Onto blockchain

Onto blockchain

Enforcement

The legal system ensures financial contracts …

… but a contract on blockchain should enforce itself.

Onto blockchain

Enforcement

The legal system ensures financial contracts …

… but a contract on blockchain should enforce itself.

Double spend

Blockchain designed to prevent spending the same money twice …

… but that's precisely how credit works.

Crypto-economics

Make the past irrefutable through cryptography.

Shape behaviour through financial incentives.

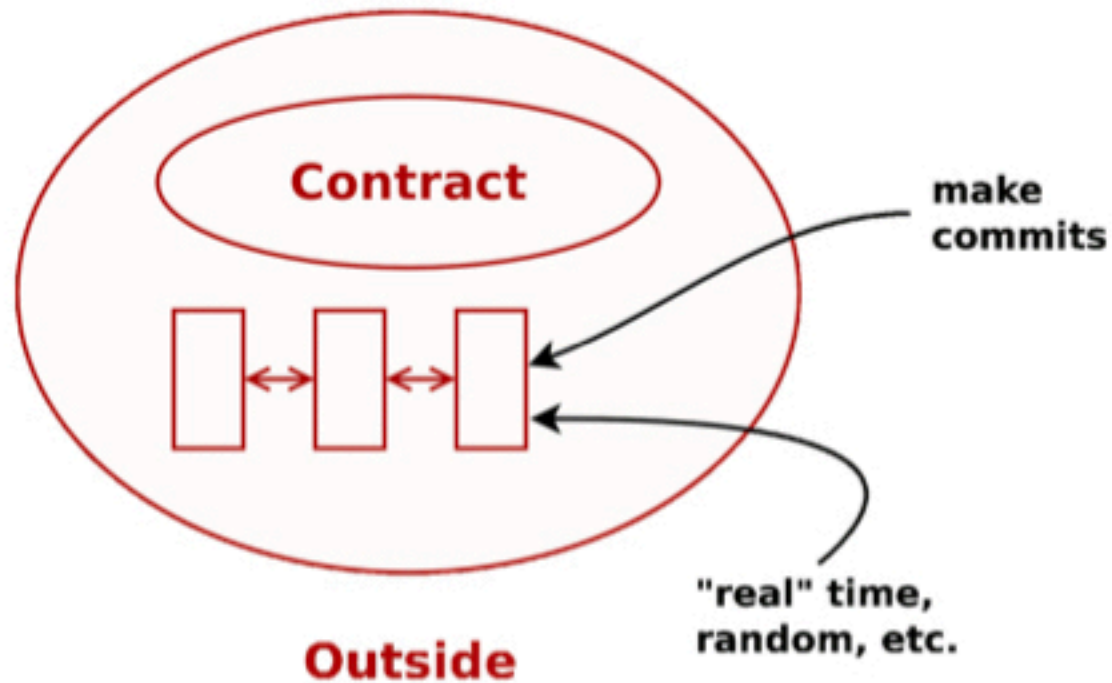
Avoid bad behaviour … and “walk away”.

Interactions with the outside world

Real values: e.g. *"the spot price of oil in Aberdeen at 12:00, 31-12-17"*.

Random values.

