to determine that the enabling messages are legal *PaxosConsensus* messages. Because *PaxosConsensus* allows a $1a$ message to be sent at any time, the only acceptor action whose enabling messages must be checked is the $Phase2b$ action. It is enabled iff the appropriate $1c$ message and $2a$ message are legal. The $1c$ message is legal iff the leader has received the necessary $1b$ messages. The acceptor therefore maintains a set of $1b$ messages that it knows have been sent, and checks that those $1b$ messages enable the sending of the $1c$ message.

A $2a$ message is legal in the *PaxosConsensus* algorithm iff (i) the corresponding $1c$ message is legal and (ii) it is the only $2a$ message that the leader sends. In the *BPCon* algorithm, there are no explicit $2a$ messages. They are implicitly sent by the acceptors when they send enough $2av$ messages.

We leave unspecified how an acceptor discovers what $1b$ messages have been sent. In the Castro-Liskov algorithm, this is done by having acceptors relay messages sent by other acceptors. An acceptor knows that a $1b$ message has been sent if it receives it directly or else receives a copy from a weak *Byzantine* quorum of acceptors. A (non-malicious) leader must determine what $1b$ messages acceptors know about so it chooses a value so that a quorum of acceptors will act on its $Phase1c$ message and cause that value to be chosen. However, this is necessary only for liveness, so we ignore this for now.

In other implementations of our algorithm, the leader sends along with the $1c$ message a proof that the necessary $1b$ messages have been sent. The easiest way to do this is to have acceptors digitally sign their $1b$ messages, so a copy of the message proves that it has been sent (by the acceptor indicated in the message's *acc* field). The necessary proofs can also be constructed using only message authenticators (like the ones used in the Castro-Liskov algorithm); how this is done is described elsewhere.

In the abstract algorithm presented here, which we call *BPCon*, we do not specify how acceptors learn what $1b$ messages have been sent. We simply introduce a variable *knowsSent* such that $knowsSent[a]$ represents the set of $1b$ messages that (good) acceptor a knows have been sent, and have an action that nondeterministically adds sent $1b$ messages to this set.

263 **--algorithm** *BPCon* **{**

****

The variables:

$maxBal[a] =$ Highest ballot in which acceptor a has participated.

$maxVBal[a]$ = Highest ballot in which acceptor a has cast a vote (sent a $2b$ message); or -1 if it hasn't cast a vote.

$maxVVal[a]$ = *Value* acceptor a has voted for in ballot $maxVBal[a]$, or *None* if $maxVBal[a] = -1$.

$2avSent[a]$ = A set of records in $[val : Value, bal : Ballot]$ describing the $2av$ messages that a has sent. A record is added to this set, and any element with a the same *val* field (and lower *bal* field) removed when a sends a $2av$ message.

$knownSent[a]$ = The set of $1b$ messages that acceptor a knows have been sent.

$bmsgs$ = The set of all messages that have been sent. See the discussion of the *msgs* variable in module *PConProof* to understand our modeling of message passing.

****

288 **variables** $maxBal = [a \in Acceptor \mapsto -1]$,
289 $maxVBal = [a \in Acceptor \mapsto -1]$,
290 $maxVVal = [a \in Acceptor \mapsto None]$,
291 $2avSent = [a \in Acceptor \mapsto \{\}]$,
292 $knowsSent = [a \in Acceptor \mapsto \{\}]$,
293 $bmsgs = \{\}$
294 **define** {
295 $sentMsgs(type, bal) \triangleq \{m \in bmsgs : m.type = type \land m.bal = bal\}$

297 $KnowsSafeAt(ac, b, v) \triangleq$

True for an acceptor ac, ballot b, and value v iff the set of $1b$ messages in $knowsSent[ac]$ implies that value v is safe at ballot b in the *PaxosConsensus* algorithm being emulated by the good acceptors. To understand the definition, see the definition of *ShowsSafeAt* in module *PConProof* and recall (a) the meaning of the *mCBal* and *mCVal* fields of a $1b$ message and (b) that the set of real acceptors in a *ByzQuorum* forms a quorum of the *PaxosConsensus* algorithm.

308 LET $S \triangleq \{m \in knowsSent[ac] : m.bal = b\}$
309 IN $\lor \exists\, BQ \in ByzQuorum :$
310 $\forall\, a \in BQ : \exists\, m \in S : \land\, m.acc = a$
311 $\land\, m.mbal = -1$
312 $\lor \exists\, c \in 0 \,..\, (b-1) :$
313 $\land \exists\, BQ \in ByzQuorum :$
314 $\forall\, a \in BQ : \exists\, m \in S : \land\, m.acc = a$
315 $\land\, m.mbal \leq c$
316 $\land\, (m.mbal = c) \Rightarrow (m.mval = v)$
317 $\land \exists\, WQ \in WeakQuorum :$
318 $\forall\, a \in WQ :$
319 $\exists\, m \in S : \land\, m.acc = a$
320 $\land \exists\, r \in m.m2av : \land\, r.bal \geq c$
321 $\land\, r.val = v$
322 }

We now describe the processes' actions as macros.

As in the *Paxos* consensus algorithm, a ballot *self* leader (good or malicious) can execute a $Phase1a$ ation at any time.

```
330  macro Phase1a(){bmsgs := bmsgs ∪ {[type ↦ "1a", bal ↦ self]} ; }
```

The acceptor's $Phase1b$ ation is similar to that of the *PaxosConsensus* algorithm.

```
336  macro Phase1b(b){
337   when (b > maxBal[self]) ∧ (sentMsgs("1a", b) ≠ {}) ;
338   maxBal[self] := b ;
339   bmsgs := bmsgs ∪ {[type  ↦ "1b", bal ↦ b, acc ↦ self,
340                      m2av ↦ 2avSent[self],
341                      mbal  ↦ maxVBal[self], mval ↦ maxVVal[self]]} ;
342   }
```

A good ballot *self* leader can send a phase $1c$ message for value v if it knows that the messages in $knowsSent[a]$ for a *Quorum* of (good) acceptors imply that they know that v is safe at ballot *self* , and that they can convince any other acceptor that the appropriate $1b$ messages have been sent to that it will also know that v is safe at ballot *self* .

A malicious ballot *self* leader can send any phase $1c$ messages it wants (including one that a good leader could send). We prove safety with a $Phase1c$ ation that allows a leader to be malicious. To prove liveness, we will have to assume a good leader that sends only correct $1c$ messages.

As in the *PaxosConsensus* algorithm, we allow a $Phase1c$ action to send a set of $Phase1c$ messages. (This is not done in the Castro-Liskov algorithm, but seems natural in light of the *PaxosConsensus* algorithm.)

```
363  macro Phase1c(){
364    with (S ∈ SUBSET [type : {"1c"}, bal : {self}, val : Value]){
365      bmsgs := bmsgs ∪ S}
366   }
```

If acceptor *self* receives a ballot b phase $1c$ message with value v, it relays v in a phase $2av$ message if

- it has not already sent a $2av$ message in this or a later ballot and
- the messages in $knowsSent[self]$ show it that v is safe at b in the non-Byzantine *Paxos* consensus algorithm being emulated.

```
378  macro Phase2av(b){
379    when  ∧ maxBal[self] ≤ b
380          ∧ ∀ r ∈ 2avSent[self] : r.bal < b ;
381              We could just as well have used r.bal ≠ b in this condition.
382    with (m ∈ {ms ∈ sentMsgs("1c", b) : KnowsSafeAt(self, b, ms.val)}){
383      bmsgs := bmsgs ∪
384                 {[type ↦ "2av", bal ↦ b, val ↦ m.val, acc ↦ self]} ;
385      2avSent[self] :=  {r ∈ 2avSent[self] : r.val ≠ m.val}
386                          ∪ {[val ↦ m.val, bal ↦ b]}
387      } ;
388    maxBal[self]  := b ;
```

```
  }
```

Acceptor *self* can send a phase $2b$ message with value v if it has received phase $2av$ messages from a *Byzantine* quorum, which implies that a quorum of good acceptors assert that this is the first $1c$ message sent by the leader and that the leader was allowed to send that message. It sets $maxBal[self]$, $maxVBal[self]$, and $maxVVal[self]$ as in the non-Byzantine algorithm.

```
  macro Phase2b(b){
    when maxBal[self] ≤ b ;
    with (v ∈ {vv ∈ Value :
                    ∃ Q ∈ ByzQuorum :
                      ∀ aa ∈ Q :
                        ∃ m ∈ sentMsgs("2av", b) : ∧ m.val = vv
                                                   ∧ m.acc = aa}){
        bmsgs := bmsgs ∪
                    {[type ↦ "2b", acc ↦ self, bal ↦ b, val ↦ v]} ;
        maxVVal[self] := v ;
      } ;
    maxBal[self] := b ;
    maxVBal[self] := b
  }
```

At any time, an acceptor can learn that some set of $1b$ messages were sent (but only if they atually were sent).

```
  macro LearnsSent(b){
    with (S ∈ SUBSET sentMsgs("1b", b)){
      knowsSent[self] := knowsSent[self] ∪ S
    }
  }
```

A malicious acceptor *self* can send any acceptor message indicating that it is from itself. Since a malicious acceptor could allow other malicious processes to forge its messages, this action could represent the sending of the message by any malicious process.

```
  macro FakingAcceptor(){
    with (m ∈ {mm ∈ 1bMessage ∪ 2avMessage ∪ 2bMessage :
                  mm.acc = self}){
        bmsgs := bmsgs ∪ {m}
    }
  }
```

We combine these individual actions into a complete algorithm in the usual way, with separate process declarations for the acceptor, leader, and fake acceptor processes.

```
  process (acceptor ∈ Acceptor){
    acc: while (TRUE){
           with (b ∈ Ballot){either Phase1b(b)or Phase2av(b)
                                 or Phase2b(b)or LearnsSent(b)}
    }
  }
```

```
  process (leader ∈ Ballot){
    ldr: while (TRUE){
           either Phase1a()or Phase1c()
         }
   }

  process (facceptor ∈ FakeAcceptor){
     facc : while (TRUE){FakingAcceptor()}
   }
}
```

Below is the TLA+ translation, as produced by the translator. (Some blank lines have been removed.)

```
\* BEGIN TRANSLATION
VARIABLES maxBal, maxVBal, maxVVal, 2avSent, knowsSent, bmsgs

\* define statement
sentMsgs(type, bal) ≜ {m ∈ bmsgs : m.type = type ∧ m.bal = bal}

KnowsSafeAt(ac, b, v) ≜
  LET S ≜ {m ∈ knowsSent[ac] : m.bal = b}
  IN  ∨ ∃ BQ ∈ ByzQuorum :
          ∀ a ∈ BQ : ∃ m ∈ S : ∧ m.acc = a
                               ∧ m.mbal = -1
      ∨ ∃ c ∈ 0 .. (b - 1) :
          ∧ ∃ BQ ∈ ByzQuorum :
              ∀ a ∈ BQ : ∃ m ∈ S : ∧ m.acc = a
                                   ∧ m.mbal ≤ c
                                   ∧ (m.mbal = c) ⇒ (m.mval = v)
          ∧ ∃ WQ ∈ WeakQuorum :
              ∀ a ∈ WQ :
                ∃ m ∈ S : ∧ m.acc = a
                          ∧ ∃ r ∈ m.m2av : ∧ r.bal ≥ c
                                           ∧ r.val = v

vars ≜ ⟨maxBal, maxVBal, maxVVal, 2avSent, knowsSent, bmsgs⟩

ProcSet ≜ (Acceptor) ∪ (Ballot) ∪ (FakeAcceptor)

Init ≜ \* Global variables
       ∧ maxBal = [a ∈ Acceptor ↦ -1]
       ∧ maxVBal = [a ∈ Acceptor ↦ -1]
       ∧ maxVVal = [a ∈ Acceptor ↦ None]
       ∧ 2avSent  = [a ∈ Acceptor ↦ {}]
       ∧ knowsSent = [a ∈ Acceptor ↦ {}]
       ∧ bmsgs = {}
```

```
acceptor(self) ≜ ∃ b ∈ Ballot :
                   ∨ ∧ (b > maxBal[self]) ∧ (sentMsgs("1a", b) ≠ {})
                     ∧ maxBal' = [maxBal EXCEPT ![self] = b]
                     ∧ bmsgs' = (bmsgs ∪ {[type  ↦ "1b", bal ↦ b, acc ↦ self,
                                           m2av ↦ 2avSent[self],
                                           mbal  ↦ maxVBal[self], mval ↦ maxVVal[self]]})
                     ∧ UNCHANGED ⟨maxVBal, maxVVal, 2avSent, knowsSent⟩
                   ∨ ∧ ∧ maxBal[self] ≤ b
                       ∧ ∀ r ∈ 2avSent[self] : r.bal < b
                     ∧ ∃ m ∈ {ms ∈ sentMsgs("1c", b) : KnowsSafeAt(self, b, ms.val)} :
                         ∧ bmsgs' = (bmsgs ∪
                                     {[type ↦ "2av", bal ↦ b, val ↦ m.val, acc ↦ self]})
                         ∧ 2avSent' = [2avSent EXCEPT ![self] = {r ∈ 2avSent[self] : r.val ≠ m.val}
                                                              ∪ {[val ↦ m.val, bal ↦ b]}]
                     ∧ maxBal' = [maxBal EXCEPT ![self] = b]
                     ∧ UNCHANGED ⟨maxVBal, maxVVal, knowsSent⟩
                   ∨ ∧ maxBal[self] ≤ b
                     ∧ ∃ v ∈ {vv     ∈ Value :
                                ∃ Q ∈ ByzQuorum :
                                  ∀ aa ∈ Q :
                                    ∃ m ∈ sentMsgs("2av", b) : ∧ m.val = vv
                                                               ∧ m.acc = aa} :
                         ∧ bmsgs' = (bmsgs ∪
                                     {[type ↦ "2b", acc ↦ self, bal ↦ b, val ↦ v]})
                         ∧ maxVVal' = [maxVVal EXCEPT ![self] = v]
                     ∧ maxBal' = [maxBal EXCEPT ![self] = b]
                     ∧ maxVBal' = [maxVBal EXCEPT ![self] = b]
                     ∧ UNCHANGED ⟨2avSent, knowsSent⟩
                   ∨ ∧ ∃ S ∈ SUBSET sentMsgs("1b", b) :
                         knowsSent' = [knowsSent EXCEPT ![self] = knowsSent[self] ∪ S]
                     ∧ UNCHANGED ⟨maxBal, maxVBal, maxVVal, 2avSent, bmsgs⟩

leader(self) ≜  ∧ ∨ ∧ bmsgs' = (bmsgs ∪ {[type ↦ "1a", bal ↦ self]})
                  ∨ ∧ ∃ S ∈ SUBSET [type : {"1c"}, bal : {self}, val : Value] :
                        bmsgs' = (bmsgs ∪ S)
                ∧ UNCHANGED ⟨maxBal, maxVBal, maxVVal, 2avSent, knowsSent⟩

facceptor(self) ≜  ∧ ∃ m ∈ {mm ∈ 1bMessage ∪ 2avMessage ∪ 2bMessage :
                               mm.acc = self} :
                       bmsgs' = (bmsgs ∪ {m})
                   ∧ UNCHANGED ⟨maxBal, maxVBal, maxVVal, 2avSent,
                                knowsSent⟩

Next ≜ (∃ self ∈ Acceptor : acceptor(self))
       ∨ (∃ self ∈ Ballot : leader(self))
       ∨ (∃ self ∈ FakeAcceptor : facceptor(self))
```

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

END TRANSLATION

As in module *PConProof*, we now rewrite the next-state relation in a form more convenient for writing proofs.

$$
\begin{array}{l}
Phase1b(self,\ b) \triangleq \\
\quad \land (b > maxBal[self]) \land (sentMsgs(\text{“1a”},\ b) \neq \{\}) \\
\quad \land maxBal' = [maxBal \text{ EXCEPT } ![self] = b] \\
\quad \land bmsgs' = bmsgs \cup \{[type \mapsto \text{“1b”},\ bal \mapsto b,\ acc \mapsto self, \\
\qquad\qquad\qquad\qquad m2av \mapsto 2avSent[self], \\
\qquad\qquad\qquad\qquad mbal \mapsto maxVBal[self],\ mval \mapsto maxVVal[self]]\} \\
\quad \land \text{UNCHANGED } \langle maxVBal,\ maxVVal,\ 2avSent,\ knowsSent\rangle \\
\\
Phase2av(self,\ b) \triangleq \\
\quad \land maxBal[self] \leq b \\
\quad \land \forall\, r \in 2avSent[self] : r.bal < b \\
\quad \land \exists\, m \in \{ms \in sentMsgs(\text{“1c”},\ b) : KnowsSafeAt(self,\ b,\ ms.val)\} : \\
\qquad \land bmsgs' = bmsgs \cup \\
\qquad\qquad\qquad \{[type \mapsto \text{“2av”},\ bal \mapsto b,\ val \mapsto m.val,\ acc \mapsto self]\} \\
\qquad \land 2avSent' = [2avSent \text{ EXCEPT} \\
\qquad\qquad\qquad\qquad ![self] = \{r \in 2avSent[self] : r.val \neq m.val\} \\
\qquad\qquad\qquad\qquad\qquad\qquad \cup \{[val \mapsto m.val,\ bal \mapsto b]\}] \\
\quad \land maxBal' = [maxBal \text{ EXCEPT } ![self] = b] \\
\quad \land \text{UNCHANGED } \langle maxVBal,\ maxVVal,\ knowsSent\rangle \\
\\
Phase2b(self,\ b) \triangleq \\
\quad \land maxBal[self] \leq b \\
\quad \land \exists\, v \in \{vv \in Value : \\
\qquad\qquad \exists\, Q \in ByzQuorum : \\
\qquad\qquad\quad \forall\, a \in Q : \\
\qquad\qquad\qquad \exists\, m \in sentMsgs(\text{“2av”},\ b) : \land m.val = vv \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \land m.acc = a\} : \\
\qquad \land bmsgs' = (bmsgs \cup \\
\qquad\qquad\qquad \{[type \mapsto \text{“2b”},\ acc \mapsto self,\ bal \mapsto b,\ val \mapsto v]\}) \\
\qquad \land maxVVal' = [maxVVal \text{ EXCEPT } ![self] = v] \\
\quad \land maxBal' = [maxBal \text{ EXCEPT } ![self] = b] \\
\quad \land maxVBal' = [maxVBal \text{ EXCEPT } ![self] = b] \\
\quad \land \text{UNCHANGED } \langle 2avSent,\ knowsSent\rangle \\
\\
LearnsSent(self,\ b) \triangleq \\
\quad \land \exists\, S \in \text{SUBSET } sentMsgs(\text{“1b”},\ b) : \\
\qquad knowsSent' = [knowsSent \text{ EXCEPT } ![self] = knowsSent[self] \cup S] \\
\quad \land \text{UNCHANGED } \langle maxBal,\ maxVBal,\ maxVVal,\ 2avSent,\ bmsgs\rangle \\
\\
Phase1a(self) \triangleq
\end{array}
$$

$\land\ bmsgs' = (bmsgs \cup \{[type \mapsto \text{“1a”},\ bal \mapsto self]\})$
$\land$ UNCHANGED $\langle maxBal,\ maxVBal,\ maxVVal,\ 2avSent,\ knowsSent\rangle$

$Phase1c(self) \triangleq$
$\land \exists\, S \in$ SUBSET $[type : \{\text{“1c”}\},\ bal : \{self\},\ val : Value] :$
$bmsgs' = (bmsgs \cup S)$
$\land$ UNCHANGED $\langle maxBal,\ maxVBal,\ maxVVal,\ 2avSent,\ knowsSent\rangle$

$FakingAcceptor(self) \triangleq$
$\land \exists\, m \in \{mm \in 1bMessage \cup 2avMessage \cup 2bMessage : mm.acc = self\} :$
$bmsgs' = (bmsgs \cup \{m\})$
$\land$ UNCHANGED $\langle maxBal,\ maxVBal,\ maxVVal,\ 2avSent,\ knowsSent\rangle$

The following lemma describes how the next-state relation *Next* can be written in terms of the actions defined above.

LEMMA $NextDef \triangleq$
$Next = \lor \exists\, self \in Acceptor :$
$\exists\, b \in Ballot : \lor Phase1b(self,\ b)$
$\lor Phase2av(self,\ b)$
$\lor Phase2b(self,\ b)$
$\lor LearnsSent(self,\ b)$
$\lor \exists\, self \in Ballot : \lor Phase1a(self)$
$\lor Phase1c(self)$
$\lor \exists\, self \in FakeAcceptor : FakingAcceptor(self)$
$\langle 1\rangle 1.\ \forall\, self : acceptor(self) = NextDef!2!1!(self)$
BY DEF $acceptor$, $Phase1b$, $Phase2av$, $Phase2b$, $LearnsSent$
$\langle 1\rangle 2.\ \forall\, self : leader(self) = NextDef!2!2!(self)$
BY DEF $leader$, $Phase1a$, $Phase1c$
$\langle 1\rangle 3.\ \forall\, self : facceptor(self) = NextDef!2!3!(self)$
BY DEF $facceptor$, $FakingAcceptor$
$\langle 1\rangle 4.$ QED
BY $\langle 1\rangle 1$, $\langle 1\rangle 2$, $\langle 1\rangle 3$
DEF $Next$, $acceptor$, $leader$, $facceptor$

THE REFINEMENT MAPPING

We define a quorum to be the set of acceptors in a *Byzantine* quorum. The quorum assumption *QA* of module *PConProof*, which we here call *QuorumTheorem*, follows easily from the definition and assumption *BQA*.

$Quorum \triangleq \{S \cap Acceptor : S \in ByzQuorum\}$

THEOREM $QuorumTheorem \triangleq$
$\land \forall\, Q1,\ Q2 \in Quorum : Q1 \cap Q2 \neq \{\}$
$\land \forall\, Q \in Quorum : Q \subseteq Acceptor$
$\langle 1\rangle 1.\ QuorumTheorem!1$
$\langle 2\rangle 1.$ SUFFICES ASSUME NEW $Q1 \in Quorum$, NEW $Q2 \in Quorum$

PROVE $Q1 \cap Q2 \neq \{\}$

OBVIOUS

$\langle 2 \rangle 2$. PICK $BQ1 \in ByzQuorum$, $BQ2 \in ByzQuorum$:

$\land\ Q1 = BQ1 \cap Acceptor$

$\land\ Q2 = BQ2 \cap Acceptor$

BY $\langle 2 \rangle 1$ DEF $Quorum$

$\langle 2 \rangle 3$. $Q1 \cap Q2 = BQ1 \cap BQ2 \cap Acceptor$

BY $\langle 2 \rangle 2$

$\langle 2 \rangle 4$. QED

BY BQA, $\langle 2 \rangle 3$

$\langle 1 \rangle 2$. $QuorumTheorem!2$

BY DEF $Quorum$

$\langle 1 \rangle 3$. QED

BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$

We now define refinement mapping under which our algorithm implements the algorithm of module *PConProof*. First, we define the set *msgs* that implements the variable of the same name in *PConProof*. There are two non-obvious parts of the definition.

1. The $1c$ messages in *msgs* should just be the ones that are legal–that is, messages whose value is safe at the indicated ballot. The obvious way to define legality is in terms of $1b$ messages that have been sent. However, this has the effect that sending a $1b$ message can add both that $1b$ message and one or more $1c$ messages to *msgs*. Proving implementation under this refinement mapping would require adding a stuttering variable. Instead, we define the $1c$ message to be legal if the set of $1b$ messages that some acceptor knows were sent confirms its legality. Thus, those $1c$ messages are added to *msgs* by the *LearnsSent* ation, which has no other effect on the refinement mapping.

2. A $2a$ message is added to *msgs* when a quorum of acceptors have reacted to it by sending a $2av$ message.

$msgsOfType(t) \triangleq \{m \in bmsgs : m.type = t\}$

$acceptorMsgsOfType(t) \triangleq \{m \in msgsOfType(t) : m.acc \in Acceptor\}$

$1bRestrict(m) \triangleq [type \mapsto \text{“1b”},\ acc \mapsto m.acc,\ bal \mapsto m.bal,$
$mbal \mapsto m.mbal,\ mval \mapsto m.mval]$

$1bmsgs \triangleq \{1bRestrict(m) : m \in acceptorMsgsOfType(\text{“1b”})\}$

$1cmsgs \triangleq \{m \in msgsOfType(\text{“1c”}) :$
$\exists\, a \in Acceptor : KnowsSafeAt(a,\ m.bal,\ m.val)\}$

$2amsgs \triangleq \{m \in [type : \{\text{“2a”}\},\ bal : Ballot,\ val : Value] :$
$\exists\, Q \in Quorum :$
$\forall\, a \in Q :$
$\exists\, m2av \in acceptorMsgsOfType(\text{“2av”}) :$
$\land\ m2av.acc = a$
$\land\ m2av.bal = m.bal$
$\land\ m2av.val = m.val\}$

$msgs \triangleq msgsOfType(\text{“1a”}) \cup 1bmsgs \cup 1cmsgs \cup 2amsgs$
$\quad \cup acceptorMsgsOfType(\text{“2b”})$

We now define *PmaxBal*, the state function with which we instantiate the variable *maxBal* of *PConProof*. The reason we don't just instantiate it with the variable *maxBal* is that $maxBal[a]$ can change when acceptor a performs a $Phase2av$ ation, which does not correspond to any acceptor action of the *PConProof* algorithm. We want $PmaxBal[a]$ to change only when a performs a $Phase1b$ or $Phase2b$ ation–that is, when it sends a $1b$ or $2b$ message. Thus, we define $PmaxBal[a]$ to be the largest *bal* field of all $1b$ and $2b$ messages sent by a .

To define *PmaxBal*, we need to define an operator *MaxBallot* so that $MaxBallot(S)$ is the largest element of S if S is non-empty a finite set consisting of ballot numbers and possibly the value -1.

$MaxBallot(S) \triangleq$
IF $S = \{\}$ THEN -1
ELSE CHOOSE $mb \in S : \forall\, x \in S : mb \geq x$

To prove that the CHOOSE in this definition actually does choose a maximum of S when S is nonempty, we need the following trivial fact. It has been checked by *TLC* with $-5 \,..\, 5$ substituted for *Int*.

AXIOM $FiniteSetHasMax \triangleq$
$\forall\, S \in$ SUBSET $Int :$
$IsFiniteSet(S) \wedge (S \neq \{\}) \Rightarrow \exists\, max \in S : \forall\, x \in S : max \geq x$

Our proofs use this property of *MaxBallot*.

THEOREM $MaxBallotProp \triangleq$
$\forall\, S \in$ SUBSET $(Ballot \cup \{-1\}) :$
$IsFiniteSet(S) \Rightarrow$
IF $S = \{\}$ THEN $MaxBallot(S) = -1$
ELSE $\wedge\, MaxBallot(S) \in S$
$\wedge\, \forall\, x \in S : MaxBallot(S) \geq x$
$\langle 1 \rangle$ SUFFICES ASSUME NEW $S \in$ SUBSET $(Ballot \cup \{-1\})$,
$IsFiniteSet(S)$
PROVE $MaxBallotProp!(S)!2$
OBVIOUS
$\langle 1 \rangle 1$.CASE $S = \{\}$
BY $\langle 1 \rangle 1$ DEF *MaxBallot*
$\langle 1 \rangle 2$.CASE $S \neq \{\}$
$\langle 2 \rangle 1$. $S \in$ SUBSET *Int*
BY DEF *Ballot* $\langle 2 \rangle 1$
$\langle 2 \rangle 2$. $\exists\, mb \in S : \forall\, x \in S : mb \geq x$
BY $\langle 2 \rangle 1$, $\langle 1 \rangle 2$, *FiniteSetHasMax*

$\langle 2 \rangle 3$. QED
BY $\langle 1 \rangle 2$, $\langle 2 \rangle 2$ DEF *MaxBallot*
$\langle 1 \rangle 3$. QED
BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$

We now prove a couple of lemmas about *MaxBallot*.

```
LEMMA MaxBallotLemma1 ≜
            ∀ S ∈ SUBSET (Ballot ∪ {−1}) :
              IsFiniteSet(S) ⇒
                ∀ y ∈ S :
                  (∀ x ∈ S : y ≥ x) ⇒ (y = MaxBallot(S))
⟨1⟩1. SUFFICES ASSUME NEW S ∈ SUBSET (Ballot ∪ {−1}),
                       IsFiniteSet(S),
                       NEW y ∈ S,
                       ∀ x ∈ S : y ≥ x
               PROVE   y = MaxBallot(S)
  OBVIOUS
⟨1⟩2. ∧ MaxBallot(S) ∈ S
      ∧ ∀ x ∈ S : MaxBallot(S) ≥ x
  BY ⟨1⟩1, MaxBallotProp
⟨1⟩3. MaxBallot(S) ≥ y
  BY  ⟨1⟩2
⟨1⟩4. y ∈ Ballot ∪ {−1}
  OBVIOUS
⟨1⟩5 y ≥ MaxBallot(S)
  BY ⟨1⟩1, ⟨1⟩2
⟨1⟩6. ∀ mbs ∈ Ballot ∪ {−1} :
        mbs ≥ y ∧ y ≥ mbs ⇒ y = mbs
  BY ⟨1⟩4, SimpleArithmetic DEF Ballot
⟨1⟩7. MaxBallot(S) ∈ Ballot ∪ {−1}
  BY ⟨1⟩2
⟨1⟩8. QED
  BY ⟨1⟩3, ⟨1⟩5, ⟨1⟩6, ⟨1⟩7

LEMMA MaxBallotLemma2 ≜
            ∀ S, T ∈ SUBSET (Ballot ∪ {−1}) :
              IsFiniteSet(S) ∧ IsFiniteSet(T) ⇒
                MaxBallot(S ∪ T) = IF MaxBallot(S) ≥ MaxBallot(T)
                                   THEN MaxBallot(S)
                                   ELSE MaxBallot(T)
⟨1⟩1. ∀ S ∈ SUBSET (Ballot ∪ {−1}) :
       IsFiniteSet(S) ⇒ (MaxBallot(S) ∈ Ballot ∪ {−1})
  BY MaxBallotProp
⟨1⟩2. ASSUME NEW S ∈ SUBSET (Ballot ∪ {−1}),
             NEW T ∈ SUBSET (Ballot ∪ {−1}),
             IsFiniteSet(S) ∧ IsFiniteSet(T),
             MaxBallot(S) ≥ MaxBallot(T)
      PROVE  MaxBallot(S ∪ T) = MaxBallot(S)
  ⟨2⟩1.CASE S = {}
    ⟨3⟩1.CASE T = {}
      BY ⟨3⟩1
```

⟨3⟩2.CASE $T \neq \{\}$
⟨4⟩1. $MaxBallot(S) = -1$
BY ⟨1⟩2, ⟨2⟩1, *MaxBallotProp*
⟨4⟩2. $MaxBallot(T) = -1$
⟨5⟩1. $\forall\, x \in Ballot \cup \{-1\} : -1 \geq x \Rightarrow x = -1$
BY *SimpleArithmetic* DEF *Ballot*
⟨5⟩2. $MaxBallot(T) \in Ballot \cup \{-1\}$
BY ⟨1⟩2, ⟨1⟩1
⟨5⟩3. QED
BY ⟨5⟩1, ⟨5⟩2, ⟨4⟩1, ⟨1⟩2
⟨4⟩3. $-1 \in T \wedge \forall\, x \in T : -1 \geq x$
BY ⟨4⟩2, ⟨3⟩2, ⟨1⟩2, *MaxBallotProp*
⟨4⟩4. QED
BY ⟨4⟩1, ⟨4⟩3, ⟨3⟩2, ⟨2⟩1, ⟨1⟩2, *MaxBallotLemma*1
⟨3⟩3. QED
BY ⟨3⟩1, ⟨3⟩2
⟨2⟩2.CASE $S \neq \{\}$
⟨3⟩1.CASE $T = \{\}$
BY ⟨3⟩1
⟨3⟩2.CASE $T \neq \{\}$
⟨4⟩1. $\wedge\, MaxBallot(S) \in S$
$\wedge\, \forall\, x \in S : MaxBallot(S) \geq x$
BY ⟨2⟩2, ⟨1⟩2, *MaxBallotProp*
⟨4⟩2. $\wedge\, MaxBallot(T) \in T$
$\wedge\, \forall\, x \in T : MaxBallot(T) \geq x$
BY ⟨3⟩2, ⟨1⟩2, *MaxBallotProp*
⟨4⟩3. $\forall\, x,\, y \in Ballot \cup \{-1\},\ TT \in \text{SUBSET}\ (Ballot \cup \{-1\}) :$
$(x \geq y) \wedge (\forall\, z \in TT : y \geq z) \Rightarrow (\forall\, z \in TT : x \geq z)$
⟨5⟩1. SUFFICES ASSUME NEW $x \in Ballot \cup \{-1\}$, NEW $y \in Ballot \cup \{-1\}$,
NEW $TT \in \text{SUBSET}\ (Ballot \cup \{-1\})$,
$(x \geq y)$, $\forall\, z \in TT : y \geq z$
PROVE $(\forall\, z \in TT : x \geq z)$
OBVIOUS
⟨5⟩2. SUFFICES ASSUME NEW $z \in TT$
PROVE $x \geq z$
OBVIOUS
⟨5⟩3. $z \in Ballot \cup \{-1\}$
BY ⟨5⟩1, ⟨5⟩2
⟨5⟩4. $y \geq z$
BY ⟨5⟩1
⟨5⟩5. QED
BY *SimpleArithmetic*, ⟨5⟩1, ⟨5⟩3, ⟨5⟩4 DEF *Ballot*
⟨4⟩4. $\wedge\, MaxBallot(S) \in Ballot \cup \{-1\}$
$\wedge\, MaxBallot(T) \in Ballot \cup \{-1\}$

