MODULE *definitions*

LOCAL INSTANCE *Integers*
LOCAL INSTANCE *Sequences*
LOCAL INSTANCE *utils*

CONSTANTS *BcNodes*, *BftNodes*, *CrossLink2Nodes*
CONSTANTS *Sigma*, *L*

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

The genesis blocks for the chains.

$BcGenesisBlock \triangleq [context\_bft \mapsto 0, hash \mapsto 0]$
$BftGenesisBlock \triangleq [headers\_bc \mapsto \langle\rangle, hash \mapsto 0]$
$CrossLink2GenesisBlock \triangleq [fin \mapsto \langle BcGenesisBlock \rangle]$

The views and tips for the chains.

$BcView(i) \triangleq bc\_chains[i]$
$BcTip(i) \triangleq BcView(i)[Len(BcView(i))]$

$BftView(i) \triangleq bft\_chains[i]$
$BftTip(i) \triangleq BftView(i)[Len(BftView(i))]$

$BcTips \triangleq \{BcTip(i) : i \in 1 \,..\, BcNodes\}$
$BftTips \triangleq \{BftTip(i) : i \in 1 \,..\, BftNodes\}$

The best chain selection functions.

$ChooseBestBcTip \triangleq Max(\{t.hash : t \in BcTips\})$
$ChooseBestBftTip \triangleq Max(\{t.hash : t \in BftTips\})$

$ChooseContextBft \triangleq Max(\{t.hash : t \in BcTips\})$

$ChooseBestBcChain \triangleq$
  CHOOSE $i \in 1 \,..\, BcNodes : Len(bc\_chains[i]) = Max(\{Len(bc\_chains[j]) : j \in 1 \,..\, BcNodes\})$

$ChooseBestBftChain \triangleq$
  CHOOSE $i \in 1 \,..\, BftNodes : Len(bft\_chains[i]) = Max(\{Len(bft\_chains[j]) : j \in 1 \,..\, BftNodes\})$

$ChooseBcView \triangleq BcView(\text{CHOOSE } i \in 1 \,..\, BcNodes : \text{TRUE})$

$ChooseBestCrosslinkChain \triangleq$
  CHOOSE $i \in 1 \,..\, CrossLink2Nodes : Len(crosslink2\_chains[i]) =$
    $Max(\{Len(crosslink2\_chains[j]) : j \in 1 \,..\, CrossLink2Nodes\})$

Definition: Computable efficiently function

$\star bftlastfinal : \star bftblock \to \star bftblock \cup \{\bot\}$

$BftLastFinal(n) \triangleq bft\_chains[n]$