A small distributed concensus protocol

Yoichi Hirai*

February 27, 2017

Contents

*i@yoichihirai.com

- This document is produced from the code available at https://github.com/pirapira/pos.
- To get updates on this project and similar ones, follow http://gitter.im/ethereum/formal-methods.

theory *MinimumAlgo*

imports *Main*

begin

1 Definition of the Protocol (Not Skippable)

In this development we do not know much about hashes. There are many hashes. Two hashes might be equal or not.

datatype *hash* = *Hash int*

Views are numbers. We actually need the fact that views are in total order. Otherwise accountable safety can be broken.

type-synonym *view* = *int*

We have two kinds of messages.

datatype *message* =
  Commit hash ∗ *view*
| *Prepare hash* ∗ *view* ∗ *view*

Sometimes we want to talk about the view of a message.

datatype *validator* = *Validator int*

type-synonym *sent* = *validator* ∗ *message*

A situation might be seen from a global point of view where every sent messages can be seen, or more likely seen from a local point of view.

record *situation* =
  Validators :: *validator set*
  Messages :: *sent set*
  PrevHash :: *hash* ⇒ *hash option*

In the next section, we are going to determine which of the validators are slashed in a situation.

We will be talking about two conflicting commits. To define 'conflicting' one needs to look at the hashes.

A situation contains information which hash is the parent of which hash. We can follow this link n-times.

fun *nth-ancestor* :: *situation* ⇒ *nat* ⇒ *hash* ⇒ *hash option*
where
nth-ancestor - 0 h = *Some h*
| *nth-ancestor s* (*Suc n*) *h* =
(*case PrevHash s h of*
None ⇒ *None*
| *Some h′* ⇒ *nth-ancestor s n h′*)

And also we are allowed to talk if two hashes are in ancestor-descendant relation. It does not matter if this is computable.

definition *is-descendant-or-self* :: *situation* ⇒ *hash* ⇒ *hash* ⇒ *bool*
where
is-descendant-or-self s x y = (∃ *n*. *nth-ancestor s n x* = *Some y*)

We can also talk if two hashes are not in ancestor-descendant relation in whichever ways.

definition *not-on-same-chain* :: *situation* ⇒ *hash* ⇒ *hash* ⇒ *bool*
where
not-on-same-chain s x y = ((¬ *is-descendant-or-self s x y*) ∧ (¬ *is-descendant-or-self s y x*))

In the slashing condition, we will be talking about two-thirds of the validators doing something.

We can lift any predicate about a validator into a predicate about a situation: two thirds of the validators satisfy the predicate.

definition *two-thirds* :: *situation* ⇒ (*validator* ⇒ *bool*) ⇒ *bool*
where
two-thirds s f =
(*2* ∗ *card* (*Validators s*) ≤ *3* ∗ *card* ({*n*. *n* ∈ *Validators s* ∧ *f n*}))

Similarly for one-third, more-than-two-thirds, and more-than-one-third.

definition *one-third* :: *situation* ⇒ (*validator* ⇒ *bool*) ⇒ *bool*
where
one-third s f =
(*card* (*Validators s*) ≤ *3* ∗ *card* ({*n*. *n* ∈ *Validators s* ∧ *f n*}))

definition *more-than-two-thirds* :: *situation* ⇒ (*validator* ⇒ *bool*) ⇒ *bool*
where
more-than-two-thirds s f =
(*2* ∗ *card* (*Validators s*) < *3* ∗ *card* ({*n*. *n* ∈ *Validators s* ∧ *f n*}))

definition *more-than-one-third* :: *situation* ⇒ (*validator* ⇒ *bool*) ⇒ *bool*
where
more-than-one-third s f =

$(card\ (Validators\ s) < 3 * card\ (\{n.\ n \in Validators\ s \wedge f\ n\}))$

definition *two-thirds-sent-message* :: *situation* ⇒ *message* ⇒ *bool*
where
two-thirds-sent-message s m =
$(two\text{-}thirds\ s\ (\lambda\ n.\ (n,\ m) \in Messages\ s)$

A hash is prepared when two-thirds of the validators have sent a certain message.

definition *prepared* :: *situation* ⇒ *hash* ⇒ *view* ⇒ *view* ⇒ *bool*
where
prepared s h v vs =
$(two\text{-}thirds\text{-}sent\text{-}message\ s\ (Prepare\ (h,\ v,\ vs)))$

A hash is committed when two-thirds of the validators have sent a certain message.

definition *committed* :: *situation* ⇒ *hash* ⇒ *bool*
where
$committed\ s\ h = (\exists\ v.\ two\text{-}thirds\text{-}sent\text{-}message\ s\ (Commit\ (h,\ v)))$

2 The Slashing Conditions (not skippable)

[i] A validator is slashed when it has sent a commit message of a hash that is not prepared yet.

definition *slashed-one* :: *situation* ⇒ *validator* ⇒ *bool*
where
slashed-one s n =
$(n \in Validators\ s \wedge$
$(\exists\ h\ v.$
$((n,\ Commit\ (h,\ v)) \in Messages\ s \wedge$
$(\neg\ (\exists\ vs.\ -1 \leq vs \wedge vs < v \wedge prepared\ s\ h\ v\ vs)\ ))))$

[ii] A validator is slashed when it has sent a prepare message whose view src is not -1 but has no supporting preparation in the view src.

definition *slashed-two* :: *situation* ⇒ *validator* ⇒ *bool*
where
slashed-two s n =
$(n \in Validators\ s \wedge$
$(\exists\ h\ v\ vs.$
$((n,\ Prepare\ (h,\ v,\ vs)) \in Messages\ s \wedge$
$vs \neq -1 \wedge$
$(\neg\ (\exists\ h\text{-}anc\ vs'.$
$-1 \leq vs' \wedge vs' < vs \wedge$
$Some\ h\text{-}anc = nth\text{-}ancestor\ s\ (nat\ (v - vs))\ h \wedge$
$prepared\ s\ h\text{-}anc\ vs\ vs')))))$

[iii] A validator is slashed when it has sent a commit message and a prepare message containing view numbers in a specific constellation.

definition *slashed-three* :: *situation* ⇒ *validator* ⇒ *bool*
where
slashed-three s n =
(*n* ∈ *Validators s* ∧
(∃ *x y v w u*.
(*n*, *Commit* (*x*, *v*)) ∈ *Messages s* ∧
(*n*, *Prepare* (*y*, *w*, *u*)) ∈ *Messages s* ∧
u < *v* ∧ *v* < *w*))

[iv] A validator is slashed when it has sent two conflicting Prepare messages at the same view.

definition *slashed-four* :: *situation* ⇒ *validator* ⇒ *bool*
where
slashed-four s n =
(*n* ∈ *Validators s* ∧
(∃ *x1 x2 v vs1 vs2*.
(*n*, *Prepare* (*x1*, *v*, *vs1*)) ∈ *Messages s* ∧
(*n*, *Prepare* (*x2*, *v*, *vs2*)) ∈ *Messages s* ∧
(*x1* ≠ *x2* ∨ *vs1* ≠ *vs2*)))