BY $\langle 1\rangle 1$, $\langle 1\rangle 2$

$\langle 4\rangle 5$. $\forall\, x \in T : \mathit{MaxBallot}(S) \geq x$

BY $\langle 1\rangle 2$, $\langle 4\rangle 2$, $\langle 4\rangle 3$, $\langle 4\rangle 4$

$\langle 4\rangle 6$. $\land\, \mathit{MaxBallot}(S) \in S \cup T$
$\land\, \forall\, x \in S \cup T : \mathit{MaxBallot}(S) \geq x$

BY $\langle 4\rangle 1$, $\langle 4\rangle 5$

$\langle 4\rangle 7$. $\mathit{IsFiniteSet}(S \cup T)$

BY $\langle 1\rangle 2$, $\mathit{UnionOfFiniteSetsFinite}$

$\langle 4\rangle 8$. QED

BY $\langle 4\rangle 6$, $\langle 4\rangle 7$, $\mathit{MaxBallotLemma}1$

$\langle 3\rangle 3$. QED

BY $\langle 3\rangle 1$, $\langle 3\rangle 2$

$\langle 2\rangle 3$. QED

BY $\langle 2\rangle 1$, $\langle 2\rangle 2$

$\langle 1\rangle 3$. SUFFICES ASSUME NEW $S \in$ SUBSET $(\mathit{Ballot} \cup \{-1\})$,
NEW $T \in$ SUBSET $(\mathit{Ballot} \cup \{-1\})$,
$\mathit{IsFiniteSet}(S) \land \mathit{IsFiniteSet}(T)$
PROVE $\mathit{MaxBallot}(S \cup T) =$ IF $\mathit{MaxBallot}(S) \geq \mathit{MaxBallot}(T)$
THEN $\mathit{MaxBallot}(S)$
ELSE $\mathit{MaxBallot}(T)$

OBVIOUS

$\langle 1\rangle 4$.CASE $\mathit{MaxBallot}(S) \geq \mathit{MaxBallot}(T)$

$\langle 2\rangle$ SUFFICES $\mathit{MaxBallot}(S \cup T) = \mathit{MaxBallot}(S)$

BY $\langle 1\rangle 3$, $\langle 1\rangle 4$

$\langle 2\rangle$ QED

BY $\langle 1\rangle 2$, $\langle 1\rangle 3$, $\langle 1\rangle 4$ added $\langle 1\rangle 3$ on 13 *Nov* 2010

$\langle 1\rangle 5$.CASE $\neg(\mathit{MaxBallot}(S) \geq \mathit{MaxBallot}(T))$

$\langle 2\rangle$ SUFFICES $\mathit{MaxBallot}(T \cup S) = \mathit{MaxBallot}(T)$

BY $\langle 1\rangle 3$, $\langle 1\rangle 5$

$\langle 2\rangle$ $\mathit{MaxBallot}(T) \geq \mathit{MaxBallot}(S)$

$\langle 3\rangle 1$. $\forall\, x,\ y \in \mathit{Ballot} \cup \{-1\} : x \geq y \lor y \geq x$

BY *SimpleArithmetic* DEF *Ballot*

$\langle 3\rangle 2$. $\land\, \mathit{MaxBallot}(S) \in \mathit{Ballot} \cup \{-1\}$
$\land\, \mathit{MaxBallot}(T) \in \mathit{Ballot} \cup \{-1\}$

BY $\langle 1\rangle 1$, $\langle 1\rangle 3$

$\langle 3\rangle 3$. QED

BY $\langle 1\rangle 5$, $\langle 3\rangle 1$, $\langle 3\rangle 2$

$\langle 2\rangle$ QED

BY $\langle 1\rangle 2$, $\langle 1\rangle 3$, $\langle 1\rangle 5$

$\langle 1\rangle 6$. QED

BY $\langle 1\rangle 4$, $\langle 1\rangle 5$

We finally come to our definition of *PmaxBal*, the state function substituted for variable *maxBal* of module *PConProof* by our refinement mapping. We also prove a couple of lemmas about *PmaxBal*.

```
1bOr2bMsgs ≜ {m ∈ bmsgs : m.type ∈ {"1b", "2b"}}

PmaxBal ≜ [a ∈ Acceptor ↦
            MaxBallot({m.bal : m ∈ {ma ∈ 1bOr2bMsgs :
                                    ma.acc = a}})]

LEMMA PmaxBalLemma1 ≜
        ∀ m : ∧ bmsgs' = bmsgs ∪ {m}
              ∧ m.type ≠ "1b" ∧ m.type ≠ "2b"
              ⇒ PmaxBal' = PmaxBal
⟨1⟩ SUFFICES ASSUME NEW m,
                    bmsgs' = bmsgs ∪ {m},
                    m.type ≠ "1b" ∧ m.type ≠ "2b"
             PROVE  PmaxBal' = PmaxBal
  OBVIOUS
⟨1⟩1. ASSUME NEW ma, ma.type ∈ {"1b", "2b"}
      PROVE ma ∈ bmsgs' ≡ ma ∈ bmsgs
  BY ⟨1⟩1
⟨1⟩2. QED
  BY ⟨1⟩1 DEF PmaxBal, 1bOr2bMsgs

LEMMA PmaxBalLemma2 ≜
        ∀ m : (bmsgs' = bmsgs ∪ {m}) ⇒
          ∀ a ∈ Acceptor : (m.acc ≠ a ⇒ PmaxBal'[a] = PmaxBal[a])
⟨1⟩ SUFFICES ASSUME NEW m,
                    bmsgs' = bmsgs ∪ {m},
                    NEW a ∈ Acceptor,
                    m.acc ≠ a
             PROVE  PmaxBal'[a] = PmaxBal[a]
  OBVIOUS
⟨1⟩1. ASSUME NEW ma, ma.acc ≠ m.acc
      PROVE ma ∈ bmsgs' ≡ ma ∈ bmsgs
  BY ⟨1⟩1
⟨1⟩2. QED
  BY ⟨1⟩1 DEF PmaxBal, 1bOr2bMsgs
```

Finally, we define the refinement mapping. As before, for any operator *op* defined in module *PConProof*, the following INSTANCE statement defines $P!op$ to be the operator obtained from *op* by the indicated substitutions, along with the implicit substitutions

```
Acceptor ← Acceptor,
Quorum ← Quorum
Value ← Value
maxVBal ← maxVBal
maxVVal ← maxVVal
msgs     ← msgs
```

$P \triangleq$ INSTANCE $PConProof$ WITH $maxBal \leftarrow PmaxBal$

We now define the inductive invariant *Inv* used in our proof. It is defined to be the conjunction of a number of separate invariants that we define first, starting with the ever-present type-correctness invariant.

$$\begin{aligned} TypeOK \triangleq\ & \land maxBal \in [Acceptor \to Ballot \cup \{-1\}] \\ & \land 2avSent \in [Acceptor \to \text{SUBSET } [val : Value,\ bal : Ballot]] \\ & \land maxVBal \in [Acceptor \to Ballot \cup \{-1\}] \\ & \land maxVVal \in [Acceptor \to Value \cup \{None\}] \\ & \land knowsSent \in [Acceptor \to \text{SUBSET } 1bMessage] \\ & \land bmsgs \subseteq BMessage \end{aligned}$$

To use the definition of *PmaxBal*, we need to know that the set of $1b$ and $2b$ messages in *bmsgs* is finite. This is asserted by the following invariant. Note that the set *bmsgs* is not necessarily finite because we allow a $Phase1c$ action to send an infinite number of $1c$ messages.

$$bmsgsFinite \triangleq IsFiniteSet(1bOr2bMsgs)$$

The following lemma is used to prove the invariance of *bmsgsFinite*.

LEMMA $FiniteMsgsLemma \triangleq$
$\quad \forall\, m : bmsgsFinite \land (bmsgs' = bmsgs \cup \{m\}) \Rightarrow bmsgsFinite'$
$\langle 1 \rangle$ SUFFICES ASSUME NEW m, $bmsgsFinite$, $bmsgs' = bmsgs \cup \{m\}$
$\quad$ PROVE $bmsgsFinite'$
$\quad$ OBVIOUS
$\langle 1 \rangle 1$.CASE $(m.type \in \{\text{“1b”},\ \text{“2b”}\})$
$\quad \langle 2 \rangle 1.\ 1bOr2bMsgs' = 1bOr2bMsgs \cup \{m\}$
$\quad\quad$ BY $\langle 1 \rangle 1$ DEF $1bOr2bMsgs$
$\quad \langle 2 \rangle 2.$ QED
$\quad\quad$ BY $\langle 2 \rangle 1$, $SingletonSetFinite$, $UnionOfFiniteSetsFinite$ DEF $bmsgsFinite$
$\langle 1 \rangle 2$.CASE $m.type \notin \{\text{“1b”},\ \text{“2b”}\}$
$\quad$ BY $\langle 1 \rangle 2$ DEF $bmsgsFinite$, $1bOr2bMsgs$
$\langle 1 \rangle 3.$ QED
$\quad$ BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$

Invariant $1bInv1$ asserts that if (good) acceptor a has $mCBal[a] \neq -1$, then there is a $1c$ message for ballot $mCBal[a]$ and value $mCVal[a]$ in the emulated execution of *PaxosConsensus*.

$$\begin{aligned} 1bInv1 \triangleq\ & \forall\, m \in bmsgs\ : \\ & \land m.type = \text{“1b”} \\ & \land m.acc \in Acceptor \\ & \Rightarrow \forall\, r \in m.m2av : \\ & \quad [type \mapsto \text{“1c”},\ bal \mapsto r.bal,\ val \mapsto r.val] \in msgs \end{aligned}$$

Invariant $1bInv2$ asserts that an acceptor sends at most one $1b$ message for any ballot.

$$\begin{aligned} 1bInv2 \triangleq\ & \forall\, m1,\ m2 \in bmsgs\ : \\ & \land m1.type = \text{“1b”} \\ & \land m2.type = \text{“1b”} \\ & \land m1.acc \in Acceptor \end{aligned}$$

$\land\ m1.acc = m2.acc$
$\land\ m1.bal = m2.bal$
$\Rightarrow\ m1 = m2$

Invariant $2avInv1$ asserts that an acceptor sends at most one $2av$ message in any ballot.

$2avInv1 \triangleq \forall\, m1,\, m2 \in bmsgs :$
$\land\ m1.type = \text{“2av”}$
$\land\ m2.type = \text{“2av”}$
$\land\ m1.acc \in Acceptor$
$\land\ m1.acc = m2.acc$
$\land\ m1.bal = m2.bal$
$\Rightarrow\ m1 = m2$

Invariant $2avInv2$ follows easily from the meaning (and setting) of $2avSent$.

$2avInv2 \triangleq \forall\, m \in bmsgs :$
$\land\ m.type = \text{“2av”}$
$\land\ m.acc \in Acceptor$
$\Rightarrow \exists\, r \in 2avSent[m.acc] :\ \land\ r.val = m.val$
$\land\ r.bal \geq m.bal$

Invariant $2avInv3$ asserts that an acceptor sends a $2av$ message only if the required $1c$ message exists in the emulated execution of *PaxosConsensus*.

$2avInv3 \triangleq \forall\, m \in bmsgs :$
$\land\ m.type = \text{“2av”}$
$\land\ m.acc \in Acceptor$
$\Rightarrow [type \mapsto \text{“1c”},\ bal \mapsto m.bal,\ val \mapsto m.val] \in msgs$

Invariant *maxBalInv* is a simple consequence of the fact that an acceptor a sets $maxBal[a]$ to b whenever it sends a $1b$, $2av$, or $2b$ message in ballot b.

$maxBalInv \triangleq \forall\, m \in bmsgs :$
$\land\ m.type \in \{\text{“1b”},\ \text{“2av”},\ \text{“2b”}\}$
$\land\ m.acc \in Acceptor$
$\Rightarrow\ m.bal \leq maxBal[m.acc]$

Invariant *accInv* asserts some simple relations between the variables local to an acceptor, as well as the fact that acceptor a sets $maxCBal[a]$ to b and $maxCVal[a]$ to v only if there is a ballot-b $1c$ message for value c in the simulated execution of the *PaxosConsensus* algorithm.

$accInv \triangleq \forall\, a \in Acceptor :$
$\forall\, r \in 2avSent[a] :$
$\land\ r.bal \leq maxBal[a]$
$\land\ [type \mapsto \text{“1c”},\ bal \mapsto r.bal,\ val \mapsto r.val] \in msgs$

Invariant *knowsSentInv* simply asserts that for any acceptor a , $knowsSent[a]$ is a set of $1b$ messages that have actually been sent.

$knowsSentInv \triangleq \forall\, a \in Acceptor : knowsSent[a] \subseteq msgsOfType(\text{“1b”})$

$Inv \triangleq$

$TypeOK \land bmsgsFinite \land 1bInv1 \land 1bInv2 \land maxBalInv \land 2avInv1 \land 2avInv2$
$\land 2avInv3 \land accInv \land knowsSentInv$

We now prove some simple lemmas that are useful for reasoning about *PmaxBal*.

LEMMA $PMaxBalLemma3 \triangleq$
ASSUME $TypeOK$,
$bmsgsFinite$,
NEW $a \in Acceptor$
PROVE LET $S \triangleq \{m.bal : m \in \{ma \in bmsgs :$
$\land\ ma.type \in \{\text{“1b”}, \text{“2b”}\}$
$\land\ ma.acc = a\}\}$
IN $\land\ IsFiniteSet(S)$
$\land\ S \in \text{SUBSET } Ballot$
⟨1⟩ DEFINE $T \triangleq \{ma \in bmsgs : \land\ ma.type \in \{\text{“1b”}, \text{“2b”}\}$
$\land\ ma.acc = a\}$
$S \triangleq \{m.bal : m \in T\}$
⟨1⟩1. $IsFiniteSet(S)$
⟨2⟩1. $IsFiniteSet(T)$
BY $SubsetOfFiniteSetFinite$ DEF $bmsgsFinite$, $1bOr2bMsgs$
⟨2⟩ DEFINE $f[m \in T] \triangleq m.bal$
⟨2⟩2. $IsFiniteSet(\{f[m] : m \in T\})$
BY ⟨2⟩1, $ImageOfFiniteSetFinite$
⟨2⟩3. QED
BY ⟨2⟩2
⟨1⟩2. ASSUME NEW $b \in S$
PROVE $b \in Ballot$
⟨2⟩1. PICK $m \in bmsgs : b = m.bal \land m.type \in \{\text{“1b”}, \text{“2b”}\}$
OBVIOUS
⟨2⟩2.CASE $m.type = \text{“1b”}$
⟨3⟩1. $m \in 1bMessage$
BY ⟨2⟩2, $BMessageLemma$ DEF $TypeOK$
⟨3⟩2. QED
BY ⟨3⟩1, ⟨2⟩1 DEF $1bMessage$
⟨2⟩3.CASE $m.type = \text{“2b”}$
⟨3⟩1. $m \in 2bMessage$
BY ⟨2⟩3, $BMessageLemma$ DEF $TypeOK$
⟨3⟩2. QED
BY ⟨3⟩1, ⟨2⟩1 DEF $2bMessage$
⟨2⟩4. QED
BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3
⟨1⟩3. QED
BY ⟨1⟩1, ⟨1⟩2

LEMMA $PmaxBalLemma4 \triangleq$
$TypeOK \land maxBalInv \land bmsgsFinite \Rightarrow$

$\forall\, a \in Acceptor : PmaxBal[a] \leq maxBal[a]$

$\langle 1 \rangle$ SUFFICES ASSUME $TypeOK$,
$maxBalInv$,
$bmsgsFinite$,
NEW $a \in Acceptor$
PROVE $PmaxBal[a] \leq maxBal[a]$
OBVIOUS

$\langle 1 \rangle$ DEFINE $SM \triangleq \{ma \in bmsgs : \land\, ma.type \in \{\text{“1b”}, \text{“2b”}\}$
$\land\, ma.acc = a\}$
$S \triangleq \{ma.bal : ma \in SM\}$

$\langle 1 \rangle 1.\ PmaxBal[a] = MaxBallot(S)$
BY DEF $PmaxBal$, $1bOr2bMsgs$

$\langle 1 \rangle 2.\ \land\, IsFiniteSet(S)$
$\land\, S \in$ SUBSET $(Ballot \cup \{-1\})$
BY $PMaxBalLemma3$

$\langle 1 \rangle 3.\ \forall\, b \in S : b \leq maxBal[a]$
BY DEF $maxBalInv$

$\langle 1 \rangle 4$.CASE $S = \{\}$
$\langle 2 \rangle 1.\ PmaxBal[a] = -1$
BY $\langle 1 \rangle 2$, $\langle 1 \rangle 1$, $\langle 1 \rangle 4$, $MaxBallotProp$
$\langle 2 \rangle 2.\ maxBal[a] \in \{-1\} \cup Ballot$
BY DEF $TypeOK$
$\langle 2 \rangle 3.\ \forall\, b \in \{-1\} \cup Ballot : -1 \leq b$
BY $SimpleArithmetic$ DEF $Ballot$
$\langle 2 \rangle 4.$ QED
BY $\langle 2 \rangle 1$, $\langle 2 \rangle 2$, $\langle 2 \rangle 3$

$\langle 1 \rangle 5$.CASE $S \neq \{\}$
$\langle 2 \rangle 1.\ MaxBallot(S) \in S$
BY $\langle 1 \rangle 2$, $\langle 1 \rangle 5$, $MaxBallotProp$
$\langle 2 \rangle 2.$ QED
BY $\langle 1 \rangle 1$, $\langle 1 \rangle 3$, $\langle 2 \rangle 1$

$\langle 1 \rangle 6.$ QED
BY $\langle 1 \rangle 4$, $\langle 1 \rangle 5$

LEMMA $PmaxBalLemma5 \triangleq$
$TypeOK \land bmsgsFinite \Rightarrow$
$\forall\, a \in Acceptor : PmaxBal[a] \in Ballot \cup \{-1\}$

$\langle 1 \rangle$ SUFFICES ASSUME $TypeOK$, $bmsgsFinite$, NEW $a \in Acceptor$
PROVE $PmaxBal[a] \in Ballot \cup \{-1\}$
OBVIOUS

$\langle 1 \rangle$ DEFINE $S \triangleq \{m.bal : m \in \{ma \in bmsgs : \land\, ma.type \in \{\text{“1b”}, \text{“2b”}\}$
$\land\, ma.acc = a\}\}$

$\langle 1 \rangle 1.\ \land\, S \subseteq (Ballot \cup \{-1\})$
$\land\, IsFiniteSet(S)$

$\langle 2 \rangle$ DEFINE $M \triangleq \{ma \in bmsgs : \land\, ma.type \in \{\text{“1b”}, \text{“2b”}\}$

$\wedge\, ma.acc = a\}$

$\langle 2\rangle 1.\ IsFiniteSet(M)$
BY $SubsetOfFiniteSetFinite$ DEF $bmsgsFinite$, $1bOr2bMsgs$

$\langle 2\rangle 2.\ IsFiniteSet(S)$
$\langle 3\rangle$ DEFINE $f[x \in M] \triangleq x.bal$
$\langle 3\rangle 1.\ S = \{f[x] : x \in M\}$
OBVIOUS
$\langle 3\rangle$ HIDE DEF f, S, M
$\langle 3\rangle 2$ QED
BY $\langle 2\rangle 1$, $\langle 3\rangle 1$, $ImageOfFiniteSetFinite$

$\langle 2\rangle 3.\ \forall\, m \in M : m.bal \in Ballot \cup \{-1\}$
$\langle 3\rangle$ SUFFICES ASSUME NEW $m \in M$
PROVE $m.bal \in Ballot \cup \{-1\}$
OBVIOUS
$\langle 3\rangle 1.\ m \in bmsgs \wedge m.type \in \{\text{“1b”}, \text{“2b”}\}$
OBVIOUS
$\langle 3\rangle 2$.CASE $m.type = \text{“1b”}$
BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $BMessageLemma$ DEF $TypeOK$, $1bMessage$
$\langle 3\rangle 3$.CASE $m.type = \text{“2b”}$
BY $\langle 3\rangle 1$, $\langle 3\rangle 3$, $BMessageLemma$ DEF $TypeOK$, $2bMessage$
$\langle 3\rangle 4$. QED
BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$
$\langle 2\rangle$ QED
BY $\langle 2\rangle 2$, $\langle 2\rangle 3$

$\langle 1\rangle 2$. QED
BY $\langle 1\rangle 1$, , $\langle 1\rangle 2$, $MaxBallotProp$ DEF $PmaxBal$, $1bOr2bMsgs$

Now comes a bunch of useful lemmas.

We first prove that $P!NextDef$ is a valid theorem and give it the name $PNextDef$. This requires proving that the assumptions of module $PConProof$ are satisfied by the refinement mapping. Note that $P!NextDef!:$ is an abbreviation for the statement of theorem $P!NextDef$ – that is, for the statement of theorem $NextDef$ of module $PConProof$ under the substitutions of the refinement mapping.

LEMMA $PNextDef \triangleq P!NextDef!:$
$\langle 1\rangle 1.\ P!QA$
BY $QuorumTheorem$
$\langle 1\rangle 2.\ P!BallotAssump$
BY $BallotAssump$ DEF $Ballot$, $P!Ballot$, $ByzAcceptor$
$\langle 1\rangle 3$. QED
BY $P!NextDef$, $\langle 1\rangle 1$, $\langle 1\rangle 2$, $NoSetContainsEverything$

The provers have a hard time dealing with all the quantifiers inside the definition of $KnowsSafeAt$. To help them, we define some formulas that allow us to break it into pieces.

```
KSet(a, b) ≜ {m ∈ knowsSent[a] : m.bal = b}
KS11a(a, S)  ≜ ∃ m ∈ S : ∧ m.acc = a
                         ∧ m.mbal = −1
KS11(BQ, S) ≜ ∀ a ∈ BQ : KS11a(a, S)
KS1(S) ≜ ∃ BQ ∈ ByzQuorum : KS11(BQ, S)
KS21BQa(v, c, a, S)  ≜ ∃ m ∈ S : ∧ m.acc = a
                                 ∧ m.mbal ≤ c
                                 ∧ (m.mbal = c) ⇒ (m.mval = v)
KS21BQ(v, c, BQ, S) ≜ ∀ a ∈ BQ  : KS21BQa(v, c, a, S)
KS21(v, c, S) ≜ ∃ BQ ∈ ByzQuorum : KS21BQ(v, c, BQ, S)
KS22WQa(v, c, a, S)   ≜ ∃ m ∈ S : ∧ m.acc = a
                                  ∧ ∃ r ∈ m.m2av : ∧ r.bal ≥ c
                                                   ∧ r.val = v
KS22WQ(v, c, WQ, S) ≜ ∀ a ∈ WQ : KS22WQa(v, c, a, S)
KS22(v, c, S) ≜ ∃ WQ ∈ WeakQuorum : KS22WQ(v, c, WQ, S)
KS2(v, b, S) ≜ ∃ c ∈ 0 .. (b − 1) : ∧ KS21(v, c, S)
                                   ∧ KS22(v, c, S)
```

The following lemma asserts the obvious relation between *KnowsSafeAt* and the top-level definitions *KS*1, *KS*2, and *KSet*. The second conjunct is, of course, the primed version of the first. To understand why it is needed, see the discussion of *MsgsTypeLemmaPrime* below.

```
LEMMA KnowsSafeAtDef ≜
        ∀ a, b, v :
          ∧ KnowsSafeAt(a, b, v) = (KS1(KSet(a, b)) ∨ KS2(v, b, KSet(a, b)))
          ∧ KnowsSafeAt(a, b, v)′ = (KS1(KSet(a, b)′) ∨ KS2(v, b, KSet(a, b)′))
  BY DEF KnowsSafeAt, KSet, KS11a, KS11, KS1, KS21BQa,
          KS21BQ, KS21, KS22WQa, KS22WQ, KS22, KS2

LEMMA MsgsTypeLemma ≜
        ∀ m ∈ msgs : ∧ (m.type = "1a") ≡ (m ∈ msgsOfType("1a"))
                     ∧ (m.type = "1b") ≡ (m ∈ 1bmsgs)
                     ∧ (m.type = "1c") ≡ (m ∈ 1cmsgs)
                     ∧ (m.type = "2a") ≡ (m ∈ 2amsgs)
                     ∧ (m.type = "2b") ≡ (m ∈ acceptorMsgsOfType("2b"))
⟨1⟩ SUFFICES ASSUME NEW m ∈ msgs
             PROVE MsgsTypeLemma!(m)
  OBVIOUS
⟨1⟩1. (m ∈ msgsOfType("1a")) ⇒ (m.type = "1a")
  BY DEF msgsOfType
⟨1⟩2. (m ∈ 1bmsgs) ⇒ (m.type = "1b")
  ⟨2⟩ ∀ mm : [type ↦ "1b", acc ↦ mm.acc, bal ↦ mm.bal,
              mbal ↦ mm.mbal, mval ↦ mm.mval].type = "1b"
    OBVIOUS
  ⟨2⟩ QED
    BY DEF 1bmsgs, 1bRestrict
⟨1⟩3. (m ∈ 1cmsgs) ⇒ (m.type = "1c")
```

BY DEF $1cmsgs$, $msgsOfType$
$\langle 1\rangle 4.\ (m \in 2amsgs) \Rightarrow (m.type = \text{“2a”})$
BY DEF $2amsgs$
$\langle 1\rangle 5.\ (m \in acceptorMsgsOfType(\text{“2b”})) \Rightarrow (m.type = \text{“2b”})$
BY DEF $acceptorMsgsOfType$, $msgsOfType$
$\langle 1\rangle 6.$ QED
BY $\langle 1\rangle 1$, $\langle 1\rangle 2$, $\langle 1\rangle 3$, $\langle 1\rangle 4$, $\langle 1\rangle 5$ DEF $msgs$

The following lemma is the primed version of *MsgsTypeLemma*. That is, its statement is just the statement of *MsgsTypeLemma* primed. It follows from *MsgsTypeLemma* by the meta-theorem that if we can prove a state-predicate F as a (top-level) theorem, then we can deduce F'. A more general meta-theorem says that if we can prove a state predicate F that does not appear within the proof of an ASSUME /PROVE , then we can deduce F'. We expect this meta-theorem will be enshrined in a proof rule when temporal-logic reasoning is implemented in *TLAPS*. Until then, we must prove F' separately from F.

LEMMA $MsgsTypeLemmaPrime \triangleq$
$\forall\, m \in msgs' :\ \land (m.type = \text{“1a”}) \equiv (m \in msgsOfType(\text{“1a”})')$
$\land (m.type = \text{“1b”}) \equiv (m \in 1bmsgs')$
$\land (m.type = \text{“1c”}) \equiv (m \in 1cmsgs')$
$\land (m.type = \text{“2a”}) \equiv (m \in 2amsgs')$
$\land (m.type = \text{“2b”}) \equiv (m \in acceptorMsgsOfType(\text{“2b”})')$
$\langle 1\rangle$ SUFFICES ASSUME NEW $m \in msgs'$
PROVE $MsgsTypeLemmaPrime!(m)$
OBVIOUS
$\langle 1\rangle 1.\ (m \in msgsOfType(\text{“1a”})') \Rightarrow (m.type = \text{“1a”})$
BY DEF $msgsOfType$
$\langle 1\rangle 2.\ (m \in 1bmsgs') \Rightarrow (m.type = \text{“1b”})$
$\langle 2\rangle\ \forall\, mm : [type \mapsto \text{“1b”},\ acc \mapsto mm.acc,\ bal \mapsto mm.bal,$
$mbal \mapsto mm.mbal,\ mval \mapsto mm.mval].type = \text{“1b”}$
OBVIOUS
$\langle 2\rangle$ QED
BY DEF $1bmsgs$, $1bRestrict$
$\langle 1\rangle 3.\ (m \in 1cmsgs') \Rightarrow (m.type = \text{“1c”})$
BY DEF $1cmsgs$, $msgsOfType$
$\langle 1\rangle 4.\ (m \in 2amsgs') \Rightarrow (m.type = \text{“2a”})$
BY DEF $2amsgs$
$\langle 1\rangle 5.\ (m \in acceptorMsgsOfType(\text{“2b”})') \Rightarrow (m.type = \text{“2b”})$
BY DEF $acceptorMsgsOfType$, $msgsOfType$
$\langle 1\rangle 6.$ QED
BY $\langle 1\rangle 1$, $\langle 1\rangle 2$, $\langle 1\rangle 3$, $\langle 1\rangle 4$, $\langle 1\rangle 5$ DEF $msgs$

The following lemma describes how *msgs* is changed by the actions of the algorithm.

LEMMA $MsgsLemma \triangleq$
$TypeOK \Rightarrow$
$\land\ \forall\, self \in Acceptor,\ b \in Ballot :$
$Phase1b(self,\ b) \Rightarrow$

$msgs' = msgs \cup$
$\{[type \mapsto$ "1b", $acc \mapsto self$, $bal \mapsto b$,
$mbal \mapsto maxVBal[self]$, $mval \mapsto maxVVal[self]]\}$
$\land \quad \forall\, self \in Acceptor,\ b \in Ballot :$
$Phase2av(self,\ b) \Rightarrow$
$\lor\ msgs' = msgs$
$\lor\ \exists\, v \in Value :$
$\land\ [type \mapsto$ "1c", $bal \mapsto b$, $val \mapsto v] \in msgs$
$\land\ msgs' = msgs \cup \{[type \mapsto$ "2a", $bal \mapsto b$, $val \mapsto v]\}$
$\land \quad \forall\, self \in Acceptor,\ b \in Ballot :$
$Phase2b(self,\ b) \Rightarrow$
$\exists\, v \in Value :$
$\land\ \exists\, Q \in ByzQuorum :$
$\forall\, a \in Q :$
$\exists\, m \in sentMsgs($"2av", $b) : \land\ m.val = v$
$\land\ m.acc = a$
$\land\ msgs' = msgs \cup$
$\{[type \mapsto$ "2b", $acc \mapsto self$, $bal \mapsto b$, $val \mapsto v]\}$
$\land\ bmsgs' = bmsgs \cup$
$\{[type \mapsto$ "2b", $acc \mapsto self$, $bal \mapsto b$, $val \mapsto v]\}$
$\land\ maxVVal' = [maxVVal$ EXCEPT $![self] = v]$
$\land \quad \forall\, self \in Acceptor,\ b \in Ballot :$
$LearnsSent(self,\ b) \Rightarrow$
$\exists\, S \in$ SUBSET $\{m \in msgsOfType($"1c"$) : m.bal = b\} :$
$msgs' = msgs \cup S$
$\land \quad \forall\, self \in Ballot :$
$Phase1a(self) \Rightarrow$
$msgs' = msgs \cup \{[type \mapsto$ "1a", $bal \mapsto self]\}$
$\land \quad \forall\, self \in Ballot :$
$Phase1c(self) \Rightarrow$
$\exists\, S \in$ SUBSET $[type : \{$"1c"$\},\ bal : \{self\},\ val : Value] :$
$\land\ \forall\, m \in S :$
$\exists\, a \in Acceptor : KnowsSafeAt(a,\ m.bal,\ m.val)$
$\land\ msgs' = msgs \cup S$
$\land \quad \forall\, self \in FakeAcceptor : FakingAcceptor(self) \Rightarrow msgs' = msgs$
$\langle 1\rangle$a. ASSUME NEW S, $bmsgs' = bmsgs \cup S$
PROVE $(\forall\, m \in S : m.type \neq$ "1a"$) \Rightarrow$
$(msgsOfType($"1a"$)' = msgsOfType($"1a"$))$
BY $\langle 1\rangle$a DEF $msgsOfType$
$\langle 1\rangle$b. ASSUME NEW S, $bmsgs' = bmsgs \cup S$
PROVE $(\forall\, m \in S : m.type \neq$ "1b"$) \Rightarrow (1bmsgs' = 1bmsgs)$
BY $\langle 1\rangle$b DEF $1bmsgs$, $1bRestrict$, $acceptorMsgsOfType$, $msgsOfType$
$\langle 1\rangle$c. ASSUME NEW S, $bmsgs' = bmsgs \cup S$
PROVE $(\forall\, m \in S : m.type \neq$ "1c"$) \land (knowsSent' = knowsSent)$
$\Rightarrow (1cmsgs' = 1cmsgs)$

