
MODULE *crosslink2*

EXTENDS *TLC*, *Naturals*, *Sequences*, *utils*

CONSTANTS *BcNodes*, *BftNodes*, *CrossLink2Nodes*

CONSTANTS *ByzBft*, *ByzCl*

CONSTANTS *Sigma*, *L*

VARIABLES *bc_chains*, *bft_chains*, *crosslink2_chains*

INSTANCE *definitions*

Init \triangleq

$\wedge bc_chains = [i \in 1 \dots BcNodes \mapsto \langle BcGenesisBlock \rangle]$
 $\wedge bft_chains = [i \in 1 \dots BftNodes \mapsto \langle BftGenesisBlock \rangle]$
 $\wedge crosslink2_chains = [i \in 1 \dots CrossLink2Nodes \mapsto CrossLink2GenesisBlock]$

HonestBc \triangleq

$\exists n \in 1 \dots BcNodes :$
 LET
 $base \triangleq bc_chains[BestBcChainIdx]$
 $bft \triangleq bft_chains[BestBftChainIdx]$
 $tip \triangleq base[Len(base)].hash$
 $next \triangleq tip + 1$ IN
 $\wedge bc_chains' = [bc_chains \text{ EXCEPT } ![n] = Append(base, [$
 $context_bft \mapsto bft[Len(bft)].hash,$
 $hash \mapsto next])]$
 $\wedge \text{UNCHANGED } \langle bft_chains, crosslink2_chains \rangle$

HonestBft \triangleq

$\exists n \in 1 \dots BftNodes :$
 LET
 $base \triangleq bft_chains[BestBftChainIdx]$
 $bc \triangleq bc_chains[BestBcChainIdx]$
 $tip \triangleq base[Len(base)].hash$
 $next \triangleq tip + 1$
 $hdrs \triangleq PruneLasts(bc, Sigma)$ IN
 $\wedge bft_chains' = [bft_chains \text{ EXCEPT } ![n] = Append(base, [$
 $headers_bc \mapsto hdrs,$
 $hash \mapsto next])]$
 $\wedge \text{UNCHANGED } \langle bc_chains, crosslink2_chains \rangle$

ByzantineBft \triangleq

$\exists n \in ByzBft :$
 LET
 $base \triangleq bft_chains[BestBftChainIdx]$

$ \begin{aligned} bc &\triangleq bc_chains[BestBcChainIdx] \\ tip &\triangleq base[Len(base)].hash \\ &\text{Byzantine node can create an arbitrary faulty block within a range} \\ byz &\triangleq tip + (\text{CHOOSE } inc \in 2 \dots 10 : \text{TRUE}) \\ hdrs &\triangleq PruneLasts(bc, Sigma) \text{IN} \\ \wedge bft_chains' &= [bft_chains \text{ EXCEPT } ![n] = Append(base, [\\ &\quad headers_bc \mapsto hdrs, \\ &\quad hash \mapsto byz])] \\ &\wedge \text{UNCHANGED } \langle bc_chains, crosslink2_chains \rangle \end{aligned} $
$ \begin{aligned} HonestCrosslink &\triangleq \\ &\exists n \in 1 \dots CrossLink2Nodes : \\ &\text{LET} \\ &\quad fin \triangleq PruneFirsts(bc_chains[BestBcChainIdx], Sigma) \\ &\quad ba \triangleq LocalBa(fin, bc_chains[BestBcChainIdx]) \\ &\text{IN} \\ &\wedge crosslink2_chains' = [crosslink2_chains \text{ EXCEPT } ![n] = [\\ &\quad fin \mapsto fin, \\ &\quad ba \mapsto ba]] \\ &\wedge \text{UNCHANGED } \langle bc_chains, bft_chains \rangle \\ &\vee \text{UNCHANGED } \langle bc_chains, bft_chains, crosslink2_chains \rangle \end{aligned} $
$ \begin{aligned} Next &\triangleq \\ &\vee HonestBc \\ &\vee HonestBft \\ &\vee HonestCrosslink \\ &\vee ByzantineBft \end{aligned} $
$Spec \triangleq Init \wedge \Box [Next]_{\langle bc_chains, bft_chains, crosslink2_chains \rangle}$
<hr/> <div style="background-color: #f0f0f0; padding: 2px;">Type checking</div> $ \begin{aligned} BcChainsTypeCheck &\triangleq bc_chains \in Seq(Seq([context_bft : Nat, hash : Nat])) \\ BftChainsTypeCheck &\triangleq bft_chains \in \\ &\quad Seq(Seq([headers_bc : Seq([context_bft : Nat, hash : Nat]), hash : Nat])) \\ CrossLink2ChainsTypeCheck &\triangleq crosslink2_chains \in \\ &\quad Seq([fin : Seq([context_bft : Nat, hash : Nat]), ba : Seq([context_bft : Nat, hash : Nat])]) \end{aligned} $ <hr/>
<div style="background-color: #f0f0f0; padding: 2px;">Assumptions</div> <p>ASSUME <i>BftThresholdOK</i></p>
<div style="background-color: #f0f0f0; padding: 2px;">Lemma: Linear Prefix</div> <p>If $A \preceq_\star C$ and $B \preceq_\star C$ then $A \star_\star B$.</p>

$BcLinearPrefix \triangleq$
 $\forall a, b, c \in 1 \dots BcNodes :$
 $\text{LET } A \triangleq bc_chains[a]$
 $\quad B \triangleq bc_chains[b]$
 $\quad C \triangleq bc_chains[c]$
 $\text{IN } IsPrefix(A, C) \wedge IsPrefix(B, C) \Rightarrow$
 $\quad IsPrefix(A, B) \vee IsPrefix(B, A)$

$BftLinearPrefix \triangleq$
 $\forall a, b, c \in 1 \dots BftNodes :$
 $\text{LET } A \triangleq bft_chains[a]$
 $\quad B \triangleq bft_chains[b]$
 $\quad C \triangleq bft_chains[c]$
 $\text{IN } IsPrefix(A, C) \wedge IsPrefix(B, C) \Rightarrow$
 $\quad IsPrefix(A, B) \vee IsPrefix(B, A)$

Definition: Agreement on a view

An execution of Π has Agreement on the view $V : Node \times Time \rightarrow \star chain$ iff for all times t, u and all Π nodes i, j (potentially the same) such that i is honest at time t and j is honest at time u , we have $V_i^t \star V_j^u$.

