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,
SequencesExt

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
PARTY,	Parties participating in the *L2* protocol
KEY,	Set of keys for each party used in the *L2* protocol
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.

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

$NoCSV \triangleq \text{CHOOSE } c : c \notin CSV$
$MaxCSV \triangleq \text{CHOOSE } c \in CSV : \forall\, y \in CSV : c \geq y$
$NoHash \triangleq \text{CHOOSE } h : h \notin HASH$
$NoSpendHeight \triangleq -1$

All keys available for use by the parties
$Keys \triangleq PARTY \times KEY$

$Input \triangleq [$
$\quad txid : TXID,$
$\quad index : VOUT,$

$sighash\_flag : SighashFlag$, Parts of transactions covered by signature
$signed\_by : Seq(Keys)$, One or more keys that have signed this input
$hash\_preimage : HASH \cup \{NoHash\}$
]

$Output \triangleq$ [
$index : VOUT$,
$type : OutputTypes$,
$keys : Seq(Keys)$, Sig from these keys is required to spend
$csv : CSV \cup \{NoCSV\}$, The CSV should have expired before spend
$hash : HASH \cup \{NoHash\}$, Pre-image required to spend
$amount : AMOUNT$
]

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

$CreateP2WKHOutput(keys, amount) \triangleq$ [
$index \mapsto 0$,
$type \mapsto$ "p2wkh",
$keys \mapsto keys$,
$csv \mapsto NoCSV$,
$hash \mapsto NoHash$,
$amount \mapsto amount$
]

$CreateMultisigOutput(keys, amount) \triangleq$ [
$index \mapsto 0$,
$type \mapsto$ "multisig",
$keys \mapsto keys$,
$csv \mapsto NoCSV$,
$hash \mapsto NoHash$,
$amount \mapsto amount$
]

$CreateMultisigWithCSVOutput(keys, amount) \triangleq$ [
$index \mapsto 0$,
$type \mapsto$ "multisig_with_csv",
$keys \mapsto keys$,
$csv \mapsto MaxCSV$,
$hash \mapsto NoHash$,

$amount \mapsto amount$
]

Add a coinbase tx spendable with a pk. No verification is required here as no prevout is being spent.

$AddP2WKHCoinbaseToMempool(id,\ keys,\ amount) \triangleq$
$\land\ id \notin mempool$
$\land\ published[id] = NoSpendHeight$
$\land\ transactions' = [transactions \text{ EXCEPT } ![id] = [inputs \mapsto \langle\rangle,$
$outputs \mapsto \langle CreateP2WKHOutput(keys,\ amount)\rangle]]$
$\land\ mempool' = mempool \cup \{id\}$
$\land$ UNCHANGED $\langle chain\_height,\ published\rangle$

Add a coinbase tx with a *multisig* output spendable by signature from all keys.

We don't do threshold signatures for simplicity.

$AddMultisigCoinbaseToMempool(id,\ keys,\ amount) \triangleq$
$\land\ id \notin mempool$
$\land\ published[id] = NoSpendHeight$
$\land\ transactions' = [transactions \text{ EXCEPT } ![id] = [inputs \mapsto \langle\rangle,$
$outputs \mapsto \langle CreateMultisigOutput(keys,\ amount)\rangle]]$
$\land\ mempool' = mempool \cup \{id\}$
$\land$ UNCHANGED $\langle chain\_height,\ published\rangle$

Confirm transaction from *mempool*.

$ConfirmMempoolTx(id) \triangleq$
$\land\ id \in mempool$
$\land\ published[id] = NoSpendHeight$
$\land$ LET $tx \triangleq transactions[id]$
IN
$\land\ chain\_height' = chain\_height + 1$ Each tx is in it's own block
$\land\ published' = [published \text{ EXCEPT } ![id] = chain\_height']$
$\land\ mempool' = mempool \setminus \{id\}$
$\land$ UNCHANGED $\langle transactions\rangle$

Create a transaction spending the given output/id, and spendable by the given key.