A validator is slashed when at least one of the above conditions [i]–[iv] hold.

definition *slashed* :: *situation* ⇒ *validator* ⇒ *bool*
where
slashed s n = (*slashed-one s n* ∨
slashed-two s n ∨
slashed-three s n ∨
slashed-four s n)

definition *one-third-slashed* :: *situation* ⇒ *bool*
where
one-third-slashed s = *one-third s* (*slashed s*)

However, since cardinality of an infinite set is defined to be zero, we should be talking about situations where the set of validators is finite.

definition *situation-has-finitely-many-validators* :: *situation* ⇒ *bool*
where
situation-has-finitely-many-validators s = (*Validators s* ≠ {} ∧ *finite* (*Validators s*))

3 Useful Lemmas for Accountable Safety (can be skipped)

lemma *card-not* [*simp*] :
finite s ⟹
card {*n* ∈ *s*. ¬ *f n*} = *card s* − *card* {*n* ∈ *s*. *f n*}
⟨*proof*⟩

lemma *not-one-third* [*simp*] :
situation-has-finitely-many-validators s $\Longrightarrow$
$(\neg$ *one-third s f*$) =$ (*more-than-two-thirds s* $(\lambda\ n.\ \neg\ f\ n)$)
⟨*proof*⟩

lemma *condition-one-positive* :
$\exists\ n.\ (n,$ *Commit* $(x,\ v)) \in$ *Messages s* $\wedge$
n $\in$ *Validators s* $\wedge$
$\neg$ *slashed s n* $\Longrightarrow$
$(\exists\, v\ vs.$
two-thirds s $(\lambda n.\ (n,$ *Prepare* $(x,\ v,\ vs)) \in$ *Messages s*$)$
$\wedge\ -\ 1 \leq vs \wedge vs < v)$
⟨*proof*⟩

lemma *condition-one-positive′* :
$\exists\ n.\ (n,$ *Commit* $(x,\ v)) \in$ *Messages s* $\wedge$
n $\in$ *Validators s* $\wedge$
$\neg$ *slashed s n* $\Longrightarrow$
$(\exists\, vs.$
two-thirds s $(\lambda n.\ (n,$ *Prepare* $(x,\ v,\ vs)) \in$ *Messages s*$)$
$\wedge\ -\ 1 \leq vs \wedge vs < v)$
⟨*proof*⟩

lemma *set-conj* [*simp*] :
$\{n \in s.\ f\ n \wedge g\ n\} = \{n \in s.\ f\ n\} \cap \{n \in s.\ g\ n\}$
⟨*proof*⟩

lemma *two-more-two-set* :
finite s $\Longrightarrow$
$2 *$ *card s* $\leq 3 *$ *card* $\{n \in s.\ f\ n\} \Longrightarrow$
$2 *$ *card s* $< 3 *$ *card* $\{n \in s.\ g\ n\} \Longrightarrow$
card s
$< 3 *$ *card* $(\{n \in s.\ f\ n\} \cap \{n \in s.\ g\ n\})$

⟨*proof*⟩

lemma *two-more-two* :
situation-has-finitely-many-validators s $\Longrightarrow$
two-thirds s f $\Longrightarrow$
more-than-two-thirds s g $\Longrightarrow$
more-than-one-third s $(\lambda\ n.\ f\ n \wedge g\ n)$

⟨*proof*⟩

lemma *card-nonzero-exists* :
card $\{n \in s.\ f\ n\} > 0 \Longrightarrow$
$\exists\ n \in s.\ f\ n$

⟨*proof*⟩

lemma *more-than-one-third-exists* :
situation-has-finitely-many-validators s ⟹
more-than-one-third s f ⟹
∃ *n* ∈ *Validators s*. *f n*

⟨*proof*⟩

lemma *two-more-two-ex* :
situation-has-finitely-many-validators s ⟹
two-thirds s f ⟹
more-than-two-thirds s g ⟹
∃ *n* ∈ *Validators s*. *f n* ∧ *g n*
⟨*proof*⟩

lemma *commit-expand*:
situation-has-finitely-many-validators s ⟹
two-thirds-sent-message s (*Commit* (*x*, *v*)) ⟹
(∃ *vs*. *prepared s x v vs* ∧ −*1* ≤ *vs* ∧ *vs* < *v*) ∨ *one-third-slashed s*
⟨*proof*⟩

lemma *card-conj-le* :
finite s ⟹
card ({*n* ∈ *s*. *f n*} ∩ {*n* ∈ *s*. *g n*})
= *card* {*n* ∈ *s*. *f n*} + *card* { *n* ∈ *s*. *g n*} − *card* ({*n* ∈ *s*. *f n*} ∪ {*n* ∈ *s*. *g n*})
⟨*proof*⟩

lemma *two-two-set* :
2 ∗ *card s* ≤ *3* ∗ *card* {*n* ∈ *s*. *f n*} ⟹
2 ∗ *card s* ≤ *3* ∗ *card* {*n* ∈ *s*. *g n*} ⟹
finite s ⟹
card s ≤ *3* ∗ *card* ({*n* ∈ *s*. *f n*} ∩ {*n* ∈ *s*. *g n*})
⟨*proof*⟩

lemma *two-two* :
situation-has-finitely-many-validators s ⟹
two-thirds s f ⟹
two-thirds s g ⟹
one-third s (λ *n*. *f n* ∧ *g n*)
⟨*proof*⟩

lemma *dependency-self* [*simp*]:
¬ *not-on-same-chain s y y*
⟨*proof*⟩

lemma *prepare-direct-conflict* :
not-on-same-chain s x y ⟹

$n \in \mathit{Validators}\ s \Longrightarrow$
$(n,\ \mathit{Prepare}\ (x,\ v2,\ vs1)) \in \mathit{Messages}\ s \Longrightarrow$
$(n,\ \mathit{Prepare}\ (y,\ v2,\ vs2)) \in \mathit{Messages}\ s \Longrightarrow \mathit{slashed\text{-}four}\ s\ n$
⟨*proof*⟩

lemma *inclusion-card-le* :
$\forall n.\ n \in \mathit{Validators}\ s \longrightarrow f\ n \longrightarrow g\ n \Longrightarrow$
$\mathit{finite}\ (\mathit{Validators}\ s) \Longrightarrow$
$\mathit{card}\ \{n \in \mathit{Validators}\ s.\ f\ n\} \leq \mathit{card}\ \{n \in \mathit{Validators}\ s.\ g\ n\}$
⟨*proof*⟩

lemma *mp-one-third* :
$\mathit{finite}\ (\mathit{Validators}\ s) \Longrightarrow$
$\forall\ n.\ n \in \mathit{Validators}\ s \longrightarrow f\ n \longrightarrow g\ n \Longrightarrow$
$\mathit{one\text{-}third}\ s\ f \Longrightarrow \mathit{one\text{-}third}\ s\ g$
⟨*proof*⟩