⟨2⟩1. $(knowsSent' = knowsSent) \Rightarrow$
$\forall\, a \in Acceptor :$
$\forall\, b,\, v : KnowsSafeAt(a,\, b,\, v)' = KnowsSafeAt(a,\, b,\, v)$
BY DEF $KnowsSafeAt$
⟨2⟩2. $(knowsSent' = knowsSent) \Rightarrow$
$\forall\, b,\, v : (\exists\, a \in Acceptor : KnowsSafeAt(a,\, b,\, v)') \equiv$
$(\exists\, a \in Acceptor : KnowsSafeAt(a,\, b,\, v))$
BY ⟨2⟩1
⟨2⟩3. $(\forall\, m \in S : m.type \neq \text{“1c”}) \Rightarrow msgsOfType(\text{“1c”})' = msgsOfType(\text{“1c”})$
BY ⟨1⟩c, ⟨2⟩3 DEF $msgsOfType$
⟨2⟩4. QED
BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3 DEF $1cmsgs$, $KnowsSafeAt$
⟨1⟩d. ASSUME NEW S, $bmsgs' = bmsgs \cup S$
PROVE $(\forall\, m \in S : m.type \neq \text{“2av”}) \Rightarrow (2amsgs' = 2amsgs)$
⟨3⟩ SUFFICES ASSUME $\forall\, m \in S : m.type \neq \text{“2av”}$
PROVE $2amsgs' = 2amsgs$
OBVIOUS
⟨3⟩1. $acceptorMsgsOfType(\text{“2av”})' = acceptorMsgsOfType(\text{“2av”})$
BY ⟨1⟩d DEF $acceptorMsgsOfType$, $msgsOfType$
⟨3⟩2. QED
BY ⟨3⟩1 DEF $2amsgs$
⟨1⟩e. ASSUME NEW S, $bmsgs' = bmsgs \cup S$
PROVE $(\forall\, m \in S : m.type \neq \text{“2b”}) \Rightarrow$
$(acceptorMsgsOfType(\text{“2b”})' = acceptorMsgsOfType(\text{“2b”}))$
BY ⟨1⟩e DEF $acceptorMsgsOfType$, $msgsOfType$
⟨1⟩ HAVE $TypeOK$
⟨1⟩1. ASSUME NEW $self \in Acceptor$, NEW $b \in Ballot$
PROVE $Phase1b(self,\, b) \Rightarrow$
$msgs' = msgs \cup$
$\{[type \mapsto \text{“1b”},\, acc \mapsto self,\, bal \mapsto b,$
$mbal \mapsto maxVBal[self],\, mval \mapsto maxVVal[self]]\}$
⟨2⟩ DEFINE $m \triangleq [type \mapsto \text{“1b”},\, acc \mapsto self,\, bal \mapsto b,$
$m2av \mapsto 2avSent[self],$
$mbal \mapsto maxVBal[self],\, mval \mapsto maxVVal[self]]$
⟨2⟩ SUFFICES ASSUME $Phase1b(self,\, b)$
PROVE $msgs' = msgs \cup \{1bRestrict(m)\}$
BY DEF $1bRestrict$
⟨2⟩1. $bmsgs' = bmsgs \cup \{m\}$
⟨3⟩1. $[type \mapsto \text{“1b”},\, bal \mapsto b,\, acc \mapsto self,\, m2av \mapsto 2avSent[self],$
$mbal \mapsto maxVBal[self],\, mval \mapsto maxVVal[self]]$
$= m$
The provers seem to be really bad at equality of records, so I had to jump through hoops to get it to prove this trivial result.
⟨4⟩ DEFINE $a0 \triangleq \text{“1b”}$

$a1 \triangleq 2avSent[self]$
$a3 \triangleq maxVBal[self]$
$a4 \triangleq maxVVal[self]$
$a5 \triangleq b$
$a6 \triangleq self$
⟨4⟩ HIDE DEF $a0$, $a1$, $a3$, $a4$, $a5$, $a6$
⟨4⟩1. $[type \mapsto a0,\ bal \mapsto a5,\ acc \mapsto a6,$
$m2av \mapsto a1,\ mbal \mapsto a3,\ mval \mapsto a4]$
$=$
$[type \mapsto a0,\ acc \mapsto a6,\ bal \mapsto a5,$
$m2av \mapsto a1,\ mbal \mapsto a3,\ mval \mapsto a4]$
OBVIOUS It takes *Isabelle* $1 - 1/2$ minutes to prove this.
⟨4⟩ USE DEF $a0$, $a1$, $a3$, $a4$, $a5$, $a6$
⟨4⟩2. QED
BY ⟨4⟩1
⟨3⟩ HIDE DEF m
⟨3⟩2. QED
BY ⟨3⟩1 DEF $Phase1b$
⟨2⟩2. $\land\ m.type = \text{“1b”}$
$\land\ m.acc = self$
OBVIOUS
⟨2⟩ HIDE DEF m
⟨2⟩3. $acceptorMsgsOfType(\text{“1b”})' = acceptorMsgsOfType(\text{“1b”}) \cup \{m\}$
BY ⟨2⟩1, ⟨2⟩2 DEF $acceptorMsgsOfType$, $msgsOfType$
⟨2⟩4. $1bmsgs' = 1bmsgs \cup \{1bRestrict(m)\}$
BY ⟨2⟩3 DEF $1bmsgs$
⟨2⟩5. $knowsSent' = knowsSent$
BY DEF $Phase1b$
⟨2⟩6. $\land\ msgsOfType(\text{“1a”})' = msgsOfType(\text{“1a”})$
$\land\ 1cmsgs' = 1cmsgs$
$\land\ 2amsgs' = 2amsgs$
$\land\ acceptorMsgsOfType(\text{“2b”})' = acceptorMsgsOfType(\text{“2b”})$
⟨3⟩ DEFINE $S \triangleq \{m\}$
⟨3⟩ $\land\ bmsgs' = bmsgs \cup S$
$\land\ \forall\, mm \in S : mm.type = \text{“1b”}$
BY ⟨2⟩1, ⟨2⟩2
⟨3⟩ HIDE DEF S
⟨3⟩ QED
BY ⟨2⟩5, ⟨1⟩a, ⟨1⟩c, ⟨1⟩d, ⟨1⟩e
⟨2⟩7. QED
BY ⟨2⟩4, ⟨2⟩6 DEF $msgs$

⟨1⟩2. ASSUME NEW $self \in Acceptor$, NEW $b \in Ballot$
PROVE $Phase2av(self,\ b) \Rightarrow$
$\lor\ msgs' = msgs$

$\lor \exists\, v \in \mathit{Value} :$
$\land [\mathit{type} \mapsto \text{“1c”},\ \mathit{bal} \mapsto b,\ \mathit{val} \mapsto v] \in \mathit{msgs}$
$\land \mathit{msgs}' = \mathit{msgs} \cup$
$\{[\mathit{type} \mapsto \text{“2a”},\ \mathit{bal} \mapsto b,\ \mathit{val} \mapsto v]\}$

$\langle 2 \rangle$ HAVE $\mathit{Phase2av}(\mathit{self},\ b)$

$\langle 2 \rangle 1$. PICK $m \in \mathit{sentMsgs}(\text{“1c”},\ b) :$
$\land\ \mathit{KnowsSafeAt}(\mathit{self},\ b,\ m.\mathit{val})$
$\land\ \mathit{bmsgs}' = \mathit{bmsgs} \cup$
$\{[\mathit{type} \mapsto \text{“2av”},\ \mathit{bal} \mapsto b,\ \mathit{val} \mapsto m.\mathit{val},\ \mathit{acc} \mapsto \mathit{self}]\}$
BY DEF $\mathit{Phase2av}$

$\langle 2 \rangle 2$. $m = [\mathit{type} \mapsto \text{“1c”},\ \mathit{bal} \mapsto b,\ \mathit{val} \mapsto m.\mathit{val}]$
$\langle 3 \rangle 1$. $\land\ m \in \mathit{bmsgs}$
$\land\ m.\mathit{type} = \text{“1c”}$
$\land\ m.\mathit{bal} = b$
BY DEF $\mathit{sentMsgs}$
$\langle 3 \rangle 2$. $m \in \mathit{BMessage}$
BY $\langle 3 \rangle 1$ DEF TypeOK
$\langle 3 \rangle 3$. $m \in 1c\mathit{Message}$
BY $\langle 3 \rangle 1$, $\langle 3 \rangle 2$, $\mathit{BMessageLemma}$
$\langle 3 \rangle 4$. $m = [\mathit{type} \mapsto m.\mathit{type},\ \mathit{bal} \mapsto m.\mathit{bal},\ \mathit{val} \mapsto m.\mathit{val}]$
BY $\langle 3 \rangle 3$ DEF $1c\mathit{Message}$
$\langle 3 \rangle 5$. QED
BY $\langle 3 \rangle 1$, $\langle 3 \rangle 4$

$\langle 2 \rangle$ DEFINE $\mathit{ma} \triangleq [\mathit{type} \mapsto \text{“2a”},\ \mathit{bal} \mapsto b,\ \mathit{val} \mapsto m.\mathit{val}]$
$\langle 2 \rangle 3$. SUFFICES ASSUME $\mathit{msgs}' \neq \mathit{msgs}$
PROVE $\land\ m \in \mathit{msgs}$
$\land\ \mathit{msgs}' = \mathit{msgs} \cup \{\mathit{ma}\}$
$\langle 3 \rangle 1$. $m.\mathit{val} \in \mathit{Value}$
$\langle 4 \rangle 1$. $\land\ m \in \mathit{bmsgs}$
$\land\ m.\mathit{type} = \text{“1c”}$
BY DEF $\mathit{sentMsgs}$
$\langle 4 \rangle 2$. $m \in \mathit{BMessage}$
BY $\langle 4 \rangle 1$ DEF TypeOK
$\langle 4 \rangle 3$. $m \in 1c\mathit{Message}$
BY $\langle 4 \rangle 1$, $\langle 4 \rangle 2$, $\mathit{BMessageLemma}$
$\langle 4 \rangle 4$. QED
BY $\langle 4 \rangle 3$ DEF $1c\mathit{Message}$
$\langle 3 \rangle 2$. QED
BY $\langle 3 \rangle 1$, $\langle 2 \rangle 2$

$\langle 2 \rangle 4$. $m \in \mathit{msgs}$
$\langle 3 \rangle 1$. $\land\ m \in \mathit{bmsgs}$
$\land\ m.\mathit{type} = \text{“1c”}$
$\land\ m.\mathit{bal} = b$

BY DEF $sentMsgs$

$\langle 3 \rangle 2.\ m \in msgsOfType(\text{“1c”})$

BY $\langle 3 \rangle 1$ DEF $msgsOfType$

$\langle 3 \rangle 3.\ KnowsSafeAt(self,\ m.bal,\ m.val)$

BY $\langle 2 \rangle 1,\ \langle 3 \rangle 1$

$\langle 3 \rangle 4.\ \exists\, a \in Acceptor : KnowsSafeAt(a,\ m.bal,\ m.val)$

BY $\langle 3 \rangle 3$

$\langle 3 \rangle 5.\ m \in 1cmsgs$

BY $\langle 2 \rangle 2,\ \langle 3 \rangle 1,\ \langle 3 \rangle 2,\ \langle 3 \rangle 4$ DEF $1cmsgs$

$\langle 3 \rangle 6.$ QED

BY $\langle 3 \rangle 5$ DEF $msgs$

$\langle 2 \rangle 5.\ msgs' = msgs \cup \{ma\}$

$\langle 3 \rangle$ DEFINE $mb \triangleq [type \mapsto \text{“2av”},\ bal \mapsto b,\ val \mapsto m.val,\ acc \mapsto self]$

$S \triangleq \{mb\}$

$\langle 3 \rangle 1.\ \land\ bmsgs' = bmsgs \cup S$

$\land\ \forall\, mm \in S : mm.type = \text{“2av”}$

BY $\langle 2 \rangle 1$

$\langle 3 \rangle 2.\ knowsSent' = knowsSent$

BY DEF $Phase2av$

$\langle 3 \rangle 3.\ \land\ msgsOfType(\text{“1a”})' = msgsOfType(\text{“1a”})$

$\land\ 1bmsgs' = 1bmsgs$

$\land\ 1cmsgs' = 1cmsgs$

$\land\ acceptorMsgsOfType(\text{“2b”})' = acceptorMsgsOfType(\text{“2b”})$

$\langle 4 \rangle$ HIDE DEF S

$\langle 4 \rangle$ QED

BY $\langle 3 \rangle 1,\ \langle 3 \rangle 2,\ \langle 1 \rangle$a, $\langle 1 \rangle$b, $\langle 1 \rangle$c, $\langle 1 \rangle$e

$\langle 3 \rangle 4.\ 2amsgs \subseteq 2amsgs'$

$\langle 4 \rangle 1.$ SUFFICES ASSUME NEW $m1 \in [type : \{\text{“2a”}\},\ bal : Ballot,\ val : Value]$,

$\exists\, Q \in Quorum :$

$\forall\, a \in Q :$

$\exists\, m2av \in acceptorMsgsOfType(\text{“2av”}) :$

$\land\ m2av.acc = a$

$\land\ m2av.bal = m1.bal$

$\land\ m2av.val = m1.val$

PROVE $\exists\, Q \in Quorum :$

$\forall\, a \in Q :$

$\exists\, m2av \in acceptorMsgsOfType(\text{“2av”})' :$

$\land\ m2av.acc = a$

$\land\ m2av.bal = m1.bal$

$\land\ m2av.val = m1.val$

BY DEF $2amsgs$

$\langle 4 \rangle 2.$ PICK $Q \in Quorum :$

$\forall\, a \in Q\ :$

$\exists\, m2av \in acceptorMsgsOfType(\text{“2av”}) :$

$\land\ m2av.acc = a$

$\land\ m2av.bal = m1.bal$
$\land\ m2av.val = m1.val$
BY $\langle 4\rangle 1$
$\langle 4\rangle$ SUFFICES ASSUME NEW $a \in Q$
PROVE $\exists\, m2av \in acceptorMsgsOfType(\text{“2av”})' :$
$\land\ m2av.acc = a$
$\land\ m2av.bal = m1.bal$
$\land\ m2av.val = m1.val$
OBVIOUS
$\langle 4\rangle 3.\ \exists\, m2av \in acceptorMsgsOfType(\text{“2av”}) :$
$\land\ m2av.acc = a$
$\land\ m2av.bal = m1.bal$
$\land\ m2av.val = m1.val$
BY $\langle 4\rangle 2$
$\langle 4\rangle 4.\ acceptorMsgsOfType(\text{“2av”}) \subseteq acceptorMsgsOfType(\text{“2av”})'$
$\langle 5\rangle 1.\ msgsOfType(\text{“2av”}) \subseteq msgsOfType(\text{“2av”})'$
BY $\langle 3\rangle 1$ DEF $msgsOfType$
$\langle 5\rangle 2.$ QED
BY $\langle 5\rangle 1$ DEF $acceptorMsgsOfType$
$\langle 4\rangle 5.$ QED
BY $\langle 4\rangle 3, \langle 4\rangle 4$
$\langle 3\rangle 5.\ msgs \subseteq msgs'$
BY $\langle 3\rangle 3, \langle 3\rangle 4$ DEF $msgs$
$\langle 3\rangle 6.$ ASSUME NEW $mp \in msgs'$, $mp \notin msgs$
PROVE $mp = ma$
$\langle 4\rangle 1.\ mp \in 2amsgs' \land mp \notin 2amsgs$
BY $\langle 3\rangle 3, \langle 3\rangle 6$ DEF $msgs$
$\langle 4\rangle 2.$ PICK $Q \in Quorum :$
$\forall\, a \in Q :$
$\exists\, m2av \in acceptorMsgsOfType(\text{“2av”})' :$
$\land\ m2av.acc = a$
$\land\ m2av.bal = mp.bal$
$\land\ m2av.val = mp.val$
BY $\langle 4\rangle 1$ DEF $2amsgs$
$\langle 4\rangle 3.\ \neg\forall\, a \in Q :$
$\exists\, m2av \in acceptorMsgsOfType(\text{“2av”}) :$
$\land\ m2av.acc = a$
$\land\ m2av.bal = mp.bal$
$\land\ m2av.val = mp.val$
BY $\langle 4\rangle 1$ DEF $2amsgs$
$\langle 4\rangle 4.$ PICK $a \in Q :$
$\neg\exists\, m2av \in acceptorMsgsOfType(\text{“2av”}) :$
$\land\ m2av.acc = a$
$\land\ m2av.bal = mp.bal$
$\land\ m2av.val = mp.val$

BY $\langle 4\rangle 3$
$\langle 4\rangle 5$. PICK $m2av \in acceptorMsgsOfType(\text{“2av”})' :$
$\wedge\ m2av.acc = a$
$\wedge\ m2av.bal = mp.bal$
$\wedge\ m2av.val = mp.val$
BY $\langle 4\rangle 2$
$\langle 4\rangle 6$. $m2av \notin acceptorMsgsOfType(\text{“2av”})$
BY $\langle 4\rangle 5$, $\langle 4\rangle 4$
$\langle 4\rangle 7$. $m2av = mb$
$\langle 5\rangle 1$. $acceptorMsgsOfType(\text{“2av”})' = acceptorMsgsOfType(\text{“2av”}) \cup \{mb\}$
BY $\langle 3\rangle 1$, $mb.type = \text{“2av”}$ DEF $acceptorMsgsOfType$, $msgsOfType$
$\langle 5\rangle 2$. QED
BY $\langle 4\rangle 6$, $\langle 5\rangle 1$
$\langle 4\rangle 8$. $mp = [type \mapsto \text{“2a”},\ bal \mapsto mp.bal,\ val \mapsto mp.val]$
$\langle 5\rangle 1$. $mp \in [type : \{\text{“2a”}\},\ bal : Ballot,\ val : Value]$
BY $\langle 4\rangle 1$ DEF $2amsgs$
$\langle 5\rangle 2$. QED
BY $\langle 5\rangle 1$
$\langle 4\rangle 9$. QED
BY $\langle 4\rangle 8$, $\langle 4\rangle 7$, $\langle 4\rangle 5$
$\langle 3\rangle 7$. QED
BY $\langle 2\rangle 3$, $\langle 3\rangle 5$, $\langle 3\rangle 6$
$\langle 2\rangle 6$. QED
BY $\langle 2\rangle 4$, $\langle 2\rangle 5$

$\langle 1\rangle 3$. ASSUME NEW $self \in Acceptor$, NEW $b \in Ballot$
PROVE $Phase2b(self,\ b) \Rightarrow$
$\exists\, v \in Value :$
$\wedge\ \exists\, Q \in ByzQuorum :$
$\forall\, a \in Q :$
$\exists\, m \in sentMsgs(\text{“2av”},\ b) :\ \wedge\ m.val = v$
$\wedge\ m.acc = a$
$\wedge\ msgs' = msgs \cup$
$\{[type \mapsto \text{“2b”},\ acc \mapsto self,\ bal \mapsto b,$
$val \mapsto v]$
$\}$
$\wedge\ bmsgs' = bmsgs \cup$
$\{[type \mapsto \text{“2b”},\ acc \mapsto self,\ bal \mapsto b,\ val \mapsto v]\}$
$\wedge\ maxVVal' = [maxVVal$ EXCEPT $![self] = v]$
$\langle 2\rangle$ HAVE $Phase2b(self,\ b)$
$\langle 2\rangle 1$. PICK $v \in Value :$
$\wedge\ \exists\, Q \in ByzQuorum :$
$\forall\, a \in Q :$
$\exists\, m \in sentMsgs(\text{“2av”},\ b) :\ \wedge\ m.val = v$
$\wedge\ m.acc = a$

$\land\ bmsgs' =$
$bmsgs \cup$
$\{[type \mapsto \text{“2b”},\ acc \mapsto self,\ bal \mapsto b,\ val \mapsto v]\}$
$\land\ maxVVal' = [maxVVal \text{ EXCEPT } ![self] = v]$
BY DEF $Phase2b$
$\langle 2\rangle$ DEFINE $bm \triangleq [type \mapsto \text{“2b”},\ acc \mapsto self,\ bal \mapsto b,\ val \mapsto v]$
$\langle 2\rangle 2.\ \land\ msgsOfType(\text{“1a”})' = msgsOfType(\text{“1a”})$
$\land\ 1bmsgs' = 1bmsgs$
$\land\ 1cmsgs' = 1cmsgs$
$\land\ 2amsgs' = 2amsgs$
$\langle 3\rangle$ DEFINE $S \triangleq \{bm\}$
$\langle 3\rangle\ \land\ bmsgs' = bmsgs \cup S$
$\land\ \forall\, m \in S : m.type = \text{“2b”}$
BY $\langle 2\rangle 1$, $bm.type = \text{“2b”}$
$\langle 3\rangle$ HIDE DEF S
$\langle 3\rangle$ QED
BY $\langle 1\rangle$a, $\langle 1\rangle$b, $knowsSent' = knowsSent$, $\langle 1\rangle$c, $\langle 1\rangle$d DEF $Phase2b$
$\langle 2\rangle 3.\ acceptorMsgsOfType(\text{“2b”})' = acceptorMsgsOfType(\text{“2b”}) \cup \{bm\}$
$\langle 3\rangle 0.\ acceptorMsgsOfType(\text{“2b”}) \subseteq acceptorMsgsOfType(\text{“2b”})'$
BY $\langle 2\rangle 1$ DEF $acceptorMsgsOfType$, $msgsOfType$
$\langle 3\rangle 1.\ bm \in acceptorMsgsOfType(\text{“2b”})'$
BY $\langle 2\rangle 1$ DEF $acceptorMsgsOfType$, $msgsOfType$
$\langle 3\rangle 3$. ASSUME NEW $m \in acceptorMsgsOfType(\text{“2b”})'$, $m \neq bm$
PROVE $m \in acceptorMsgsOfType(\text{“2b”})$
$\langle 4\rangle 1.\ \land\ m \in bmsgs'$
$\land\ m.type = \text{“2b”}$
$\land\ m.acc \in Acceptor$
BY DEF $acceptorMsgsOfType$, $msgsOfType$
$\langle 4\rangle$a. $bmsgs' = bmsgs \cup \{bm\}$
BY $\langle 2\rangle 1$
$\langle 4\rangle 2.\ m \in bmsgs$
BY $\langle 4\rangle 1$, $\langle 2\rangle 1$, $bmsgs' = bmsgs \cup \{bm\}$, $\langle 3\rangle 3$
$\langle 4\rangle 3$. QED
BY $\langle 4\rangle 1$, $\langle 4\rangle 2$ DEF $acceptorMsgsOfType$, $msgsOfType$
$\langle 3\rangle 5$. QED
BY $\langle 3\rangle 0$, $\langle 3\rangle 1$, $\langle 3\rangle 3$

$\langle 2\rangle 4.\ msgs' = msgs \cup \{bm\}$
BY $\langle 2\rangle 2$, $\langle 2\rangle 3$ DEF $msgs$
$\langle 2\rangle 5$. QED
BY $\langle 2\rangle 1$, $\langle 2\rangle 4$, $\langle 2\rangle 5$

$\langle 1\rangle 4$. ASSUME NEW $self \in Acceptor$, NEW $b \in Ballot$
PROVE $LearnsSent(self,\ b) \Rightarrow$
$\exists\, S \in \text{SUBSET } \{m \in msgsOfType(\text{“1c”}) : m.bal = b\} :$

$msgs' = msgs \cup S$

$\langle 2 \rangle$ HAVE $LearnsSent(self,\ b)$

$\langle 2 \rangle 1. \land msgsOfType(\text{“1a”})' = msgsOfType(\text{“1a”})$
$\land\ 1bmsgs' = 1bmsgs$
$\land\ 2amsgs' = 2amsgs$
$\land\ acceptorMsgsOfType(\text{“2b”})' = acceptorMsgsOfType(\text{“2b”})$

$\langle 3 \rangle 1. \land bmsgs' = bmsgs \cup \{\}$
$\land\ \forall\, m \in \{\} : m.type = \text{“x”}$
BY DEF $LearnsSent$

$\langle 3 \rangle 2. \langle 2 \rangle 1!1$
BY $\langle 3 \rangle 1$, $\langle 1 \rangle$a

$\langle 3 \rangle 3. \langle 2 \rangle 1!2$
BY $\langle 3 \rangle 1$, $\langle 1 \rangle$b

$\langle 3 \rangle 4. \langle 2 \rangle 1!3$
BY $\langle 3 \rangle 1$, $\langle 1 \rangle$d

$\langle 3 \rangle 5. \langle 2 \rangle 1!4$
BY $\langle 3 \rangle 1$, $\langle 1 \rangle$e

$\langle 3 \rangle 6.$ QED
BY $\langle 3 \rangle 2$, $\langle 3 \rangle 3$, $\langle 3 \rangle 4$, $\langle 3 \rangle 5$

$\langle 2 \rangle 2. \land 1cmsgs \subseteq 1cmsgs'$
$\land\ \forall\, m \in 1cmsgs' \setminus 1cmsgs :$
$m \in msgsOfType(\text{“1c”}) \land m.bal = b$

$\langle 3 \rangle 1.\ bmsgs' = bmsgs$
BY DEF $LearnsSent$

$\langle 3 \rangle 2.$ PICK $S \in$ SUBSET $sentMsgs(\text{“1b”},\ b) :$
$knowsSent' = [knowsSent \text{ EXCEPT } ![self] = knowsSent[self] \cup S]$
BY DEF $LearnsSent$

$\langle 3 \rangle 3.$ ASSUME NEW $m \in 1cmsgs$
PROVE $m \in 1cmsgs'$

$\langle 4 \rangle 1. \forall\, a \in Acceptor : knowsSent[a] \subseteq knowsSent'[a]$
BY $\langle 3 \rangle 2$ DEF $TypeOK$

$\langle 4 \rangle 2. \forall\, a \in Acceptor :$
$\forall\, bb,\ v : KnowsSafeAt(a,\ bb,\ v) \Rightarrow KnowsSafeAt(a,\ bb,\ v)'$

$\langle 5 \rangle$ SUFFICES ASSUME NEW $a \in Acceptor$, NEW bb, NEW v,
$KnowsSafeAt(a,\ bb,\ v)$
PROVE $KnowsSafeAt(a,\ bb,\ v)'$
OBVIOUS

$\langle 5 \rangle 1. \forall\, T,\ U : (T \subseteq U) \Rightarrow \land\ KS1(T) \Rightarrow KS1(U)$
$\land\ KS2(v,\ bb,\ T) \Rightarrow KS2(v,\ bb,\ U)$

$\langle 6 \rangle$ SUFFICES ASSUME NEW T, NEW U, $T \subseteq U$
PROVE $\langle 5 \rangle 1!(T,\ U)!2$
OBVIOUS

$\langle 6 \rangle 1. \langle 5 \rangle 1!(T,\ U)!2!1$

$\langle 7 \rangle$ SUFFICES ASSUME NEW BQ
PROVE $KS11(BQ,\ T) \Rightarrow KS11(BQ,\ U)$

BY DEF $KS1$

$\langle 7 \rangle$ SUFFICES ASSUME NEW ac
PROVE $KS11a(ac,\ T) \Rightarrow KS11a(ac,\ U)$

BY DEF $KS11$

$\langle 7 \rangle$ QED

BY DEF $KS11a$

$\langle 6 \rangle 2.\langle 5 \rangle 1!(T,\ U)!2!2$

$\langle 7 \rangle 1$ ASSUME NEW c
PROVE $KS21(v,\ c,\ T) \Rightarrow KS21(v,\ c,\ U)$

$\langle 8 \rangle$ SUFFICES ASSUME NEW BQ
PROVE $KS21BQ(v,\ c,\ BQ,\ T) \Rightarrow KS21BQ(v,\ c,\ BQ,\ U)$

BY DEF $KS21$

$\langle 8 \rangle$ SUFFICES ASSUME NEW ac
PROVE $KS21BQa(v,\ c,\ ac,\ T) \Rightarrow KS21BQa(v,\ c,\ ac,\ U)$

BY DEF $KS21BQ$

$\langle 8 \rangle$ QED

BY DEF $KS21BQa$

$\langle 7 \rangle 2$ ASSUME NEW c
PROVE $KS22(v,\ c,\ T) \Rightarrow KS22(v,\ c,\ U)$

$\langle 8 \rangle$ SUFFICES ASSUME NEW WQ
PROVE $KS22\,WQ(v,\ c,\ WQ,\ T) \Rightarrow KS22\,WQ(v,\ c,\ WQ,\ U)$

BY DEF $KS22$

$\langle 8 \rangle$ SUFFICES ASSUME NEW ac
PROVE $KS22\,WQa(v,\ c,\ ac,\ T) \Rightarrow KS22\,WQa(v,\ c,\ ac,\ U)$

BY DEF $KS22\,WQ$

$\langle 8 \rangle$ QED

BY DEF $KS22\,WQa$

$\langle 7 \rangle 3$. QED

BY $\langle 7 \rangle 1$, $\langle 7 \rangle 2$ DEF $KS2$

$\langle 6 \rangle 3$. QED

BY $\langle 6 \rangle 1$, $\langle 6 \rangle 2$

$\langle 5 \rangle 2$. $KSet(a,\ bb) \subseteq KSet(a,\ bb)'$

BY $\langle 4 \rangle 1$ DEF $KSet$

$\langle 5 \rangle 3$. QED

BY $\langle 5 \rangle 1$, $\langle 5 \rangle 2$, *KnowsSafeAtDef*

$\langle 4 \rangle 3$. $msgsOfType(\text{“1c”})' = msgsOfType(\text{“1c”})$

BY DEF *msgsOfType*, *LearnsSent*

$\langle 4 \rangle 4$. QED

BY $\langle 4 \rangle 2$, $\langle 4 \rangle 3$ DEF $1cmsgs$

$\langle 3 \rangle 4$. ASSUME NEW $m \in 1cmsgs'$, $m \notin 1cmsgs$
PROVE $m \in msgsOfType(\text{“1c”}) \land m.bal = b$

$\langle 4 \rangle 1$. $m \in msgsOfType(\text{“1c”})$

BY DEF $1cmsgs$, *msgsOfType*, *LearnsSent*

$\langle 4 \rangle 2$. PICK $a \in Acceptor : KnowsSafeAt(a,\ m.bal,\ m.val)'$

BY DEF $1cmsgs$

⟨4⟩3. $\neg KnowsSafeAt(a,\ m.bal,\ m.val)$
BY ⟨3⟩4, ⟨4⟩1 DEF $1cmsgs$
⟨4⟩4. $\forall\, aa \in Acceptor,\ bb \in Ballot :$
$\forall\, mm \in KSet(aa,\ bb)' :$
$mm \notin KSet(aa,\ bb) \Rightarrow bb = b$
⟨5⟩1. SUFFICES ASSUME NEW $aa \in Acceptor$,
NEW $bb \in Ballot$,
NEW $mm \in KSet(aa,\ bb)'$,
$mm \notin KSet(aa,\ bb)$
PROVE $bb = b$
OBVIOUS
⟨5⟩2. ASSUME NEW $m1 \in knowsSent[aa]'$, $m1 \notin knowsSent[aa]$
PROVE $m1.bal = b$
⟨6⟩1. $knowsSent' = [knowsSent$ EXCEPT $![self] = knowsSent[self] \cup S]$
BY DEF $LearnsSent$
⟨6⟩2.CASE $aa \neq self$
BY ⟨6⟩2, ⟨5⟩2 DEF $TypeOK$, $LearnsSent$
⟨6⟩3.CASE $aa = self$
BY ⟨6⟩1, ⟨6⟩3, ⟨5⟩2 DEF $TypeOK$, $sentMsgs$
⟨6⟩4. QED
BY ⟨6⟩2, ⟨6⟩3
⟨5⟩3. QED
BY ⟨5⟩1, ⟨5⟩2 DEF $KSet$
⟨4⟩5. $m.bal \in Ballot$
BY ⟨4⟩1, $BMessageLemma$ DEF $1cMessage$, $msgsOfType$, $TypeOK$
⟨4⟩7.CASE $KS1(KSet(a,\ m.bal)') \wedge \neg KS1(KSet(a,\ m.bal))$
⟨5⟩1. PICK $BQ \in ByzQuorum : KS11(BQ,\ KSet(a,\ m.bal)')$
BY ⟨4⟩7 DEF $KS1$
⟨5⟩2. $\neg KS11(BQ,\ KSet(a,\ m.bal))$
BY ⟨4⟩7 DEF $KS1$
⟨5⟩3. PICK $aa \in BQ : \neg KS11a(aa,\ KSet(a,\ m.bal))$
BY ⟨5⟩2 DEF $KS11$
⟨5⟩4. $KS11a(aa,\ KSet(a,\ m.bal)')$
BY ⟨5⟩1 DEF $KS11$
⟨5⟩5. PICK $mm \in KSet(a,\ m.bal)' : mm.acc = aa \wedge mm.mbal = -1$
BY ⟨5⟩4 DEF $KS11a$
⟨5⟩6. $mm \notin KSet(a,\ m.bal)$
BY ⟨5⟩5, ⟨5⟩3 DEF $KS11a$
⟨5⟩7. $m.bal = b$
BY ⟨4⟩4, ⟨4⟩5, ⟨5⟩5, ⟨5⟩6
⟨5⟩8. QED
BY ⟨4⟩1, ⟨5⟩7
⟨4⟩8.CASE $KS2(m.val,\ m.bal,\ KSet(a,\ m.bal)') \wedge \neg KS2(m.val,\ m.bal,\ KSet(a,\ m.bal))$
⟨5⟩1. PICK $c \in 0\,..\,(m.bal - 1) : \wedge KS21(m.val,\ c,\ KSet(a,\ m.bal))'$
$\wedge KS22(m.val,\ c,\ KSet(a,\ m.bal))'$

BY $\langle 4\rangle 8$ DEF $KS2$

$\langle 5\rangle 2$. CASE $\neg KS21(m.val,\ c,\ KSet(a,\ m.bal))$

$\langle 6\rangle 1$. PICK $BQ \in ByzQuorum : KS21BQ(m.val,\ c,\ BQ,\ KSet(a,\ m.bal))'$

BY $\langle 5\rangle 1$ DEF $KS21$

$\langle 6\rangle 2$. PICK $aa \in BQ$:

$$\neg\exists\, mm \in KSet(a,\ m.bal) : \begin{array}{l} \wedge\ mm.acc = aa \\ \wedge\ mm.mbal \leq c \\ \wedge\ (mm.mbal = c) \Rightarrow (mm.mval = m.val) \end{array}$$

BY $\langle 5\rangle 2$ DEF $KS21$, $KS21BQ$, $KS21BQa$

$\langle 6\rangle 3$. PICK $mm \in KSet(a,\ m.bal)'$: $\begin{array}{l} \wedge\ mm.acc = aa \\ \wedge\ mm.mbal \leq c \\ \wedge\ (mm.mbal = c) \Rightarrow (mm.mval = m.val) \end{array}$

BY $\langle 6\rangle 1$ DEF $KS21BQ$, $KS21BQa$

$\langle 6\rangle 4$. $mm \notin KSet(a,\ m.bal)$

BY $\langle 6\rangle 2$, $\langle 6\rangle 3$

$\langle 6\rangle 5$. $mm.bal = m.bal$

BY DEF $KSet$

$\langle 6\rangle 6$. QED

BY $\langle 6\rangle 4$, $\langle 6\rangle 5$, $\langle 4\rangle 4$, $\langle 4\rangle 5$, $\langle 4\rangle 1$