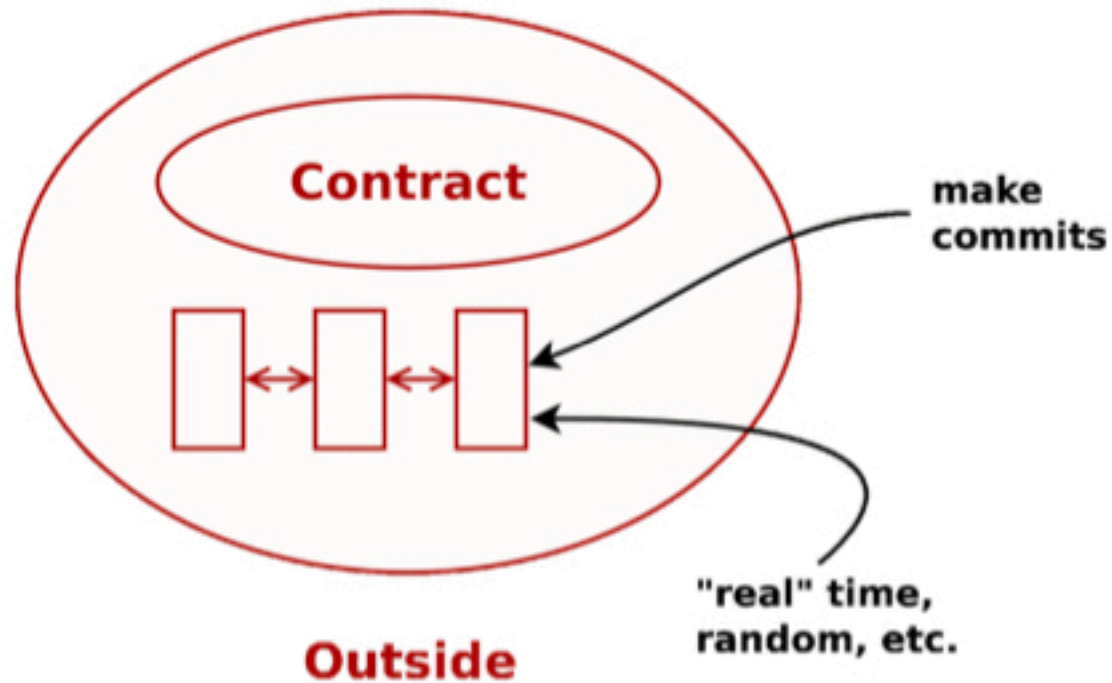
Participants sign-off … ?



Commitments

Commit a certain amount of cash for a finite time.

Need to avoid “walk away” …



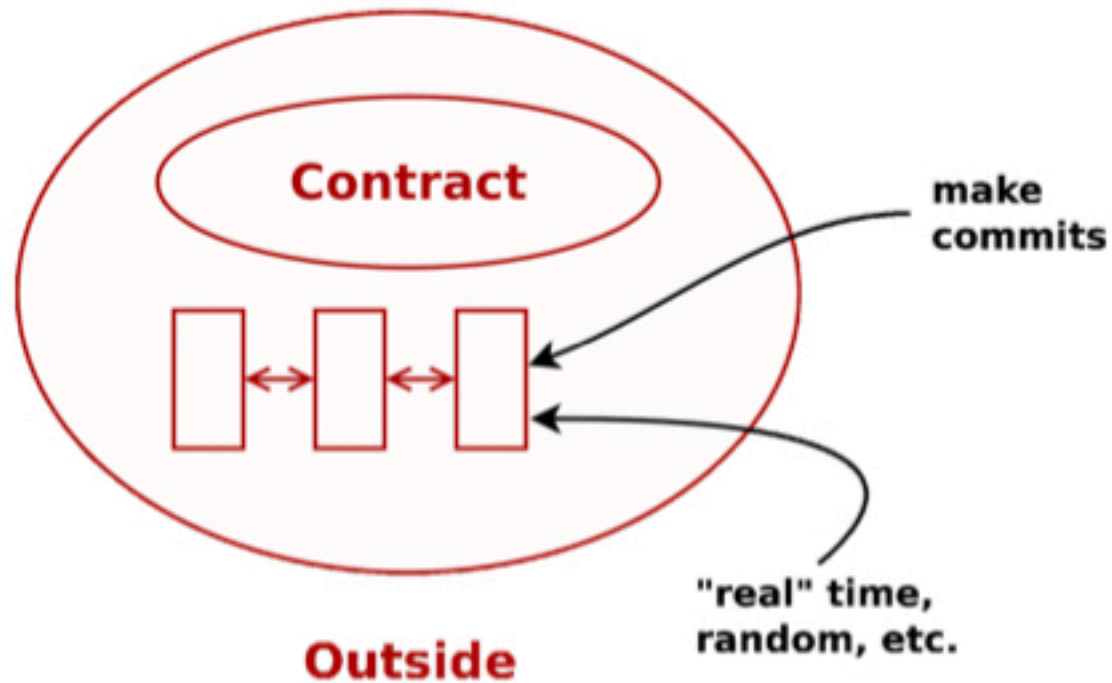
Commitments and timeouts

Commit a certain amount of cash for a finite time.