lemma *mp-two-thirds* :
$\mathit{finite}\ (\mathit{Validators}\ s) \Longrightarrow$
$\forall\ n.\ n \in \mathit{Validators}\ s \longrightarrow f\ n \longrightarrow g\ n \Longrightarrow$
$\mathit{two\text{-}thirds}\ s\ f \Longrightarrow \mathit{two\text{-}thirds}\ s\ g$
⟨*proof*⟩

lemma *safety-case1'* :
$\mathit{situation\text{-}has\text{-}finitely\text{-}many\text{-}validators}\ s \Longrightarrow$
$\mathit{not\text{-}on\text{-}same\text{-}chain}\ s\ x\ y \Longrightarrow$
$\mathit{two\text{-}thirds}\ s\ (\lambda n.\ (n,\ \mathit{Prepare}\ (x,\ v2,\ vs1)) \in \mathit{Messages}\ s) \Longrightarrow$
$\mathit{two\text{-}thirds}\ s\ (\lambda n.\ (n,\ \mathit{Prepare}\ (y,\ v2,\ vs2)) \in \mathit{Messages}\ s) \Longrightarrow \mathit{one\text{-}third}\ s\ (\mathit{slashed}\ s)$
⟨*proof*⟩

lemma *safety-case1* :
$\mathit{situation\text{-}has\text{-}finitely\text{-}many\text{-}validators}\ s \Longrightarrow$
$\mathit{not\text{-}on\text{-}same\text{-}chain}\ s\ x\ y \Longrightarrow$
$\mathit{prepared}\ s\ x\ v2\ vs1 \Longrightarrow$
$\mathit{prepared}\ s\ y\ v2\ vs2 \Longrightarrow$
$\mathit{one\text{-}third\text{-}slashed}\ s$
⟨*proof*⟩

lemma *not-on-same-chain-sym* [*simp*] :
$\mathit{not\text{-}on\text{-}same\text{-}chain}\ s\ x\ y = \mathit{not\text{-}on\text{-}same\text{-}chain}\ s\ y\ x$
⟨*proof*⟩

lemma *commit-prepare* :
$\mathit{situation\text{-}has\text{-}finitely\text{-}many\text{-}validators}\ s \Longrightarrow$
$\mathit{two\text{-}thirds}\ s\ (\lambda n.\ (n,\ \mathit{Commit}\ (y,\ v)) \in \mathit{Messages}\ s) \Longrightarrow$
$(\exists\, vs.\ \mathit{prepared}\ s\ y\ v\ vs \wedge -1 \leq vs \wedge vs < v) \vee \mathit{one\text{-}third\text{-}slashed}\ s$

⟨*proof*⟩

lemma *one-third-prepared-conflict* :
$x \neq y \Longrightarrow$
one-third s
$(\lambda n.\ (n,\ Prepare\ (y,\ c\text{-}view,\ vs)) \in Messages\ s \wedge (n,\ Prepare\ (x,\ c\text{-}view,\ vs1))$
$\in Messages\ s) \Longrightarrow$
situation-has-finitely-many-validators s $\Longrightarrow$
one-third s (*slashed s*)
⟨*proof*⟩

lemma *prepared-conflict* :
prepared s y c-view vs $\Longrightarrow$
situation-has-finitely-many-validators s $\Longrightarrow$
$x \neq y \Longrightarrow$
prepared s x c-view vs1 $\Longrightarrow$
one-third-slashed s
⟨*proof*⟩

lemma *commit-prepared* :
situation-has-finitely-many-validators s $\Longrightarrow$
$x \neq y \Longrightarrow$
two-thirds-sent-message s (*Commit* (*y*, *c-view*)) $\Longrightarrow$
prepared s x c-view vs1 $\Longrightarrow$
one-third-slashed s
⟨*proof*⟩

lemma *condition-three-again* :
situation-has-finitely-many-validators s $\Longrightarrow$
vs1 $<$ *c-view* $\Longrightarrow$
c-view $<$ *v* $\Longrightarrow$
one-third s $(\lambda n.\ (n,\ Commit\ (y,\ c\text{-}view)) \in Messages\ s \wedge (n,\ Prepare\ (x,\ v,$
vs1)) $\in Messages\ s) \Longrightarrow$
one-third-slashed s
⟨*proof*⟩

lemma *between-concrete* :
c-view $<$ *v* $\Longrightarrow$
two-thirds-sent-message s (*Commit* (*y*, *c-view*)) $\Longrightarrow$
prepared s x v vs1 $\Longrightarrow$
vs1 $<$ *c-view* $\Longrightarrow$
situation-has-finitely-many-validators s $\Longrightarrow$
one-third-slashed s
⟨*proof*⟩

lemma *between-case* :
c-view $\leq$ *v* $\Longrightarrow$
situation-has-finitely-many-validators s $\Longrightarrow$
two-thirds-sent-message s (*Commit* (*y*, *c-view*)) $\Longrightarrow$

prepared s x v vs1 $\Longrightarrow$ $-$ *1* $\leq$ *vs1* $\Longrightarrow$ *c-view* $\neq$ *v* $\Longrightarrow$ *vs1* $<$ *c-view* $\Longrightarrow$ *one-third-slashed s*
⟨*proof*⟩

lemma *ancestors-ancestor* :
$\forall$ *m x y.*
nth-ancestor s n x = *Some y* $\longrightarrow$
nth-ancestor s m y = *nth-ancestor s* (*n* + *m*) *x*

⟨*proof*⟩

lemma *nat-min-min* :
vs1 $<$ *v* $\Longrightarrow$
$\neg$ *vs1* $<$ *c-view* $\Longrightarrow$
(*nat* (*v* $-$ *vs1*) + *nat* (*vs1* $-$ *c-view*)) = *nat* (*v* $-$ *c-view*)
⟨*proof*⟩

lemma *ancestor-ancestor* :
nth-ancestor s (*nat* (*v* $-$ *c-view*)) *x* $\neq$ *Some y* $\Longrightarrow$
vs1 $<$ *v* $\Longrightarrow$
$\neg$ *vs1* $<$ *c-view* $\Longrightarrow$
$\neg$ *c-view* $\leq$ $-$ *1* $\Longrightarrow$
$-$ *1* $\leq$ *vs′* $\Longrightarrow$
vs′ $<$ *vs1* $\Longrightarrow$
Some h-anc = *nth-ancestor s* (*nat* (*v* $-$ *vs1*)) *x* $\Longrightarrow$
nth-ancestor s (*nat* (*vs1* $-$ *c-view*)) *h-anc* $\neq$ *Some y*

