

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	<i>CSV</i> ,	Set of <i>CSV</i> values
	<i>VOUT</i> ,	Set of <i>vout</i> values
	<i>TXID</i> ,	Set of transaction ids
	<i>AMOUNT</i> ,	Set of amounts that can be used
	<i>KEY</i> ,	Set of all keys used for signatures
	<i>HASH</i>	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", "multisig-with_csv"}

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

MaxCSV \triangleq CHOOSE $c \in CSV : \forall y \in CSV : c \geq y$

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

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

Input \triangleq [
 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* \cup {*NoHash*}
]

$Output \triangleq [$
 $\quad index : VOUT,$
 $\quad type : OutputTypes,$
 $\quad keys : Seq(KEY),$
 $\quad csv : CSV \cup \{NoCSV\},$
 $\quad hash : HASH \cup \{NoHash\},$
 $\quad amount : AMOUNT$
 $\quad]$

Sig from these keys is required to spend
 The CSV should have expired before spend
 Pre-image required to spend

VARIABLES
 $\quad chain_height,$
 $\quad transactions,$
 $\quad mempool,$
 $\quad published$

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

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

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

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

$$\begin{aligned} \text{AddP2WKHCoinbaseToMempool}(id, key, amount) &\triangleq \\ &\wedge id \notin mempool \\ &\wedge id \notin published \\ &\wedge transactions' = [transactions \text{ EXCEPT } ![id] = [inputs \mapsto \langle \rangle, \\ &\quad outputs \mapsto \langle \text{CreateP2WKHOutput}(\langle key \rangle, amount) \rangle]] \\ &\wedge mempool' = mempool \cup \{id\} \\ &\wedge \text{UNCHANGED } \langle chain_height, published \rangle \end{aligned}$$

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

We don't do threshold signatures for simplicity.

$$\begin{aligned} \text{AddMultisigCoinbaseToMempool}(id, keys, amount) &\triangleq \\ &\wedge id \notin mempool \\ &\wedge id \notin published \\ &\wedge transactions' = [transactions \text{ EXCEPT } ![id] = [inputs \mapsto \langle \rangle, \\ &\quad outputs \mapsto \langle \text{CreateMultisigOutput}(keys, amount) \rangle]] \\ &\wedge mempool' = mempool \cup \{id\} \\ &\wedge \text{UNCHANGED } \langle chain_height, published \rangle \end{aligned}$$

Confirm coinbase transaction from *mempool*.

$$\begin{aligned} \text{ConfirmCoinbaseMempoolTx} &\triangleq \\ \exists id \in \text{DOMAIN } transactions : & \\ &\wedge id \in mempool \\ &\wedge id \notin published \\ &\wedge \text{LET } tx \triangleq transactions[id] \\ &\text{IN} \\ &\wedge tx.inputs = \langle \rangle \quad \begin{array}{l} \text{A coinbase } tx, \text{ has no inputs.} \\ \text{We are not dealing with blocks, so we} \\ \text{ignore the block index coinbase check} \end{array} \\ &\wedge published' = published \cup \{id\} \\ &\wedge mempool' = mempool \setminus \{id\} \\ &\wedge chain_height' = chain_height + 1 \quad \text{Each } tx \text{ is in its own block} \\ &\wedge \text{UNCHANGED } \langle transactions \rangle \end{aligned}$$

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

$$\begin{aligned} \text{CreateP2WKHTx}(spending, output, id, key, amount) &\triangleq [\\ &inputs \mapsto \langle [txid \mapsto spending, \\ &\quad index \mapsto output.index, \\ &\quad sighash_flag \mapsto \text{"all"}, \\ &\quad signed_by \mapsto output.keys, \\ &\quad hash_preimage \mapsto NoHash] \rangle, \\ &outputs \mapsto \langle \text{CreateP2WKHOutput}(\langle key \rangle, amount) \rangle \\ &] \end{aligned}$$

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

$$\text{CreateMultisigTx}(\text{spending}, \text{output}, \text{id}, \text{keys}, \text{amount}) \triangleq [$$

$$\text{inputs} \mapsto \langle [\text{txid} \mapsto \text{spending},$$

$$\text{index} \mapsto \text{output.index},$$

$$\text{sighash_flag} \mapsto \text{"all"},$$

$$\text{signed_by} \mapsto \text{output.keys},$$

$$\text{hash_preimage} \mapsto \text{NoHash}] \rangle,$$

$$\text{outputs} \mapsto \langle \text{CreateMultisigOutput}(\text{keys}, \text{amount}) \rangle$$

$$]$$

$$\text{CreateMultisigWithCSVTx}(\text{spending}, \text{output}, \text{id}, \text{keys}, \text{amount}) \triangleq [$$

$$\text{inputs} \mapsto \langle [\text{txid} \mapsto \text{spending},$$

$$\text{index} \mapsto \text{output.index},$$

$$\text{sighash_flag} \mapsto \text{"all"},$$

$$\text{signed_by} \mapsto \text{output.keys},$$

$$\text{hash_preimage} \mapsto \text{NoHash}] \rangle,$$

$$\text{outputs} \mapsto \langle \text{CreateMultisigWithCSVOutput}(\text{keys}, \text{amount}) \rangle$$

$$]$$

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.

$$\text{AddSpendTxToMempool}(\text{id}, \text{key}, \text{amount}, \text{input_type}, \text{output_type}) \triangleq$$

$$\exists s \in \text{published} :$$

$$\exists o \in \text{ToSet}(\text{transactions}[s].\text{outputs}) :$$

$$\wedge \text{id} \notin \text{mempool}$$

$$\wedge \text{id} \notin \text{published}$$

$$\wedge o.\text{type} = \text{input_type} \quad \text{Select published tx of input_type}$$

$$\wedge \text{transactions}' = [\text{transactions} \text{ EXCEPT } ![\text{id}] =$$

$$\text{CASE } (\text{output_type} = \text{"p2wkh"}) \rightarrow$$

$$\text{CreateP2WKHTx}(s, o, \text{id}, \text{key}, \text{amount})$$

$$\square (\text{output_type} = \text{"multisig"}) \rightarrow$$

$$\text{CreateMultisigTx}(s, o, \text{id}, \langle \text{key}, \text{ChooseKey}(\text{key}) \rangle, \text{amount})$$

$$\square (\text{output_type} = \text{"multisig_with_csv"}) \rightarrow$$

$$\text{CreateMultisigWithCSVTx}(s, o, \text{id}, \langle \text{key}, \text{ChooseKey}(\text{key}) \rangle, \text{amount})$$

$$]$$

$$\wedge \text{mempool}' = \text{mempool} \cup \{\text{id}\}$$

$$\wedge \text{UNCHANGED } \langle \text{chain_height}, \text{published} \rangle$$