Need to avoid "walk away" …

We don't *require* a commitment: can only *ask for* one …

… and only wait a bounded time for the commitment to be made.



Marlowe in a nutshell

An EDSL as a Haskell data type.

Executable small-step semantics.

Implemented on Cardano "mockchain".

Analysis and proof.

Compile from original DSLs.

Meadow simulation.

Escrow – timeouts

```
(When (Or (majority_chose refund)
          (majority_chose pay))
      (Choice (majority_chose pay)
              (Pay alice bob AvailableMoney)
              redeem_original))
```

Escrow – timeouts

```
(When (Or (majority_chose refund)
          (majority_chose pay))
      90
      (Choice (majority_chose pay)
              (Pay alice bob AvailableMoney)
              redeem_original))
```

Escrow – timeouts

Timeout: waiting ends at this block height.

```
(When (Or (majority_chose refund)
          (majority_chose pay))
      90
      (Choice (majority_chose pay)
              (Pay alice bob AvailableMoney)
              redeem_original))
```

Escrow – timeouts

```
(When (Or (majority_chose refund)
          (majority_chose pay))
      90
      (Choice (majority_chose pay)
              (Pay alice bob AvailableMoney)
              redeem_original)
      redeem_original)
```

Escrow – timeouts

```
(When (Or (majority_chose refund)
          (majority_chose pay))
      90
      (Choice (majority_chose p
              (Pay alice bob Av
              redeem_original))
      redeem_original)
```

Do this at timeout, if trigger hasn't happened.

Escrow – commitments

```
(When (Or (majority_chose refund)
          (majority_chose pay))
      90
      (Choice (majority_chose pay)
              (Pay alice bob AvailableMoney)
              redeem_original)
      redeem_original)
```

Escrow – commitments

```
(When (Or (majority_chose refund)
          (majority_chose pay))
      90
      (Choice (majority_chose pay)
              (Pay alice bob AvailableMoney)
              redeem_original)
      redeem_original)
```

Escrow – commitments

```
(CommitCash id1 alice 15000 10 100
                (When (Or (majority_chose refund)
                          (majority_chose pay))
                      90
                      (Choice (majority_chose pay)
                              (Pay alice bob AvailableMoney)
                              redeem_original)
                      redeem_original)

                Null)
```

Escrow – commitments

Waits for Alice to make a commitment of 15k ADA

```
(CommitCash id1 alice 15000 10 100
                (When (Or (majority_chose refund)
                          (majority_chose pay))
                      90
                      (Choice (majority_chose pay)
                              (Pay alice bob AvailableMoney)
                              redeem_original)
                      redeem_original)

                Null)
```

Escrow – commitments

Waits for Alice to make a commitment of 15k ADA

Commitment until block height 100

```
(CommitCash id1 alice 15000 10 100
                (When (Or (majority_chose refund)
                          (majority_chose pay))
                      90
                      (Choice (majority_chose pay)
                              (Pay alice bob AvailableMoney)
                              redeem_original)
                      redeem_original)

                Null)
```

Escrow – commitments

Waits for Alice to make a commitment of 15k ADA

Commitment until block height 100

Can be redeemed after that

```
(CommitCash id1 alice 15000 10 100
                (When (Or (majority_chose refund)
                          (majority_chose pay))
                      90
                      (Choice (majority_chose pay)
                              (Pay alice bob AvailableMoney)
                              redeem_original)
                      redeem_original)

                Null)
```