⟨*proof*⟩

lemma *the-induction* :
nat (*v* $-$ *c-view*) $\leq$ *Suc n* $\Longrightarrow$
situation-has-finitely-many-validators s $\Longrightarrow$
nth-ancestor s (*nat* (*v* $-$ *c-view*)) *x* $\neq$ *Some y* $\Longrightarrow$
two-thirds-sent-message s (*Commit* (*y*, *c-view*)) $\Longrightarrow$
prepared s x v vs1 $\Longrightarrow$
vs1 $<$ *v* $\Longrightarrow$
$\neg$ *vs1* $<$ *c-view* $\Longrightarrow$
$\neg$ *c-view* $\leq$ $-$ *1* $\Longrightarrow$
$\forall$ *x y v.*
nat (*v* $-$ *c-view*) $\leq$ *n* $\longrightarrow$
c-view $\leq$ *v* $\longrightarrow$
situation-has-finitely-many-validators s $\longrightarrow$
nth-ancestor s (*nat* (*v* $-$ *c-view*)) *x* $\neq$ *Some y* $\longrightarrow$
two-thirds-sent-message s (*Commit* (*y*, *c-view*)) $\longrightarrow$
($\forall$ *vs1. prepared s x v vs1* $\longrightarrow$ $-$ *1* $\leq$ *vs1* $\longrightarrow$ *vs1* $<$ *v* $\longrightarrow$ *one-third-slashed s*) $\Longrightarrow$
one-third-slashed s
⟨*proof*⟩

The following lemma is a core of the accountable safety proof. It requires the mathematical induction.

lemma *safety-sub-ind'* :
 ∀ *c-view s x y v vs1*.
  n ≥ *nat* (*v* − *c-view*) ⟶
  v ≥ *c-view* ⟶
  situation-has-finitely-many-validators s ⟶
  nth-ancestor s (*nat* (*v* − *c-view*)) *x* ≠ *Some y* ⟶
  two-thirds-sent-message s (*Commit* (*y*, *c-view*)) ⟶
  prepared s x v vs1 ⟶
  − *1* ≤ *vs1* ⟶ *vs1* < *v* ⟶ *one-third-slashed s*
⟨*proof*⟩

lemma *safety-sub-ind''* :
 n = *nat* (*v* − *c-view*) ⟹
  v ≥ *c-view* ⟹
  situation-has-finitely-many-validators s ⟹
  nth-ancestor s n x ≠ *Some y* ⟹
  two-thirds-sent-message s (*Commit* (*y*, *c-view*)) ⟹
  prepared s x v vs1 ⟹
  − *1* ≤ *vs1* ⟹ *vs1* < *v* ⟹ *one-third-slashed s*
⟨*proof*⟩

lemma *not-on-chain-not-ancestor* [*simp*] :
 not-on-same-chain s x y ⟹
  nth-ancestor s m x ≠ *Some y*
⟨*proof*⟩

lemma *safety-sub-ind* :
 situation-has-finitely-many-validators s ⟶
  not-on-same-chain s x y ⟶
  two-thirds-sent-message s (*Commit* (*x*, *v1*)) ⟶
  two-thirds-sent-message s (*Commit* (*y*, *v2*)) ⟶
  prepared s x v1' vs1 ⟶
  prepared s y v2' vs2 ⟶
  v1' ≥ *v2* ∨ *v2'* ≥ *v1* ⟶
  − *1* ≤ *vs1* ⟶ *vs1* < *v1'* ⟶ − *1* ≤ *vs2* ⟶ *vs2* < *v2'* ⟶ *one-third-slashed s*
⟨*proof*⟩

lemma *safety-sub-closer* :
 situation-has-finitely-many-validators s ⟶
  not-on-same-chain s x y ⟶
  two-thirds-sent-message s (*Commit* (*x*, *v1*)) ⟶
  two-thirds-sent-message s (*Commit* (*y*, *v2*)) ⟶
  prepared s x v1 vs1 ⟶
  prepared s y v2 vs2 ⟶
  v2 ≤ *v1* ∨ *v1* ≤ *v2* ⟶

$-1 \leq vs1 \longrightarrow vs1 < v1 \longrightarrow -1 \leq vs2 \longrightarrow vs2 < v2 \longrightarrow$ *one-third-slashed s*
⟨*proof*⟩

lemma *view-total* [*simp*]:
$(v2 :: view) \leq v1 \vee v1 \leq v2$
⟨*proof*⟩

lemma *safety-sub'* :
situation-has-finitely-many-validators s $\Longrightarrow$
not-on-same-chain s x y $\Longrightarrow$
two-thirds-sent-message s (*Commit* (*x*, *v1*)) $\Longrightarrow$
two-thirds-sent-message s (*Commit* (*y*, *v2*)) $\Longrightarrow$
prepared s x v1 vs1 $\Longrightarrow$
prepared s y v2 vs2 $\Longrightarrow$
$-1 \leq vs1 \Longrightarrow vs1 < v1 \Longrightarrow -1 \leq vs2 \Longrightarrow vs2 < v2 \Longrightarrow$ *one-third-slashed s*
⟨*proof*⟩

lemma *accountable-safety-sub* :
situation-has-finitely-many-validators s $\Longrightarrow$
$\exists$ *v1 vs1*. *two-thirds-sent-message s* (*Commit* (*x*, *v1*)) $\wedge$ *prepared s x v1 vs1* $\wedge$
$-1 \leq vs1 \wedge vs1 < v1 \Longrightarrow$
$\exists$ *v2 vs2*. *two-thirds-sent-message s* (*Commit* (*y*, *v2*)) $\wedge$ *prepared s y v2 vs2* $\wedge$
$-1 \leq vs2 \wedge vs2 < v2 \Longrightarrow$
not-on-same-chain s x y $\Longrightarrow$
one-third-slashed s

⟨*proof*⟩

4 Accountable Safety (don't skip)

lemma *accountable-safety* :
situation-has-finitely-many-validators s $\Longrightarrow$
committed s x $\Longrightarrow$ *committed s y* $\Longrightarrow$
not-on-same-chain s x y $\Longrightarrow$ *one-third-slashed s*
⟨*proof*⟩

5 More Terminology for Liveness

definition *view-of-message* :: *message* $\Rightarrow$ *view*
where
view-of-message m = (*case m of*
Commit (*h*, *v*) $\Rightarrow$ *v*
| *Prepare* (*h*, *v*, *v-src*) $\Rightarrow$ *v*)

definition *message-has-valid-view* :: *message* $\Rightarrow$ *bool*
where
message-has-valid-view m = (*case m of*
Commit (*h*,*v*) $\Rightarrow$ $0 \leq v$

