MODULE *Operators*

This module defines common operations that can be applied to the representation of the network that is used by the $p2p$ algorithm.

LOCAL INSTANCE *Integers*
LOCAL INSTANCE *Sequences*
LOCAL INSTANCE *Utils*
VARIABLE *the_network*

Given a block collection and a hash, returns the block with the given hash.

$$FindBlockByHash(block\_collection,\ hash) \triangleq \text{CHOOSE } b \in block\_collection : b.hash = hash$$

Update the peer set of a local peer with a new remote peer address establishing a connection.

$$\begin{array}{l} UpdatePeerSet(local\_peer\_address,\ remote\_peer\_address) \triangleq\ [i \in 1\ ..\ Len(the\_network) \mapsto \\ \quad \text{IF } the\_network[i].peer = local\_peer\_address \text{ THEN} \\ \qquad [the\_network[i] \text{ EXCEPT } !.peer\_set = @ \cup \{remote\_peer\_address\}] \\ \quad \text{ELSE} \\ \qquad the\_network[i] \\ ] \end{array}$$

Given a block collection, a start height and an end height, returns the blocks in the given range.

$$\begin{array}{l} FindBlocks(block\_collection,\ start\_height,\ end\_height) \triangleq \\ \quad \{b \in block\_collection : \\ \qquad \land\ b.height \geq start\_height \\ \qquad \land\ b.height \leq end\_height \\ \quad \} \end{array}$$

Get the full data of a peer from the network given a peer address.

$$GetPeerFromNetwork(peer\_address) \triangleq \text{CHOOSE } peer \in ToSet(the\_network) : peer.peer = peer\_address$$

Get the chain tip of a peer given a peer address.

$$\begin{array}{l} GetPeerTipByAddress(peer\_address) \triangleq \\ \quad \text{LET } peer\_blocks \triangleq (\text{CHOOSE } peer \in ToSet(the\_network) : peer.peer = peer\_address).blocks \\ \quad \text{IN} \quad \text{IF } peer\_blocks = \{\} \text{ THEN} \\ \qquad\quad [height \mapsto 0,\ block \mapsto \text{“serialized block data 0”},\ hash \mapsto \text{“blockhash 0”}] \\ \quad \text{ELSE} \\ \qquad \text{CHOOSE } block \in peer\_blocks : block.height = Max(\{b.height : b \in peer\_blocks\}) \end{array}$$

Get the chain tip of a peer given a peer index of the network.

$$\begin{array}{l} GetPeerTipByIndex(peer\_index) \triangleq \\ \quad \text{IF } the\_network[peer\_index].blocks = \{\} \text{ THEN} \\ \qquad [height \mapsto 0,\ block \mapsto \text{“serialized block data 0”},\ hash \mapsto \text{“blockhash 0”}] \\ \quad \text{ELSE} \\ \qquad \text{CHOOSE } block \in the\_network[peer\_index].blocks : block.height = \\ \qquad\quad Max(\{b.height : b \in the\_network[peer\_index].blocks\}) \end{array}$$

Get the chain tip of a peer given a peer index of the network and the network.

$$GetPeerTipByIndexAndNetwork(peer\_index,\ network) \triangleq$$

```
    IF network[peer_index].blocks = {} THEN
        [height ↦ 0, block ↦ "serialized block data 0", hash ↦ "blockhash 0"]
     ELSE
        CHOOSE block ∈ network[peer_index].blocks : block.height =
            Max({b.height : b ∈ network[peer_index].blocks})

Get the blocks of a peer given a peer address.
GetPeerBlocks(peer_address) ≜ (CHOOSE peer ∈ ToSet(the_network) : peer.peer = peer_address).blocks
```