Escrow – commitments

```
(CommitCash id1 alice 15000 10 100
                (When (Or (majority_chose refund)
                          (majority_chose pay))
                      90
                      (Choice (majority_chose pay)
                              (Pay alice bob AvailableMoney)
                              redeem_original)
                      redeem_original)

                Null)
```

Waits for Alice to make a commitment of 15k ADA

Commitment until block height 100

Can be redeemed after that

Only wait until block height 10…

Escrow – commitments

```
(CommitCash id1 alice 15000 10 100
            (When (Or (majority_chose refund)
                      (majority_chose pay))
                  90
                  (Choice (majority_chose pay)
                          (Pay alice bob AvailableMoney)
                          redeem_original)
                  redeem_original)

            Null)
```

Waits for Alice to make a commitment of 15k ADA

Commitment until block height 100

Can be redeemed after that

Only wait until block height 10…

… and if no commitment do this.

Deposit incentive

```
CommitCash com1 alice ada100 10 200
           (CommitCash com2 bob ada20 20 200
               (When (PersonChoseSomething choice1 alice) 100
                     (Both (RedeemCC com1 Null)
                           (RedeemCC com2 Null))
                     (Pay pay1 bob alice ada20 200
                          (Both (RedeemCC com1 Null)
                                (RedeemCC com2 Null))))
                (RedeemCC com1 Null))
          Null
```

Deposit incentive

```
CommitCash com1 alice ada100 10 200
           (CommitCash com2 bob ada20 20 200
               (When (PersonChoseSomething choice1 alice) 100
                     (Both (RedeemCC com1 Null)
                           (RedeemCC com2 Null))
                     (Pay pay1 bob alice ada20 200
                          (Both (RedeemCC com1 Null)
                                (RedeemCC com2 Null))))
               (RedeemCC com1 Null))
```

Wait until time 10 for alice to commit 100 ADA until time 200.