$CreateP2WKHTx(spending,\ output,\ id,\ output\_key,\ amount) \triangleq$ [
$inputs \mapsto \langle[txid \mapsto spending,$
$index \mapsto output.index,$
$sighash\_flag \mapsto$ "all",
$signed\_by \mapsto output.keys,$
$hash\_preimage \mapsto NoHash]\rangle,$
$outputs \mapsto \langle CreateP2WKHOutput(output\_key,\ amount)\rangle$
]

Create a transaction spending the given output/id, and spendable by as a *multisig* of the given keys.

$$
\begin{array}{l}
CreateMultisigTx(spending,\ output,\ id,\ output\_keys,\ amount) \triangleq\ [\\
\quad inputs \mapsto \langle [txid \mapsto spending, \\
\qquad\qquad index \mapsto output.index, \\
\qquad\qquad sighash\_flag \mapsto \text{“all”}, \\
\qquad\qquad signed\_by \mapsto output.keys, \\
\qquad\qquad hash\_preimage \mapsto NoHash] \rangle, \\
\quad outputs \mapsto \langle CreateMultisigOutput(output\_keys,\ amount) \rangle \\
]
\end{array}
$$

$$
\begin{array}{l}
CreateMultisigWithCSVTx(spending,\ output,\ id,\ output\_keys,\ amount) \triangleq\ [\\
\quad inputs \mapsto \langle [txid \mapsto spending, \\
\qquad\qquad index \mapsto output.index, \\
\qquad\qquad sighash\_flag \mapsto \text{“all”}, \\
\qquad\qquad signed\_by \mapsto output.keys, \\
\qquad\qquad hash\_preimage \mapsto NoHash] \rangle, \\
\quad outputs \mapsto \langle CreateMultisigWithCSVOutput(output\_keys,\ amount) \rangle \\
]
\end{array}
$$

Add a new transaction to *mempool*.

The transaction is created and added to *mempool*.

The transaction is constructed such that it is a valid transaction.

input_type specifies the type of published output to select to spend.

output_type specifies the type of new output to create.

$$
\begin{array}{l}
AddSpendTxToMempool(id,\ output\_keys,\ amount,\ input\_type,\ output\_type) \triangleq \\
\quad \exists\, s \in \text{DOMAIN } published : \\
\qquad \wedge\ published[s] \neq NoSpendHeight \\
\qquad \wedge \\
\qquad\quad \exists\, o \in ToSet(transactions[s].outputs) : \\
\qquad\qquad \wedge\ id \notin mempool \\
\qquad\qquad \wedge\ o.type = input\_type \qquad \text{Select published } tx \text{ of } input\_type \\
\qquad\qquad \wedge\ transactions' = [transactions \text{ EXCEPT } ![id] = \\
\qquad\qquad\qquad\quad \text{CASE } (output\_type = \text{“p2wkh”}) \rightarrow \\
\qquad\qquad\qquad\qquad\qquad CreateP2WKHTx(s,\ o,\ id,\ output\_keys,\ amount) \\
\qquad\qquad\qquad\quad \square\ (output\_type = \text{“multisig”}) \rightarrow \\
\qquad\qquad\qquad\qquad\qquad CreateMultisigTx(s,\ o,\ id,\ output\_keys,\ amount) \\
\qquad\qquad\qquad\quad \square\ (output\_type = \text{“multisig\_with\_csv”}) \rightarrow \\
\qquad\qquad\qquad\qquad\qquad CreateMultisigWithCSVTx(s,\ o,\ id,\ output\_keys,\ amount) \\
\qquad\qquad\quad] \\
\qquad\qquad \wedge\ mempool' = mempool \cup \{id\} \\
\qquad\qquad \wedge \text{ UNCHANGED } \langle chain\_height,\ published \rangle
\end{array}
$$