$\langle 5\rangle 3$. CASE $\neg KS22(m.val,\ c,\ KSet(a,\ m.bal))$

$\langle 6\rangle 1$. PICK $WQ \in WeakQuorum : KS22WQ(m.val,\ c,\ WQ,\ KSet(a,\ m.bal))'$

BY $\langle 5\rangle 1$ DEF $KS22$

$\langle 6\rangle 2$. PICK $aa \in WQ$:

$$\neg\exists\, mm \in KSet(a,\ m.bal) : \begin{array}{l} \wedge\ mm.acc = aa \\ \wedge\ \exists\, r \in mm.m2av : \\ \quad \wedge\ r.bal \geq c \\ \quad \wedge\ r.val = m.val \end{array}$$

$\langle 7\rangle 1$. $\neg(\forall\, aa \in WQ :$

$$\exists\, m\_1 \in KSet(a,\ m.bal) : \begin{array}{l} \wedge\ m\_1.acc = aa \\ \wedge\ \exists\, r \in m\_1.m2av : \begin{array}{l} \wedge\ r.bal \geq c \\ \wedge\ r.val = m.val) \end{array} \end{array}$$

BY $\langle 5\rangle 3$ DEF $KS22$, $KS22WQ$, $KS22WQa$

$\langle 7\rangle 2$. QED

BY $\langle 7\rangle 1$

$\langle 6\rangle 3$. PICK $mm \in KSet(a,\ m.bal)'$: $\begin{array}{l} \wedge\ mm.acc = aa \\ \wedge\ \exists\, r \in mm.m2av : \\ \quad \wedge\ r.bal \geq c \\ \quad \wedge\ r.val = m.val \end{array}$

BY $\langle 6\rangle 1$ DEF $KS22WQ$, $KS22WQa$

$\langle 6\rangle 4$. $mm \notin KSet(a,\ m.bal)$

BY $\langle 6\rangle 2$, $\langle 6\rangle 3$

$\langle 6\rangle 5$. $mm.bal = m.bal$

BY DEF $KSet$

$\langle 6\rangle 6$. QED

BY $\langle 6\rangle 4$, $\langle 6\rangle 5$, $\langle 4\rangle 4$, $\langle 4\rangle 5$, $\langle 4\rangle 1$

$\langle 5\rangle 4$. QED

BY $\langle 5\rangle 2$, $\langle 5\rangle 3$, $\langle 4\rangle 8$ DEF $KS2$

$\langle 4\rangle$ QED

$\langle 5\rangle 1$. $KS1(KSet(a,\ m.bal)') \lor KS2(m.val,\ m.bal,\ KSet(a,\ m.bal)')$

$\langle 6\rangle$ $KnowsSafeAt(a,\ m.bal,\ m.val)' = (KS1(KSet(a,\ m.bal)') \lor KS2(m.val,\ m.bal,\ KSet(a,\ m.bal)'))$

BY $KnowsSafeAtDef$

$\langle 6\rangle$ QED

BY $\langle 4\rangle 2$

$\langle 5\rangle 2$. $(\neg KS1(KSet(a,\ m.bal))) \land (\neg KS2(m.val,\ m.bal,\ KSet(a,\ m.bal)))$

BY $\langle 4\rangle 3$, $KnowsSafeAtDef$

$\langle 5\rangle 3$. QED

BY $\langle 4\rangle 7$, $\langle 4\rangle 8$, $\langle 5\rangle 1$, $\langle 5\rangle 2$

$\langle 3\rangle 5$. QED

BY $\langle 3\rangle 3$, $\langle 3\rangle 4$

$\langle 2\rangle 3$. QED

$\langle 3\rangle$ DEFINE $S \triangleq 1cmsgs' \setminus 1cmsgs$

$\langle 3\rangle 1$. $1cmsgs' = 1cmsgs \cup S$

BY $\langle 2\rangle 2$

$\langle 3\rangle 2$. $S \in$ SUBSET $\{m \in msgsOfType(\text{“1c”}) : m.bal = b\}$

BY $\langle 2\rangle 2$

$\langle 3\rangle$ HIDE DEF S

$\langle 3\rangle 3$. $msgs' = msgs \cup S$

BY $\langle 3\rangle 1$, $\langle 2\rangle 1$ DEF $msgs$

$\langle 3\rangle 4$. QED

BY $\langle 3\rangle 3$, $\langle 3\rangle 2$

$\langle 1\rangle 5$. ASSUME NEW $self \in Ballot$

PROVE $Phase1a(self) \Rightarrow$

$msgs' = msgs \cup \{[type \mapsto \text{“1a”},\ bal \mapsto self]\}$

$\langle 2\rangle$ DEFINE $m \triangleq [type \mapsto \text{“1a”},\ bal \mapsto self]$

$\langle 2\rangle$ SUFFICES ASSUME $Phase1a(self)$

PROVE $msgs' = msgs \cup \{m\}$

BY $\langle 1\rangle 5$

$\langle 2\rangle 1$. $\land\ bmsgs' = bmsgs \cup \{m\}$

$\land\ m.type = \text{“1a”}$

BY DEF $Phase1a$

$\langle 2\rangle 2$. $\land\ 1bmsgs' = 1bmsgs$

$\land\ 1cmsgs' = 1cmsgs$

$\land\ 2amsgs' = 2amsgs$

$\land\ acceptorMsgsOfType(\text{“2b”})' = acceptorMsgsOfType(\text{“2b”})$

$\langle 3\rangle$ DEFINE $S \triangleq \{m\}$

$\langle 3\rangle$ $\land\ bmsgs' = bmsgs \cup S$

$\land\ \forall\, mm \in S : mm.type = \text{“1a”}$

$\land\ knowsSent' = knowsSent$

BY $\langle 2\rangle 1$ DEF $Phase1a$

⟨3⟩ HIDE DEF S
⟨3⟩ QED
BY ⟨1⟩b, ⟨1⟩c, ⟨1⟩d, ⟨1⟩e
⟨2⟩3. $msgsOfType(\text{“1a”})' = msgsOfType(\text{“1a”}) \cup \{m\}$
BY ⟨2⟩1 DEF $msgsOfType$
⟨2⟩4. QED
BY ⟨2⟩2, ⟨2⟩3 DEF $msgs$
⟨1⟩6. ASSUME NEW $self \in Ballot$
PROVE $Phase1c(self) \Rightarrow$
$\exists\, S \in \text{SUBSET } [type : \{\text{“1c”}\},\ bal : \{self\},\ val : Value] :$
$\land \forall\, m \in S :$
$\exists\, a \in Acceptor : KnowsSafeAt(a,\ m.bal,\ m.val)$
$\land\ msgs' = msgs \cup S$
⟨2⟩1. SUFFICES ASSUME NEW $S \in \text{SUBSET } [type : \{\text{“1c”}\},\ bal : \{self\},\ val : Value]$,
$bmsgs' = (bmsgs \cup S)$,
$knowsSent' = knowsSent$
PROVE ⟨1⟩6!2!2
BY DEF $Phase1c$
⟨2⟩ DEFINE $SS \triangleq \{m \in S : \exists\, a \in Acceptor : KnowsSafeAt(a,\ m.bal,\ m.val)\}$
⟨2⟩ SUFFICES $msgs' = msgs \cup SS$
BY ⟨2⟩1
⟨2⟩2. $\forall\, m \in S : m.type = \text{“1c”}$
BY ⟨2⟩1
⟨2⟩3. $\land\ msgsOfType(\text{“1a”})' = msgsOfType(\text{“1a”})$
$\land\ 1bmsgs' = 1bmsgs$
$\land\ 2amsgs' = 2amsgs$
$\land\ acceptorMsgsOfType(\text{“2b”})' = acceptorMsgsOfType(\text{“2b”})$
BY ⟨2⟩1, ⟨2⟩2, ⟨1⟩a, ⟨1⟩b, ⟨1⟩d, ⟨1⟩e
⟨2⟩4. $1cmsgs' = 1cmsgs \cup SS$
⟨3⟩1. $msgsOfType(\text{“1c”})' = msgsOfType(\text{“1c”}) \cup S$
BY ⟨2⟩1 DEF $msgsOfType$
⟨3⟩2. $1cmsgs' =$
$\{m \in msgsOfType(\text{“1c”}) :$
$\exists\, a \in Acceptor : KnowsSafeAt(a,\ m.bal,\ m.val)'\}$
$\cup$
$\{m \in S : \exists\, a \in Acceptor : KnowsSafeAt(a,\ m.bal,\ m.val)'\}$
BY ⟨3⟩1 DEF $1cmsgs$
⟨3⟩3. $\forall\, m : \forall\, a \in Acceptor :$
$KnowsSafeAt(a,\ m.bal,\ m.val)' = KnowsSafeAt(a,\ m.bal,\ m.val)$
BY ⟨2⟩1 DEF $KnowsSafeAt$
⟨3⟩4. $1cmsgs' =$
$\{m \in msgsOfType(\text{“1c”}) :$
$\exists\, a \in Acceptor : KnowsSafeAt(a,\ m.bal,\ m.val)\}$
$\cup$
$\{m \in S : \exists\, a \in Acceptor : KnowsSafeAt(a,\ m.bal,\ m.val)\}$

BY $\langle 2\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$

$\langle 3\rangle 5$. QED

BY $\langle 3\rangle 4$ DEF $1cmsgs$

$\langle 2\rangle 5$. QED

BY $\langle 2\rangle 3$, $\langle 2\rangle 4$ DEF $msgs$

$\langle 1\rangle 7$. ASSUME NEW $self \in FakeAcceptor$

PROVE $FakingAcceptor(self) \Rightarrow msgs' = msgs$

$\langle 2\rangle$ SUFFICES ASSUME $FakingAcceptor(self)$

PROVE $msgs' = msgs$

OBVIOUS

$\langle 2\rangle 1$. PICK $m \in 1bMessage \cup 2avMessage \cup 2bMessage$:

$\land\ m.acc = self$

$\land\ bmsgs' = bmsgs \cup \{m\}$

BY DEF $FakingAcceptor$

$\langle 2\rangle 2$. $m.type \in \{$"1b", "2av", "2b"$\}$

BY DEF $1bMessage$, $2avMessage$, $2bMessage$

$\langle 2\rangle 3$. $\land\ msgsOfType($"1a"$)' = msgsOfType($"1a"$)$

$\land\ 1cmsgs' = 1cmsgs$

$\langle 3\rangle$ $S \triangleq \{m\}$

$\langle 3\rangle 1$. $bmsgs' = bmsgs \cup S$

BY $\langle 2\rangle 1$

$\langle 3\rangle 2$. $\land \forall\, mm \in S : mm.type \neq$ "1a"

$\land \forall\, mm \in S : mm.type \neq$ "1c"

BY $\langle 2\rangle 1$, $\langle 2\rangle 2$

$\langle 3\rangle$ HIDE DEF S

$\langle 3\rangle 3$. $knowsSent' = knowsSent$

BY DEF $FakingAcceptor$

$\langle 3\rangle 4$. QED

BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$, $\langle 1\rangle$a, $\langle 1\rangle$c

$\langle 2\rangle 4$. $m.acc \notin Acceptor$

BY $\langle 2\rangle 1$, BQA

$\langle 2\rangle 5$. $1bmsgs' = 1bmsgs$

$\langle 3\rangle 1$. $acceptorMsgsOfType($"1b"$)' = acceptorMsgsOfType($"1b"$)$

BY $\langle 2\rangle 1$, $\langle 2\rangle 4$ DEF $acceptorMsgsOfType$, $msgsOfType$

$\langle 3\rangle 2$. QED

BY $\langle 3\rangle 1$ DEF $1bmsgs$

$\langle 2\rangle 6$. $2amsgs' = 2amsgs$

$\langle 3\rangle 1$. $acceptorMsgsOfType($"2av"$)' = acceptorMsgsOfType($"2av"$)$

BY $\langle 2\rangle 1$, $\langle 2\rangle 4$ DEF $acceptorMsgsOfType$, $msgsOfType$

$\langle 3\rangle 2$. QED

BY $\langle 3\rangle 1$ DEF $2amsgs$

$\langle 2\rangle 7$. $acceptorMsgsOfType($"2b"$)' = acceptorMsgsOfType($"2b"$)$

BY $\langle 2\rangle 1$, $\langle 2\rangle 4$ DEF $acceptorMsgsOfType$, $msgsOfType$

$\langle 2\rangle 8$. QED

BY $\langle 2\rangle 3$, $\langle 2\rangle 5$, $\langle 2\rangle 6$, $\langle 2\rangle 7$ DEF *msgs*

$\langle 1\rangle 9$. QED

BY $\langle 1\rangle 1$, $\langle 1\rangle 2$, $\langle 1\rangle 3$, $\langle 1\rangle 4$, $\langle 1\rangle 5$, $\langle 1\rangle 6$, $\langle 1\rangle 7$

Finally, we come to the proof of invariance of our inductive invariant *Inv*. Because *TLAPS* does not yet do temporal reasoning, we omit the proofs of the obviously true temporal-logic steps.

THEOREM $Spec \Rightarrow \Box Inv$

$\langle 1\rangle 1$. $Init \Rightarrow Inv$

$\langle 2\rangle$ SUFFICES ASSUME $Init$
PROVE Inv

OBVIOUS

$\langle 2\rangle$ USE DEF $Init$

$\langle 2\rangle 1$. $TypeOK$

BY DEF $TypeOK$

$\langle 2\rangle 2$. $bmsgsFinite$

BY $EmptySetFinite$ DEF $bmsgsFinite$, $1bOr2bMsgs$

$\langle 2\rangle 3$. $1bInv1$

BY DEF $1bInv1$

$\langle 2\rangle 4$. $1bInv2$

BY DEF $1bInv2$

$\langle 2\rangle 5$. $maxBalInv$

BY DEF $maxBalInv$

$\langle 2\rangle 6$. $2avInv1$

BY DEF $2avInv1$

$\langle 2\rangle 7$. $2avInv2$

BY DEF $2avInv2$

$\langle 2\rangle 8$. $2avInv3$

BY DEF $2avInv3$

$\langle 2\rangle 9$. $accInv$

BY DEF $accInv$

$\langle 2\rangle 10$. $knowsSentInv$

BY DEF $knowsSentInv$

$\langle 2\rangle 11$. QED

BY $\langle 2\rangle 1$, $\langle 2\rangle 2$, $\langle 2\rangle 3$, $\langle 2\rangle 4$, $\langle 2\rangle 5$, $\langle 2\rangle 6$, $\langle 2\rangle 7$, $\langle 2\rangle 8$, $\langle 2\rangle 9$, $\langle 2\rangle 10$
DEF Inv

$\langle 1\rangle 2$. $Inv \wedge [Next]_{vars} \Rightarrow Inv'$

$\langle 2\rangle$ SUFFICES ASSUME Inv, $[Next]_{vars}$
PROVE Inv'

OBVIOUS

$\langle 2\rangle 1$. ASSUME NEW $self \in Acceptor$,
NEW $b \in Ballot$,
$\vee\ Phase1b(self,\ b)$
$\vee\ Phase2av(self,\ b)$
$\vee\ Phase2b(self,\ b)$

$\lor$ $LearnsSent(self,\ b)$

PROVE Inv'

$\langle 3\rangle 1$.CASE $Phase1b(self,\ b)$

$\langle 4\rangle$ USE $Phase1b(self,\ b)$

$\langle 4\rangle$ DEFINE $mb \triangleq [type \mapsto$ "1b", $bal \mapsto b,\ acc \mapsto self,$
$m2av \mapsto 2avSent[self],$
$mbal \mapsto maxVBal[self],\ mval \mapsto maxVVal[self]]$
$mc \triangleq [type \mapsto$ "1b", $acc \mapsto self,\ bal \mapsto b,$
$mbal \mapsto maxVBal[self],\ mval \mapsto maxVVal[self]]$

$\langle 4\rangle 1.\ msgs' = msgs \cup \{mc\}$

BY $MsgsLemma$ DEF Inv

$\langle 4\rangle 2.\ TypeOK'$

$\langle 5\rangle 1.\ mb \in BMessage$

BY DEF Inv, $TypeOK$, $BMessage$, $1bMessage$, $ByzAcceptor$

$\langle 5\rangle 2$. QED

BY $\langle 5\rangle 1$ DEF $TypeOK$, Inv, $Phase1b$

$\langle 4\rangle 3.\ bmsgsFinite'$

BY $FiniteMsgsLemma$ DEF Inv, $bmsgsFinite$, $Phase1b$

$\langle 4\rangle 4.\ 1bInv1'$

$\langle 5\rangle 1$. SUFFICES ASSUME NEW $m \in bmsgs'$,
NEW $r \in m.m2av$,
$1bInv1!(m)!1$
PROVE $[type \mapsto$ "1c",
$bal \mapsto r.bal,\ val \mapsto r.val] \in msgs'$

BY DEF $1bInv1$

$\langle 5\rangle 2$.CASE $m = mb$

BY $\langle 4\rangle 1$, $\langle 5\rangle 1$, $\langle 5\rangle 2$ DEF Inv, $accInv$

$\langle 5\rangle 3$.CASE $m \in bmsgs$

BY $\langle 5\rangle 1$, $\langle 5\rangle 3$, $\langle 4\rangle 1$ DEF Inv, $1bInv1$

$\langle 5\rangle 4$. QED

BY $\langle 5\rangle 2$, $\langle 5\rangle 3$ DEF $Phase1b$

$\langle 4\rangle 5.\ 1bInv2'$

$\langle 5\rangle 1$. ASSUME NEW $m1 \in bmsgs$, NEW $m2 \in bmsgs$,
$1bInv2!(m1,\ m2)!1$
PROVE $m1 = m2$

BY $\langle 5\rangle 1$ DEF Inv, $1bInv2$

$\langle 5\rangle 2$. ASSUME NEW $m1 \in bmsgs'$,
$1bInv2!(m1,\ mb)!1$
PROVE $m1 = mb$

$\langle 6\rangle 1$. SUFFICES ASSUME $m1 \neq mb$
PROVE FALSE

OBVIOUS

$\langle 6\rangle 2.\ m1 \in bmsgs$

BY $\langle 6\rangle 1$ DEF $Phase1b$

$\langle 6\rangle 3.\ m1.bal \leq maxBal[self]$

BY $\langle 5\rangle 2$, $\langle 6\rangle 2$ DEF Inv, $maxBalInv$

$\langle 6\rangle 4$. $b > maxBal[self]$

BY DEF $Phase1b$

$\langle 6\rangle 5$. $m1.bal \in Ballot \cup \{-1\}$

BY $\langle 5\rangle 2$, $\langle 6\rangle 2$ DEF Inv, $TypeOK$, $1bMessage$

$\langle 6\rangle 6$. $maxBal[self] \in Ballot \cup \{-1\}$

BY DEF Inv, $TypeOK$

$\langle 6\rangle 7$. $\forall\, m1bal,\ maxbalself \in Ballot \cup \{-1\} :$
$b > maxbalself \land m1bal \leq maxbalself \Rightarrow m1bal \neq b$

BY $SimpleArithmetic$ DEF $Ballot$

$\langle 6\rangle 8$. $m1.bal \neq b$

BY $\langle 6\rangle 3$, $\langle 6\rangle 4$, $\langle 6\rangle 5$, $\langle 6\rangle 6$, $\langle 6\rangle 7$

$\langle 6\rangle 9$. QED

BY $\langle 5\rangle 2$, $\langle 6\rangle 1$, $\langle 6\rangle 8$

$\langle 5\rangle 3$. QED

$\langle 6\rangle$ SUFFICES ASSUME NEW $m1 \in bmsgs'$, NEW $m2 \in bmsgs'$,
$1bInv2!(m1,\ m2)!1$
PROVE $m1 = m2$

BY DEF $1bInv2$

$\langle 6\rangle 1$.CASE $m1 \neq mb \land m2 \neq mb$

BY $\langle 6\rangle 1$, $\langle 5\rangle 1$ DEF $Phase1b$

$\langle 6\rangle 2$.CASE $m1 = mb$

BY $\langle 5\rangle 2$, $\langle 6\rangle 2$

$\langle 6\rangle 3$.CASE $m2 = mb$

BY $\langle 5\rangle 2$, $\langle 6\rangle 3$

$\langle 6\rangle 4$. QED

BY $\langle 6\rangle 1$, $\langle 6\rangle 2$, $\langle 6\rangle 3$

$\langle 4\rangle 6$. $maxBalInv'$

$\langle 5\rangle 1$. SUFFICES ASSUME NEW $m \in bmsgs'$,
$m.type \in \{$"1b", "2av", "2b"$\}$,
$m.acc \in Acceptor$
PROVE $m.bal \leq maxBal'[m.acc]$

BY DEF $maxBalInv$

$\langle 5\rangle 2$. $\land \forall\, x \in Ballot \cup \{-1\} :$
$(b > x) \Rightarrow (x \leq b)$
$\land \forall\, x,\ y,\ z \in Ballot \cup \{-1\} :$
$(x \leq y) \land (y \leq z) \Rightarrow (x \leq z)$
$\land \forall\, x \in Ballot \cup \{-1\} : x \leq x$

BY $SimpleArithmetic$ DEF $Ballot$

$\langle 5\rangle 3$. ASSUME NEW $a \in Acceptor$
PROVE $\land\ maxBal[a] \leq maxBal'[a]$
$\land\ maxBal[a] \in Ballot \cup \{-1\}$
$\land\ maxBal'[a] \in Ballot \cup \{-1\}$

$\langle 6\rangle 1$.$\langle 5\rangle 3!2!2 \land \langle 5\rangle 3!2!3$

BY $\langle 4\rangle 2$ DEF Inv, $TypeOK$

⟨6⟩2. $maxBal[a] \leq maxBal'[a]$
⟨7⟩1.CASE $a = self$
BY ⟨5⟩2, ⟨6⟩1, ⟨7⟩1 DEF $Phase1b$, Inv, $TypeOK$
⟨7⟩2.CASE $a \neq self$
BY ⟨5⟩2, ⟨7⟩2, ⟨5⟩3 DEF $Phase1b$, Inv, $TypeOK$
⟨7⟩3. QED
BY ⟨7⟩1, ⟨7⟩2
⟨6⟩3. QED
BY ⟨6⟩1, ⟨6⟩2
⟨5⟩4.CASE $m \in bmsgs$
⟨6⟩1. $m.bal \leq maxBal[m.acc]$
BY ⟨5⟩1, ⟨5⟩4 DEF Inv, $maxBalInv$
⟨6⟩2. $m.bal \in Ballot \cup \{-1\}$
⟨7⟩1.CASE $m.type =$ "1b"
BY ⟨5⟩4, ⟨7⟩1, $BMessageLemma$ DEF Inv, $TypeOK$, $1bMessage$
⟨7⟩2.CASE $m.type =$ "2av"
BY ⟨5⟩4, ⟨7⟩2, $BMessageLemma$ DEF Inv, $TypeOK$, $2avMessage$
⟨7⟩3.CASE $m.type =$ "2b"
BY ⟨5⟩4, ⟨7⟩3, $BMessageLemma$ DEF Inv, $TypeOK$, $2bMessage$
⟨7⟩4. QED
BY ⟨7⟩1, ⟨7⟩2, ⟨7⟩3, ⟨5⟩1
⟨6⟩3. QED
⟨7⟩1. $\land\ maxBal[m.acc] \leq maxBal'[m.acc]$
$\land\ maxBal[m.acc] \in Ballot \cup \{-1\}$
$\land\ maxBal'[m.acc] \in Ballot \cup \{-1\}$
BY ⟨5⟩1, ⟨5⟩3
⟨7⟩2. QED
BY ⟨6⟩1, ⟨6⟩2, ⟨7⟩1, ⟨5⟩2
⟨5⟩5.CASE $m = mb$
BY ⟨5⟩2, ⟨5⟩5 DEF Inv, $TypeOK$, $Phase1b$
⟨5⟩6. QED
BY ⟨5⟩4, ⟨5⟩5 DEF $Phase1b$
⟨4⟩7. $2avInv1'$
⟨5⟩1. SUFFICES ASSUME NEW $m1 \in bmsgs'$, NEW $m2 \in bmsgs'$,
$2avInv1!(m1, m2)!1$
PROVE $2avInv1!(m1, m2)!2'$
BY DEF $2avInv1$
⟨5⟩2. $m1 \neq mb \land m2 \neq mb$
BY ⟨5⟩1, $mb.type =$ "1b"
⟨5⟩3. $m1 \in bmsgs \land m2 \in bmsgs$
BY ⟨5⟩2 DEF $Phase1b$
⟨5⟩4. QED
BY ⟨5⟩1, ⟨5⟩3 DEF Inv, $2avInv1$
⟨4⟩8. $2avInv2'$
⟨5⟩1. SUFFICES ASSUME NEW $m \in bmsgs'$,

$2avInv2!(m)!1$
PROVE $\exists\, r \in 2avSent'[m.acc] :$
$\wedge\ r.val = m.val$
$\wedge\ r.bal \geq m.bal$
BY DEF $2avInv2$
$\langle 5\rangle 2.\ m \neq mb$
BY $\langle 5\rangle 1$, $mb.type =$ "1b"
$\langle 5\rangle 3.\ m \in bmsgs$
BY $\langle 5\rangle 2$ DEF $Phase1b$
$\langle 5\rangle 4.$ QED
BY $\langle 5\rangle 1$, $\langle 5\rangle 3$ DEF $Phase1b$, Inv, $2avInv2$
$\langle 4\rangle 9.\ 2avInv3'$
$\langle 5\rangle 1.$ SUFFICES ASSUME NEW $m \in bmsgs'$,
$2avInv3!(m)!1$
PROVE $2avInv3!(m)!2'$
BY DEF $2avInv3$
$\langle 5\rangle 2.\ m \neq mb$
BY $\langle 5\rangle 1$, $mb.type =$ "1b"
$\langle 5\rangle 3.\ m \in bmsgs$
BY $\langle 5\rangle 2$ DEF $Phase1b$
$\langle 5\rangle 4.$ QED
BY $\langle 5\rangle 1$, $\langle 5\rangle 3$, $\langle 4\rangle 1$ DEF $Phase1b$, Inv, $2avInv3$
$\langle 4\rangle 10.\ accInv'$
$\langle 5\rangle$ SUFFICES ASSUME NEW $a \in Acceptor$,
NEW $r \in 2avSent[a]$
PROVE $\wedge\ r.bal \leq maxBal'[a]$
$\wedge\ [type \mapsto$ "1c", $bal \mapsto r.bal$, $val \mapsto r.val] \in msgs'$
BY DEF $accInv$, $Phase1b$
$\langle 5\rangle 1.\ \wedge\ r.bal \leq maxBal[a]$
$\wedge\ [type \mapsto$ "1c", $bal \mapsto r.bal$, $val \mapsto r.val] \in msgs$
BY DEF Inv, $accInv$
$\langle 5\rangle 2.\ [type \mapsto$ "1c", $bal \mapsto r.bal$, $val \mapsto r.val] \in msgs'$
BY $\langle 5\rangle 1$, $MsgsLemma$ DEF Inv
$\langle 5\rangle 3.\ maxBal[a] \leq maxBal'[a]$
$\langle 6\rangle 1.$CASE $a = self$
$\langle 7\rangle 1.\ \forall\, maxbal \in Ballot \cup \{-1\} :$
$b > maxbal \Rightarrow maxbal \leq b$
BY $SimpleArithmetic$ DEF $Ballot$
$\langle 7\rangle 2.$ QED
BY $\langle 6\rangle 1$, $\langle 7\rangle 1$ DEF $Phase1b$, Inv, $TypeOK$
$\langle 6\rangle 2.$CASE $a \neq self$
$\langle 7\rangle 1.\ \forall\, maxbal \in Ballot \cup \{-1\} : maxbal \leq maxbal$
BY $SimpleArithmetic$ DEF $Ballot$
$\langle 7\rangle 2.$ QED

BY $\langle 5\rangle 3$, $\langle 6\rangle 2$, $\langle 7\rangle 1$ DEF $Phase1b$, Inv, $TypeOK$

$\langle 6\rangle 3$. QED

BY $\langle 6\rangle 1$, $\langle 6\rangle 2$

$\langle 5\rangle 4$. $r.bal \leq maxBal'[a]$

$\langle 6\rangle 1$. $\forall\, rbal,\, maxb,\, maxbp \in Ballot \cup \{-1\}$:
$rbal \leq maxb \land maxb \leq maxbp \Rightarrow\ rbal \leq maxbp$

BY *SimpleArithmetic* DEF $Ballot$

$\langle 6\rangle 2$. $r.bal \in Ballot$

$\langle 7\rangle 1$. $2avSent[a] \in$ SUBSET $[val : Value,\, bal : Ballot]$

BY DEF Inv, $TypeOK$

$\langle 7\rangle 2$. $r \in [val : Value,\, bal : Ballot]$

BY $\langle 7\rangle 1$

$\langle 7\rangle 3$. QED

BY $\langle 7\rangle 2$

$\langle 6\rangle 3$. $\land\, maxBal[a] \in Ballot \cup \{-1\}$
$\land\, maxBal'[a] \in Ballot \cup \{-1\}$

BY $\langle 4\rangle 2$ DEF Inv, $TypeOK$

$\langle 6\rangle 4$. QED

BY $\langle 6\rangle 1$, $\langle 6\rangle 2$, $\langle 6\rangle 3$, $\langle 5\rangle 1$, $\langle 5\rangle 3$

$\langle 5\rangle 5$. QED

BY $\langle 5\rangle 2$, $\langle 5\rangle 4$

$\langle 4\rangle 11$. $knowsSentInv'$

$\langle 5\rangle 1$. $msgsOfType(\text{“1b”}) \subseteq msgsOfType(\text{“1b”})'$

BY DEF $Phase1b$, $msgsOfType$

$\langle 5\rangle 2$. QED

BY $\langle 5\rangle 1$ DEF Inv, $knowsSentInv$, $Phase1b$

$\langle 4\rangle 12$. QED

BY $\langle 4\rangle 2$, $\langle 4\rangle 3$, $\langle 4\rangle 4$, $\langle 4\rangle 5$, $\langle 4\rangle 6$, $\langle 4\rangle 7$, $\langle 4\rangle 8$, $\langle 4\rangle 9$,
$\langle 4\rangle 10$, $\langle 4\rangle 11$ DEF Inv

$\langle 3\rangle 2$.CASE $Phase2av(self,\, b)$

$\langle 4\rangle$ USE $Phase2av(self,\, b)$

$\langle 4\rangle 1$. PICK $mc \in sentMsgs(\text{“1c”},\, b)$:
$\land\, KnowsSafeAt(self,\, b,\, mc.val)$
$\land\, bmsgs' = bmsgs \cup$
$\{[type \mapsto \text{“2av”},\, bal \mapsto b,$
$val \mapsto mc.val,\, acc \mapsto self]\}$
$\land\, 2avSent' = [2avSent$ EXCEPT
$![self] = \{r \in 2avSent[self] : r.val \neq mc.val\}$
$\cup \{[val \mapsto mc.val,\, bal \mapsto b]\}]$

BY DEF $Phase2av$

$\langle 4\rangle 2$. $mc = [type \mapsto \text{“1c”},\, bal \mapsto mc.bal,\, val \mapsto mc.val]$

Follows from $\langle 4\rangle 1$ and def of *sentMsgs* (domain of *mc*).
but provers are unable to prove this easily.

