MODULE *LNContracsUsingBitcoinTransactions*

This spec captures the behaviour of commitment transactions on the two sides of a Lightning channel.

We model the various kinds of outputs a commitment transactions will have over its lifetime.

The state of the commitment transaction changes in reponse to the various actions like supersede, publish, etc are taken by parties.

We also do not deal with the communication protocol between nodes for creating and updating commitment transactions. This spec will only focus on the various commitment transaction and their lifecycle in response to interaction between parties and the blockchain.

We ignore the details of how transactions are signed and just mark transactions as signed. This lets us focus on the specifying the behaviour of the commitment transactions without dealing with lower level complexities.

The model defines the intial balance from *alice* to bob. TLA+ will handle situations where channels are balanced and when all the balance is on the other side.

TODO: Add actions for closing channels. Currenly we only have support for breach *tx* and the corresponding breach remedy txs.

TODO: Add *HTLCs*.

EXTENDS *Integers*,
TLC,
Sequences,
FiniteSets,
BitcoinTransactions

CONSTANTS
InitialBalance Initial balances for *alice* and bob

VARIABLES
commitment_txs, Commitment txs held by each party. Not yet broadcast.
breach_remedy_txs *BR* txs held by each party. Not yet broadcast.

$SeqToSet(s) \triangleq \{s[i] : i \in \text{DOMAIN } s\}$

Current channel contracts only ever have two parties

$Party \triangleq \{\text{“alice”}, \text{“bob”}\}$

$vars \triangleq \langle chain\_height, transactions, mempool, published, commitment\_txs, breach\_remedy\_txs\rangle$

$Init \triangleq$
$\quad \wedge transactions = [id \in TXID \mapsto [inputs \mapsto \langle\rangle, outputs \mapsto \langle\rangle]]$

$\land\ commitment\_txs = [p \in Party \mapsto \langle\rangle]$
$\land\ breach\_remedy\_txs = [p \in Party \mapsto \langle\rangle]$
$\land\ chain\_height = 0$
$\land\ mempool = \{\}$
$\land\ published = [id \in TXID \mapsto NoSpendHeight]$

$TypeOK \triangleq$
$\land\quad transactions \in [TXID \to [inputs : Seq(Input),\ outputs : Seq(Output)]]$
$\land\quad commitment\_txs \in [Party \to Seq(TXID)]$
$\land\quad breach\_remedy\_txs \in [Party \to Seq(TXID)]$
$\land\quad mempool \in \text{SUBSET } TXID$
$\land\quad published \in [TXID \to Int]$

$ChooseKey(k) \triangleq \text{CHOOSE } e \in KEY : e \neq k$

$CreateFundingTx(id,\ keys,\ amount) \triangleq$
$\land\ AddMultisigCoinbaseToMempool(id,\ keys,\ amount)$
$\land\ \text{UNCHANGED } \langle commitment\_txs,\ breach\_remedy\_txs\rangle$

$ConfirmTx(id) \triangleq$
$\land\ ConfirmMempoolTx(id)$
$\land\ \text{UNCHANGED } \langle commitment\_txs,\ breach\_remedy\_txs\rangle$

$Next \triangleq$
$\lor \exists\, keys \in KEY \times KEY,\ id \in TXID,\ amount \in AMOUNT :$
$\quad\lor CreateFundingTx(id,\ keys,\ amount)$
$\lor \exists\, id \in TXID : ConfirmTx(id)$

$Spec \triangleq$
$\land\ Init$
$\land\ \Box[Next]_{\langle vars\rangle}$