$|$ *Prepare* $(h,\ v,\ v\text{-}src) \Rightarrow -1 \leq v)$

definition *view-of-sent-message* :: $(validator * message) \Rightarrow view$
where
view-of-sent-message $=$ *view-of-message o snd*

definition *no-invalid-view* :: *situation* $\Rightarrow$ *bool*
where
no-invalid-view s $=$
$(\forall\ n\ m.\ (n,\ m) \in Messages\ s \longrightarrow$
message-has-valid-view m$)$

definition *finite-messages* :: *situation* $\Rightarrow$ *bool*
where
finite-messages s $=$ *finite* $(Messages\ s)$

definition *new-descendant-available* :: *situation* $\Rightarrow$ *bool*
where
new-descendant-available s $=$
$(\forall\ n\ h\ v\ diff.$
$(n,\ Commit\ (h,\ v)) \in Messages\ s \longrightarrow$
$(\exists\ h\text{-}new.\ nth\text{-}ancestor\ s\ diff\ h\text{-}new = Some\ h \wedge \neg\ committed\ s\ h\text{-}new))$

definition *authors* :: $(validator * message)\ set \Rightarrow validator\ set$
where
authors ms $= \{n.\ \exists\ m.\ (n,\ m) \in ms\}$

definition *unslashed-validators* :: *situation* $\Rightarrow$ *validator set*
where
unslashed-validators s $= \{n \in Validators\ s.\ \neg\ slashed\ s\ n\}$

definition *unslashed-can-extend* :: *situation* $\Rightarrow$ *situation* $\Rightarrow$ *bool*
where
unslashed-can-extend s s-new $=$
$(\exists\ new\text{-}messages.$
authors new-messages $\subseteq$ *unslashed-validators s* $\wedge$
Validators s-new $=$ *Validators s* $\wedge$
Messages s-new $=$ *Messages s* $\cup$ *new-messages* $\wedge$
PrevHash s-new $=$ *PrevHash s-new*$)$

definition *no-new-slashed* :: *situation* $\Rightarrow$ *situation* $\Rightarrow$ *bool*
where
no-new-slashed s s-new $=$
$(\forall\ n.\ n \in Validators\ s \longrightarrow slashed\ s\text{-}new\ n \longrightarrow slashed\ s\ n)$

6 Useful Lemmas for Plausible Liveness (skippable)

definition *no-commits-by-honest* :: *situation* $\Rightarrow$ *bool*

where
no-commits-by-honest s =
(∀ *n* ∈ *Validators s*. (∀ *h v*.
(*n*, *Commit* (*h*, *v*)) ∈ *Messages s* ⟶ *slashed s n*
))

definition *no-messages-by-honest* :: *situation* ⇒ *bool*
where
no-messages-by-honest s =
(∀ *n* ∈ *Validators s*. (∀ *m*. (*n*, *m*) ∈ *Messages s* ⟶ *slashed s n*))

definition *some-commits-by-honest-at* :: *situation* ⇒ *view* ⇒ *bool*
where
some-commits-by-honest-at s v =
(∃ *n* ∈ *Validators s*.
¬ *slashed s n* ∧
(∃ *h*. (*n*, *Commit* (*h*, *v*)) ∈ *Messages s*))

definition *some-messages-by-honest-at* :: *situation* ⇒ *view* ⇒ *bool*
where
some-messages-by-honest-at s v =
(∃ *n* ∈ *Validators s*.
¬ *slashed s n* ∧
(∃ *m*. *view-of-message m* = *v* ∧
(*n*, *m*) ∈ *Messages s*))

definition *no-commits-by-honest-after* :: *situation* ⇒ *view* ⇒ *bool*
where
no-commits-by-honest-after s v-latest =
(∀ *n* ∈ *Validators s*. (∀ *h v*.
(*n*, *Commit* (*h*, *v*)) ∈ *Messages s* ⟶
v ≤ *v-latest* ∨ *slashed s n*
))

definition *no-messages-by-honest-after* :: *situation* ⇒ *view* ⇒ *bool*
where
no-messages-by-honest-after s v-latest =
(∀ *n* ∈ *Validators s*. (∀ *m*.
(*n*, *m*) ∈ *Messages s* ⟶
view-of-message m ≤ *v-latest* ∨ *slashed s n*))

lemma *some-commits-by-honest-intro* :
∃ *n*∈*Validators s*. (∃ *h v*. (*n*, *Commit* (*h*, *v*)) ∈ *Messages s*) ∧ ¬ *slashed s n* ⟹
{*M1*. *some-commits-by-honest-at s M1*} ≠ {}
⟨*proof*⟩

lemma *some-messages-by-honest-intro* :

$\exists\, n{\in}\mathit{Validators}\ s.\ (\exists\, m.\ (n,\ m) \in \mathit{Messages}\ s) \wedge \neg\ \mathit{slashed}\ s\ n \Longrightarrow$
$\{\mathit{M1}.\ \mathit{some\text{-}messages\text{-}by\text{-}honest\text{-}at}\ s\ \mathit{M1}\} \neq \{\}$
⟨*proof*⟩

lemma *finite-commits-by-honest* :
finite-messages $s \Longrightarrow$
finite $\{\mathit{M1}.\ \mathit{some\text{-}commits\text{-}by\text{-}honest\text{-}at}\ s\ \mathit{M1}\}$
⟨*proof*⟩

lemma *finite-messages-by-honest* :
finite-messages $s \Longrightarrow$
finite $\{\mathit{M1}.\ \mathit{some\text{-}messages\text{-}by\text{-}honest\text{-}at}\ s\ \mathit{M1}\}$
⟨*proof*⟩

lemma *view-some-arithmetics* :
$(v :: \mathit{view}) \leq x \vee v \neq x$
⟨*proof*⟩

lemma *finite-views-have-max* :
$\mathit{finite}\ (\mathit{views} :: \mathit{view\ set}) \Longrightarrow \mathit{views} \neq \{\} \Longrightarrow$
$\exists\ \mathit{v\text{-}max}.$
$\mathit{v\text{-}max} \in \mathit{views} \wedge (\forall\ v.\ v \leq \mathit{v\text{-}max} \vee v \notin \mathit{views})$
⟨*proof*⟩

lemma *M1-prop-sub2* :
$\exists\ \mathit{v\text{-}max}.\ \mathit{v\text{-}max} \in \{\mathit{M1}.\ \mathit{some\text{-}commits\text{-}by\text{-}honest\text{-}at}\ s\ \mathit{M1}\}$
$\wedge (\forall\ v.\ v \leq \mathit{v\text{-}max} \vee v \notin \{\mathit{M1}.\ \mathit{some\text{-}commits\text{-}by\text{-}honest\text{-}at}\ s\ \mathit{M1}\})$
$\Longrightarrow$
$\exists\, \mathit{M1}.\ \mathit{M1} = -\ 1 \wedge (\forall\, n{\in}\mathit{Validators}\ s.\ (\exists\, h\ v.\ (n,\ \mathit{Commit}\ (h,\ v)) \in \mathit{Messages}\ s)$
$\longrightarrow \mathit{slashed}\ s\ n) \vee$
$\mathit{some\text{-}commits\text{-}by\text{-}honest\text{-}at}\ s\ \mathit{M1} \wedge \mathit{no\text{-}commits\text{-}by\text{-}honest\text{-}after}\ s\ \mathit{M1}$
⟨*proof*⟩

lemma *M1-properties* :
finite-messages $s \Longrightarrow$
$\exists\ \mathit{M1}.$
$(\mathit{M1} = -1 \wedge \mathit{no\text{-}commits\text{-}by\text{-}honest}\ s)$
$\vee\ (\mathit{some\text{-}commits\text{-}by\text{-}honest\text{-}at}\ s\ \mathit{M1} \wedge \mathit{no\text{-}commits\text{-}by\text{-}honest\text{-}after}\ s\ \mathit{M1})$

