

MODULE *Operators*

This module defines comment operations that can be applied to the network 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.

$UpdatePeerSet(local_peer_address, remote_peer_address) \triangleq [i \in 1 \dots Len(the_network) \mapsto$
 IF $the_network[i].peer = local_peer_address$ THEN
 $[the_network[i] \text{ EXCEPT } !.peer_set = @ \cup \{remote_peer_address\}]$
 ELSE
 $the_network[i]$

]

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

$FindBlocks(block_collection, start_height, end_height) \triangleq$
 $\{b \in block_collection :$
 $\wedge b.height \geq start_height$
 $\wedge b.height \leq end_height$
 $\}$

Get the peer a peer from the network given a peer address.

$GetPeerFromNetwork(peer_address) \triangleq \text{CHOOSE } peer \in ToSet(the_network) : peer.peer = peer_address$

$Max(S) \triangleq \text{CHOOSE } x \in S : \forall y \in S : x \geq y$

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

$GetPeerTipByAddress(peer_address) \triangleq$
 LET $peer_blocks \triangleq (\text{CHOOSE } peer \in ToSet(the_network) : peer.peer = peer_address).blocks$
 IN IF $peer_blocks = \{\}$ THEN
 $[height \mapsto 0, block \mapsto \text{"serialized block data 0"}, hash \mapsto \text{"blockhash 0"}]$
 ELSE
 $\text{CHOOSE } block \in peer_blocks : block.height = Max(\{b.height : b \in peer_blocks\})$

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

$GetPeerTipByIndex(peer_index) \triangleq$
 IF $the_network[peer_index].blocks = \{\}$ THEN
 $[height \mapsto 0, block \mapsto \text{"serialized block data 0"}, hash \mapsto \text{"blockhash 0"}]$
 ELSE
 $\text{CHOOSE } block \in the_network[peer_index].blocks : block.height =$
 $Max(\{b.height : b \in the_network[peer_index].blocks\})$

Get the chain tip of a peer given a peer index in the network.
 $GetPeerTipByIndexAndNetwork(peer_index, network) \triangleq$
 IF $network[peer_index].blocks = \{\}$ THEN
 $[height \mapsto 0, block \mapsto \text{"serialized block data 0"}, hash \mapsto \text{"blockhash 0"}]$
 ELSE
 CHOOSE $block \in network[peer_index].blocks : block.height =$
 $Max(\{b.height : b \in network[peer_index].blocks\})$
 Get the blocks of a peer given a peer address.
 $GetPeerBlocks(peer_address) \triangleq (CHOOSE peer \in ToSet(the_network) : peer.peer = peer_address).blocks$
