MODULE *BitcoinTransactions*

This spec captures the actions and states of bitcoin transactions in the context of the bitcoin blockchain. These actions will be used by the *LN* Contracts spec and other layer two contract specifications.

The focus of this module is to provide:

1. Way to generate transactions that accept input and generate outputs

2. Confirm transactions so that outputs can be spent.

3. Most importantly - provide a way to verify spend conditions without building the entire cryptography machinery. This enables spec authors to focus on what the conditions achieve instead of how those conditions are achieved.

Goal A: Move environment / bitcoin transaction actions and variables from Contracts to here

EXTENDS *Sequences*,
Integers,
TLC

Define constants so that we can define finite sets for inputs, outputs and txids etc.

CONSTANTS *CSV*, Set of *CSV* values
VOUT, Set of vout values
TXID, Set of transaction ids
AMOUNT, Set of amounts that can be used
KEY, Set of all keys used for signatures
HASH Set of all hash preimages

$SighashFlag \triangleq \{\text{“all”}, \text{“none”}, \text{“single”}, \text{“anyonecanpay”}\}$

Set of output types supported for building contracts.

Each output type will have to provide a means to verify an input trying to spend it.

$OutputTypes \triangleq \{\text{“p2wkh”}, \text{“multisig”}, \text{“multisig\_with\_csv”}, \text{“hash\_lock”}\}$

$NoCSV \triangleq \text{CHOOSE } c : c \notin CSV$
$NoHash \triangleq \text{CHOOSE } h : h \notin HASH$

$Input \triangleq [$
$\quad index : VOUT,$
$\quad sighash\_flag : SighashFlag,$ Parts of transactions covered by signature
$\quad signed\_by : Seq(KEY),$ One or more keys that have signed this input
$\quad hash\_preimage : HASH$
$]$

$Output \triangleq [$
$\quad index : VOUT,$
$\quad type : OutputTypes,$
$\quad csv : CSV \cup \{NoCSV\},$
$\quad hash : HASH \cup \{NoHash\},$

$amount : AMOUNT$
]

VARIABLES
$chain\_height$,
$transactions$,
$mempool$,
$published$

$vars \triangleq \langle chain\_height,\ transactions,\ mempool,\ published \rangle$

$Init \triangleq$
$\land\ transactions = [id \in TXID \mapsto [inputs \mapsto \langle\rangle,\ outputs \mapsto \langle\rangle]]$
$\land\ chain\_height = 0$
$\land\ mempool = \{\}$
$\land\ published = \{\}$

$TypeOK \triangleq$
$\land\quad transactions \in [TXID \to [inputs : Seq(Input),\ outputs : Seq(Output)]]$
$\land\quad mempool \in \text{SUBSET } TXID$
$\land\ published \in \text{SUBSET } Tx$

$CreateP2PKHOutput(key,\ amount) \triangleq$ [
$index \mapsto 0$,
$type \mapsto$ "p2wkh",
$csv \mapsto NoCSV$,
$hash \mapsto NoHash$,
$amount \mapsto amount$
]

Add a new coinbase tx to $mempool$. No verification is required here as no prevout is being spent.
$AddCoinbaseToMempool(id,\ key,\ amount) \triangleq$
$\land\ id \notin mempool$
$\land\ id \notin published$
$\land\ transactions' = [transactions \text{ EXCEPT } ![id] = [inputs \mapsto \langle\rangle,$
$outputs \mapsto \langle CreateP2PKHOutput(key,\ amount)\rangle]]$
$\land\ mempool' = mempool \cup \{id\}$
$\land$ UNCHANGED $\langle chain\_height,\ published\rangle$

Confirm coinbase transaction from $mempool$.
$ConfirmCoinbaseMempoolTx \triangleq$
$\exists\, id \in \text{DOMAIN } transactions :$

$\land\ id \in mempool$
$\land\ id \notin published$
$\land$ LET $tx \triangleq transactions[id]$
IN
$\land\ tx.inputs = \langle\rangle$ A coinbase tx, has no inputs. We are not dealing with blocks, so we ignore the block index coinbase check
$\land\ published' = published \cup \{id\}$
$\land\ mempool' = mempool \setminus \{id\}$
$\land\ chain\_height' = chain\_height + 1$
$\land$ UNCHANGED $\langle transactions \rangle$

$Next \triangleq$
$\lor \exists\, k \in KEY,\ id \in TXID,\ a \in AMOUNT :$
$\lor\ AddCoinbaseToMempool(id,\ k,\ a)$
$\lor\ ConfirmCoinbaseMempoolTx$

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