⟨*proof*⟩

lemma *M2-prop-sub2* :
$\exists\ \mathit{v\text{-}max}.\ \mathit{v\text{-}max} \in \{\mathit{M1}.\ \mathit{some\text{-}messages\text{-}by\text{-}honest\text{-}at}\ s\ \mathit{M1}\}$
$\wedge (\forall\ v.\ v \leq \mathit{v\text{-}max} \vee v \notin \{\mathit{M1}.\ \mathit{some\text{-}messages\text{-}by\text{-}honest\text{-}at}\ s\ \mathit{M1}\})$
$\Longrightarrow$
$\exists\, \mathit{M2}.\ \mathit{M2} = -\ 1 \wedge (\forall\, n{\in}\mathit{Validators}\ s.\ (\exists\, m.\ (n,\ m) \in \mathit{Messages}\ s) \longrightarrow \mathit{slashed}\ s$

$n) \vee$
$\quad$ *some-messages-by-honest-at s M2* $\wedge$ *no-messages-by-honest-after s M2*
⟨*proof*⟩

lemma *M2-properties*:
finite-messages s $\Longrightarrow$
$\exists$ *M2*.
(*M2* $= -1 \wedge$ *no-messages-by-honest s*)
$\vee$ (*some-messages-by-honest-at s M2* $\wedge$ *no-messages-by-honest-after s M2*)

⟨*proof*⟩

lemma *no-messages-no-commits* [*simp*] :
no-messages-by-honest s $\Longrightarrow$ *no-commits-by-honest s*
⟨*proof*⟩

lemma *no-messages-then-no-messages-after* [*simp*] :
no-messages-by-honest s $\Longrightarrow$ *no-messages-by-honest-after s m*
⟨*proof*⟩

lemma *no-messages-denies-somit-commits-at* [*simp*] :
no-messages-by-honest s $\Longrightarrow$
$\neg$ *some-commits-by-honest-at s m*
⟨*proof*⟩

lemma *no-messages-denies-some-messages-at* [*simp*] :
no-messages-by-honest s $\Longrightarrow$
$\neg$ *some-messages-by-honest-at s m*
⟨*proof*⟩

lemma *no-commits-denies-some-commits-at* [*simp*] :
no-commits-by-honest s $\Longrightarrow$
no-commits-by-honest-after s M1
⟨*proof*⟩

definition *liveness-witness* :: *hash* $\Rightarrow$ *view* $\Rightarrow$ *view* $\Rightarrow$ *validator set* $\Rightarrow$ (*validator* $*$ *message*) *set*
where
liveness-witness h M1 M2 ns =
$\{(n,$ *Prepare* $(h,$ *M2* $+ 1,$ *M1*$))$ $\mid n.\ n \in ns\} \cup$
$\{(n,$ *Commit* $(h,$ *M2* $+ 1))$ $\mid n.\ n \in ns\}$

lemma *author-of-witness* [*simp*] :
authors (*liveness-witness h-new M1 M2 ns*) $=$ *ns*
⟨*proof*⟩

lemma *unslashed-can-use-witness* [*simp*]:
unslashed-can-extend s

$(\!|\mathit{Validators} = \mathit{Validators}\ s,$
$\mathit{Messages} = \mathit{Messages}\ s \cup \mathit{liveness\text{-}witness}\ \mathit{h\text{-}new}\ \mathit{M1}\ \mathit{M2}\ \{n \in \mathit{Validators}\ s.\ \neg \mathit{slashed}\ s\ n\},$
$\mathit{PrevHash} = \mathit{PrevHash}\ s|\!)$
⟨*proof*⟩

lemma *more-than-two-thirds-mp* :
$\mathit{finite}\ (\mathit{Validators}\ s) \Longrightarrow$
$\forall\ n.\ n \in \mathit{Validators}\ s \longrightarrow f\ n \longrightarrow g\ n \Longrightarrow$
$\mathit{more\text{-}than\text{-}two\text{-}thirds}\ s\ f \Longrightarrow \mathit{more\text{-}than\text{-}two\text{-}thirds}\ s\ g$
⟨*proof*⟩

lemma *witness-commits-inner* :
$\mathit{Validators}\ s \neq \{\} \wedge \mathit{finite}\ (\mathit{Validators}\ s) \Longrightarrow$
$\mathit{more\text{-}than\text{-}two\text{-}thirds}\ s\ (\lambda n.\ \neg\ \mathit{slashed}\ s\ n) \Longrightarrow$
$2 * \mathit{card}\ (\mathit{Validators}\ s)$
$\leq 3 * \mathit{card}\ \{n \in \mathit{Validators}\ s.\ (n,\ \mathit{Commit}\ (\mathit{h\text{-}new},\ \mathit{M2} + 1)) \in \mathit{Messages}\ s \vee n \in \mathit{Validators}\ s \wedge \neg\ \mathit{slashed}\ s\ n\}$
⟨*proof*⟩

lemma *witness-commits* [*simp*] :
$\mathit{situation\text{-}has\text{-}finitely\text{-}many\text{-}validators}\ s \Longrightarrow$
$\neg\ \mathit{one\text{-}third\text{-}slashed}\ s \Longrightarrow$
$\mathit{committed}$
$(\!|\mathit{Validators} = \mathit{Validators}\ s,$
$\mathit{Messages} = \mathit{Messages}\ s \cup \mathit{liveness\text{-}witness}\ \mathit{h\text{-}new}\ \mathit{M1}\ \mathit{M2}\ \{n \in \mathit{Validators}\ s.\ \neg \mathit{slashed}\ s\ n\},$
$\mathit{PrevHash} = \mathit{PrevHash}\ s|\!)$
$\mathit{h\text{-}new}$
⟨*proof*⟩

lemma *two-thirds-transfer* [*simp*] :
$\mathit{two\text{-}thirds}\ (\!|\ \mathit{Validators} = \mathit{Validators}\ s,\ \mathit{Messages} = X,\ \mathit{PrevHash} = Y\ |\!)\ g =$
$\mathit{two\text{-}thirds}\ s\ g$
⟨*proof*⟩

lemma *prepared-transfer* :
$\mathit{finite}\ (\mathit{Validators}\ s) \Longrightarrow$
$\mathit{prepared}\ s\ \mathit{ha}\ v\ \mathit{vs} \Longrightarrow$
$\mathit{prepared}$
$(\!|\mathit{Validators} = \mathit{Validators}\ s,$
$\mathit{Messages} = \mathit{Messages}\ s \cup X,$
$\mathit{PrevHash} = \mathit{PrevHash}\ s|\!)$
$\mathit{ha}\ v\ \mathit{vs}$
⟨*proof*⟩

lemma *witness-has-certain-view* :
$(\mathit{na},\ \mathit{Commit}\ (\mathit{ha},\ v)) \in \mathit{liveness\text{-}witness}\ \mathit{h\text{-}new}\ \mathit{M1}\ \mathit{M2}\ \{n \in \mathit{Validators}\ s.\ \neg\ \mathit{slashed}\ s\ n\} \Longrightarrow$