$\langle 5\rangle 1$. $\land\, mc \in [type : \{\text{“1c”}\},\, bal : Ballot,\, val : Value]$
$\land\, mc.type = \text{“1c”}$

BY $\langle 4\rangle 1$, *BMessageLemma* DEF *sentMsgs*, *Inv*, *TypeOK*, *1cMessage*

$\langle 5\rangle$ DEFINE $mcx \triangleq [type \mapsto$ "1c", $bal \mapsto mc.bal$, $val \mapsto mc.val]$

$\langle 5\rangle 2. \land mc = [i \in \{$"type", "bal", "val"$\} \mapsto mc[i]]$
$\land mcx = [i \in \{$"type", "bal", "val"$\} \mapsto mcx[i]]$

BY $\langle 5\rangle 1$

$\langle 5\rangle 3. \forall i \in \{$"type", "bal", "val"$\} : mc[i] = mcx[i]$

$\langle 6\rangle$ SUFFICES ASSUME NEW $i \in \{$"type", "bal", "val"$\}$
PROVE $mc[i] = mcx[i]$

OBVIOUS

$\langle 6\rangle 1$.CASE $i =$ "type"

BY $\langle 5\rangle 1$, $\langle 6\rangle 1$

$\langle 6\rangle 2$.CASE $i =$ "bal"

BY $\langle 6\rangle 2$

$\langle 6\rangle 3$.CASE $i =$ "val"

BY $\langle 6\rangle 3$

$\langle 6\rangle 4$. QED

BY $\langle 6\rangle 1$, $\langle 6\rangle 2$, $\langle 6\rangle 3$

$\langle 5\rangle 4$. QED

BY $\langle 5\rangle 2$, $\langle 5\rangle 3$

$\langle 4\rangle$ DEFINE $mb \triangleq [type \mapsto$ "2av", $bal \mapsto b$,
$val \mapsto mc.val$, $acc \mapsto self]$
$mmc(v) \triangleq [type \mapsto$ "1c", $bal \mapsto b$, $val \mapsto v]$
$ma(v) \triangleq [type \mapsto$ "2a", $bal \mapsto b$, $val \mapsto v]$

$\langle 4\rangle 3. \lor msgs' = msgs$
$\lor \exists v \in Value :$
$\land mmc(v) \in msgs$
$\land msgs' = msgs \cup \{ma(v)\}$

$\langle 5\rangle 1$. *MsgsLemma*!2!2

BY *MsgsLemma* DEF *Inv*

$\langle 5\rangle 2$. QED

BY $\langle 5\rangle 1$

$\langle 4\rangle 4. msgs \subseteq msgs'$

BY $\langle 4\rangle 3$

$\langle 4\rangle 5. TypeOK'$

$\langle 5\rangle 1. mc \in 1cMessage$

BY $mc.type =$ "1c", *BMessageLemma* DEF *sentMsgs*, *Inv*, *TypeOK*

$\langle 5\rangle 2. mb.val \in Value$

$\langle 6\rangle 2$. QED

BY $\langle 5\rangle 1$ DEF *1cMessage*

$\langle 5\rangle 3$. *TypeOK*!1$'$

BY DEF *Inv*, *TypeOK*, *Phase2av*

$\langle 5\rangle 4$. *TypeOK*!2$'$

$\langle 6\rangle 1. 2avSent[self] \in$ SUBSET $[val : Value, bal : Ballot]$

BY DEF *Inv*, *TypeOK*

⟨6⟩2. $\{r \in 2avSent[self] : r.val \neq mc.val\}$
$\cup \{[val \mapsto mc.val,\ bal \mapsto b]\}$
$\in$ SUBSET $[val : Value,\ bal : Ballot]$
BY ⟨6⟩1, ⟨5⟩1, $mc.val \in Value$ DEF $1cMessage$
⟨6⟩3. QED
BY ⟨4⟩1, ⟨6⟩2 DEF Inv, $TypeOK$
⟨5⟩5. $TypeOK!3' \wedge TypeOK!4' \wedge TypeOK!5'$
BY DEF Inv, $TypeOK$, $Phase2av$
⟨5⟩6. $TypeOK!6'$
⟨6⟩1. $mb \in 2avMessage$
BY ⟨5⟩2 DEF $2avMessage$, $ByzAcceptor$
⟨6⟩2. QED
BY ⟨4⟩1, ⟨6⟩1 DEF Inv, $TypeOK$, $BMessage$
⟨5⟩7. QED
BY ⟨5⟩3, ⟨5⟩4, ⟨5⟩5, ⟨5⟩6 DEF $TypeOK$
⟨4⟩6. $bmsgsFinite'$
BY ⟨4⟩1, $FiniteMsgsLemma$ DEF Inv, $bmsgsFinite$
⟨4⟩7. $1bInv1'$
⟨5⟩1. SUFFICES ASSUME NEW $m \in bmsgs'$,
$1bInv1!(m)!1$
PROVE $1bInv1!(m)!2'$
BY DEF $1bInv1$
⟨5⟩2. $m \neq mb$
BY ⟨5⟩1, $mb.type =$ "2av"
⟨5⟩3. $m \in bmsgs$
BY ⟨4⟩1, ⟨5⟩2 DEF $Phase2av$
⟨5⟩4. QED
BY ⟨5⟩1, ⟨5⟩3, ⟨4⟩4 DEF $Phase2av$, Inv, $1bInv1$
⟨4⟩8. $1bInv2'$
⟨5⟩1. SUFFICES ASSUME NEW $m1 \in bmsgs'$, NEW $m2 \in bmsgs'$,
$1bInv2!(m1,\ m2)!1$
PROVE $1bInv2!(m1,\ m2)!2'$
BY DEF $1bInv2$
⟨5⟩2. $m1 \neq mb \wedge m2 \neq mb$
BY ⟨5⟩1, $mb.type =$ "2av"
⟨5⟩3. $m1 \in bmsgs \wedge m2 \in bmsgs$
BY ⟨4⟩1, ⟨5⟩2
⟨5⟩4. QED
BY ⟨5⟩1, ⟨5⟩3 DEF Inv, $1bInv2$
⟨4⟩9. $maxBalInv'$
⟨5⟩1. SUFFICES ASSUME NEW $m \in bmsgs'$,
$m.type \in \{$"1b", "2av", "2b"$\}$,
$m.acc \in Acceptor$
PROVE $m.bal \leq maxBal'[m.acc]$
BY DEF $maxBalInv$

⟨5⟩2. $\land \forall\, x,\ y,\ z \in Ballot \cup \{-1\} :$
$(x \leq y) \land (y \leq z) \Rightarrow (x \leq z)$
$\land \forall\, x \in Ballot \cup \{-1\} : x \leq x$
BY *SimpleArithmetic* DEF *Ballot*
⟨5⟩3. ASSUME NEW $a \in Acceptor$
PROVE $\land\ maxBal[a] \leq maxBal'[a]$
$\land\ maxBal[a] \in Ballot \cup \{-1\}$
$\land\ maxBal'[a] \in Ballot \cup \{-1\}$
⟨6⟩1.⟨5⟩3!2!2 $\land$ ⟨5⟩3!2!3
BY DEF *Inv*, *TypeOK*, *Phase*2*av*
⟨6⟩2. $maxBal[a] \leq maxBal'[a]$
⟨7⟩1.CASE $a = self$
BY ⟨6⟩1, ⟨7⟩1 DEF *Phase*2*av*, *Inv*, *TypeOK*
⟨7⟩2.CASE $a \neq self$
BY ⟨5⟩2, ⟨7⟩2, ⟨6⟩1 DEF *Phase*2*av*, *Inv*, *TypeOK*
⟨7⟩3. QED
BY ⟨7⟩1, ⟨7⟩2
⟨6⟩3. QED
BY ⟨6⟩1, ⟨6⟩2
⟨5⟩4.CASE $m \in bmsgs$
⟨6⟩1. $m.bal \leq maxBal[m.acc]$
BY ⟨5⟩1, ⟨5⟩4 DEF *Inv*, *maxBalInv*
⟨6⟩2. $m.bal \in Ballot \cup \{-1\}$
⟨7⟩1.CASE $m.type =$ "1b"
BY ⟨5⟩4, ⟨7⟩1, *BMessageLemma* DEF *Inv*, *TypeOK*, 1*bMessage*
⟨7⟩2.CASE $m.type =$ "2av"
BY ⟨5⟩4, ⟨7⟩2, *BMessageLemma* DEF *Inv*, *TypeOK*, 2*avMessage*
⟨7⟩3.CASE $m.type =$ "2b"
BY ⟨5⟩4, ⟨7⟩3, *BMessageLemma* DEF *Inv*, *TypeOK*, 2*bMessage*
⟨7⟩4. QED
BY ⟨7⟩1, ⟨7⟩2, ⟨7⟩3, ⟨5⟩1
⟨6⟩3. QED
⟨7⟩1. $\land\ maxBal[m.acc] \leq maxBal'[m.acc]$
$\land\ maxBal[m.acc] \in Ballot \cup \{-1\}$
$\land\ maxBal'[m.acc] \in Ballot \cup \{-1\}$
BY ⟨5⟩1, ⟨5⟩3
⟨7⟩2. QED
BY ⟨6⟩1, ⟨6⟩2, ⟨7⟩1, ⟨5⟩2
⟨5⟩5.CASE $m = mb$
⟨6⟩1. $b \leq b$
BY *SimpleArithmetic* DEF *Ballot*
⟨6⟩2. QED
⟨7⟩1. $m.bal = b \land m.acc = self$
BY ⟨5⟩5
⟨7⟩2. $maxBal'[self] = b$

BY DEF Inv, $TypeOK$, $Phase2av$

$\langle 7 \rangle 3$. QED

BY $\langle 6 \rangle 1$, $\langle 7 \rangle 1$, $\langle 7 \rangle 2$ DEF Inv, $TypeOK$, $Phase2av$

$\langle 5 \rangle 6$. QED

BY $\langle 5 \rangle 4$, $\langle 5 \rangle 5$, $\langle 4 \rangle 1$

$\langle 4 \rangle 10$. $2avInv1'$

$\langle 5 \rangle 1$. ASSUME NEW $m1 \in bmsgs$, NEW $m2 \in bmsgs$, $2avInv1!(m1, m2)!1$ PROVE $m1 = m2$

BY $\langle 5 \rangle 1$ DEF Inv, $2avInv1$

$\langle 5 \rangle 2$. ASSUME NEW $m1 \in bmsgs'$, $2avInv1!(m1, mb)!1$ PROVE $m1 = mb$

$\langle 6 \rangle 1$. SUFFICES ASSUME $m1 \neq mb$ PROVE FALSE

OBVIOUS

$\langle 6 \rangle 2$. $m1 \in bmsgs$

BY $\langle 6 \rangle 1$, $\langle 4 \rangle 1$

$\langle 6 \rangle 3$. PICK $r \in 2avSent[self] : r.bal \geq m1.bal$

BY $\langle 5 \rangle 2$, $\langle 6 \rangle 2$ DEF Inv, $2avInv2$

$\langle 6 \rangle 4$. $r.bal < b$

BY DEF $Phase2av$

$\langle 6 \rangle 5$. $m1.bal \in Ballot \cup \{-1\}$

BY $\langle 5 \rangle 2$, $\langle 6 \rangle 2$ DEF Inv, $TypeOK$, $1bMessage$

$\langle 6 \rangle 6$. $r \in [val : Value, bal : Ballot]$

BY DEF Inv, $TypeOK$

$\langle 6 \rangle 7$. $r.bal \in Ballot$

BY $\langle 6 \rangle 6$ DEF Inv, $TypeOK$

$\langle 6 \rangle 8$. $\forall\, m1bal,\ maxcbalself \in Ballot \cup \{-1\} :$
$maxcbalself < b \land maxcbalself \geq m1bal \Rightarrow m1bal \neq b$

BY $SimpleArithmetic$ DEF $Ballot$

$\langle 6 \rangle 9$. $m1.bal \neq b$

BY $\langle 6 \rangle 3$, $\langle 6 \rangle 4$, $\langle 6 \rangle 5$, $\langle 6 \rangle 7$, $\langle 6 \rangle 8$

$\langle 6 \rangle 10$. QED

BY $\langle 5 \rangle 2$, $\langle 6 \rangle 9$

$\langle 5 \rangle 3$. QED

$\langle 6 \rangle$ SUFFICES ASSUME NEW $m1 \in bmsgs'$, NEW $m2 \in bmsgs'$, $2avInv1!(m1, m2)!1$ PROVE $m1 = m2$

BY DEF $2avInv1$

$\langle 6 \rangle 1$.CASE $m1 \neq mb \land m2 \neq mb$

BY $\langle 6 \rangle 1$, $\langle 5 \rangle 1$, $\langle 4 \rangle 1$ DEF $Phase2av$

$\langle 6 \rangle 2$.CASE $m1 = mb$

BY $\langle 5 \rangle 2$, $\langle 6 \rangle 2$

⟨6⟩3. CASE $m2 = mb$
BY ⟨5⟩2, ⟨6⟩3
⟨6⟩4. QED
BY ⟨6⟩1, ⟨6⟩2, ⟨6⟩3
⟨4⟩11. $2avInv2'$
⟨5⟩1. SUFFICES ASSUME NEW $m \in bmsgs'$,
$2avInv2!(m)!1$
PROVE $\exists\, r \in 2avSent'[m.acc] : \wedge\, r.val = m.val$
$\wedge\, r.bal \geq m.bal$
BY DEF $2avInv2$
⟨5⟩2. CASE $m.acc = self$
⟨6⟩1. CASE $m = mb$
⟨7⟩ DEFINE $r \triangleq [val \mapsto mc.val,\ bal \mapsto b]$
⟨7⟩1. $r \in 2avSent'[self]$
BY ⟨4⟩1 DEF Inv, $TypeOK$
⟨7⟩2. $b \geq b$
BY $SimpleArithmetic$ DEF $Ballot$
⟨7⟩3. QED
BY ⟨7⟩1, ⟨6⟩1, $mb.bal = b$, ⟨7⟩2
⟨6⟩2. CASE $m \neq mb$
⟨7⟩1. $m \in bmsgs$
BY ⟨4⟩1, ⟨6⟩2
⟨7⟩2. PICK $r \in 2avSent[m.acc] : \wedge\, r.val = m.val$
$\wedge\, r.bal \geq m.bal$
BY ⟨5⟩1, ⟨7⟩1 DEF Inv, $2avInv2$
⟨7⟩3. CASE $r.val = mc.val$
⟨8⟩1. $r.bal \leq maxBal[self]$
BY ⟨5⟩2 DEF Inv, $accInv$
⟨8⟩2. $b \geq m.bal$
⟨9⟩1. $\forall\, rbal,\ mbal,\ maxbal \in Ballot \cup \{-1\} :$
$rbal \geq mbal \wedge rbal \leq maxbal \wedge maxbal \leq b$
$\Rightarrow b \geq mbal$
BY $SimpleArithmetic$ DEF $Ballot$
⟨9⟩2. $r.bal \in Ballot$
⟨10⟩1. $r \in [val : Value,\ bal : Ballot]$
BY ⟨5⟩2 DEF Inv, $TypeOK$
⟨10⟩2. QED
BY ⟨10⟩1
⟨9⟩3. $maxBal[self] \in Ballot \cup \{-1\}$
BY DEF Inv, $TypeOK$
⟨9⟩4. $m.bal \in Ballot$
BY ⟨5⟩1, ⟨5⟩2, ⟨7⟩1, $BMessageLemma$ DEF Inv, $TypeOK$, $2avMessage$
⟨9⟩5. QED
BY ⟨9⟩1, ⟨9⟩2, ⟨9⟩3, ⟨9⟩4, ⟨7⟩2, ⟨8⟩1 DEF $Phase2av$
⟨8⟩ DEFINE $rr \triangleq [val \mapsto mc.val,\ bal \mapsto b]$

⟨8⟩ $rr \in 2avSent'[m.acc]$
BY ⟨4⟩1, ⟨5⟩2 DEF Inv, $TypeOK$
⟨8⟩3. WITNESS $rr \in 2avSent'[m.acc]$
⟨8⟩4. QED
BY ⟨8⟩2, ⟨7⟩2, ⟨7⟩3
⟨7⟩4.CASE $r.val \neq mc.val$
BY ⟨7⟩2, ⟨4⟩1, ⟨5⟩2, ⟨7⟩4 DEF Inv, $TypeOK$
⟨7⟩5. QED
BY ⟨7⟩3, ⟨7⟩4
⟨6⟩3. QED
BY ⟨6⟩1, ⟨6⟩2
⟨5⟩3.CASE $m.acc \neq self$
⟨6⟩1. $m \in bmsgs$
BY ⟨5⟩3, $mb.acc = self$, ⟨4⟩1
⟨6⟩2. $m.acc \in Acceptor$
BY $BMessageLemma$, ⟨6⟩1, ⟨5⟩1 DEF Inv, $TypeOK$, $2avMessage$
⟨6⟩3. $2avSent'[m.acc] = 2avSent[m.acc]$
BY ⟨6⟩2, ⟨4⟩1, ⟨5⟩3 DEF Inv, $TypeOK$
⟨6⟩4. PICK $r \in 2avSent[m.acc] : \land r.val = m.val$
$\land r.bal \geq m.bal$
BY ⟨6⟩1, ⟨6⟩2, ⟨5⟩1 DEF Inv, $2avInv2$
⟨6⟩5. QED
BY ⟨6⟩3, ⟨6⟩4
⟨5⟩4. QED
BY ⟨5⟩2, ⟨5⟩3
⟨4⟩12. $2avInv3'$
⟨5⟩1. SUFFICES ASSUME NEW $m \in bmsgs'$,
$2avInv3!(m)!1$
PROVE $2avInv3!(m)!2'$
BY DEF $2avInv3$
⟨5⟩2.CASE $m \in bmsgs$
BY ⟨5⟩1, ⟨5⟩2, ⟨4⟩4 DEF Inv, $2avInv3$
⟨5⟩3.CASE $m = mb$
⟨6⟩1. $mc \in msgs$
BY ⟨4⟩1 DEF $sentMsgs$, $msgs$, $1cmsgs$, $msgsOfType$
⟨6⟩2. $mc \in msgs'$
BY ⟨6⟩1, ⟨4⟩4
⟨6⟩3. $mc.bal = m.bal \land mc.val = m.val$
BY ⟨5⟩3 DEF $sentMsgs$
⟨6⟩4. $mc = [type \mapsto$ "1c", $bal \mapsto m.bal$, $val \mapsto m.val]$
BY ⟨4⟩2, ⟨6⟩3
⟨6⟩5. QED
BY ⟨6⟩1, ⟨6⟩4, ⟨4⟩4
⟨5⟩4. QED
BY ⟨5⟩2, ⟨5⟩3, ⟨4⟩1

⟨4⟩13. $accInv'$
⟨5⟩1. SUFFICES ASSUME NEW $a \in Acceptor$,
NEW $r \in 2avSent'[a]$
PROVE $\land\ r.bal \leq maxBal'[a]$
$\land\ [type \mapsto$ "1c", $bal \mapsto r.bal,\ val \mapsto r.val]$
$\in msgs'$
BY DEF $accInv$
⟨5⟩2. $maxBal[a] \leq maxBal'[a]$
⟨6⟩1.CASE $a = self$
BY ⟨6⟩1 DEF Inv, $TypeOK$, $Phase2av$
⟨6⟩2.CASE $a \neq self$
⟨7⟩1. $\forall\, mbal \in Ballot \cup \{-1\} : mbal \leq mbal$
BY $SimpleArithmetic$ DEF $Ballot$
⟨7⟩2. QED
BY ⟨6⟩2, ⟨7⟩1 DEF Inv, $TypeOK$, $Phase2av$
⟨6⟩3. QED
BY ⟨6⟩1, ⟨6⟩2
⟨5⟩3.CASE $r \in 2avSent[a]$
⟨6⟩1. $\land\ r.bal \leq maxBal[a]$
$\land\ [type \mapsto$ "1c", $bal \mapsto r.bal,\ val \mapsto r.val] \in msgs'$
BY ⟨5⟩3, ⟨4⟩4 DEF Inv, $accInv$
⟨6⟩2. $\forall\, rbal,\ maxbal,\ maxbalp \in Ballot \cup \{-1\} :$
$rbal \leq maxbal \land maxbal \leq maxbalp \Rightarrow rbal \leq maxbalp$
BY $SimpleArithmetic$ DEF $Ballot$
⟨6⟩3. $r.bal \in Ballot$
⟨7⟩1. $r \in [val : Value,\ bal : Ballot]$
BY ⟨5⟩3 DEF Inv, $TypeOK$
⟨7⟩2. QED
BY ⟨7⟩1
⟨6⟩4. $\land\ maxBal[a] \in Ballot \cup \{-1\}$
$\land\ maxBal'[a] \in Ballot \cup \{-1\}$
BY ⟨4⟩5 DEF Inv, $TypeOK$
⟨6⟩5. $r.bal \leq maxBal'[a]$
BY ⟨6⟩1, ⟨6⟩2, ⟨6⟩3, ⟨6⟩4, ⟨5⟩2
⟨6⟩6. QED
BY ⟨6⟩1, ⟨6⟩5
⟨5⟩4.CASE $r \notin 2avSent[a]$
⟨6⟩1. $a = self$
⟨7⟩1. $2avSent'[a] \neq 2avSent[a]$
BY ⟨5⟩4
⟨7⟩2. QED
BY ⟨4⟩1, ⟨7⟩1 DEF Inv, $TypeOK$
⟨6⟩2. $r = [val \mapsto mc.val,\ bal \mapsto b]$
BY ⟨6⟩1, ⟨5⟩4, ⟨4⟩1 DEF Inv, $TypeOK$

```
2581            ⟨6⟩3. [type ↦ "1c", bal ↦ r.bal, val ↦ r.val] ∈ msgs′
2582              ⟨7⟩1. ∧ mc ∈ msgsOfType("1c")
2583                    ∧ mc.bal = b
2584                    ∧ KnowsSafeAt(self, mc.bal, mc.val)
2585                BY ⟨4⟩1 DEF sentMsgs, msgsOfType
2586              ⟨7⟩2. mc ∈ msgs′
2587                BY ⟨7⟩1, ⟨4⟩4 DEF msgs, 1cmsgs
2588              ⟨7⟩3. mc = ⟨6⟩3!1
2589                BY ⟨4⟩2, ⟨6⟩2, ⟨4⟩1 DEF sentMsgs ⟨7⟩4
2590              ⟨7⟩4. QED
2591                BY ⟨7⟩2, ⟨7⟩3
2592            ⟨6⟩4. ∧ maxBal′[a] = b
2593                  ∧ r.bal = b
2594              BY ⟨6⟩1, ⟨6⟩2 DEF Phase2av, Inv, TypeOK
2595            ⟨6⟩5. b ≤ b
2596              BY SimpleArithmetic DEF Ballot
2597            ⟨6⟩6. QED
2598              BY ⟨6⟩3, ⟨6⟩4, ⟨6⟩5
2599          ⟨5⟩5. QED
2600            BY ⟨5⟩3, ⟨5⟩4
2601        ⟨4⟩14. knowsSentInv′
2602          ⟨5⟩1. msgsOfType("1b")′ = msgsOfType("1b")
2603            ⟨6⟩1. ASSUME NEW m ∈ msgsOfType("1b")
2604                  PROVE  m ∈ msgsOfType("1b")′
2605              BY ⟨4⟩1 DEF msgsOfType
2606            ⟨6⟩2. ASSUME NEW m ∈ msgsOfType("1b")′
2607                  PROVE  m ∈ msgsOfType("1b")
2608              BY m.type = "1b", mb.type = "2av", ⟨4⟩1 DEF msgsOfType
2609            ⟨6⟩3. QED
2610              BY ⟨6⟩1, ⟨6⟩2
2611          ⟨5⟩2. QED
2612            BY ⟨5⟩1 DEF Phase2av, Inv, knowsSentInv, msgsOfType
2613        ⟨4⟩15. QED
2614        BY ⟨4⟩5, ⟨4⟩6, ⟨4⟩7, ⟨4⟩8, ⟨4⟩9, ⟨4⟩10, ⟨4⟩11, ⟨4⟩12,
2615           ⟨4⟩13, ⟨4⟩14 DEF Inv
2616      ⟨3⟩3.CASE Phase2b(self, b)
2617        ⟨4⟩ USE Phase2b(self, b)
2618        ⟨4⟩1. PICK v ∈ Value :
2619                ∧  ∃ Q ∈ ByzQuorum :
2620                     ∀ a ∈ Q :
2621                       ∃ m ∈ sentMsgs("2av", b) : ∧ m.val = v
2622                                                  ∧ m.acc = a
2623                ∧  msgs′ = msgs ∪
2624                             {[type ↦ "2b", acc ↦ self, bal ↦ b, val ↦ v]}
2625                ∧  bmsgs′ = (bmsgs ∪
```

$\{[type \mapsto$ "2b", $acc \mapsto self,\ bal \mapsto b,\ val \mapsto v]\})$

$\land\ \ maxVVal' = [maxVVal$ EXCEPT $![self] = v]$

$\langle 5\rangle 1.\ MsgsLemma!2!3$

BY $MsgsLemma$ DEF Inv

$\langle 5\rangle 2.\ MsgsLemma!2!3!(self,\ b)$

BY $\langle 5\rangle 1$

$\langle 5\rangle$ DEFINE $exp(v) \triangleq MsgsLemma!2!3!(self,\ b)!2!(v)$

$\langle 5\rangle 3.$ SUFFICES $\exists\, v \in Value : exp(v)$

OBVIOUS

$\langle 5\rangle 4.$ QED

BY $\langle 5\rangle 2$

$\langle 4\rangle$ DEFINE $mb \triangleq [type \mapsto$ "2b", $acc \mapsto self,\ bal \mapsto b,\ val \mapsto v]$

$\langle 4\rangle 2.\ TypeOK'$

$\langle 5\rangle 1.\ TypeOK!1' \land\ \ TypeOK!3' \land\ \ TypeOK!5'$

BY DEF $Phase2b$, Inv, $TypeOK$

$\langle 5\rangle 2.\ TypeOK!2'$

BY DEF Inv, $TypeOK$, $Phase2b$

$\langle 5\rangle 3.\ TypeOK!4'$

BY $\langle 4\rangle 1$ DEF Inv, $TypeOK$

$\langle 5\rangle 4.\ bmsgs' \subseteq BMessage$

BY $\langle 4\rangle 1$ DEF Inv, $TypeOK$, $BMessage$, $2bMessage$, $ByzAcceptor$

$\langle 5\rangle 5.$ QED

BY $\langle 5\rangle 1$, $\langle 5\rangle 2$, $\langle 5\rangle 3$, $\langle 5\rangle 4$ DEF $TypeOK$

$\langle 4\rangle 3.\ bmsgsFinite'$

BY $\langle 4\rangle 1$, $FiniteMsgsLemma$ DEF Inv, $bmsgsFinite$

$\langle 4\rangle 4.\ 1bInv1'$

$\langle 5\rangle 1.$ SUFFICES ASSUME NEW $m \in bmsgs'$,
$1bInv1!(m)!1$
PROVE $1bInv1!(m)!2'$

BY DEF $1bInv1$

$\langle 5\rangle 2.\ m \neq mb$

BY $\langle 5\rangle 1$, $mb.type =$ "2b"

$\langle 5\rangle 3.\ m \in bmsgs$

BY $\langle 4\rangle 1$, $\langle 5\rangle 2$

$\langle 5\rangle 4.$ QED

BY $\langle 4\rangle 1$, $\langle 5\rangle 1$, $\langle 5\rangle 3$, $\langle 4\rangle 4$ DEF Inv, $1bInv1$

$\langle 4\rangle 5.\ 1bInv2'$

$\langle 5\rangle 1.$ SUFFICES ASSUME NEW $m1 \in bmsgs'$, NEW $m2 \in bmsgs'$,
$1bInv2!(m1,\ m2)!1$
PROVE $1bInv2!(m1,\ m2)!2'$

BY DEF $1bInv2$

$\langle 5\rangle 2.\ m1 \neq mb \land m2 \neq mb$

BY $\langle 5\rangle 1$, $mb.type \ \ =$ "2b"

$\langle 5\rangle 3.\ m1 \in bmsgs \land m2 \in bmsgs$

BY $\langle 4\rangle 1$, $\langle 5\rangle 2$

⟨5⟩4. QED
BY ⟨5⟩1, ⟨5⟩3 DEF Inv, $1bInv2$
⟨4⟩6. $maxBalInv'$
The following copied almost exactly from proof for $Phase2b(self, b)$
⟨5⟩1. SUFFICES ASSUME NEW $m \in bmsgs'$,
$m.type \in \{$"1b", "2av", "2b"$\}$,
$m.acc \in Acceptor$
PROVE $m.bal \leq maxBal'[m.acc]$
BY DEF $maxBalInv$
⟨5⟩2. $\land \forall\, x,\, y,\, z \in Ballot \cup \{-1\} :$
$(x \leq y) \land (y \leq z) \Rightarrow (x \leq z)$
$\land \forall\, x \in Ballot \cup \{-1\} : x \leq x$
BY $SimpleArithmetic$ DEF $Ballot$
⟨5⟩3. ASSUME NEW $a \in Acceptor$
PROVE $\land\, maxBal[a] \leq maxBal'[a]$
$\land\, maxBal[a] \in Ballot \cup \{-1\}$
$\land\, maxBal'[a] \in Ballot \cup \{-1\}$
⟨6⟩1.⟨5⟩3!2!2 $\land$ ⟨5⟩3!2!3
BY ⟨4⟩2 DEF Inv, $TypeOK$, $Phase2b$
⟨6⟩2. $maxBal[a] \leq maxBal'[a]$
⟨7⟩1.CASE $a = self$
BY ⟨6⟩1, ⟨7⟩1 DEF $Phase2b$, Inv, $TypeOK$
⟨7⟩2.CASE $a \neq self$
BY ⟨5⟩2, ⟨7⟩2, ⟨6⟩1 DEF $Phase2b$, Inv, $TypeOK$
⟨7⟩3. QED
BY ⟨7⟩1, ⟨7⟩2
⟨6⟩3. QED
BY ⟨6⟩1, ⟨6⟩2
⟨5⟩4.CASE $m \in bmsgs$
⟨6⟩1. $m.bal \leq maxBal[m.acc]$
BY ⟨5⟩1, ⟨5⟩4 DEF Inv, $maxBalInv$
⟨6⟩2. $m.bal \in Ballot \cup \{-1\}$
⟨7⟩1.CASE $m.type =$ "1b"
BY ⟨5⟩4, ⟨7⟩1, $BMessageLemma$ DEF Inv, $TypeOK$, $1bMessage$
⟨7⟩2.CASE $m.type =$ "2av"
BY ⟨5⟩4, ⟨7⟩2, $BMessageLemma$ DEF Inv, $TypeOK$, $2avMessage$
⟨7⟩3.CASE $m.type =$ "2b"
BY ⟨5⟩4, ⟨7⟩3, $BMessageLemma$ DEF Inv, $TypeOK$, $2bMessage$
⟨7⟩4. QED
BY ⟨7⟩1, ⟨7⟩2, ⟨7⟩3, ⟨5⟩1
⟨6⟩3. QED
⟨7⟩1. $\land\, maxBal[m.acc] \leq maxBal'[m.acc]$
$\land\, maxBal[m.acc] \in Ballot \cup \{-1\}$
$\land\, maxBal'[m.acc] \in Ballot \cup \{-1\}$

BY $\langle 5\rangle 1$, $\langle 5\rangle 3$

$\langle 7\rangle 2$. QED

BY $\langle 6\rangle 1$, $\langle 6\rangle 2$, $\langle 7\rangle 1$, $\langle 5\rangle 2$

$\langle 5\rangle 5$.CASE $m = mb$

$\langle 6\rangle 1$. $b \leq b$

BY *SimpleArithmetic* DEF *Ballot*

$\langle 6\rangle 2$. QED

$\langle 7\rangle 1$. $m.bal = b \land m.acc = self$

BY $\langle 5\rangle 5$

$\langle 7\rangle 2$. $maxBal'[self] = b$

BY DEF *Inv*, *TypeOK*, *Phase*2*b*

$\langle 7\rangle 3$. QED

BY $\langle 6\rangle 1$, $\langle 7\rangle 1$, $\langle 7\rangle 2$ DEF *Inv*, *TypeOK*, *Phase*2*b*

$\langle 5\rangle 6$. QED

BY $\langle 5\rangle 4$, $\langle 5\rangle 5$, $\langle 4\rangle 1$

$\langle 4\rangle 7$. $2avInv1'$

$\langle 5\rangle 1$. SUFFICES ASSUME NEW $m1 \in bmsgs'$, NEW $m2 \in bmsgs'$,
$2avInv1!(m1, m2)!1$
PROVE $2avInv1!(m1, m2)!2'$

BY DEF $2avInv1$

$\langle 5\rangle 2$. $m1 \neq mb \land m2 \neq mb$

BY $\langle 5\rangle 1$, $mb.type$ = "2b"

$\langle 5\rangle 3$. $m1 \in bmsgs \land m2 \in bmsgs$

BY $\langle 4\rangle 1$, $\langle 5\rangle 2$

$\langle 5\rangle 4$. QED

BY $\langle 5\rangle 1$, $\langle 5\rangle 3$ DEF *Inv*, $2avInv1$

$\langle 4\rangle 8$. $2avInv2'$

$\langle 5\rangle 1$. SUFFICES ASSUME NEW $m \in bmsgs'$,
$2avInv2!(m)!1$
PROVE $\exists\, r \in 2avSent'[m.acc] : \land r.val = m.val$
$\land r.bal \geq m.bal$

BY DEF $2avInv2$, *Phase*2*b*, *Inv*, *TypeOK*

$\langle 5\rangle 2$. $m \neq mb$

BY $\langle 5\rangle 1$, $mb.type$ = "2b"

$\langle 5\rangle 3$. $m \in bmsgs$

BY $\langle 4\rangle 1$, $\langle 5\rangle 2$

$\langle 5\rangle 4$. $2avSent'[m.acc] = 2avSent[m.acc]$

BY $\langle 5\rangle 1$ DEF *Inv*, *TypeOK*, *Phase*2*b*

$\langle 5\rangle 5$. QED

BY $\langle 5\rangle 1$, $\langle 5\rangle 3$, $\langle 5\rangle 4$ DEF *Inv*, $2avInv2$ BY $\langle 5\rangle 4$, $\langle 5\rangle 5$

$\langle 4\rangle 9$. $2avInv3'$

$\langle 5\rangle 1$. SUFFICES ASSUME NEW $m \in bmsgs'$,
$2avInv3!(m)!1$
PROVE $2avInv3!(m)!2'$

BY DEF $2avInv3$

⟨5⟩2. $m \neq mb$
BY ⟨5⟩1, $mb.type = \text{“2b”}$
⟨5⟩3. $m \in bmsgs$
BY ⟨4⟩1, ⟨5⟩2
⟨5⟩4. QED
BY ⟨5⟩1, ⟨5⟩3, ⟨4⟩1 DEF Inv, $2avInv3$
⟨4⟩10. $accInv'$
Proof copied with a few changes from that of $Phase1b$
⟨5⟩ SUFFICES ASSUME NEW $a \in Acceptor$,
NEW $r \in 2avSent[a]$
PROVE $\land\, r.bal \leq maxBal'[a]$
$\land\, [type \mapsto \text{“1c”},\ bal \mapsto r.bal,\ val \mapsto r.val]$
$\in msgs'$
BY DEF $accInv$, $Phase2b$
⟨5⟩1. $\land\, r.bal \leq maxBal[a]$
$\land\, [type \mapsto \text{“1c”},\ bal \mapsto r.bal,\ val \mapsto r.val] \in msgs$
BY DEF Inv, $accInv$
⟨5⟩2. $[type \mapsto \text{“1c”},\ bal \mapsto r.bal,\ val \mapsto r.val] \in msgs'$
BY ⟨5⟩1, $MsgsLemma$ DEF Inv
⟨5⟩3. $maxBal[a] \leq maxBal'[a]$
⟨6⟩1.CASE $a = self$
⟨7⟩1. $\forall\, maxbal \in Ballot \cup \{-1\} :$
$b > maxbal \Rightarrow maxbal \leq b$
BY $SimpleArithmetic$ DEF $Ballot$
⟨7⟩2. QED
BY ⟨6⟩1, ⟨7⟩1 DEF $Phase2b$, Inv, $TypeOK$
⟨6⟩2.CASE $a \neq self$
⟨7⟩1. $\forall\, maxbal \in Ballot \cup \{-1\} : maxbal \leq maxbal$
BY $SimpleArithmetic$ DEF $Ballot$
⟨7⟩2. $maxBal'[a] = maxBal[a]$
BY ⟨6⟩2 DEF $Phase2b$, Inv, $TypeOK$
⟨7⟩3. QED
BY ⟨7⟩1, ⟨7⟩2 DEF $Phase2b$, Inv, $TypeOK$
⟨6⟩3. QED
BY ⟨6⟩1, ⟨6⟩2
⟨5⟩4. $r.bal \leq maxBal'[a]$
⟨6⟩1. $\forall\, rbal,\ maxb,\ maxbp \in Ballot \cup \{-1\} :$
$rbal \leq maxb \land maxb \leq maxbp \Rightarrow\ rbal \leq maxbp$
BY $SimpleArithmetic$ DEF $Ballot$
⟨6⟩2. $r.bal \in Ballot$
⟨7⟩1. $2avSent[a] \in$ SUBSET $[val : Value,\ bal : Ballot]$
BY DEF Inv, $TypeOK$
⟨7⟩2. $r \in [val : Value,\ bal : Ballot]$
BY ⟨7⟩1
⟨7⟩3. QED

