MODULE *Contracts*

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 supercede, spend, revoke etc are taken.

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.

EXTENDS *Integers*,
TLC,
Sequences

CONSTANTS

CSV,	The csv value to use in contracts
Height	The height up to which we run the spec

Channel contracts only ever have two parties

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

For the first revocation we only need two keys per party

$NumKey \triangleq 2$

Set of all keys

$Key \triangleq \forall\, p \in Party,\, k \in 0 \,..\, NumKey - 1 : \langle p,\, k \rangle$

Value to capture missing *CSV* in output

$NoCSV \triangleq \text{CHOOSE } c : c \notin 0 \,..\, CSV$

Multisig outputs without *CSV* encumberance

$MultiSigOutput \triangleq \forall\, a,\, b \in Party \times Party : \langle a,\, b,\, NoCSV \rangle$

Multisig outputs with *CSV* encumberance

$MultiSigWithCSVOutput \triangleq \forall\, a,\, b \in Party \times Party : \langle a,\, b,\, CSV \rangle$

P2PKH outputs, without encumbrance

$P2PKH \triangleq Key$

$AllOutput \triangleq MultiSigOutput \cup MultiSigWithCSVOutput \cup P2PKH$

$NoOutput \triangleq \text{CHOOSE } o : o \notin AllOutput$

Set of all signatures for all commit txs. The signature in real world is related to the commit transaction, however, leave out this complication of how the signature is generated. If there is a signature by a key on a tx, it is assumed it is correctly signed as per bitcoin's requirements

$Sig \triangleq \forall\, p \in Party,\, k \in 0 \,..\, NumKey - 1 : \langle p,\, k \rangle$

Value to capture unsigned transactions

$NoSig \triangleq \text{CHOOSE } s : s \notin Sig$

Define the commitment tx type. We don't have *HTLCs* yet. We also don't filter outputs to the party here. We leave that for actions, or we'll add the filter when needed.

Commitment transactions are different for different parties and that is captured in *commitment_txs*. The *Party* here is simply to make it easier to know immediately which is the local party.

$$CommitmentTx \triangleq [\; outputs \mapsto Seq(AllOutput),\; local\_sig \mapsto Sig \cup NoSig,\; remote\_sig \mapsto Sig \cup NoSig \;]$$

VARIABLES

commitment_txs The set of all commiment transactions for both parties

$vars \triangleq \langle commitment\_txs \rangle$

$Init \triangleq$

$\wedge\, commitment\_txs = \forall\, p \in Party : [p \to \langle\rangle]$

$TypeInvariant \triangleq$

$\wedge\, commitment\_txs \in [Party \to Seq(CommitmentTx)]$