$v = M2 + 1$

⟨*proof*⟩

lemma *witness-has-certain-hash* :
$(na,\ Commit\ (ha,\ v)) \in liveness\text{-}witness\ h\text{-}new\ M1\ M2\ \{n \in Validators\ s.\ \neg\ slashed\ s\ n\} \Longrightarrow$
$ha = h\text{-}new$

⟨*proof*⟩

lemma *more-than-two-thirds-imply-two-thirds* :
more-than-two-thirds s f $\Longrightarrow$
two-thirds s f
⟨*proof*⟩

lemma *witness-prepares*:
situation-has-finitely-many-validators s $\Longrightarrow$
$\neg$ *one-third-slashed s* $\Longrightarrow$
prepared
$(\!|\ Validators = Validators\ s,$
$Messages = Messages\ s \cup liveness\text{-}witness\ h\text{-}new\ M1\ M2\ \{n \in Validators\ s.\ \neg\ slashed\ s\ n\},$
$PrevHash = PrevHash\ s\ |\!)$
h-new $(M2 + 1)\ M1$
⟨*proof*⟩

lemma *commit-can-be-after-neg-one*:
situation-has-finitely-many-validators s $\Longrightarrow$
$n \in Validators\ s \Longrightarrow$
$\neg$ *slashed s n* $\Longrightarrow$
$(n,\ Commit\ (h,\ M1)) \in Messages\ s \Longrightarrow$
$-\ 1 \leq M1$

⟨*proof*⟩

lemma *witness-not-slashed-one* :
situation-has-finitely-many-validators s $\Longrightarrow$
$\neg$ *one-third-slashed s* $\Longrightarrow$
no-messages-by-honest-after s M2 $\Longrightarrow$
$n \in Validators\ s \Longrightarrow$
$\neg$ *slashed s n* $\Longrightarrow$
$(n,\ Commit\ (h,\ M1)) \in Messages\ s \Longrightarrow$
$na \in Validators\ s \Longrightarrow$
slashed-one
$(\!|\ Validators = Validators\ s,$
$Messages = Messages\ s \cup liveness\text{-}witness\ h\text{-}new\ M1\ M2\ \{n \in Validators\ s.\ \neg\ slashed\ s\ n\},$

PrevHash = PrevHash s⦈
na ⟹
slashed-one s na
⟨*proof*⟩

lemma *nth-ancestor-transfers* [*simp*] :
∀ *N M s ha*.
nth-ancestor
(⦇*Validators* = *N*,
Messages = *M*,
PrevHash = *PrevHash s*⦈
n ha =
nth-ancestor s n ha
⟨*proof*⟩

lemma *witness-prepares-certain-hash* :
(*na*, *Prepare* (*ha*, *v*, *vs*)) ∈ *liveness-witness h-new M1 M2 validators* ⟹
ha = *h-new*
⟨*proof*⟩

lemma *witness-prepares-certain-view* :
(*na*, *Prepare* (*ha*, *v*, *vs*)) ∈ *liveness-witness h-new M1 M2 validators* ⟹
v = *M2* + *1*
⟨*proof*⟩

lemma *witness-commits-certain-view* :
(*n*, *Commit* (*h*, *v*)) ∈ *liveness-witness h-new M1 M2 validators* ⟹
v = *M2* + *1*
⟨*proof*⟩

lemma *witness-prepares-certain-view-src* :
(*na*, *Prepare* (*ha*, *v*, *vs*)) ∈ *liveness-witness h-new M1 M2 validators* ⟹
vs = *M1*
⟨*proof*⟩

lemma *it-is-somebody-that-prepares* :
situation-has-finitely-many-validators s ⟹
¬ *one-third-slashed s* ⟹
prepared s h v v-src ⟹
∃ *n*. *n* ∈ *Validators s* ∧
¬ *slashed s n* ∧
(*n*, *Prepare* (*h*, *v*, *v-src*)) ∈ *Messages s*

⟨*proof*⟩

lemma *slashed-two-transfers* :
situation-has-finitely-many-validators s ⟹
¬ *one-third-slashed s* ⟹
no-messages-by-honest-after s M2 ⟹

$n \in$ *Validators* $s \Longrightarrow$
$\neg$ *slashed* s $n \Longrightarrow$
$(n,$ *Commit* $(h,$ *M1*$)) \in$ *Messages* $s \Longrightarrow$
nth-ancestor s (*nat* (*M2* + 1 − *M1*)) *h-new* = *Some* $h \Longrightarrow$
$na \in$ *Validators* $s \Longrightarrow$
slashed-two
(| *Validators* = *Validators* s,
Messages = *Messages* $s \cup$ *liveness-witness h-new M1 M2* $\{n \in$ *Validators* $s. \neg$ *slashed* s $n\}$,
PrevHash = *PrevHash* s |)
$na \Longrightarrow$
slashed-two s na
⟨*proof*⟩

lemma *no-prepare-after* :
no-messages-by-honest-after s *M2* $\Longrightarrow$
$na \in$ *Validators* $s \Longrightarrow$
$(na,$ *Prepare* $(y,$ $w,$ $u)) \in$ *Messages* $s \Longrightarrow$
$\neg$ *slashed* s $na \Longrightarrow$ $w \leq$ *M2*
⟨*proof*⟩

lemma *slashed-intro-three* :
slashed-three s $n \Longrightarrow$ *slashed* s n
⟨*proof*⟩

lemma *no-commit-after* :
no-commits-by-honest-after s *M1* $\Longrightarrow$
$na \in$ *Validators* $s \Longrightarrow$
$(na,$ *Commit* $(x,$ $v)) \in$ *Messages* $s \Longrightarrow \neg$ *slashed* s $na \Longrightarrow$ $v \leq$ *M1*
⟨*proof*⟩

lemma *slashed-three-transfers* :
situation-has-finitely-many-validators $s \Longrightarrow$
$\neg$ *one-third-slashed* $s \Longrightarrow$
no-commits-by-honest-after s *M1* $\Longrightarrow$
no-messages-by-honest-after s *M2* $\Longrightarrow$
$na \in$ *Validators* $s \Longrightarrow$
slashed-three
(| *Validators* = *Validators* s,
Messages = *Messages* $s \cup$ *liveness-witness h-new M1 M2* $\{n \in$ *Validators* $s. \neg$ *slashed* s $n\}$,
PrevHash = *PrevHash* s |)
$na \Longrightarrow$
slashed s na