BY $\langle 7\rangle 2$
$\langle 6\rangle 3.\ \wedge\ maxBal[a] \in Ballot \cup \{-1\}$
$\wedge\ maxBal'[a] \in Ballot \cup \{-1\}$
BY $\langle 4\rangle 2$ DEF Inv, $TypeOK$
$\langle 6\rangle 4.$ QED
BY $\langle 6\rangle 1$, $\langle 6\rangle 2$, $\langle 6\rangle 3$, $\langle 5\rangle 1$, $\langle 5\rangle 3$
$\langle 5\rangle 5.$ QED
BY $\langle 5\rangle 2$, $\langle 5\rangle 4$
$\langle 4\rangle 11.\ knowsSentInv'$
$\langle 5\rangle 1.\ msgsOfType(\text{“1b”}) \subseteq msgsOfType(\text{“1b”})'$
BY $\langle 4\rangle 1$ DEF $Phase2b$, $msgsOfType$
$\langle 5\rangle 2.$ QED
BY $\langle 5\rangle 1$ DEF Inv, $knowsSentInv$, $Phase2b$
$\langle 4\rangle 12.$ QED
BY $\langle 4\rangle 2$, $\langle 4\rangle 3$, $\langle 4\rangle 4$, $\langle 4\rangle 5$, $\langle 4\rangle 6$, $\langle 4\rangle 7$, $\langle 4\rangle 8$, $\langle 4\rangle 9$,
$\langle 4\rangle 10$, $\langle 4\rangle 11$ DEF Inv
$\langle 3\rangle 4.$CASE $LearnsSent(self,\ b)$
$\langle 4\rangle$ USE $LearnsSent(self,\ b)$
$\langle 4\rangle 1.$ PICK $MS :\ \wedge\ MS \subseteq \{m \in msgsOfType(\text{“1c”}) : m.bal = b\}$
$\wedge\ msgs' = msgs \cup MS$
BY $MsgsLemma$ DEF Inv
$\langle 4\rangle 2.$ PICK $S :$
$\wedge\ \ S \subseteq sentMsgs(\text{“1b”},\ b)$
$\wedge\ \ knowsSent' =$
$[knowsSent$ EXCEPT $![self] = knowsSent[self] \cup S]$
BY DEF $LearnsSent$
$\langle 4\rangle 3.\ TypeOK'$
$\langle 5\rangle 1.\ knowsSent' \in [Acceptor \to$ SUBSET $1bMessage]$
$\langle 6\rangle$ DEFINE $ks(a) \triangleq$ IF $a = self$ THEN $knowsSent[self] \cup S$
ELSE $knowsSent[a]$
$\langle 6\rangle$ SUFFICES ASSUME NEW $a \in Acceptor$
PROVE $ks(a) \in$ SUBSET $1bMessage$
BY $\langle 4\rangle 2$ DEF Inv, $TypeOK$
$\langle 6\rangle 1.$CASE $a \neq self$
BY $\langle 6\rangle 1$ DEF Inv, $TypeOK$
$\langle 6\rangle 2.$CASE $a = self$
$\langle 7\rangle$ SUFFICES ASSUME NEW $m \in S$
PROVE $m \in 1bMessage$
BY $\langle 6\rangle 2$ DEF Inv, $TypeOK$
$\langle 7\rangle$ QED
BY $\langle 4\rangle 2$, $BMessageLemma$ DEF $sentMsgs$, Inv, $TypeOK$
$\langle 6\rangle 3.$ QED
BY $\langle 6\rangle 1$, $\langle 6\rangle 2$
$\langle 5\rangle 2.$ QED
BY $\langle 5\rangle 1$ DEF Inv, $TypeOK$, $LearnsSent$

⟨4⟩4. $bmsgsFinite'$
BY DEF $LearnsSent$, Inv, $bmsgsFinite$, $1bOr2bMsgs$
⟨4⟩5. $1bInv1'$
BY ⟨4⟩1 DEF $LearnsSent$, Inv, $1bInv1$
⟨4⟩6. $1bInv2'$
BY DEF $LearnsSent$, Inv, $1bInv2$
⟨4⟩7. $maxBalInv'$
BY DEF $LearnsSent$, Inv, $maxBalInv$
⟨4⟩8. $2avInv1'$
BY DEF $LearnsSent$, Inv, $2avInv1$
⟨4⟩9. $2avInv2'$
BY DEF $LearnsSent$, Inv, $2avInv2$
⟨4⟩10. $2avInv3'$
BY ⟨4⟩1 DEF $LearnsSent$, Inv, $2avInv3$
⟨4⟩11. $accInv'$
BY ⟨4⟩1 DEF $LearnsSent$, Inv, $accInv$
⟨4⟩12. $knowsSentInv'$
⟨5⟩ SUFFICES ASSUME NEW $a \in Acceptor$
PROVE $knowsSent'[a] \subseteq msgsOfType(\text{“1b”})$
BY DEF $LearnsSent$, $knowsSentInv$, $msgsOfType$
⟨5⟩1.CASE $a \neq self$
BY ⟨4⟩2 DEF Inv, $TypeOK$, $knowsSentInv$, $sentMsgs$, $msgsOfType$
⟨5⟩2.CASE $a = self$
BY ⟨4⟩2 DEF Inv, $TypeOK$, $knowsSentInv$, $sentMsgs$, $msgsOfType$
⟨5⟩3. QED
BY ⟨5⟩1, ⟨5⟩2
⟨4⟩13. QED
BY ⟨4⟩3, ⟨4⟩4, ⟨4⟩5, ⟨4⟩6, ⟨4⟩7, ⟨4⟩8, ⟨4⟩9, ⟨4⟩10,
⟨4⟩11, ⟨4⟩12 DEF Inv
⟨3⟩5. QED
BY ⟨2⟩1, ⟨3⟩1, ⟨3⟩2, ⟨3⟩3, ⟨3⟩4
⟨2⟩2. ASSUME NEW $self \in Ballot$,
$\lor\ Phase1a(self)$
$\lor\ Phase1c(self)$
PROVE Inv'
⟨3⟩1.CASE $Phase1a(self)$
⟨4⟩ USE $Phase1a(self)$
⟨4⟩ DEFINE $ma \triangleq [type \mapsto \text{“1a”},\ bal \mapsto self]$
⟨4⟩1. $msgs' = msgs \cup \{ma\}$
BY $MsgsLemma$ DEF Inv
⟨4⟩2. $TypeOK'$
⟨5⟩1. $bmsgs' \subseteq BMessage$
BY DEF $Phase1a$, Inv, $TypeOK$, $BMessage$, $1aMessage$
⟨5⟩2. QED
BY ⟨5⟩1 DEF Inv, $TypeOK$, $Phase1a$

⟨4⟩3. $bmsgsFinite'$
BY $FiniteMsgsLemma$ DEF Inv, $bmsgsFinite$, $Phase1a$
⟨4⟩4. $1bInv1'$
BY ⟨4⟩1 DEF $Phase1a$, Inv, $1bInv1$
⟨4⟩5. $1bInv2'$
⟨5⟩1. SUFFICES ASSUME NEW $m1 \in bmsgs'$, NEW $m2 \in bmsgs'$,
$1bInv2!(m1, m2)!1$
PROVE $1bInv2!(m1, m2)!2'$
BY DEF $1bInv2$
⟨5⟩2. $m1 \neq ma \land m2 \neq ma$
BY ⟨5⟩1, $ma.type = \text{“1a”}$
⟨5⟩3. $m1 \in bmsgs \land m2 \in bmsgs$
BY ⟨5⟩2 DEF $Phase1a$
⟨5⟩4. QED
BY ⟨5⟩1, ⟨5⟩3 DEF Inv, $1bInv2$
⟨4⟩6. $maxBalInv'$
BY DEF $Phase1a$, Inv, $maxBalInv$
⟨4⟩7. $2avInv1'$
⟨5⟩1. SUFFICES ASSUME NEW $m1 \in bmsgs'$, NEW $m2 \in bmsgs'$,
$2avInv1!(m1, m2)!1$
PROVE $2avInv1!(m1, m2)!2'$
BY DEF $2avInv1$
⟨5⟩2. $m1 \neq ma \land m2 \neq ma$
BY ⟨5⟩1, $ma.type = \text{“1a”}$
⟨5⟩3. $m1 \in bmsgs \land m2 \in bmsgs$
BY ⟨5⟩2 DEF $Phase1a$
⟨5⟩4. QED
BY ⟨5⟩1, ⟨5⟩3 DEF Inv, $2avInv1$
⟨4⟩8. $2avInv2'$
BY DEF $Phase1a$, Inv, $2avInv2$
⟨4⟩9. $2avInv3'$
BY ⟨4⟩1 DEF $Phase1a$, Inv, $2avInv3$
⟨4⟩10. $accInv'$
BY ⟨4⟩1 DEF $Phase1a$, Inv, $accInv$
⟨4⟩11. $knowsSentInv'$
BY DEF Inv, $knowsSentInv$, $msgsOfType$, $Phase1a$
⟨4⟩12. QED
BY ⟨4⟩2, ⟨4⟩3, ⟨4⟩4, ⟨4⟩5, ⟨4⟩6, ⟨4⟩7, ⟨4⟩8, ⟨4⟩9,
⟨4⟩10, ⟨4⟩11 DEF Inv
⟨3⟩2.CASE $Phase1c(self)$
⟨4⟩ USE $Phase1c(self)$
⟨4⟩1. PICK $S : \land S \in \text{SUBSET } [type : \{\text{“1c”}\}, bal : \{self\}, val : Value]$
$\land bmsgs' = bmsgs \cup S$
BY DEF $Phase1c$
⟨4⟩2. PICK $MS :$

$\land\ MS \in \text{SUBSET}\ [type : \{\text{“1c”}\},\ bal : \{self\},\ val : Value]$
$\land\ \forall\, m \in MS :$
$\exists\, a \in Acceptor : KnowsSafeAt(a,\ m.bal,\ m.val)$
$\land\ msgs' = msgs \cup MS$
BY $MsgsLemma$ DEF Inv
$\langle 4\rangle 3.\ TypeOK'$
$\langle 5\rangle 1.\ bmsgs' \subseteq BMessage$
BY $\langle 4\rangle 1$ DEF Inv, $TypeOK$, $BMessage$, $1cMessage$
$\langle 5\rangle 2.$ QED
BY $\langle 5\rangle 1$ DEF Inv, $TypeOK$, $Phase1c$
$\langle 4\rangle 4.\ bmsgsFinite'$
$\langle 5\rangle 1.\ 1bOr2bMsgs' = 1bOr2bMsgs$
BY $\langle 4\rangle 1$ DEF $1bOr2bMsgs$
$\langle 5\rangle 2.$ QED
BY $\langle 5\rangle 1$ DEF $bmsgsFinite$, Inv
$\langle 4\rangle 5.\ 1bInv1'$
BY $\langle 4\rangle 2$ DEF $Phase1c$, Inv, $1bInv1$
$\langle 4\rangle 6.\ 1bInv2'$
$\langle 5\rangle 1.$ SUFFICES ASSUME NEW $m1 \in bmsgs'$, NEW $m2 \in bmsgs'$,
$1bInv2!(m1,\ m2)!1$
PROVE $1bInv2!(m1,\ m2)!2'$
BY DEF $1bInv2$
$\langle 5\rangle 2.\ m1 \notin S \land m2 \notin S$
$\langle 6\rangle 1.\ \forall\, m \in S : m.type = \text{“1c”}$
BY $\langle 4\rangle 1$
$\langle 6\rangle 2.$ QED
BY $\langle 6\rangle 1$, $\langle 5\rangle 1$
$\langle 5\rangle 3.\ m1 \in bmsgs \land m2 \in bmsgs$
BY $\langle 5\rangle 2$, $\langle 4\rangle 1$
$\langle 5\rangle 4.$ QED
BY $\langle 5\rangle 1$, $\langle 5\rangle 3$ DEF Inv, $1bInv2$
$\langle 4\rangle 7.\ maxBalInv'$
BY DEF $Phase1c$, Inv, $maxBalInv$
$\langle 4\rangle 8.\ 2avInv1'$
$\langle 5\rangle 1.$ SUFFICES ASSUME NEW $m1 \in bmsgs'$, NEW $m2 \in bmsgs'$,
$2avInv1!(m1,\ m2)!1$
PROVE $2avInv1!(m1,\ m2)!2'$
BY DEF $2avInv1$
$\langle 5\rangle 2.\ m1 \notin S \land m2 \notin S$
$\langle 6\rangle 1.\ \forall\, m \in S : m.type = \text{“1c”}$
BY $\langle 4\rangle 1$
$\langle 6\rangle 2.$ QED
BY $\langle 6\rangle 1$, $\langle 5\rangle 1$
$\langle 5\rangle 3.\ m1 \in bmsgs \land m2 \in bmsgs$
BY $\langle 5\rangle 2$, $\langle 4\rangle 1$

2986 ⟨5⟩4. QED
2987 BY ⟨5⟩1, ⟨5⟩3 DEF Inv, $2avInv1$
2988 ⟨4⟩9. $2avInv2'$
2989 BY DEF $Phase1c$, Inv, $2avInv2$
2990 ⟨4⟩10. $2avInv3'$
2991 BY ⟨4⟩2 DEF $Phase1c$, Inv, $2avInv3$
2992 ⟨4⟩11. $accInv'$
2993 BY ⟨4⟩2 DEF $Phase1c$, Inv, $accInv$
2994 ⟨4⟩12. $knowsSentInv'$
2995 BY DEF Inv, $knowsSentInv$, $msgsOfType$, $Phase1c$
2996 ⟨4⟩13. QED
2997 BY ⟨4⟩3, ⟨4⟩4, ⟨4⟩5, ⟨4⟩6, ⟨4⟩7, ⟨4⟩8, ⟨4⟩9, ⟨4⟩10,
2998 ⟨4⟩11, ⟨4⟩12 DEF Inv
2999 ⟨3⟩3. QED
3000 BY ⟨3⟩1, ⟨3⟩2, ⟨2⟩2
3001 ⟨2⟩3. ASSUME NEW $self \in FakeAcceptor$,
3002 $FakingAcceptor(self)$
3003 PROVE
3004 Inv'
3005 ⟨3⟩ USE $FakingAcceptor(self)$
3006 ⟨3⟩1. PICK $m \in 1bMessage \cup 2avMessage \cup 2bMessage$:
3007 $\land\ m.acc \notin Acceptor$
3008 $\land\ bmsgs' = bmsgs \cup \{m\}$
3009 BY BQA DEF $FakingAcceptor$
3010 ⟨3⟩2. $msgs' = msgs$
3011 BY $MsgsLemma$ DEF Inv
3012 ⟨3⟩3. $TypeOK'$
3013 BY ⟨3⟩1 DEF Inv, $TypeOK$, $BMessage$, $FakingAcceptor$
3014 ⟨3⟩4. $bmsgsFinite'$
3015 BY ⟨3⟩1, $FiniteMsgsLemma$ DEF Inv, $TypeOK$
3016 ⟨3⟩5. $1bInv1'$
3017 ⟨4⟩1. SUFFICES ASSUME NEW $mm \in bmsgs'$,
3018 $1bInv1!(mm)!1$
3019 PROVE $1bInv1!(mm)!2'$
3020 BY DEF $1bInv1$
3021 ⟨4⟩2. $mm \in bmsgs$
3022 BY ⟨4⟩1, ⟨3⟩1
3023 ⟨4⟩3. QED
3024 BY ⟨4⟩1, ⟨4⟩2, ⟨3⟩2 DEF Inv, $1bInv1$
3025 ⟨3⟩6. $1bInv2'$
3026 ⟨4⟩1. SUFFICES ASSUME NEW $m1 \in bmsgs'$, NEW $m2 \in bmsgs'$,
3027 $1bInv2!(m1, m2)!1$
3028 PROVE $m1 = m2$
3029 BY DEF $1bInv2$
3030 ⟨4⟩2. $m1 \in bmsgs \land m2 \in bmsgs$

BY $\langle 4\rangle 1$, $\langle 3\rangle 1$

$\langle 4\rangle 3$. QED

BY $\langle 4\rangle 1$, $\langle 4\rangle 2$ DEF Inv, $1bInv2$

$\langle 3\rangle 7$. $maxBalInv'$

$\langle 4\rangle 1$. SUFFICES ASSUME NEW $mm \in bmsgs'$,
$maxBalInv!(mm)!1$
PROVE $maxBalInv!(mm)!2'$

BY DEF $maxBalInv$

$\langle 4\rangle 2$. $mm \in bmsgs$

BY $\langle 4\rangle 1$, $\langle 3\rangle 1$

$\langle 4\rangle 3$. QED

BY $\langle 4\rangle 1$, $\langle 4\rangle 2$ DEF Inv, $maxBalInv$, $FakingAcceptor$

$\langle 3\rangle 8$. $2avInv1'$

$\langle 4\rangle 1$. SUFFICES ASSUME NEW $m1 \in bmsgs'$, NEW $m2 \in bmsgs'$,
$2avInv1!(m1, m2)!1$
PROVE $m1 = m2$

BY DEF $2avInv1$

$\langle 4\rangle 2$. $m1 \in bmsgs \land m2 \in bmsgs$

BY $\langle 4\rangle 1$, $\langle 3\rangle 1$

$\langle 4\rangle 3$. QED

BY $\langle 4\rangle 1$, $\langle 4\rangle 2$ DEF Inv, $2avInv1$

$\langle 3\rangle 9$. $2avInv2'$

$\langle 4\rangle 1$. SUFFICES ASSUME NEW $mm \in bmsgs'$,
$2avInv2!(mm)!1$
PROVE $2avInv2!(mm)!2'$

BY DEF $2avInv2$

$\langle 4\rangle 2$. $mm \in bmsgs$

BY $\langle 4\rangle 1$, $\langle 3\rangle 1$

$\langle 4\rangle 3$. QED

BY $\langle 4\rangle 1$, $\langle 4\rangle 2$ DEF Inv, $2avInv2$, $FakingAcceptor$

$\langle 3\rangle 10$. $2avInv3'$

$\langle 4\rangle 1$. SUFFICES ASSUME NEW $mm \in bmsgs'$,
$2avInv3!(mm)!1$
PROVE $2avInv3!(mm)!2'$

BY DEF $2avInv3$

$\langle 4\rangle 2$. $mm \in bmsgs$

BY $\langle 4\rangle 1$, $\langle 3\rangle 1$

$\langle 4\rangle 3$. QED

BY $\langle 4\rangle 1$, $\langle 4\rangle 2$, $\langle 3\rangle 2$ DEF Inv, $2avInv3$

$\langle 3\rangle 11$. $accInv'$

BY $\langle 3\rangle 2$ DEF Inv, $accInv$, $FakingAcceptor$

$\langle 3\rangle 12$. $knowsSentInv'$

$\langle 4\rangle$ SUFFICES ASSUME NEW $a \in Acceptor$
PROVE $knowsSent'[a] \subseteq msgsOfType(\text{“1b”})'$

BY DEF $knowsSentInv$

⟨4⟩1. $msgsOfType(\text{“1b”}) \subseteq msgsOfType(\text{“1b”})'$
BY ⟨3⟩1 DEF $msgsOfType$
⟨4⟩2. QED
BY ⟨4⟩1 DEF $FakingAcceptor$, Inv, $knowsSentInv$
⟨3⟩13. QED
BY ⟨3⟩3, ⟨3⟩4, ⟨3⟩5, ⟨3⟩6, ⟨3⟩7, ⟨3⟩8, ⟨3⟩9, ⟨3⟩10,
⟨3⟩11, ⟨3⟩12 DEF Inv
⟨2⟩4. ASSUME UNCHANGED $vars$
PROVE Inv'
⟨3⟩ USE UNCHANGED $vars$ DEF Inv, $vars$
⟨3⟩ $msgs = msgs'$
BY DEF $msgs$, $msgsOfType$, $1bmsgs$, $1bRestrict$, $acceptorMsgsOfType$, $1cmsgs$,
$KnowsSafeAt$, $2amsgs$
⟨3⟩ QED
BY DEF $TypeOK$, $bmsgsFinite$, $1bOr2bMsgs$, $1bInv1$, $1bInv2$,
$maxBalInv$, $2avInv1$, $2avInv2$, $2avInv3$, $accInv$, $knowsSentInv$, $msgsOfType$
⟨2⟩5. QED
BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3, ⟨2⟩4, $NextDef$

⟨1⟩3. QED
BY ⟨1⟩1, ⟨1⟩2, $RuleInv1$ DEF $Spec$
PROOF OMITTED

We next use the invariance of Inv to prove that algorithm $BPCon$ implements algorithm $PaxosConsensus$ under the refinement mapping defined by the INSTANCE statement above. Again, we must omit the trivial temporal logic proofs until temporal logic reasoning is implemented in $TLAPS$.

THEOREM $Spec \Rightarrow P!Spec$
⟨1⟩1. $Init \Rightarrow P!Init$
⟨2⟩ SUFFICES ASSUME $Init$
PROVE $P!Init$
OBVIOUS
⟨2⟩1. $MaxBallot(\{\}) = -1$
BY $MaxBallotProp$, $EmptySetFinite$
⟨2⟩2. $P!Init!1$
BY ⟨2⟩1 DEF $Init$, $PmaxBal$, $1bOr2bMsgs$
⟨2⟩3. $P!Init!2 \land P!Init!3$
BY DEF $Init$, $None$, $P!None$
⟨2⟩4. $msgs = \{\}$
⟨3⟩1. $msgsOfType(\text{“1a”}) = \{\} \land acceptorMsgsOfType(\text{“2b”}) = \{\}$
BY DEF $Init$, $msgsOfType$, $acceptorMsgsOfType$
⟨3⟩2. $1bmsgs = \{\} \land 1cmsgs = \{\}$
BY DEF $Init$, $msgsOfType$, $acceptorMsgsOfType$, $1bmsgs$, $1cmsgs$
⟨3⟩3. $2amsgs = \{\}$
⟨4⟩1. $\{\} \notin Quorum$

3126 BY BQA DEF $Quorum$
3127 $\langle 4\rangle 2$. $acceptorMsgsOfType(\text{“2av”}) = \{\}$
3128 BY DEF $Init$, $msgsOfType$, $acceptorMsgsOfType$
3129 $\langle 4\rangle$ QED
3130 BY $\langle 4\rangle 1$, $\langle 4\rangle 2$ DEF $2amsgs$
3131 $\langle 3\rangle 4$. QED
3132 BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$ DEF $msgs$
3133 $\langle 2\rangle 5$. QED
3134 BY $\langle 2\rangle 2$, $\langle 2\rangle 3$, $\langle 2\rangle 4$ DEF $P!Init$

3136 $\langle 1\rangle 2$. $Inv \wedge Inv' \wedge [Next]_{vars} \Rightarrow [P!Next]_P!vars$
3137 $\langle 2\rangle$ $InvP \triangleq Inv'$ We probably don't need to assume Inv'
3138 $\langle 2\rangle 1$. UNCHANGED $vars \Rightarrow$ UNCHANGED $P!vars$
3139 $\langle 3\rangle$ SUFFICES ASSUME UNCHANGED $vars$
3140 PROVE UNCHANGED $P!vars$
3141 OBVIOUS
3142 $\langle 3\rangle 1$. UNCHANGED $\langle maxVBal, maxVVal\rangle$
3143 BY DEF $vars$
3144 $\langle 3\rangle 2$. UNCHANGED $PmaxBal$
3145 BY DEF $vars$, $PmaxBal$, $1bOr2bMsgs$
3146 $\langle 3\rangle 3$. UNCHANGED $msgs$
3147 $\langle 4\rangle$ USE DEF $vars$
3148 $\langle 4\rangle 1$. UNCHANGED $\langle msgsOfType(\text{“1a”}), acceptorMsgsOfType(\text{“2b”}), 1bmsgs\rangle$
3149 BY DEF $msgsOfType$, $acceptorMsgsOfType$, $1bmsgs$, $2amsgs$
3150 $\langle 4\rangle 2$. UNCHANGED $1cmsgs$
3151 BY DEF $1cmsgs$, $msgsOfType$, $KnowsSafeAt$
3152 $\langle 4\rangle 3$. UNCHANGED $2amsgs$
3153 BY DEF $2amsgs$, $msgsOfType$, $acceptorMsgsOfType$
3154 $\langle 4\rangle 4$. QED
3155 BY $\langle 4\rangle 1$, $\langle 4\rangle 2$, $\langle 4\rangle 3$ DEF $msgs$
3156 $\langle 3\rangle 4$. QED
3157 BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$ DEF $P!vars$
3158 $\langle 2\rangle$ SUFFICES ASSUME Inv, $InvP$, $Next$
3159 PROVE $P!TLANext \vee P!vars' = P!vars$
3160 $\langle 3\rangle 1$. $Inv \wedge [Next]_{vars} \Rightarrow Inv \wedge [Next]_{vars}$
3161 BY DEF Inv
3162 $\langle 3\rangle 2$. UNCHANGED $vars \Rightarrow$ UNCHANGED $P!vars$
3163 $\langle 4\rangle$ HAVE UNCHANGED $vars$
3164 $\langle 4\rangle$ USE DEF $vars$
3165 $\langle 4\rangle 1$. UNCHANGED $PmaxBal$
3166 BY DEF $PmaxBal$, $1bOr2bMsgs$
3167 $\langle 4\rangle 2$. UNCHANGED $msgs$
3168 $\langle 5\rangle 1$. $\wedge$ UNCHANGED $msgsOfType(\text{“1a”})$
3169 $\wedge$ UNCHANGED $acceptorMsgsOfType(\text{“2b”})$
3170 BY DEF $msgsOfType$, $acceptorMsgsOfType$

⟨5⟩2. UNCHANGED $1bmsgs$
BY DEF $1bmsgs$, $msgsOfType$, $acceptorMsgsOfType$
⟨5⟩3. UNCHANGED $1cmsgs$
BY DEF $1cmsgs$, $msgsOfType$, $KnowsSafeAt$
⟨5⟩4. UNCHANGED $2amsgs$
BY DEF $2amsgs$, $msgsOfType$, $acceptorMsgsOfType$
⟨5⟩5. QED
BY ⟨5⟩1, ⟨5⟩2, ⟨5⟩3, ⟨5⟩4 DEF $msgs$
⟨4⟩3. QED
BY ⟨4⟩1, ⟨4⟩2 DEF $P!vars$, $PmaxBal$, $1bOr2bMsgs$
⟨3⟩3. $Inv \land P!TLANext \Rightarrow P!Next$
BY ⟨3⟩1, ⟨3⟩2, $PNextDef$
DEF Inv, $P!ProcSet$, $P!Init$, $Ballot$, $P!Ballot$
⟨3⟩ QED
BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3, $P!NextDef$, $NextDef$ DEF Inv
⟨2⟩ HIDE DEF $InvP$
⟨2⟩2. $\forall\, a \in Acceptor : PmaxBal[a] \in Ballot \cup \{-1\}$
⟨3⟩ SUFFICES ASSUME NEW $a \in Acceptor$
PROVE $PmaxBal[a] \in Ballot \cup \{-1\}$
OBVIOUS
⟨3⟩ DEFINE $S \triangleq \{m.bal : m \in \{ma \in bmsgs :$
$\land\, ma.type \in \{\text{“1b”}, \text{“2b”}\}$
$\land\, ma.acc = a\}\}$
⟨3⟩1. $PmaxBal[a] = MaxBallot(S)$
BY DEF $PmaxBal$, $1bOr2bMsgs$
⟨3⟩2. $\land\, IsFiniteSet(S)$
$\land\, S \in$ SUBSET $Ballot$
BY $PMaxBalLemma3$ DEF Inv
⟨3⟩3.CASE $S = \{\}$
BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3, $MaxBallotProp$
⟨3⟩4.CASE $S \neq \{\}$
BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩4, $MaxBallotProp$
⟨3⟩5. QED
BY ⟨3⟩3, ⟨3⟩4
⟨2⟩3. ASSUME NEW $self \in Acceptor$, NEW $b \in Ballot$,
$Phase1b(self,\, b)$
PROVE $P!TLANext \lor P!vars' = P!vars$
⟨3⟩1. $\land\, P!sentMsgs(\text{“1a”},\, b) \neq \{\}$
$\land\, msgs' = msgs \cup \{[type \mapsto \text{“1b”},\, acc \mapsto self,\, bal \mapsto b,$
$mbal \mapsto maxVBal[self],\, mval \mapsto maxVVal[self]]\}$
⟨4⟩1.⟨3⟩1!2
BY ⟨2⟩3, $MsgsLemma$ DEF Inv
⟨4⟩2. PICK $m \in sentMsgs(\text{“1a”},\, b) : m.type = \text{“1a”} \land m.bal = b$
BY ⟨2⟩3 DEF $Phase1b$, $sentMsgs$
⟨4⟩3. $m \in msgsOfType(\text{“1a”})$

BY $\langle 4\rangle 2$ DEF $sentMsgs$, $msgsOfType$
$\langle 4\rangle 4$. $m \in msgs$
BY $\langle 4\rangle 3$ DEF $msgs$
$\langle 4\rangle 5$. $m \in P!sentMsgs(\text{“1a”}, b)$
BY $\langle 4\rangle 4$, $\langle 4\rangle 2$ DEF $P!sentMsgs$
$\langle 4\rangle 6$. QED
BY $\langle 4\rangle 1$, $\langle 4\rangle 5$
$\langle 3\rangle 2$. UNCHANGED $\langle maxVBal, maxVVal\rangle$
BY $\langle 2\rangle 3$ DEF $Phase1b$
$\langle 3\rangle 3$. $\land\ b > PmaxBal[self]$
$\land\ PmaxBal' = [PmaxBal$ EXCEPT $![self] = b]$
$\langle 4\rangle 1$. $b > PmaxBal[self]$
$\langle 5\rangle 1$. $b > maxBal[self]$
BY $\langle 2\rangle 3$ DEF $Phase1b$
$\langle 5\rangle 2$. $maxBal[self] \in Ballot \cup \{-1\}$
BY DEF Inv, $TypeOK$
$\langle 5\rangle 3$. $\forall\, pmb,\ mb \in Ballot \cup \{-1\}:$
$b > mb \land pmb \leq mb \Rightarrow b > pmb$
BY $SimpleArithmetic$ DEF $Ballot$
$\langle 5\rangle 4$. QED
BY $\langle 5\rangle 1$, $\langle 5\rangle 2$, $\langle 5\rangle 3$, $PmaxBalLemma4$, $\langle 2\rangle 2$ DEF Inv
$\langle 4\rangle$ DEFINE $m \triangleq [type \mapsto \text{“1b”},\ bal \mapsto b,\ acc \mapsto self,$
$m2av \mapsto 2avSent[self],$
$mbal \mapsto maxVBal[self],\ mval \mapsto maxVVal[self]]$
$mA(a) \triangleq \{ma \in bmsgs :\ \land\ ma.type \in \{\text{“1b”}, \text{“2b”}\}$
$\land\ ma.acc = a\}$
$S(a) \triangleq \{ma.bal : ma \in mA(a)\}$
$\langle 4\rangle 2$. $bmsgs' = bmsgs \cup \{m\}$
BY $\langle 2\rangle 3$ DEF $Phase1b$
$\langle 4\rangle 3$. $mA(self)' = mA(self) \cup \{m\}$
$\langle 5\rangle 1$. $mA(self)' = mA(self) \cup \{ma \in \{m\} :\ \land\ ma.type \in \{\text{“1b”}, \text{“2b”}\}$
$\land\ ma.acc = self\}$
BY $\langle 4\rangle 2$
$\langle 5\rangle 2$. $m.type = \text{“1b”} \land m.acc = self$
OBVIOUS
$\langle 5\rangle 3$. $\forall\, ma \in \{m\} : ma.type = \text{“1b”} \land ma.acc = self$
BY $\langle 5\rangle 2$
$\langle 5\rangle 4$. QED
BY $\langle 5\rangle 1$, $\langle 5\rangle 3$
$\langle 4\rangle 4$. $\land\ PmaxBal = [a \in Acceptor \mapsto MaxBallot(S(a))]$
$\land\ PmaxBal' = [a \in Acceptor \mapsto MaxBallot(S(a))']$
BY DEF $PmaxBal$, $1bOr2bMsgs$
$\langle 4\rangle$ HIDE DEF mA
$\langle 4\rangle 5$. $S(self)' = S(self) \cup \{b\}$
BY $\langle 4\rangle 3$, $m.bal = b$