similarly for bob

if alice chooses to before time 100, both people get their money back

otherwise, alice gets her money and the 20 ADA from bob

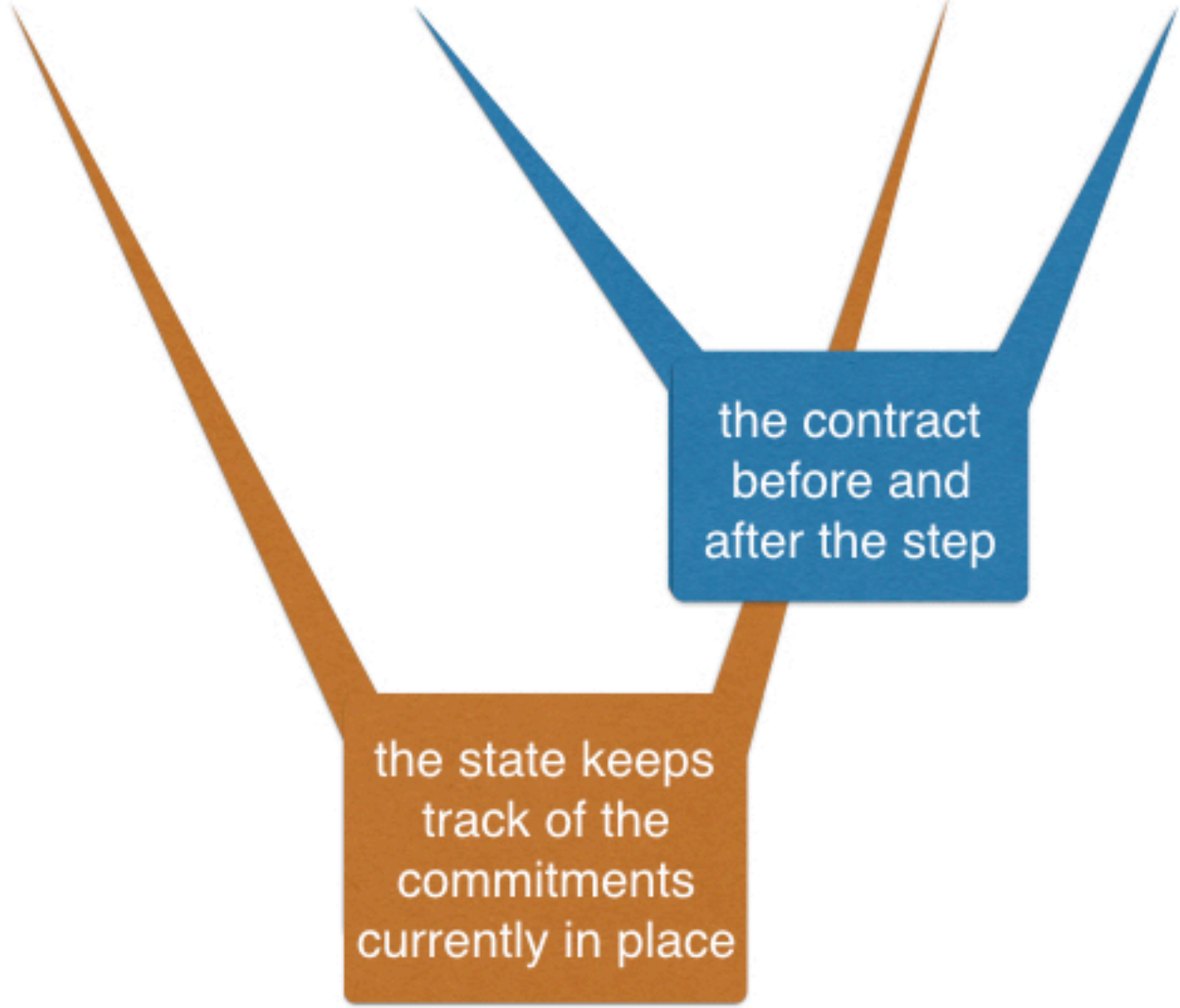