⟨*proof*⟩

lemma *slashed-four-transfers* :
no-messages-by-honest-after s M2 $\Longrightarrow$
na $\in$ *Validators s* $\Longrightarrow$
slashed-four
(| *Validators* = *Validators s*,
Messages = *Messages s* $\cup$ *liveness-witness h-new M1 M2* {*n* $\in$ *Validators s*. $\neg$ *slashed s n*},
PrevHash = *PrevHash s* |)
na $\Longrightarrow$
slashed s na
⟨*proof*⟩

lemma *slashed-destruct* :
slashed s n $\Longrightarrow$
slashed-one s n $\vee$ *slashed-two s n* $\vee$ *slashed-three s n* $\vee$ *slashed-four s n*
⟨*proof*⟩

lemma *slashed-intro-two* :
slashed-two s na $\Longrightarrow$ *slashed s na*
⟨*proof*⟩

lemma *slashed-intro-four* :
slashed-four s na $\Longrightarrow$ *slashed s na*
⟨*proof*⟩

lemma *witness-not-guilty* [*simp*]:
situation-has-finitely-many-validators s $\Longrightarrow$
finite-messages s $\Longrightarrow$
$\neg$ *one-third-slashed s* $\Longrightarrow$
$\neg$ *no-messages-by-honest s* $\Longrightarrow$
$\neg$ *no-commits-by-honest s* $\Longrightarrow$
no-commits-by-honest-after s M1 $\Longrightarrow$
some-messages-by-honest-at s M2 $\Longrightarrow$
no-messages-by-honest-after s M2 $\Longrightarrow$
n $\in$ *Validators s* $\Longrightarrow$
$\neg$ *slashed s n* $\Longrightarrow$
(*n*, *Commit* (*h*, *M1*)) $\in$ *Messages s* $\Longrightarrow$
nth-ancestor s (*nat* (*M2* + 1 − *M1*)) *h-new* = *Some h* $\Longrightarrow$
$\neg$ *committed s h-new* $\Longrightarrow$
no-new-slashed s
(| *Validators* = *Validators s*,
Messages = *Messages s* $\cup$ *liveness-witness h-new M1 M2* {*n* $\in$ *Validators s*. $\neg$ *slashed s n*},
PrevHash = *PrevHash s* |)
⟨*proof*⟩

lemma *no-messages-cannot-commit* :
situation-has-finitely-many-validators s $\Longrightarrow$

$\neg$ *one-third-slashed s* $\Longrightarrow$ *no-messages-by-honest s* $\Longrightarrow$ $\neg$ *committed s h*
⟨*proof*⟩

lemma *corner-kick* :
situation-has-finitely-many-validators s $\Longrightarrow$
finite-messages s $\Longrightarrow$
$\neg$ *one-third-slashed s* $\Longrightarrow$
no-messages-by-honest s $\Longrightarrow$
no-new-slashed s
(| *Validators* = *Validators s*,
Messages = *Messages s* $\cup$ *liveness-witness* (*Hash 0*) (− *1*) (− *1*) {*n* $\in$ *Validators s*. $\neg$ *slashed s n*},
PrevHash = *PrevHash s* |)
⟨*proof*⟩

lemma *at-least-neg-one* :
no-invalid-view s $\Longrightarrow$
some-messages-by-honest-at s M2 $\Longrightarrow$
− *1* $\leq$ *M2*
⟨*proof*⟩

lemma *no-commit-new-slashed-three*:
no-invalid-view s $\Longrightarrow$
situation-has-finitely-many-validators s $\Longrightarrow$
$\neg$ *one-third-slashed s* $\Longrightarrow$
no-commits-by-honest s $\Longrightarrow$
no-messages-by-honest-after s M2 $\Longrightarrow$
n $\in$ *Validators s* $\Longrightarrow$
$\neg$ *slashed s n* $\Longrightarrow$
slashed-three
(| *Validators* = *Validators s*,
Messages = *Messages s* $\cup$ *liveness-witness* (*Hash 0*) (− *1*) *M2* {*n* $\in$ *Validators s*. $\neg$ *slashed s n*},
PrevHash = *PrevHash s* |)
n $\Longrightarrow$
False
⟨*proof*⟩

lemma *no-commit-new-slashed-four*:
no-invalid-view s $\Longrightarrow$
situation-has-finitely-many-validators s $\Longrightarrow$
$\neg$ *one-third-slashed s* $\Longrightarrow$
no-commits-by-honest s $\Longrightarrow$
no-messages-by-honest-after s M2 $\Longrightarrow$
n $\in$ *Validators s* $\Longrightarrow$
$\neg$ *slashed s n* $\Longrightarrow$
slashed-four
(| *Validators* = *Validators s*,
Messages = *Messages s* $\cup$ *liveness-witness* (*Hash 0*) (− *1*) *M2* {*n* $\in$ *Validators*

$s.\ \neg\ slashed\ s\ n\},$
$PrevHash = PrevHash\ s\rparr$
$n \Longrightarrow$
$False$
$\langle proof \rangle$

lemma *two-thirds-sent-message-transfers* :
finite (*Validators s*) $\Longrightarrow$
two-thirds-sent-message s m $\Longrightarrow$
two-thirds-sent-message
$\lparr Validators = Validators\ s,$
$Messages =$
$Messages\ s \cup X,$
$PrevHash = PrevHash\ s\rparr\ m$
$\langle proof \rangle$

lemma *no-commit-new-slashed-two*:
no-invalid-view s $\Longrightarrow$
situation-has-finitely-many-validators s $\Longrightarrow$
$\neg$ *one-third-slashed s* $\Longrightarrow$
no-commits-by-honest s $\Longrightarrow$
some-messages-by-honest-at s M2 $\Longrightarrow$
no-messages-by-honest-after s M2 $\Longrightarrow$
$n \in Validators\ s \Longrightarrow$
$\neg\ slashed\ s\ n \Longrightarrow$
slashed-two
$\lparr Validators = Validators\ s,$
$Messages = Messages\ s \cup liveness\text{-}witness\ (Hash\ 0)\ (-\ 1)\ M2\ \{n \in Validators\ s.\ \neg\ slashed\ s\ n\},$
$PrevHash = PrevHash\ s\rparr$
$n \Longrightarrow$
$False$
$\langle proof \rangle$

lemma *corner-kick2* :
no-invalid-view s $\Longrightarrow$
situation-has-finitely-many-validators s $\Longrightarrow$
new-descendant-available s $\Longrightarrow$
$\neg$ *one-third-slashed s* $\Longrightarrow$
no-commits-by-honest s $\Longrightarrow$
some-messages-by-honest-at s M2 $\Longrightarrow$
no-messages-by-honest-after s M2 $\Longrightarrow$
$\exists\ s\text{-}new\ h\text{-}new.$
$\neg\ committed\ s\ h\text{-}new \wedge$
$unslashed\text{-}can\text{-}extend\ s\ s\text{-}new \wedge committed\ s\text{-}new\ h\text{-}new \wedge no\text{-}new\text{-}slashed\ s\ s\text{-}new$
$\langle proof \rangle$

7 Plausible Liveness (don't skip)

lemma *plausible-liveness* :
  situation-has-finitely-many-validators s $\Longrightarrow$
   no-invalid-view s $\Longrightarrow$
   new-descendant-available s $\Longrightarrow$
   finite-messages s $\Longrightarrow$
   $\neg$ *one-third-slashed s* $\Longrightarrow$
   $\exists$ *s-new h-new*.
     $\neg$ *committed s h-new* $\wedge$
     unslashed-can-extend s s-new $\wedge$
     committed s-new h-new $\wedge$
     no-new-slashed s s-new

⟨*proof*⟩

end