⟨4⟩6. $MaxBallot(S(self) \cup \{b\}) = b$
⟨5⟩1. $b > MaxBallot(S(self))$
BY ⟨4⟩1, ⟨4⟩4
⟨5⟩ DEFINE $SS \triangleq S(self) \cup \{b\}$
⟨5⟩2. $b \in SS$
OBVIOUS
⟨5⟩3. $IsFiniteSet(S(self))$
⟨6⟩1. $IsFiniteSet(1bOr2bMsgs)$
BY DEF Inv, $bmsgsFinite$
⟨6⟩2. $IsFiniteSet(mA(self))$
BY ⟨6⟩1, $SubsetOfFiniteSetFinite$ DEF mA, $1bOr2bMsgs$
⟨6⟩ DEFINE $f[ma \in mA(self)] \triangleq ma.bal$
⟨6⟩3. $S(self) = \{f[ma] : ma \in mA(self)\}$
OBVIOUS
⟨6⟩ HIDE DEF f
⟨6⟩4. QED
BY ⟨6⟩2, ⟨6⟩3, $ImageOfFiniteSetFinite$
⟨5⟩4. $IsFiniteSet(SS)$
BY ⟨5⟩3, $SingletonSetFinite$, $UnionOfFiniteSetsFinite$
⟨5⟩5. $SS \subseteq Ballot \cup \{-1\}$
⟨6⟩1. ASSUME NEW $mm \in mA(self)$
PROVE $mm.bal \in Ballot \cup \{-1\}$
⟨7⟩1. $mm \in bmsgs \wedge mm.type \in \{$"1b", "2b"$\}$
BY DEF mA
⟨7⟩2.CASE $mm.type =$ "1b"
BY ⟨7⟩1, ⟨7⟩2, $BMessageLemma$ DEF Inv, $TypeOK$, $1bMessage$
⟨7⟩3.CASE $mm.type =$ "2b"
BY ⟨7⟩1, ⟨7⟩2, $BMessageLemma$ DEF Inv, $TypeOK$, $2bMessage$
⟨7⟩4. QED
BY ⟨7⟩1, ⟨7⟩2, ⟨7⟩3
⟨6⟩2. QED
BY ⟨6⟩1
⟨5⟩6. $\forall x \in SS : b \geq x$
⟨6⟩1. $b \geq b$
BY $SimpleArithmetic$ DEF $Ballot$
⟨6⟩2. SUFFICES ASSUME NEW $x \in S(self)$
PROVE $b \geq x$
BY ⟨6⟩1
⟨6⟩3. $S(self) \neq \{\}$
OBVIOUS
⟨6⟩ HIDE DEF S
⟨6⟩4. $\wedge MaxBallot(S(self)) \in S(self)$
$\wedge MaxBallot(S(self)) \geq x$
BY ⟨5⟩3, ⟨5⟩5, ⟨6⟩3, $MaxBallotProp$, $IsFiniteSet(S(self))$
⟨6⟩5. $b > MaxBallot(S(self))$

BY $\langle 5\rangle 1$, $\langle 4\rangle 4$

$\langle 6\rangle 6$. $\wedge\ MaxBallot(S(self)) \in Ballot \cup \{-1\}$
$\wedge\ x \in Ballot \cup \{-1\}$

BY $\langle 5\rangle 5$, $\langle 6\rangle 4$

$\langle 6\rangle 7$. $\forall\, mbs \in Ballot \cup \{-1\} :$
$b > mbs \wedge mbs \geq x \Rightarrow b \geq x$

BY $\langle 6\rangle 6$, *SimpleArithmetic* DEF *Ballot*

$\langle 6\rangle 8$. QED

BY $\langle 5\rangle 1$, $\langle 6\rangle 4$, $\langle 6\rangle 6$, $\langle 6\rangle 7$

$\langle 5\rangle 7$. QED

$\langle 6\rangle$ HIDE DEF *SS*

$\langle 6\rangle 1$. $b = MaxBallot(SS)$

BY $\langle 5\rangle 2$, $\langle 5\rangle 4$, $\langle 5\rangle 5$, $\langle 5\rangle 6$, *MaxBallotLemma*1

$\langle 6\rangle 2$. QED

BY $\langle 6\rangle 1$ DEF *SS*

$\langle 4\rangle 7$. $\forall\, a \in Acceptor : a \neq self \Rightarrow S(a)' = S(a)$

$\langle 5\rangle$ USE DEF *mA*

$\langle 5\rangle 1$. SUFFICES ASSUME NEW $a \in Acceptor$, $a \neq self$
PROVE $S(a)' = S(a)$

OBVIOUS

$\langle 5\rangle 2$. $m.acc \neq a$

BY $\langle 5\rangle 1$

$\langle 5\rangle 3$. $mA(a)' = mA(a)$

$\langle 6\rangle 1$. ASSUME NEW $mm \in mA(a)'$
PROVE $mm \in mA(a)$

BY $\langle 4\rangle 2$, $\langle 5\rangle 2$, $mm.acc = a$ $\langle 7\rangle 3$

$\langle 6\rangle 2$. ASSUME NEW $mm \in mA(a)$
PROVE $mm \in mA(a)'$

BY $\langle 4\rangle 2$

$\langle 6\rangle 3$. QED

BY $\langle 6\rangle 1$, $\langle 6\rangle 2$

$\langle 5\rangle 4$. QED

BY $\langle 5\rangle 3$

$\langle 4\rangle 8$. $PmaxBal' = [PmaxBal$ EXCEPT $![self] = b]$

$\langle 5\rangle 1$. SUFFICES ASSUME NEW $a \in Acceptor$
PROVE $PmaxBal'[a] =$ IF $a = self$ THEN b
ELSE $PmaxBal[a]$

BY DEF *PmaxBal*, *1bOr2bMsgs*

$\langle 5\rangle 2$. $(a \neq self) \Rightarrow MaxBallot(S(a))' = MaxBallot(S(a))$

BY $\langle 4\rangle 7$

$\langle 5\rangle 3$. $(a \neq self) \Rightarrow PmaxBal'[a] = PmaxBal[a]$

BY $\langle 4\rangle 4$, $\langle 4\rangle 7$, $\langle 5\rangle 2$ DEF *PmaxBal*, *1bOr2bMsgs*

$\langle 5\rangle$ QED

BY $\langle 5\rangle 2$, $\langle 4\rangle 4$, $\langle 4\rangle 5$, $\langle 4\rangle 6$

3351 ⟨4⟩9. QED
3352 BY ⟨4⟩1, ⟨4⟩8
3353 ⟨3⟩ DEFINE $m \triangleq [type \mapsto$ "1b", $acc \mapsto self$, $bal \mapsto b$,
3354 $mbal \mapsto maxVBal[self]$, $mval \mapsto maxVVal[self]]$
3355 ⟨3⟩4. QED
3356 BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3 DEF $P!TLANext$, $P!Ballot$, $Ballot$, $P!Phase1b$
3357 ⟨2⟩4. ASSUME NEW $self \in Acceptor$, NEW $b \in Ballot$,
3358 $Phase2av(self, b)$
3359 PROVE $P!TLANext \lor P!vars' = P!vars$
3360 ⟨3⟩1. $PmaxBal' = PmaxBal$
3361 ⟨4⟩ DEFINE $mm(m) \triangleq [type \mapsto$ "2av", $bal \mapsto b$,
3362 $val \mapsto m.val$, $acc \mapsto self]$
3363 ⟨4⟩1. PICK $m : bmsgs' = bmsgs \cup \{mm(m)\}$
3364 BY ⟨2⟩4 DEF $Phase2av$
3365 ⟨4⟩2. $mm(m).type =$ "2av"
3366 OBVIOUS
3367 ⟨4⟩ QED
3368 BY ⟨4⟩1, ⟨4⟩2, $PmaxBalLemma1$
3369 ⟨3⟩2.CASE $msgs' = msgs$
3370 BY ⟨3⟩1, ⟨3⟩2, ⟨2⟩4 DEF $Phase2av$, $P!vars$
3371 ⟨3⟩3.CASE $\land msgs' \neq msgs$
3372 $\land \exists v \in Value :$
3373 $\land [type \mapsto$ "1c", $bal \mapsto b$, $val \mapsto v] \in msgs$
3374 $\land msgs' = msgs \cup \{[type \mapsto$ "2a", $bal \mapsto b$, $val \mapsto v]\}$
3375 ⟨4⟩1. PICK $v \in Value :$
3376 $\land [type \mapsto$ "1c", $bal \mapsto b$, $val \mapsto v] \in msgs$
3377 $\land msgs' = msgs \cup \{[type \mapsto$ "2a", $bal \mapsto b$, $val \mapsto v]\}$
3378 BY ⟨3⟩3
3379 ⟨4⟩2. $P!sentMsgs($"2a", $b) = \{\}$
3380 ⟨5⟩ DEFINE $m2a \triangleq [type \mapsto$ "2a", $bal \mapsto b$, $val \mapsto v]$
3382 ⟨5⟩1. SUFFICES ASSUME NEW $m \in P!sentMsgs($"2a", $b)$
3383 PROVE $m = m2a$
3384 ⟨6⟩1. $\forall m \in P!sentMsgs($"2a", $b) : m \in msgs$
3385 BY DEF $P!sentMsgs$
3386 ⟨6⟩2. QED
3387 BY ⟨3⟩3, ⟨4⟩1, ⟨6⟩1
3388 ⟨5⟩2. $\land m \in 2amsgs$
3389 $\land m.type =$ "2a"
3390 $\land m.bal = b$
3391 BY $MsgsTypeLemma$ DEF $P!sentMsgs$
3392 ⟨5⟩3. PICK $Q \in Quorum :$
3393 $\forall a \in Q \quad :$
3394 $\exists mav \in acceptorMsgsOfType($"2av"$) :$
3395 $\land mav.acc = a$

$\land\ mav.bal = b$
$\land\ mav.val = m.val$
BY $\langle 5\rangle 2$ DEF $2amsgs$
$\langle 5\rangle 4$. PICK $Q2 \in Quorum$:
$\forall\, a \in Q2$:
$\exists\, m2av \in acceptorMsgsOfType(\text{“2av”})'$:
$\land\ m2av.acc = a$
$\land\ m2av.bal = b$
$\land\ m2av.val = v$
$\langle 6\rangle 1$. $\land\ m2a.type = \text{“2a”}$
$\land\ m2a.bal = b$
$\land\ m2a.val = v$
OBVIOUS
$\langle 6\rangle 2$. $m2a \in 2amsgs'$
BY $\langle 4\rangle 1$, $\langle 6\rangle 1$, $m2a \in msgs'$, $MsgsTypeLemmaPrime$
$\langle 6\rangle$ HIDE DEF $m2a$
$\langle 6\rangle 3$. QED
BY $\langle 6\rangle 1$, $\langle 6\rangle 2$ DEF $2amsgs$
$\langle 5\rangle 5$. PICK $a \in Q \cap Q2 : a \in Acceptor$
BY $QuorumTheorem$
$\langle 5\rangle 6$. PICK $mav \in acceptorMsgsOfType(\text{“2av”})$:
$\land\ mav.acc = a$
$\land\ mav.bal = b$
$\land\ mav.val = m.val$
BY $\langle 5\rangle 3$, $\langle 5\rangle 5$
$\langle 5\rangle 7$. PICK $m2av \in acceptorMsgsOfType(\text{“2av”})'$:
$\land\ m2av.acc = a$
$\land\ m2av.bal = b$
$\land\ m2av.val = v$
BY $\langle 5\rangle 4$, $\langle 5\rangle 5$
$\langle 5\rangle 8$. $mav \in acceptorMsgsOfType(\text{“2av”})'$
BY $\langle 2\rangle 4$ DEF $acceptorMsgsOfType$, $msgsOfType$, $Phase2av$
$\langle 5\rangle 9$. $m2av = mav$
BY $\langle 5\rangle 5$, $\langle 5\rangle 6$, $\langle 5\rangle 7$, $\langle 5\rangle 8$ DEF $2avInv1$, $InvP$, Inv, $acceptorMsgsOfType$, $msgsOfType$
$\langle 5\rangle 10$. $m = [type \mapsto \text{“2a”},\ bal \mapsto b,\ val \mapsto m.val]$
BY $\langle 5\rangle 2$ DEF $2amsgs$
$\langle 5\rangle 11$. QED
BY $\langle 5\rangle 6$, $\langle 5\rangle 7$, $\langle 5\rangle 9$, $\langle 5\rangle 10$
$\langle 4\rangle 3$. $P!Phase2a(b,\ v)$
BY $\langle 2\rangle 4$, $\langle 3\rangle 1$, $\langle 4\rangle 1$, $\langle 4\rangle 2$ DEF $P!Phase2a$, $Phase2av$
$\langle 4\rangle 4$. QED
BY $\langle 4\rangle 3$ DEF $P!TLANext$, $Ballot$, $P!Ballot$
$\langle 3\rangle 4$. QED
BY $\langle 3\rangle 2$, $\langle 3\rangle 3$, $MsgsLemma$, $\langle 2\rangle 4$ DEF Inv
$\langle 2\rangle 5$. ASSUME NEW $self \in Acceptor$, NEW $b \in Ballot$,

$Phase2b(self,\ b)$
PROVE $P!TLANext \lor P!vars' = P!vars$
⟨3⟩ USE ⟨2⟩5
⟨3⟩1. $b \geq PmaxBal[self]$
⟨4⟩1. $PmaxBal[self] \leq maxBal[self]$
BY $PmaxBalLemma4$ DEF Inv
⟨4⟩2. $maxBal[self] \leq b$
BY DEF $Phase2b$
⟨4⟩3. $\forall\, pmb,\ mb \in Ballot \cup \{-1\} :$
$pmb \leq mb \land mb \leq b \Rightarrow\ b \geq pmb$
BY $SimpleArithmetic$ DEF $Ballot$
⟨4⟩4. QED
BY ⟨4⟩1, ⟨4⟩2, ⟨4⟩3, $PmaxBalLemma5$ DEF Inv, $TypeOK$
⟨3⟩2. PICK $v \in Value$:
$\land \exists\, Q \in ByzQuorum :$
$\forall\, a \in Q :$
$\exists\, m \in sentMsgs(\text{“2av”},\ b) :\ \land\ m.val = v$
$\land\ m.acc = a$
$\land\ msgs' = msgs \cup$
$\{[type \mapsto \text{“2b”},\ acc \mapsto self,\ bal \mapsto b,\ val \mapsto v]\}$
$\land\ bmsgs' = bmsgs \cup$
$\{[type \mapsto \text{“2b”},\ acc \mapsto self,\ bal \mapsto b,\ val \mapsto v]\}$
$\land\ maxVVal' = [maxVVal \text{ EXCEPT } ![self] = v]$
⟨4⟩1. $MsgsLemma!2!3$
BY $MsgsLemma$ DEF Inv
⟨4⟩2. $MsgsLemma!2!3!(self,\ b)!2$
BY ⟨4⟩1
⟨4⟩3. QED
BY ⟨4⟩2
⟨3⟩ DEFINE $m \triangleq [type \mapsto \text{“2a”},\ bal \mapsto b,\ val \mapsto v]$
$m2b \triangleq [type \mapsto \text{“2b”},\ acc \mapsto self,\ bal \mapsto b,\ val \mapsto v]$
⟨3⟩3. $m \in P!sentMsgs(\text{“2a”},\ b)$
⟨4⟩1. $m \in 2amsgs$
⟨5⟩1. $m \in [type : \{\text{“2a”}\},\ bal : Ballot,\ val : Value]$
OBVIOUS
⟨5⟩2. PICK $Q \in Quorum :$
$\forall\, a \in Q\quad :$
$\exists\, mm \in sentMsgs(\text{“2av”},\ b)\ :\ \land\ mm.val = v$
$\land\ mm.acc = a$
⟨6⟩1. $\forall\, BQ \in ByzQuorum : \exists\, Q \in Quorum : Q \subseteq BQ$
BY DEF $Quorum$
⟨6⟩2. QED
BY ⟨3⟩2, ⟨6⟩1 * need to first pick BQ in $ByzQuorum$
* and then let $Q \triangleq BQ \cap Acceptor$
⟨5⟩3. ASSUME NEW $a \in Q$

PROVE $\exists\, m2av \in acceptorMsgsOfType(\text{“2av”}) :$
$\wedge\, m2av.acc = a$
$\wedge\, m2av.bal = m.bal$
$\wedge\, m2av.val = m.val$

$\langle 6\rangle 1$. PICK $m2av \in sentMsgs(\text{“2av”},\, b) :\ \wedge\, m2av.val = v$
$\wedge\, m2av.acc = a$
BY $\langle 5\rangle 2$

$\langle 6\rangle 2$. $m2av \in acceptorMsgsOfType(\text{“2av”})$
BY $\langle 6\rangle 1$ DEF $sentMsgs$, $Quorum$, $acceptorMsgsOfType$, $msgsOfType$

$\langle 6\rangle$ WITNESS $m2av \in acceptorMsgsOfType(\text{“2av”})$

$\langle 6\rangle$ QED
BY $\langle 6\rangle 1$ DEF $sentMsgs$

$\langle 5\rangle$ QED
BY $\langle 5\rangle 1$, $\langle 5\rangle 3$ DEF $2amsgs$

$\langle 4\rangle 2$. QED
BY $\langle 4\rangle 1$ DEF $P!sentMsgs$, $msgs$

$\langle 3\rangle 4$. $PmaxBal' = [PmaxBal$ EXCEPT $![self] = b]$

$\langle 4\rangle 1$. ASSUME NEW $a \in Acceptor$,
$a \neq self$
PROVE $PmaxBal'[a] = PmaxBal[a]$

$\langle 5\rangle 1$. $bmsgs' = bmsgs \cup \{m2b\} \wedge m2b.acc = self$
BY $\langle 3\rangle 2$

$\langle 5\rangle 2$. QED
BY $\langle 4\rangle 1$, $\langle 5\rangle 1$, $PmaxBalLemma2$

$\langle 4\rangle 2$. $PmaxBal'[self] = b$

$\langle 5\rangle$ DEFINE $S \triangleq \{mm.bal : mm \in \{ma \in bmsgs :$
$\wedge\, ma.type \in \{\text{“1b”},\ \text{“2b”}\}$
$\wedge\, ma.acc = self\}\}$
$T \triangleq S \cup \{m2b.bal\}$

$\langle 5\rangle 1$. $IsFiniteSet(S) \wedge (S \in$ SUBSET $Ballot)$
BY $PMaxBalLemma3$ DEF Inv

$\langle 5\rangle 2$. $IsFiniteSet(T) \wedge (T \in$ SUBSET $Ballot)$
BY $\langle 5\rangle 1$, $OnePlusFinite$

$\langle 5\rangle 3$. $PmaxBal[self] = MaxBallot(S)$
BY DEF $PmaxBal$, $1bOr2bMsgs$

$\langle 5\rangle 4$. $PmaxBal'[self] = MaxBallot(T)$
BY $\langle 3\rangle 2$ DEF $PmaxBal$, $1bOr2bMsgs$

$\langle 5\rangle$ HIDE DEF S, T

$\langle 5\rangle 5$.CASE $S = \{\}$

$\langle 6\rangle 1$. $T = \{b\} \cup \{\}$
BY $\langle 5\rangle 5$ DEF T

$\langle 6\rangle 2$. $\wedge\, b \geq b$
$\wedge\, b \geq -1$
BY $SimpleArithmetic$ DEF $Ballot$

$\langle 6\rangle 3$. $MaxBallot(\{b\}) = b$

BY $\langle 6\rangle 2$, *SingletonSetFinite*, *MaxBallotLemma*1
$\langle 6\rangle 4$. $MaxBallot(\{\}) = -1$
BY *EmptySetFinite*, *MaxBallotProp*
$\langle 6\rangle$ QED
BY *SingletonSetFinite*, *EmptySetFinite*, $\langle 7\rangle 1$, $\langle 6\rangle 1$, $\langle 6\rangle 2$, $\langle 6\rangle 3$, $\langle 6\rangle 4$, *MaxBallotLemma*2, $\langle 5\rangle 4$
$\langle 5\rangle 6$.CASE $S \neq \{\}$
$\langle 6\rangle 1$. $\forall\, bb \in S : PmaxBal[self] \geq bb$
BY $\langle 5\rangle 1$, $\langle 5\rangle 3$, *MaxBallotProp*
$\langle 6\rangle 2$. ASSUME NEW $bb \in S$
PROVE $b \geq bb$
$\langle 7\rangle 1$. $\forall\, pmb \in Ballot \cup \{-1\} :$
$b \geq pmb \wedge pmb \geq bb \Rightarrow b \geq bb$
BY *SimpleArithmetic* DEF *Ballot*
$\langle 7\rangle 2$. QED
BY $\langle 6\rangle 1$, $\langle 7\rangle \langle 5\rangle 1$, $\langle 7\rangle 1$, $\langle 3\rangle 1$, *PmaxBalLemma*5 DEF *Inv*
$\langle 6\rangle 3$. $b \geq b$
BY *SimpleArithmetic* DEF *Ballot*
$\langle 6\rangle 4$. $\forall\, bb \in T : b \geq bb$
BY $\langle 6\rangle 2$, $\langle 6\rangle 3$, $m2b.bal = b$ DEF T
$\langle 6\rangle 5$. $b = MaxBallot(T)$
BY $\langle 5\rangle 2$, $\langle 6\rangle 4$, *MaxBallotLemma*1 DEF T
$\langle 6\rangle 6$. QED
BY $\langle 6\rangle 5$, $\langle 5\rangle 4$
$\langle 5\rangle 7$. QED
BY $\langle 5\rangle 5$, $\langle 5\rangle 6$
$\langle 4\rangle 3$. QED
BY $\langle 4\rangle 1$, $\langle 4\rangle 2$ DEF *PmaxBal*, $1bOr2bMsgs$
$\langle 3\rangle 5$. $\wedge\, maxVBal' = [maxVBal$ EXCEPT $![self] = b]$
$\wedge\, maxVVal' = [maxVVal$ EXCEPT $![self] = m.val]$
BY $\langle 3\rangle 2$, $m.val = v$ DEF *Phase*2*b*
$\langle 3\rangle 6$. QED
$\langle 4\rangle 1$. $P!Phase2b(self, b)$
$\langle 5\rangle 1$. $P!Phase2b(self, b)!2$
$\langle 6\rangle$ USE $\langle 3\rangle 3$
$\langle 6\rangle 1$. WITNESS $m \in P!sentMsgs(\text{“2a”}, b)$
$\langle 6\rangle 2$. QED
BY $\langle 3\rangle 4$, $\langle 3\rangle 5$, $\langle 3\rangle 2$
$\langle 5\rangle 2$. QED
BY $\langle 5\rangle 1$, $\langle 3\rangle 1$ DEF $P!Phase2b$
$\langle 4\rangle 2$. QED
BY $\langle 4\rangle 1$ DEF $P!TLANext$, *Ballot*, $P!Ballot$
$\langle 2\rangle 6$. ASSUME NEW $self \in Acceptor$, NEW $b \in Ballot$,
$LearnsSent(self, b)$
PROVE $P!TLANext \vee P!vars' = P!vars$
$\langle 3\rangle$ USE $LearnsSent(self, b)$

⟨3⟩1. PICK $SM \in$ SUBSET $\{m \in msgsOfType(\text{“1c”}) : m.bal = b\}$:
$msgs' = msgs \cup SM$
BY $MsgsLemma$ DEF Inv
⟨3⟩ DEFINE $S \triangleq \{m.val : m \in SM\}$
⟨3⟩2. $S \in$ SUBSET $Value$
⟨4⟩ SUFFICES ASSUME NEW $m \in SM$
PROVE $m.val \in Value$
OBVIOUS
⟨4⟩ QED
BY $m \in msgsOfType(\text{“1c”})$, $BMessageLemma$ DEF Inv, $TypeOK$, $msgsOfType$, $1cMessage$
⟨3⟩3. $msgs' = msgs \cup \{[type \mapsto \text{“1c”},\ bal \mapsto b,\ val \mapsto v] : v \in S\}$
⟨4⟩ SUFFICES ASSUME NEW m
PROVE $m \in SM \equiv \land m = [type \mapsto \text{“1c”},\ bal \mapsto b,\ val \mapsto m.val]$
$\land\ m.val \in S$
BY ⟨3⟩1 DEF $msgsOfType$
⟨4⟩1. ASSUME NEW $mm \in SM$
PROVE $\land mm = [type \mapsto \text{“1c”},\ bal \mapsto b,\ val \mapsto mm.val]$
$\land\ mm.val \in S$
⟨5⟩1. $\land mm \in bmsgs$
$\land\ mm.type = \text{“1c”}$
$\land\ mm.bal = b$
BY ⟨4⟩1 DEF $msgsOfType$
⟨5⟩2. $mm \in 1cMessage$
BY ⟨5⟩1, $BMessageLemma$ DEF Inv, $TypeOK$
⟨5⟩3. $mm = [type \mapsto mm.type,\ bal \mapsto mm.bal,\ val \mapsto mm.val]$
BY ONLY ⟨5⟩2 DEF $1cMessage$
⟨5⟩4. $mm.val \in S$
OBVIOUS
⟨5⟩5. QED
BY ⟨5⟩1, ⟨5⟩3, ⟨4⟩1,⟨5⟩4 DEF $1cMessage$
⟨4⟩2. ASSUME $\land m = [type \mapsto \text{“1c”},\ bal \mapsto b,\ val \mapsto m.val]$
$\land\ m.val \in S$
PROVE $m \in SM$
⟨5⟩1. PICK $mm \in SM : mm.val = m.val$
BY ⟨4⟩2
⟨5⟩2. $mm = [type \mapsto \text{“1c”},\ bal \mapsto b,\ val \mapsto mm.val]$
BY ⟨4⟩1
⟨5⟩3. QED
BY ⟨4⟩2, ⟨5⟩1, ⟨5⟩2
⟨4⟩3. QED
BY ⟨4⟩1, ⟨4⟩2
⟨3⟩4. ASSUME NEW $v \in S$
PROVE $\exists\, Q \in Quorum : P!ShowsSafeAt(Q,\ b,\ v)$
⟨4⟩1. ASSUME NEW $ac \in Acceptor$,
$KnowsSafeAt(ac,\ b,\ v)'$

PROVE $\exists\, Q \in Quorum : P!ShowsSafeAt(Q,\, b,\, v)$

$\langle 5\rangle 1.\ bmsgs' = bmsgs$
BY DEF $LearnsSent$

$\langle 5\rangle$ DEFINE $Q(BQ) \triangleq BQ \cap Acceptor$
$SS \triangleq \{m \in knowsSent'[ac] : m.bal = b\}$
$SQ(BQ) \triangleq \{1bRestrict(mm) :$
$mm \in \{m \in SS : m.acc \in Q(BQ)\}\}$
$Q1b(BQ) \triangleq \{m \in P!sentMsgs(\text{“1b”},\, b) : m.acc \in Q(BQ)\}$

$\langle 5\rangle 2.$ ASSUME NEW $BQ \in ByzQuorum$,
$\forall\, a \in BQ : \exists\, m \in SS : m.acc = a$
PROVE $SQ(BQ) = Q1b(BQ)$

$\langle 6\rangle 1.$ ASSUME NEW $m \in P!sentMsgs(\text{“1b”},\, b)$,
$m.acc \in Q(BQ)$
PROVE $m \in SQ(BQ)$

$\langle 7\rangle 1. \land m \in 1bmsgs$
$\land m.type = \text{“1b”}$
$\land m.bal = b$
BY $MsgsTypeLemma$ DEF $P!sentMsgs$, $msgs$

$\langle 7\rangle 2.$ PICK $m1 \in bmsgs : \land m1.type = \text{“1b”}$
$\land m1.acc \in Acceptor$
$\land m = 1bRestrict(m1)$
BY $\langle 7\rangle 1$ DEF $1bmsgs$, $acceptorMsgsOfType$, $msgsOfType$

$\langle 7\rangle 3.\ m1.bal = b$
BY $\langle 7\rangle 1$, $\langle 7\rangle 2$ DEF $1bRestrict$

$\langle 7\rangle 4.$ PICK $m2 \in knowsSent[ac]' :$
$\land m2.bal = b$
$\land m2.acc = m.acc$
BY $\langle 5\rangle 2$, $\langle 6\rangle 1$

$\langle 7\rangle 5.\ m2 \in bmsgs \land m2.type = \text{“1b”}$
BY $\langle 5\rangle 1$ DEF $InvP$, Inv, $knowsSentInv$, $msgsOfType$

$\langle 7\rangle 6. \land m1.acc = m.acc$
$\land m1.bal = b$
BY $\langle 7\rangle 1$, $\langle 7\rangle 2$ DEF $1bRestrict$

$\langle 7\rangle 7.\ m1 = m2$
BY $\langle 7\rangle 2$, $\langle 7\rangle 4$, $\langle 7\rangle 5$, $\langle 7\rangle 6$, $\langle 7\rangle 7$ DEF Inv, $1bInv2$

$\langle 7\rangle 8.\ m2 \in SS \land m2.acc \in Q(BQ)$
BY $\langle 7\rangle 7$, $\langle 6\rangle 1$, $\langle 7\rangle 2$, $\langle 7\rangle 4$

$\langle 7\rangle 9.$ QED
BY $\langle 7\rangle 8$, $\langle 7\rangle 7$, $\langle 7\rangle 2$

$\langle 6\rangle 2.$ ASSUME NEW $m \in SS$,
$m.acc \in Q(BQ)$
PROVE $1bRestrict(m) \in Q1b(BQ)$

$\langle 7\rangle 1. \land m \in bmsgs$
$\land m.bal = b$
$\land m.type = \text{“1b”}$

BY $\langle 5\rangle 1$ DEF $InvP$, Inv, $knowsSentInv$, $msgsOfType$

$\langle 7\rangle 2.\ m \in acceptorMsgsOfType(\text{“1b"})$

BY $\langle 7\rangle 1$, $\langle 6\rangle 2$ DEF $acceptorMsgsOfType$, $msgsOfType$

$\langle 7\rangle 3.\ 1bRestrict(m) \in msgs$

BY $\langle 7\rangle 2$ DEF $msgs$, $1bmsgs$

$\langle 7\rangle 4.$ QED

BY $\langle 6\rangle 2$, $\langle 7\rangle 1$, $\langle 7\rangle 3$ DEF $P!sentMsgs$, $1bRestrict$

$\langle 6\rangle 3.$ QED

BY $\langle 6\rangle 1$, $\langle 6\rangle 2$ DEF $Q1b$, SQ

$\langle 5\rangle 3.$CASE $KnowsSafeAt(ac,\ b,\ v)!1!1'$

$\langle 6\rangle 1.$ PICK $BQ \in ByzQuorum : KnowsSafeAt(ac,\ b,\ v)!1!1!(BQ)'$

OBVIOUS

$\langle 6\rangle 2.\ Q(BQ) \in Quorum$

BY DEF $Quorum$

$\langle 6\rangle 3.\ \forall\, a \in Q(BQ) : \exists\, m \in SS : \land\, m.acc = a$
$\land\, m.mbal = -1$

BY $\langle 6\rangle 1$

$\langle 6\rangle 4.\ \forall\, a \in Q(BQ) : \exists\, m \in SQ(BQ) : \land\, m.acc = a$
$\land\, m.mbal = -1$

$\langle 7\rangle$ HIDE DEF SS

$\langle 7\rangle$ TAKE $a \in Q(BQ)$

$\langle 7\rangle 1.$ PICK $m \in SS : m.acc = a \land m.mbal = -1$

BY $\langle 6\rangle 3$

$\langle 7\rangle 2.\ 1bRestrict(m) \in SQ(BQ)$

BY $\langle 7\rangle 1$

$\langle 7\rangle$ WITNESS $1bRestrict(m) \in SQ(BQ)$

$\langle 7\rangle 3.$ QED

BY $\langle 7\rangle 1$ DEF $1bRestrict$

$\langle 6\rangle 5.\ \forall\, m \in SQ(BQ) : m.mbal = -1$

$\langle 7\rangle$ TAKE $mm \in SQ(BQ)$

$\langle 7\rangle 1.$ PICK $m \in SS : \land\, m \in knowsSent[ac]'$
$\land\, m.bal = b$
$\land\, m.acc \in Q(BQ)$
$\land\, mm = 1bRestrict(m)$

OBVIOUS

$\langle 7\rangle 2.$ PICK $mm1 \in SQ(BQ) : \land\, mm1.acc = m.acc$
$\land\, mm1.mbal = -1$

BY $\langle 7\rangle 1$, $\langle 6\rangle 4$

$\langle 7\rangle 3.$ PICK $m1 \in SS : \land\, m1 \in knowsSent[ac]'$
$\land\, m1.bal = b$
$\land\, m1.acc \in Q(BQ)$
$\land\, mm1 = 1bRestrict(m1)$