action if bob didn't commit in time

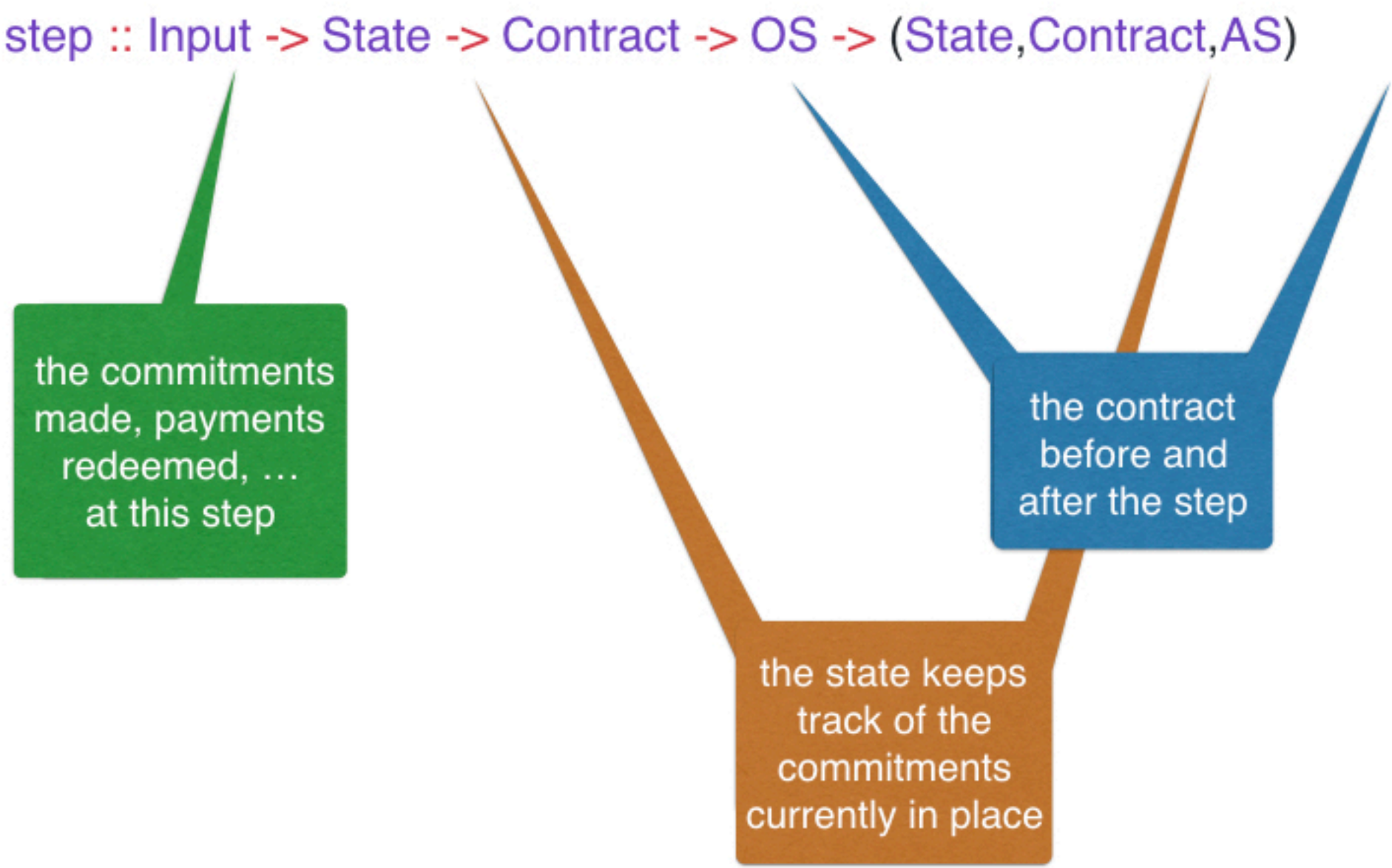

```
step :: Input -> State -> Contract -> OS -> (State,Contract,AS)
```

