$\qquad\qquad$ MODULE $crosslink2$ $\qquad\qquad$

EXTENDS TLC, $Naturals$, $Sequences$, $utils$

CONSTANTS $BcNodes$, $BftNodes$, $CrossLink2Nodes$
CONSTANTS $Sigma$, L

VARIABLES bc_chains, bft_chains, $crosslink2\_chains$

INSTANCE $definitions$

$Init \triangleq$
$\quad\wedge bc\_chains = [i \in 1 .. BcNodes \mapsto \langle BcGenesisBlock \rangle]$
$\quad\wedge bft\_chains = [i \in 1 .. BftNodes \mapsto \langle BftGenesisBlock \rangle]$
$\quad\wedge crosslink2\_chains = [i \in 1 .. CrossLink2Nodes \mapsto CrossLink2GenesisBlock]$

$Next \triangleq$
$\quad\vee \exists\, n \in 1 .. BcNodes :$
$\quad\quad\wedge bc\_chains' = [bc\_chains \text{ EXCEPT } ![n] = Append($
$\quad\quad\quad bc\_chains[ChooseBestBcChain], [$
$\quad\quad\quad\quad context\_bft \mapsto ChooseContextBft,$
$\quad\quad\quad\quad hash \mapsto ChooseBestBcTip + 1])]$
$\quad\quad\wedge \text{UNCHANGED } \langle bft\_chains, crosslink2\_chains \rangle$
$\quad\vee \exists\, m \in 1 .. BftNodes :$
$\quad\quad\wedge bft\_chains' = [bft\_chains \text{ EXCEPT } ![m] = Append($
$\quad\quad\quad bft\_chains[ChooseBestBftChain], [$
$\quad\quad\quad\quad headers\_bc \mapsto PruneLasts(ChooseBcView, Sigma),$
$\quad\quad\quad\quad hash \mapsto ChooseBestBftTip + 1])]$
$\quad\quad\wedge \text{UNCHANGED } \langle bc\_chains, crosslink2\_chains \rangle$
$\quad\vee \exists\, c \in 1 .. CrossLink2Nodes :$
$\quad\quad\wedge crosslink2\_chains' = [crosslink2\_chains \text{ EXCEPT } ![c] = [$
$\quad\quad\quad fin \mapsto PruneFirsts(bc\_chains[ChooseBestBcChain], Sigma)]]$
$\quad\quad\wedge \text{UNCHANGED } \langle bc\_chains, bft\_chains \rangle$
$\quad\vee \text{UNCHANGED } \langle bc\_chains, bft\_chains, crosslink2\_chains \rangle$

$Spec \triangleq Init \wedge \Box[Next]_{\langle bc\_chains,\, bft\_chains,\, crosslink2\_chains \rangle}$

Type checking

$BcChainsTypeCheck \triangleq bc\_chains \in Seq(Seq([context\_bft : Nat, hash : Nat]))$
$BftChainsTypeCheck \triangleq bft\_chains \in$
$\quad Seq(Seq([headers\_bc : Seq([context\_bft : Nat, hash : Nat]), hash : Nat]))$
$CrossLink2ChainsTypeCheck \triangleq crosslink2\_chains \in$
$\quad Seq([fin : Seq([context\_bft : Nat, hash : Nat])])$

If $A \preceq_\star C$ and $B \preceq_\star C$ then $A \stackrel{*}{\preceq}_\star B$.

$BcLinearPrefix \stackrel{\Delta}{=}$
  $\forall\, a,\, b,\, c \in 1\,.\,.\, BcNodes :$
    LET $A \stackrel{\Delta}{=} bc\_chains[a]$
      $B \stackrel{\Delta}{=} bc\_chains[b]$
      $C \stackrel{\Delta}{=} bc\_chains[c]$
    IN $IsPrefix(A,\, C) \land IsPrefix(B,\, C) \Rightarrow$
      $IsPrefix(A,\, B) \lor IsPrefix(B,\, A)$

$BftLinearPrefix \stackrel{\Delta}{=}$
  $\forall\, a,\, b,\, c \in 1\,.\,.\, BftNodes :$
    LET $A \stackrel{\Delta}{=} bft\_chains[a]$
      $B \stackrel{\Delta}{=} bft\_chains[b]$
      $C \stackrel{\Delta}{=} bft\_chains[c]$
    IN $IsPrefix(A,\, C) \land IsPrefix(B,\, C) \Rightarrow$
      $IsPrefix(A,\, B) \lor IsPrefix(B,\, A)$

An execution of Π has Agreement on the view $V : Node \times Time \to \star chain$ 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 $V_i^t \stackrel{*}{\preceq}_\star V_j^u$.

$BcViewAgreement \stackrel{\Delta}{=}$
  $\forall\, i,\, j \in 1\,.\,.\, BcNodes :$
    $\lor IsPrefix(bc\_chains[i],\, bc\_chains[j])$
    $\lor IsPrefix(bc\_chains[j],\, bc\_chains[i])$

$BftViewAgreement \stackrel{\Delta}{=}$
  $\forall\, i,\, j \in 1\,.\,.\, BftNodes :$
    $\lor IsPrefix(bft\_chains[i],\, bft\_chains[j])$
    $\lor IsPrefix(bft\_chains[j],\, bft\_chains[i])$

An execution of $\Pi_{\star bft}$ has Final Agreement iff for all $bftvalid$ blocks C in honest view at time t and $C\prime$ in honest view at time $t\prime$, we have $bftlastfinal(C) \stackrel{*}{\preceq}_{bft} \star bftlastfinal(C\prime)$.

$BftFinalAgreement \stackrel{\Delta}{=}$
  $\forall\, i,\, j \in 1\,.\,.\, BftNodes :$
    $\lor IsPrefix(BftLastFinal(i),\, BftLastFinal(j))$
    $\lor IsPrefix(BftLastFinal(j),\, BftLastFinal(i))$

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 $ch_i^t \lceil_{\star bc}^{\sigma} \preceq_{\star bc} ch_j^u$.

$BcPrefixConsistency \triangleq$
    $\forall\, i, j \in 1\,..\,BcNodes :$
        $Len(bc\_chains[i]) \leq Len(bc\_chains[j]) \Rightarrow$
            $IsPrefix(PruneFirsts(bc\_chains[i],\ Sigma),\ bc\_chains[j])$

Definition: Prefix Agreement

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

$BcPrefixAgreement \triangleq$
    $\forall\, i \in 1\,..\,BcNodes :$
        $IsPrefix(PruneFirsts(bc\_chains[i],\ Sigma),\ bc\_chains[i])$

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.

$LocalFinalizationLinearity \triangleq \Box[$
    $\forall\, i \in 1\,..\,CrossLink2Nodes :$
        $BcLinear(crosslink2\_chains[i].fin,\ crosslink2\_chains'[i].fin)]_{crosslink2\_chains}$

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 $fin_i \preceq ch_i^r \lceil_{\star bc}^{\sigma}$

$LocalFinDepth \triangleq$
    $\forall\, i \in 1\,..\,CrossLink2Nodes :$
        $IsPrefix(crosslink2\_chains[i].fin,\ bc\_chains[ChooseBestBcChain])$

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 $fin_i^t \underset{bc}{\preceq\!\succeq} fin_j^u$.

$AssuredFinality \triangleq$
    $\forall\, i, j \in 1\,..\,CrossLink2Nodes :$
        $\vee\ IsPrefix(crosslink2\_chains[i].fin,\ crosslink2\_chains[j].fin)$
        $\vee\ IsPrefix(crosslink2\_chains[j].fin,\ crosslink2\_chains[i].fin)$