$BcViewAgreement \triangleq$
 $\forall i, j \in 1 \dots BcNodes :$
 $\quad \vee IsPrefix(bc_chains[i], bc_chains[j])$
 $\quad \vee IsPrefix(bc_chains[j], bc_chains[i])$

$BftViewAgreement \triangleq$
 $\forall i, j \in HonestBftNodes :$
 $\quad \vee IsPrefix(bft_chains[i], bft_chains[j])$
 $\quad \vee IsPrefix(bft_chains[j], bft_chains[i])$

Definition: Final agreement

An execution of $\Pi_{\star bft}$ has Final Agreement iff for all *bftvalid* blocks C in honest view at time t and C' in honest view at time t' , we have $bftlastfinal(C) \star_{bft} bftlastfinal(C')$.

$BftFinalAgreement \triangleq$
 $\forall i, j \in HonestBftNodes :$
 $\quad \vee IsPrefix(BftLastFinal(i), BftLastFinal(j))$
 $\quad \vee IsPrefix(BftLastFinal(j), BftLastFinal(i))$

Definition: Prefix Consistency

An execution of $\Pi_{\star bc}$ has Prefix Consistency at confirmation depth σ , iff for all times $t \leq u$ and all nodes i, j (potentially the same) such that i is honest at time t and j is honest at time u , we have that $ch_i^t \upharpoonright_{\star bc}^\sigma \preceq_{\star bc} ch_j^u$.

$BcPrefixConsistency \triangleq$
 $\forall i, j \in 1 \dots BcNodes :$
 $\quad Len(bc_chains[i]) \leq Len(bc_chains[j]) \Rightarrow$

$$IsPrefix(PruneFirsts(bc_chains[i], Sigma), bc_chains[j])$$

Definition: Prefix Agreement

An execution of $\Pi_{\star bc}$ has Prefix Agreement at confirmation depth σ iff it has Agreement on the view $(i, t) \mapsto ch_i^t \upharpoonright_{\star bc}^\sigma$.

$$BcPrefixAgreement \triangleq$$

$$\forall i \in 1 \dots BcNodes :$$

$$IsPrefix(PruneFirsts(bc_chains[i], Sigma), bc_chains[i])$$

Definition: *-linear

A function $S : I \rightarrow \star block$ is *-linear iff for every $t, u \in I$ where $t \leq u$ we have $S(t) \preceq_\star S(u)$

$$BcLinear(T, U) \triangleq IsPrefix(T, U)$$

Definition: Local finalization linearity

Node i has Local finalization linearity up to time t iff the time series of $\star bc$ -blocks $fin_i^{r \leq t}$ is $\star bc$ -linear.

$$LocalFinalizationLinearity \triangleq \square[$$

$$\forall i \in 1 \dots CrossLink2Nodes :$$

$$BcLinear(crosslink2_chains[i].fin, crosslink2_chains'[i].fin)]_{crosslink2_chains}$$

Lemma: Local fin-depth

In any execution of Crosslink 2, for any node i that is honest at time t , there exists a time $r \leq t$ such that $fin_i \preceq ch_i^r \upharpoonright_{\star bc}^\sigma$

$$LocalFinDepth \triangleq$$

$$\forall i \in 1 \dots CrossLink2Nodes :$$

$$IsPrefix(crosslink2_chains[i].fin, bc_chains[BestBcChainIdx])$$

Definition: Assured Finality

An execution of Crosslink 2 has Assured Finality iff for all times t, u and all nodes i, j (potentially the same) such that i is honest at time t and j is honest at time u , we have $fin_i^t \not\preceq_{bc} fin_j^u$.

$$AssuredFinality \triangleq$$

$$\forall i, j \in 1 \dots CrossLink2Nodes :$$

$$\vee IsPrefix(crosslink2_chains[i].fin, crosslink2_chains[j].fin)$$

$$\vee IsPrefix(crosslink2_chains[j].fin, crosslink2_chains[i].fin)$$

Theorem: Ledger prefix property

For any node i that is honest at time t , and any confirmation depth μ , $fin_i^t \preceq (ba_\mu)_i^t$

$$LedgerPrefixProperty \triangleq$$

$$\forall i \in 1 \dots CrossLink2Nodes :$$

$$IsPrefix(crosslink2_chains[i].fin, crosslink2_chains[i].ba)$$

Lemma: Local ba-depth

In any execution of Crosslink 2, for any confirmation depth $\mu \leq \sigma$ and any node i that is honest at time t , there exists a time $r \leq t$ such that $(ba_\mu)_i^t \preceq_{bc} ch_i^r \upharpoonright_{bc}^\mu$.

$$\begin{aligned}
LocalBaDepth \triangleq & \square[\\
& \forall i \in 1 \dots CrossLink2Nodes : \\
& \quad IsPrefix(crosslink2_chains'[i].ba, PruneFirsts(bc_chains[BestBcChainIdx], Sigma)) \\
&]_{crosslink2_chains}
\end{aligned}$$