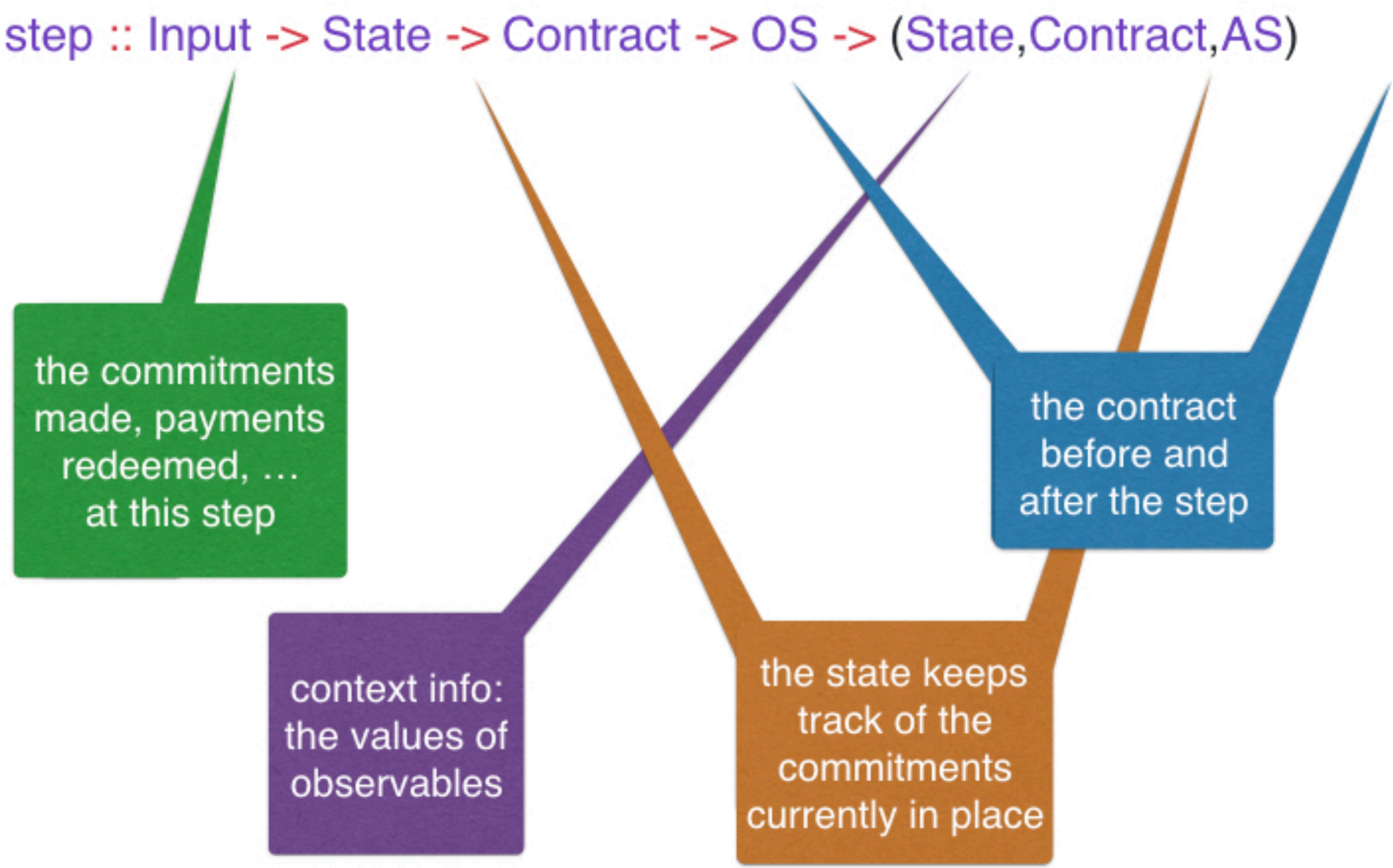

```
step :: Input -> State -> Contract -> OS -> (State,Contract,AS)
```

