Invariance Proof of Algorithm Voting

$Inv \triangleq VotesSafe \land OneValuePerBallot.$

$VotesSafe \triangleq VotedFor(a, b, v) \Rightarrow SafeAt(b, v)$

$SafeAt(b, v) \triangleq$

$\quad \forall c \in 0..(b-1):$

$\quad\quad \exists Q \in Quorums:$

$\quad\quad\quad \forall a \in Q: VotedFor(a, v, c) \lor WontVoteIn(a, c)$

1. $Init \Rightarrow Inv$

PROOF: By the definitions of $Init$ and Inv, since $Init$ implies $votes = \emptyset$ and $maxBal = -1$.

2. $Inv \land Next \Rightarrow Inv'$

2.1. CASE $IncreaseMaxBal(a, b)$

3.1. $VotesSafe'$

PROOF: $IncreaseMaxBal$ implies votes keep unchanged in next state. $VotedFor(a, b, v) \Leftrightarrow VotedFor(a, b, v)'$, and $WontVoteIn(a, c) \Rightarrow WontVoteIn(a, c)'$.$(VotedFor(a, b, v) \Rightarrow SafeAt(b, v)) \Rightarrow (VotedFor(a, b, v)' \Rightarrow SafeAt(b, v)')$

3.2. $OneValuePerBallot'$

PROOF: $IncreaseMaxBal$ implies votes keep unchanged in next state. Obviously satisfying one value per ballot.

2.2. CASE $VoteFor$

3.1. $VotesSafe'$

PROOF:ASSUME $VotedFor(aa, bb, vv)'$, to prove $SafeAt(bb, vv)'$. If (b, v) has been voted for, then $VotesSafe'$ satisfies because it's stable. If (b, v) has not been voted for, than $bb = b \land vv = v$, $VotesSafe'$ satisfies because of $ShowsSafeAt(Q, b, v)$ in $VoteFor$.

3.2. $OneValuePerBallot'$

PROOF: $VoteFor$ maintain the property of one value per ballot.

3. $Inv \Rightarrow Consistency$

PROOF: ASSUME $chosen(b1, v1) \land chosen(b2, v2)$, to prove $v1 = v2$. $(b2, v2)$ satisfy $SafeAt(b2, v2)$, and obviously $v1 = v2$ because of the definition of $SafeAt(b, v)$.

4. Q.E.D

PROOF: By 1, 2 and 3.