OBVIOUS

$\langle 7\rangle 4.\ m.acc \in Acceptor$

BY $\langle 7\rangle 1$

$\langle 7\rangle 5.\ \land\ m \in bmsgs \land m1 \in bmsgs$
$\qquad\qquad \land\ m.type = \text{“1b”} \land m1.type = \text{“1b”}$
BY $\langle 5\rangle 1$, $\langle 7\rangle 1$, $\langle 7\rangle 3$ DEF $InvP$, Inv, $knowsSentInv$, $msgsOfType$
$\langle 7\rangle 6.\ m.acc = m1.acc$
BY $\langle 7\rangle 2$, $\langle 7\rangle 3$ DEF $1bRestrict$
$\langle 7\rangle 7.\ m = m1$
BY $\langle 7\rangle 5$, $\langle 7\rangle 4$, $\langle 7\rangle 1$, $\langle 7\rangle 3$, $\langle 7\rangle 6$ DEF Inv, $1bInv2$
$\langle 7\rangle 8.\ m.mbal = -1$
BY $\langle 7\rangle 7$, $\langle 7\rangle 2$, $\langle 7\rangle 3$ DEF $1bRestrict$
$\langle 7\rangle 9.$ QED
BY $\langle 7\rangle 8$, $\langle 7\rangle 1$ DEF $1bRestrict$
$\langle 6\rangle 6.\ SQ(BQ) = Q1b(BQ)$
BY $\langle 5\rangle 2$, $\langle 6\rangle 1$
$\langle 6\rangle$ HIDE DEF SS, Q, SQ
$\langle 6\rangle$ WITNESS $Q(BQ) \in Quorum$
$\langle 6\rangle 7.$ QED
BY $\langle 6\rangle 4$, $\langle 6\rangle 5$, $\langle 6\rangle 6$ DEF $P!ShowsSafeAt$
$\langle 5\rangle 4.$CASE $KnowsSafeAt(ac,\ b,\ v)!1!2'$
$\langle 6\rangle 1.$ PICK $c \in 0\,..\,(b-1) : KnowsSafeAt(ac,\ b,\ v)!1!2!(c)'$
BY $\langle 5\rangle 4$
$\langle 6\rangle 2.$ PICK $BQ \in ByzQuorum :$
$\qquad \forall\, a \in BQ : \exists\, m \in SS :\ \land\ m.acc = a$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land\ m.mbal \leq c$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land\ (m.mbal = c) \Rightarrow (m.mval = v)$
BY $\langle 6\rangle 1$
$\langle 6\rangle 3.\ SQ(BQ) = Q1b(BQ)$
BY $\langle 6\rangle 2$, $\langle 5\rangle 2$
$\langle 6\rangle 4.\ P!ShowsSafeAt(Q(BQ),\ b,\ v)!1!1$
$\langle 7\rangle 1.$ SUFFICES ASSUME NEW $a \in Q(BQ)$
$\qquad\qquad$ PROVE $\exists\, m \in Q1b(BQ) : m.acc = a$
OBVIOUS
$\langle 7\rangle 2.$ PICK $m \in SS : m.acc = a$
BY $\langle 6\rangle 2$
$\langle 7\rangle 3.\ 1bRestrict(m) \in SQ(BQ)$
BY $\langle 7\rangle 2$
$\langle 7\rangle 4.\ 1bRestrict(m).acc = a$
BY $\langle 7\rangle 2$ DEF $1bRestrict$
$\langle 7\rangle 5.$ QED
BY $\langle 6\rangle 3$, $\langle 7\rangle 3$, $\langle 7\rangle 4$
$\langle 6\rangle 5.$ PICK $m1c \in msgs :$
$\qquad \land\ m1c = [type \mapsto \text{“1c”},\ bal \mapsto m1c.bal,\ val \mapsto v]$
$\qquad \land\ m1c.bal \geq c$
$\qquad \land\ m1c.bal \in Ballot$
$\langle 7\rangle 1.$ PICK $WQ \in WeakQuorum :$
$\qquad \forall\, a \in WQ : \exists\, m \in SS :\ \land\ m.acc = a$

$\land \exists\, r \in m.m2av :$
$\land\, r.bal \geq c$
$\land\, r.val = v$

BY $\langle 6\rangle 1$

$\langle 7\rangle 2$. PICK $a \in WQ$, $m \in SS$:
$\land\, a \in Acceptor$
$\land\, m.acc = a$
$\land \exists\, r \in m.m2av : \land\, r.bal \geq c$
$\land\, r.val = v$

$\langle 8\rangle 1$. PICK $a \in WQ : a \in Acceptor$
BY BQA

$\langle 8\rangle 2$. $\land\, a \in Acceptor$
$\land\, \langle 7\rangle 1!(WQ)!(a)$
BY $\langle 7\rangle 1$, $\langle 8\rangle 1$

$\langle 8\rangle 3$. QED
BY $\langle 8\rangle 2$

$\langle 7\rangle 3$. $\land\, m.bal = b$
$\land\, m \in bmsgs$
$\land\, m.type =$ "1b"
BY $\langle 5\rangle 1$ DEF $InvP$, Inv, $knowsSentInv$, $msgsOfType$

$\langle 7\rangle 4$. PICK $r \in m.m2av : \land\, r.bal \geq c$
$\land\, r.val = v$
BY $\langle 7\rangle 2$

$\langle 7\rangle 5$. $r.bal \in Ballot$
BY $\langle 7\rangle 2$, $\langle 7\rangle 3$, $BMessageLemma$ DEF Inv, $TypeOK$, $1bMessage$ $\langle 8\rangle 2$

$\langle 7\rangle 6$. $[type \mapsto$ "1c", $bal \mapsto r.bal$, $val \mapsto r.val] \in msgs$
BY $\langle 7\rangle 2$, $\langle 7\rangle 3$, $\langle 7\rangle 4$ DEF Inv, $1bInv1$

$\langle 7\rangle 7$. QED
BY $\langle 7\rangle 2$, $\langle 7\rangle 4$, $\langle 7\rangle 5$, $\langle 7\rangle 6$

$\langle 6\rangle 6$. ASSUME NEW $m \in Q1b(BQ)$
PROVE $\land\, m1c.bal \geq m.mbal$
$\land\, (m1c.bal = m.mbal) \Rightarrow (m.mval = v)$

$\langle 7\rangle 1$. $m.acc \in Q(BQ)$
OBVIOUS

$\langle 7\rangle 2$. PICK $mm \in SS : \land\, mm.acc = m.acc$
$\land\, mm.mbal \leq c$
$\land\, (mm.mbal = c) \Rightarrow (mm.mval = v)$
BY $\langle 6\rangle 2$

$\langle 7\rangle 3$. PICK $mm2 \in SS : \land\, mm2.acc = m.acc$
$\land\, m = 1bRestrict(mm2)$

$\langle 8\rangle 1$. PICK $mm2 \in SS : m = 1bRestrict(mm2)$
BY $\langle 6\rangle 3$

$\langle 8\rangle 2$ QED
BY $\langle 8\rangle 1$ DEF $1bRestrict$

$\langle 7\rangle 4$. $\land\, mm = mm2$

$\land\ mm2.mbal \in Ballot \cup \{-1\}$

$\langle 8\rangle 1.\ \land\ mm \in knowsSent'[ac]$
$\land\ mm.bal = b$
$\land\ mm2 \in knowsSent'[ac]$
$\land\ mm2.bal = b$
OBVIOUS

$\langle 8\rangle 2.\ \land\ mm \in bmsgs$
$\land\ mm2 \in bmsgs$
$\land\ mm.type =$ "1b"
$\land\ mm2.type =$ "1b"
BY $\langle 5\rangle 1$, $\langle 8\rangle 1$ DEF $InvP$, Inv, $knowsSentInv$, $msgsOfType$

$\langle 8\rangle 3.\ mm.acc = mm2.acc$
BY $\langle 7\rangle 2$, $\langle 7\rangle 3$

$\langle 8\rangle 4.\ mm.acc \in Acceptor$
BY $\langle 7\rangle 1$, $\langle 7\rangle 2$

$\langle 8\rangle 5.\ mm2.mbal \in Ballot \cup \{-1\}$
BY $\langle 8\rangle 2$, $BMessageLemma$ DEF Inv, $TypeOK$, $1bMessage$

$\langle 8\rangle 6.$ QED
BY $\langle 8\rangle 1$, $\langle 8\rangle 2$, $\langle 8\rangle 3$, $\langle 8\rangle 4$, $\langle 8\rangle 5$ DEF Inv, $1bInv2$

$\langle 7\rangle 5.\ \land\ m.mbal \leq c$
$\land\ (m.mbal = c) \Rightarrow (m.mval = v)$
$\land\ m.mbal \in Ballot \cup \{-1\}$
BY $\langle 7\rangle 2$, $\langle 7\rangle 3$, $\langle 7\rangle 4$ DEF $1bRestrict$

$\langle 7\rangle 6.\ m1c.bal \geq m.mbal$
$\langle 8\rangle\ \forall\, m1cbal,\ mmbal \in Ballot \cup \{-1\} :$
$mmbal \leq c \land m1cbal \geq c \Rightarrow m1cbal \geq mmbal$
BY $SimpleArithmetic$ DEF $Ballot$

$\langle 8\rangle$ QED
BY $\langle 6\rangle 5$, $\langle 7\rangle 5$

$\langle 7\rangle 7.$ ASSUME $m1c.bal = m.mbal$
PROVE $m.mval = v$
$\langle 8\rangle\ \forall\, m1cbal,\ mmbal \in Ballot \cup \{-1\} :$
$mmbal \leq c \land m1cbal \geq c \land mmbal = m1cbal \Rightarrow mmbal = c$
BY $SimpleArithmetic$ DEF $Ballot$

$\langle 8\rangle$ QED
BY $\langle 7\rangle 5$, $\langle 6\rangle 5$, $\langle 7\rangle 7$

$\langle 7\rangle 8.$ QED
BY $\langle 7\rangle 6$, $\langle 7\rangle 7$

$\langle 6\rangle 7.$ QED

$\langle 7\rangle 1.\ Q(BQ) \in Quorum$
BY DEF $Quorum$

$\langle 7\rangle 2.\ P!ShowsSafeAt(Q(BQ),\ b,\ v)!1!2!2!(m1c)$
BY $\langle 6\rangle 5$, $\langle 6\rangle 6$

$\langle 7\rangle$ QED
BY $\langle 7\rangle 1$, $\langle 6\rangle 4$, $\langle 7\rangle 2$ DEF $P!ShowsSafeAt$

```
3846          ⟨5⟩5. QED
3847            BY ⟨4⟩1, ⟨5⟩3, ⟨5⟩4  DEF KnowsSafeAt
3848        ⟨4⟩2. ∃ a ∈ Acceptor : KnowsSafeAt(a, b, v)′
3849          ⟨5⟩1. PICK m ∈ SM : m.val = v
3850            OBVIOUS
3851          ⟨5⟩2. ∧ m ∈ msgs′
3852                ∧ m.type = "1c"
3853                ∧ m.bal = b
3854            BY ⟨3⟩1  DEF msgsOfType
3855          ⟨5⟩3. m ∈ 1cmsgs′
3856            BY ⟨5⟩2, MsgsTypeLemmaPrime
3857          ⟨5⟩ QED
3858            BY ⟨5⟩1, ⟨5⟩2, ⟨5⟩3  DEF 1cmsgs
3859        ⟨4⟩3. QED
3860          BY ⟨4⟩1, ⟨4⟩2
3861      ⟨3⟩5. PmaxBal′ = PmaxBal
3862        BY  DEF LearnsSent, PmaxBal, 1bOr2bMsgs
3863      ⟨3⟩6. QED
3864        BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3, ⟨3⟩4, ⟨3⟩5
3865           DEF LearnsSent, P!Phase1c, P!TLANext, Ballot, P!Ballot
3866    ⟨2⟩7. ASSUME NEW self ∈ Ballot,
3867                  Phase1a(self)
3869          PROVE  P!TLANext ∨ P!vars′ = P!vars
3870      ⟨3⟩ USE Phase1a(self)
3871      ⟨3⟩1. msgs′ = msgs ∪ {[type ↦ "1a", bal ↦ self]}
3872        BY MsgsLemma DEF Inv
3873      ⟨3⟩2. UNCHANGED ⟨PmaxBal, maxVBal, maxVVal⟩
3874        BY  DEF Phase1a, PmaxBal, 1bOr2bMsgs
3875      ⟨3⟩3. P!Phase1a(self)
3876        BY ⟨3⟩1, ⟨3⟩2  DEF P!Phase1a
3877      ⟨3⟩4. QED
3878        BY ⟨3⟩3  DEF P!TLANext, Ballot, P!Ballot
3879    ⟨2⟩8. ASSUME NEW self ∈ Ballot,
3880                  Phase1c(self)
3881          PROVE  P!TLANext ∨ P!vars′ = P!vars
3882      ⟨3⟩ USE Phase1c(self)
3883      ⟨3⟩1. PICK SS ∈ SUBSET [type : {"1c"}, bal : {self},
3884                              val : Value] :
3885                  ∧ ∀ m ∈ SS :
3886                      ∃ a  ∈ Acceptor : KnowsSafeAt(a, m.bal, m.val)
3887                  ∧ msgs′ = msgs ∪ SS
3888        BY MsgsLemma DEF Inv
3889      ⟨3⟩ DEFINE S ≜ {m.val : m ∈ SS}
3890      ⟨3⟩2. SS = {[type ↦ "1c", bal ↦ self,
```

$val \mapsto v] : v \in S\}$

⟨4⟩1. ASSUME NEW $m \in SS$
PROVE $m \in \{[type \mapsto$ "1c", $bal \mapsto self$, $val \mapsto v] : v \in S\}$

⟨5⟩1. $m = [type \mapsto$ "1c", $bal \mapsto self$, $val \mapsto m.val]$

⟨6⟩1. $\exists\, a \in \{$"1c"$\}, b \in \{self\}, v \in Value :$
$m = [type \mapsto a, bal \mapsto b, val \mapsto v]$
OBVIOUS

⟨6⟩2. PICK $v \in Value :$
$m = [type \mapsto$ "1c", $bal \mapsto self$, $val \mapsto v]$
BY ⟨6⟩1

⟨6⟩3. $[type \mapsto$ "1c", $bal \mapsto self$, $val \mapsto v].val = v$
OBVIOUS

⟨6⟩4. QED
BY ⟨6⟩2, ⟨6⟩3

⟨5⟩2. $m.val \in S$
BY DEF S

⟨5⟩3. QED
BY ⟨5⟩1, ⟨5⟩2

⟨4⟩2. ASSUME NEW $m \in \{[type \mapsto$ "1c", $bal \mapsto self$, $val \mapsto v] : v \in S\}$
PROVE $m \in SS$

⟨5⟩1. PICK $v \in S :$
$m = [type \mapsto$ "1c", $bal \mapsto self$, $val \mapsto v]$
OBVIOUS

⟨5⟩2. PICK $mm \in SS : mm.val = v$
OBVIOUS

⟨5⟩3. $mm \in [type : \{$"1c"$\}, bal : \{self\}, val : Value]$
OBVIOUS

⟨5⟩4. $mm = [type \mapsto$ "1c", $bal \mapsto self$, $val \mapsto v]$
BY ⟨5⟩2, ⟨5⟩3

⟨5⟩5. QED
BY ⟨5⟩1, ⟨5⟩4

⟨4⟩3. QED
BY ⟨4⟩1, ⟨4⟩2

⟨3⟩3. ASSUME NEW $v \in S$
PROVE $\exists\, Q \in Quorum : P!ShowsSafeAt(Q, self, v)$

⟨4⟩ DEFINE $m \triangleq [type \mapsto$ "1c", $bal \mapsto self$, $val \mapsto v]$

⟨4⟩1. $m \in SS$
BY ⟨3⟩2

⟨4⟩2. PICK $a \in Acceptor : KnowsSafeAt(a, self, v)$
BY ⟨4⟩1, ⟨3⟩1

⟨4⟩ DEFINE $SK \triangleq \{mm \in knowsSent[a] : mm.bal = self\}$

⟨4⟩3. ASSUME NEW $BQ \in ByzQuorum$,
$\forall\, ac \in BQ : \exists\, mm \in SK : mm.acc = ac$

PROVE $P!ShowsSafeAt(BQ \cap Acceptor,\ self,\ v)!1!1$

$\langle 5\rangle$ DEFINE $Q \triangleq BQ \cap Acceptor$
$Q1b \triangleq \{mm \in P!sentMsgs(\text{“1b”},\ self) : mm.acc \in Q\}$

$\langle 5\rangle$ SUFFICES ASSUME NEW $ac \in BQ \cap Acceptor$
PROVE $\exists\, mm \in Q1b : mm.acc = ac$

OBVIOUS

$\langle 5\rangle 1.\ \forall\, mm \in acceptorMsgsOfType(\text{“1b”}) :$
$(mm.bal = self) \Rightarrow$
$(1bRestrict(mm) \in P!sentMsgs(\text{“1b”},\ self))$

$\langle 6\rangle$ SUFFICES ASSUME NEW $mm \in acceptorMsgsOfType(\text{“1b”}),$
$mm.bal = self$
PROVE $1bRestrict(mm) \in P!sentMsgs(\text{“1b”},\ self)$

OBVIOUS

$\langle 6\rangle 1.\ \wedge\, 1bRestrict(mm).type = \text{“1b”}$
$\wedge\, 1bRestrict(mm).bal = self$
BY DEF $1bRestrict,\ acceptorMsgsOfType,\ msgsOfType$

$\langle 6\rangle 2.\ 1bRestrict(mm) \in msgs$
BY DEF $1bmsgs,\ msgs$

$\langle 6\rangle 3.$ QED
BY $\langle 6\rangle 1, \langle 6\rangle 2$ DEF $P!sentMsgs$

$\langle 5\rangle 2.$ PICK $mm \in SK : mm.acc = ac$
BY $\langle 4\rangle 3$

$\langle 5\rangle 3.\ mm \in msgsOfType(\text{“1b”}) \wedge mm.bal = self$
BY DEF $Inv,\ knowsSentInv$

$\langle 5\rangle 4.\ 1bRestrict(mm) \in P!sentMsgs(\text{“1b”},\ self)$
BY $\langle 5\rangle 1, \langle 5\rangle 2, \langle 5\rangle 3$ DEF $acceptorMsgsOfType$

$\langle 5\rangle 5.\ 1bRestrict(mm).acc = ac$
BY $\langle 5\rangle 2$ DEF $1bRestrict$

$\langle 5\rangle 6.$ QED
BY $\langle 5\rangle 4, \langle 5\rangle 5$

$\langle 4\rangle 4.$CASE $KnowsSafeAt(a,\ self,\ v)!1!1$

$\langle 5\rangle 1.$ PICK $BQ \in ByzQuorum :$
$\forall\, ac \in BQ : \exists\, mm \in SK :\ \wedge\, mm.acc = ac$
$\wedge\, mm.mbal = -1$
BY $\langle 4\rangle 4$

$\langle 5\rangle$ DEFINE $Q \triangleq BQ \cap Acceptor$
$Q1b \triangleq \{mm \in P!sentMsgs(\text{“1b”},\ self) : mm.acc \in Q\}$

$\langle 5\rangle 2.\ P!ShowsSafeAt(Q,\ self,\ v)!1!1$

$\langle 6\rangle 1.\ \forall\, ac \in BQ : \exists\, mm \in SK :\ mm.acc = ac$
BY $\langle 5\rangle 1$

$\langle 6\rangle 2.$ QED
BY $\langle 6\rangle 1, \langle 4\rangle 3$

$\langle 5\rangle 3.$ ASSUME NEW $mm \in Q1b$
PROVE $mm.mbal = -1$

$\langle 6\rangle 1.\ \wedge\, mm \in 1bmsgs$

$\land\ mm.bal = self$
$\land\ mm.acc \in Q$
BY $MsgsTypeLemma$ DEF $P!sentMsgs$
$\langle 6\rangle 2$. PICK $mb \in bmsgs : \land\ mb.type = \text{“1b”}$
$\land\ mm = 1bRestrict(mb)$
BY $\langle 6\rangle 1$ DEF $1bmsgs$, $acceptorMsgsOfType$, $msgsOfType$
$\langle 6\rangle 3. \land\ mb.bal = self$
$\land\ mb.acc \in Q$
BY $\langle 6\rangle 1$, $\langle 6\rangle 2$ DEF $1bRestrict$
$\langle 6\rangle 4$. PICK $mk \in SK : (mk.acc = mb.acc) \land (mk.mbal = -1)$
BY $\langle 6\rangle 3$, $\langle 5\rangle 1$
$\langle 6\rangle 5. (mk \in bmsgs) \land (mk.bal = self) \land (mk.type = \text{“1b”})$
BY DEF Inv, $knowsSentInv$, $msgsOfType$
$\langle 6\rangle 6. mk = mb$
BY $\langle 6\rangle 2$, $\langle 6\rangle 3$, $\langle 6\rangle 4$, $\langle 6\rangle 5$ DEF Inv, $1bInv2$
$\langle 6\rangle 7$. QED
BY $\langle 6\rangle 6$, $\langle 6\rangle 4$, $\langle 6\rangle 2$ DEF $1bRestrict$
$\langle 5\rangle 4$. QED
$\langle 6\rangle$ USE DEF $Quorum$
$\langle 6\rangle$ WITNESS $Q \in Quorum$
$\langle 6\rangle$ QED
BY $\langle 5\rangle 2$, $\langle 5\rangle 3$ DEF $P!ShowsSafeAt$, $Quorum$
$\langle 4\rangle 5$.CASE $KnowsSafeAt(a,\ self,\ v)!1!2$
$\langle 5\rangle 1$. PICK $c \in 0\ ..\ (self - 1) : KnowsSafeAt(a,\ self,\ v)!1!2!(c)$
BY $\langle 4\rangle 5$
$\langle 5\rangle 2$. PICK $BQ \in ByzQuorum : KnowsSafeAt(a,\ self,\ v)!1!2!(c)!1!(BQ)$
BY $\langle 5\rangle 1$
$\langle 5\rangle$ DEFINE $Q \triangleq BQ \cap Acceptor$
$Q1b \triangleq \{mm \in P!sentMsgs(\text{“1b”},\ self) : mm.acc \in Q\}$
$\langle 5\rangle 3.\ P!ShowsSafeAt(Q,\ self,\ v)!1!1$
$\langle 6\rangle 1. \forall\, ac \in BQ : \exists\, mm \in SK : mm.acc = ac$
BY $\langle 5\rangle 2$
$\langle 6\rangle 2$. QED
BY $\langle 6\rangle 1$, $\langle 4\rangle 3$
$\langle 5\rangle 4$. PICK $WQ \in WeakQuorum : KnowsSafeAt(a,\ self,\ v)!1!2!(c)!2!(WQ)$
BY $\langle 5\rangle 1$
$\langle 5\rangle 5$. PICK $ac \in WQ \cap Acceptor :$
$KnowsSafeAt(a,\ self,\ v)!1!2!(c)!2!(WQ)!(ac)$
$\langle 6\rangle 1. \exists\, ac \in WQ \cap Acceptor :$ TRUE
BY BQA
$\langle 6\rangle 2$. QED
BY $\langle 6\rangle 1$, $\langle 5\rangle 4$
$\langle 5\rangle 6$. PICK $mk \in SK : \land\ mk.acc = ac$
$\land \exists\, r \in mk.m2av : \land\ r.bal \geq c$
$\land\ r.val = v$

BY $\langle 5\rangle 5$

$\langle 5\rangle 7$. PICK $r \in mk.m2av : \land r.bal \geq c$
$\land r.val = v$

BY $\langle 5\rangle 6$

$\langle 5\rangle 8$. $(mk \in bmsgs) \land (mk.type = \text{“1b”}) \land (mk.bal = self)$

BY DEF Inv, $knowsSentInv$, $msgsOfType$

$\langle 5\rangle$ DEFINE $mc \triangleq [type \mapsto \text{“1c”},\ bal \mapsto r.bal,\ val \mapsto v]$

$\langle 5\rangle 9$. $mc \in msgs$

$\langle 6\rangle 1$. $[type \mapsto \text{“1c”},\ bal \mapsto r.bal,\ val \mapsto r.val] \in msgs$

BY $\langle 5\rangle 6$, $\langle 5\rangle 8$ DEF Inv, $1bInv1$

$\langle 6\rangle 2$. QED

BY $\langle 6\rangle 1$, $\langle 5\rangle 7$

$\langle 5\rangle 10$. ASSUME NEW $mq \in Q1b$
PROVE $\land mc.bal \geq mq.mbal$
$\land (mc.bal = mq.mbal) \Rightarrow (mq.mval = v)$

$\langle 6\rangle 1$. $\land mq \in msgs$
$\land mq.type = \text{“1b”}$
$\land mq.bal = self$
$\land mq.acc \in Q$

BY DEF $P!sentMsgs$

$\langle 6\rangle 2$. PICK $mbq \in acceptorMsgsOfType(\text{“1b”}) :$
$mq = 1bRestrict(mbq)$

BY $\langle 6\rangle 1$, $MsgsTypeLemma$ DEF $1bmsgs$

$\langle 6\rangle 3$. $\land mbq \in bmsgs$
$\land mbq.type = \text{“1b”}$
$\land mbq.bal = self$
$\land mbq.mbal = mq.mbal$
$\land mbq.mval = mq.mval$
$\land mbq.acc \in Q$

BY $\langle 6\rangle 1$, $\langle 6\rangle 2$ DEF $acceptorMsgsOfType$, $msgsOfType$, $1bRestrict$

$\langle 6\rangle 4$. PICK $ab \in BQ :$
$\exists\, mcq \in SK : \land mcq.acc = mbq.acc$
$\land mcq.mbal \leq c$
$\land (mcq.mbal = c) \Rightarrow (mcq.mval = v)$

BY $\langle 5\rangle 2$, $\langle 6\rangle 3$

$\langle 6\rangle 5$. PICK $mcq : \land mcq \in knowsSent[a]$
$\land mcq.bal = self$
$\land mcq.acc = mbq.acc$
$\land mcq.mbal \leq c$
$\land (mcq.mbal = c) \Rightarrow (mcq.mval = v)$

BY $\langle 6\rangle 4$

$\langle 6\rangle 6$. $(mcq \in bmsgs) \land (mcq.type = \text{“1b”})$

BY $\langle 6\rangle 5$ DEF Inv, $knowsSentInv$, $msgsOfType$

$\langle 6\rangle 7$. $mcq = mbq$

BY $\langle 6\rangle 3$, $\langle 6\rangle 5$, $\langle 6\rangle 6$ DEF Inv, $1bInv2$

⟨6⟩8. $\land\ mc.bal \geq mcq.mbal$
$\land\ (mc.bal = mcq.mbal) \Rightarrow (mcq.mval = v)$
⟨7⟩1. $mc.bal \in Ballot \cup \{-1\} \land mc.bal \geq c$
⟨8⟩1. $mc.bal = r.bal$
OBVIOUS
⟨8⟩2. $mk \in bmsgs \land mk.type =$ "1b"
BY DEF Inv, $knowsSentInv$, $msgsOfType$
⟨8⟩3. $r.bal \in Ballot$
⟨9⟩1. $mk \in 1bMessage$
BY ⟨8⟩2, $BMessageLemma$ DEF Inv, $TypeOK$, $1bMessage$
⟨9⟩2. QED
BY ⟨9⟩1 DEF $1bMessage$
⟨8⟩4. QED
BY ⟨8⟩1, ⟨8⟩3, ⟨5⟩7
⟨7⟩2. $mcq.mbal \in Ballot \cup \{-1\}$
⟨8⟩ $mcq \in bmsgs \land mcq.type =$ "1b"
BY ⟨6⟩5 DEF Inv, $knowsSentInv$, $msgsOfType$
⟨8⟩ QED
BY $BMessageLemma$ DEF Inv, $TypeOK$, $1bMessage$
⟨7⟩3. $\forall\, mcbal,\ mcqmbal \in Ballot \cup \{-1\}$:
$\land\ \ mcbal \geq c$
$\land\ \ mcqmbal \leq c$
$\Rightarrow\ \land mcbal \geq mcqmbal$
$\land\ (mcbal = mcqmbal) \Rightarrow (mcqmbal = c)$
BY $SimpleArithmetic$ DEF $Ballot$
⟨7⟩4. $\land\ mc.bal \geq mcq.mbal$
$\land\ (mc.bal = mcq.mbal) \Rightarrow (mcq.mbal = c)$
BY ⟨7⟩1, ⟨7⟩2, ⟨7⟩3, ⟨6⟩5
⟨7⟩5. QED
BY ⟨7⟩4, ⟨6⟩5
⟨6⟩9. QED
BY ⟨6⟩3, ⟨6⟩7, ⟨6⟩8
⟨5⟩11. QED
⟨6⟩ $Q \in Quorum$
BY DEF $Quorum$
⟨6⟩ WITNESS $Q \in Quorum$
⟨6⟩ QED
BY ⟨5⟩3, ⟨5⟩9, ⟨5⟩10 DEF $P!ShowsSafeAt$
⟨4⟩6. QED
BY ⟨4⟩2, ⟨4⟩4, ⟨4⟩5 DEF $KnowsSafeAt$
⟨3⟩4. $P!Phase1c(self,\ S)$
BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3 DEF $P!Phase1c$, $Phase1c$, $PmaxBal$, $1bOr2bMsgs$
⟨3⟩5. QED
BY ⟨3⟩4 DEF $P!TLANext$, $Ballot$, $P!Ballot$
⟨2⟩9. ASSUME NEW $self \in FakeAcceptor$,

4116 $FakingAcceptor(self)$
4117 PROVE $P!TLANext \lor P!vars' = P!vars$
4118 $\langle 3\rangle$ USE $FakingAcceptor(self)$
4119 $\langle 3\rangle 1.\ msgs' = msgs$
4120 BY $MsgsLemma$ DEF Inv
4121 $\langle 3\rangle 2.\ PmaxBal' = PmaxBal$
4122 $\langle 4\rangle 1.$ PICK $mm : \land bmsgs' = bmsgs \cup \{mm\}$
4123 $\land mm.acc = self$
4124 BY DEF $FakingAcceptor$
4125 $\langle 4\rangle$ DEFINE $S(b,\ a) \triangleq \{ma \in b : \land ma.type \in \{\text{“1b”},\ \text{“2b”}\}$
4126 $\land ma.acc = a\}$
4127 $\langle 4\rangle 2.$ ASSUME NEW $a \in Acceptor$
4128 PROVE $S(bmsgs,\ a) = S(bmsgs',\ a)$
4129 $\langle 5\rangle 1. \forall m : (m.acc = a) \Rightarrow (m \in bmsgs' \equiv m \in bmsgs)$
4130 BY $\langle 4\rangle 1$, BQA
4131 $\langle 5\rangle 2.$ QED
4132 BY $\langle 5\rangle 1$
4133 $\langle 4\rangle$ QED
4134 BY $\langle 4\rangle 2$ DEF $PmaxBal$, $1bOr2bMsgs$
4135 $\langle 3\rangle 3.\ P!vars' = P!vars$
4136 BY $\langle 3\rangle 1$, $\langle 3\rangle 2$ DEF $P!vars$, $FakingAcceptor$
4137 $\langle 3\rangle 4.$ QED
4138 BY $\langle 3\rangle 3$
4139 $\langle 2\rangle 10.$ QED
4140 BY $\langle 2\rangle 3$, $\langle 2\rangle 4$, $\langle 2\rangle 5$, $\langle 2\rangle 6$, $\langle 2\rangle 7$, $\langle 2\rangle 8$, $\langle 2\rangle 9$, $NextDef$

4142 $\langle 1\rangle 3.$ QED
4143 PROOF OMITTED

4145 ⊢———————————————————————————————————————⊣

To see how learning is implemented, we must describe how to determine that a value has been chosen. This is done by the following definition of *chosen* to be the set of chosen values.

4151 $chosen \triangleq \{v \in Value : \exists\, BQ \in ByzQuorum,\ b \in Ballot :$
4152 $\forall\, a \in BQ : \exists\, m \in msgs : \land m.type = \text{“2b”}$
4153 $\land m.acc = a$
4154 $\land m.bal = b$
4155 $\land m.val = v\}$

The correctness of our definition of *chosen* is expressed by the following theorem, which asserts that if a value is in *chosen* , then it is also in the set *chosen* of the emulated execution of the *PaxosConsensus* algorithm.

The state function *chosen* does not necessarily equal the corresponding state function of the *PaxosConsensus* algorithm. It requires every (real or fake) acceptor in a *ByzQuorum* to vote for (send $2b$ messages) for a value v in the same ballot for v to be in *chosen* for the *BPCon* algorithm, but it requires only that every (real) acceptor in a *Quorum* vote for v in the same ballot for v to be in the set *chosen* of the emulated execution of *PaxosConsensus*.

Liveness for *BPCon* requires that, under suitable assumptions, some value is eventually in *chosen* . Since we can't assume that a fake acceptor does anything useful, liveness requires the assumption that there is a *ByzQuorum* composed entirely of real acceptors (the second conjunct of assumption *BQLA*.

THEOREM $chosen \subseteq P!chosen$

Note: I had to define *ch* and *Pch* instead of using subexpression names because of a bug in tlapm that has since been fixed.

$\langle 1 \rangle$ DEFINE $ch(v) \triangleq \exists\, BQ \in ByzQuorum,\ b \in Ballot :$
$\forall\, a \in BQ : \exists\, m \in msgs :\ \wedge\, m.type = \text{“2b”}$
$\wedge\, m.acc = a$
$\wedge\, m.bal = b$
$\wedge\, m.val = v$

$Pch(v) \triangleq \exists\, Q \in Quorum,\ b \in P!Ballot :$
$\forall\, a \in Q : \exists\, m \in msgs :\ \wedge\, m.type = \text{“2b”}$
$\wedge\, m.acc = a$
$\wedge\, m.bal = b$
$\wedge\, m.val = v$

$\langle 1 \rangle$ SUFFICES ASSUME NEW $v \in Value$, $ch(v)$
PROVE $Pch(v)$
BY DEF $chosen$, $P!chosen$

$\langle 1 \rangle$1. PICK $BQ \in ByzQuorum,\ b \in Ballot : ch(v)!(BQ,\ b)$
OBVIOUS
$\langle 1 \rangle$ DEFINE $Q \triangleq BQ \cap Acceptor$
$\langle 1 \rangle$2. $Q \in Quorum$
BY DEF $Quorum$
$\langle 1 \rangle$ WITNESS $Q \in Quorum$
$\langle 1 \rangle$ USE DEF $P!Ballot$, $Ballot$
$\langle 1 \rangle$ WITNESS $b \in P!Ballot$
$\langle 1 \rangle$3. QED
BY $\langle 1 \rangle$1

* Modification History
* Last modified Sat *Nov* 16 22:20:34 *CST* 2019 by *hengxin*
* Last modified *Tue Feb* 08 11:53:20 *PST* 2011 by lamport