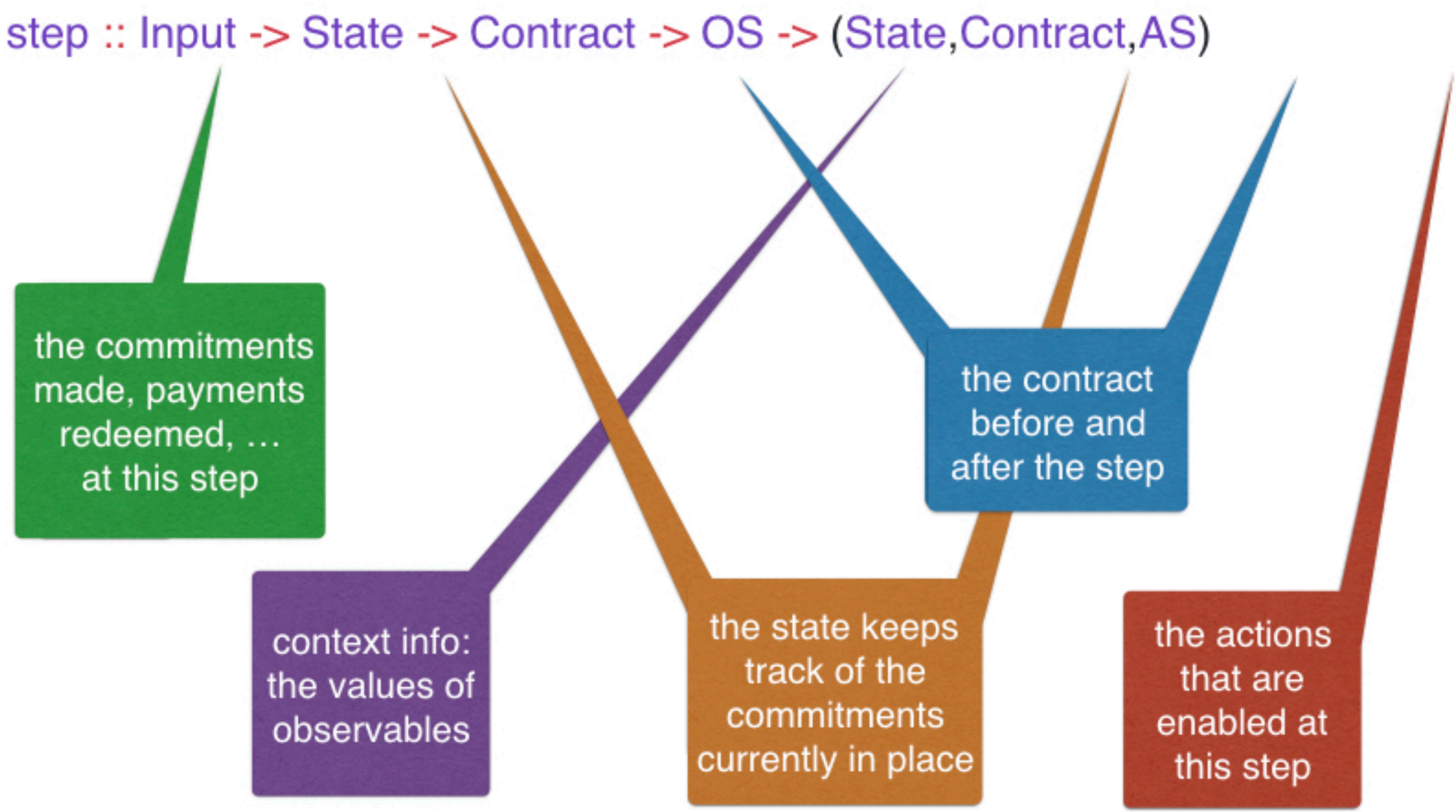
the contract before and after the step

```
step :: Input -> State -> Contract -> OS -> (State,Contract,AS)
```




step :: Input -> State -> Contract -> OS -> (State,Contract,AS)
the commitments made, payments redeemed, … at this step
the state keeps track of the commitments currently in place
the contract before and after the step


step :: Input -> State -> Contract -> OS -> (State,Contract,AS)
the commitments made, payments redeemed, … at this step
context info: the values of observables
the state keeps track of the commitments currently in place
the contract before and after the step


step :: Input -> State -> Contract -> OS -> (State,Contract,AS)
the commitments made, payments redeemed, … at this step
context info: the values of observables
the state keeps track of the commitments currently in place
the contract before and after the step
the actions that are enabled at this step

The Contract data type

```
data Contract =
    Null |
    CommitCash IdentCC Person Cash Timeout Timeout Contract Contract |
    RedeemCC IdentCC Contract |
    Pay IdentPay Person Person Cash Timeout Contract |
    Both Contract Contract |
    Choice Observation Contract Contract |
    When Observation Timeout Contract Contract
            deriving (Eq,Ord,Show,Read)
```

The Contract data type

```
data Contract =
  Null |
  CommitCash IdentCC Person Cash Timeout Timeout Contract Contract |
```

CommitCash idCC p n t1 t2 k1 k2

For this contract to make progress, either

- before the timeout t1 the user p makes a cash commitment of value n and timeout t2 with the identifier idCC: generate SuccessfulCommit action, continue as k1;
- or timeout t1 exceeded and continue as k2.

Otherwise it is quiescent. At timeout t2 remaining committed cash can be redeemed, and fullStep enables that.

The Contract data type

```
data Contract =
```

When obs expi k1 k2

Will progress either

- when the observation obs becomes true, and continues as k1, or
- when the timeout expi reached, and continues as k2.

```
  Choice Observation Contract Contract |
  When Observation Timeout Contract Contract
        deriving (Eq,Ord,Show,Read)
```