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
          KEY,          Set of all keys used for signatures
          HASH          Set of all hash preimages
```

$SighashFlag \triangleq \{$"all", "none", "single", "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.

$OutputTypes \triangleq \{$"p2wkh", "multisig", "multisig_with_csv", "hash_lock"$\}$

$OutputTypes \triangleq \{$"p2wkh", "multisig"$\}$

$NoCSV \triangleq$ CHOOSE $c : c \notin CSV$

$NoHash \triangleq$ CHOOSE $h : h \notin HASH$

```
Input ≜ [
    txid : TXID,
    index : VOUT,
    sighash_flag : SighashFlag,        Parts of transactions covered by signature
    signed_by : Seq(KEY),              One or more keys that have signed this input
    hash_preimage : HASH ∪ {NoHash}
]
```

```
Output ≜ [
    $index : VOUT$,
    $type : OutputTypes$,
    $keys : Seq(KEY)$,                 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
    $chain\_height$,
    $transactions$,
    $mempool$,
    $published$

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

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

$TypeOK \triangleq$
    $\land$   $transactions \in [TXID \rightarrow [inputs : Seq(Input), outputs : Seq(Output)]]$
    $\land$   $mempool \in$ SUBSET $TXID$
    $\land$   $published \in$ SUBSET $TXID$
```

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

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

]

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

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

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

We don't do threshold signatures for simplicity.

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

Confirm coinbase transaction from *mempool*.

```
$ConfirmCoinbaseMempoolTx \triangleq$
    $\exists id \in \text{DOMAIN } transactions :$
        $\land id \in mempool$
        $\land id \notin published$
        $\land \text{LET } tx \triangleq transactions[id]$
          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
            $\land published' = published \cup \{id\}$
            $\land mempool' = mempool \setminus \{id\}$
            $\land chain\_height' = chain\_height + 1$   Each tx is in it's own block
        $\land \text{UNCHANGED } \langle transactions \rangle$
```

Create a *p2wkh* transaction spending the given output/*id*, and spendable by the given key.

```
$CreateP2WKHTx(spending, output, id, key, amount) \triangleq [$
    $inputs \mapsto \langle [txid \mapsto spending,$
                $index \mapsto output.index,$
                $sighash\_flag \mapsto$ "all",
                $signed\_by \mapsto output.keys,$
```

```
                    $hash\_preimage \mapsto NoHash]\rangle,$
    $outputs \mapsto \langle CreateP2WKHOutput(\langle key\rangle, amount)\rangle$
]
```

Add a *p2wkh* transaction to *mempool*. The transaction is created and added to *mempool*. The transaction is constructed such that it is a valid transaction.

```
$AddSpendP2WKHToMempool(id, key, amount) \triangleq$
    $\exists s \in published :$
      $\exists o \in ToSet(transactions[s].outputs) :$
         $\land id \notin mempool$
         $\land id \notin published$
         $\land o.type = $ "p2wkh"                              Spending a *p2wkh*
         $\land transactions' = [transactions$ EXCEPT $![id] =$
                              $CreateP2WKHTx(s, o, id, key, amount)]$
         $\land mempool' = mempool \cup \{id\}$
         $\land$ UNCHANGED $\langle chain\_height, published\rangle$
```

```
$Next \triangleq$
    $\lor \exists k \in KEY, id \in TXID, a \in AMOUNT :$
        $\lor AddP2WKHCoinbaseToMempool(id, k, a)$
    $\lor \exists keys \in KEY \times KEY, id \in TXID, amount \in AMOUNT :$
        $\lor AddMultisigCoinbaseToMempool(id, keys, amount)$
    $\lor \exists id \in TXID, a \in AMOUNT, k \in KEY :$
        $AddSpendP2WKHToMempool(id, k, a)$
    $\lor ConfirmCoinbaseMempoolTx$

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