

```

1 |----- MODULE Ledger -----|
  High level specification of DLT Ledger, expressed as a single state machine without MVCC
  validation
6 | EXTENDS Sequences, Integers, TLAPS

  Constants
11 | CONSTANTS State, InitState a set of states, and
12 | ASSUME InitStateAxiom  $\triangleq$  InitState  $\in$  State the designated initial state.
13 | NULL  $\triangleq$  CHOOSE x : x  $\notin$  BOOLEAN

  State variables of this module
18 | VARIABLES state, current state of the ledger state machine.
19 |             chain, blockchain, a list of received transactions.
20 |             index index of the blockchain.
21 | vars  $\triangleq$   $\langle$ state, chain, index $\rangle$ 

  Datatype definition
27 | TotalFunc(S1, S2)  $\triangleq$  [S1  $\rightarrow$  S2  $\setminus$   $\{\{\}\}$ ] a set of total function from S1 to S2

  Operation is a function from a state to a state, can be non-deterministic, but required to be total.
33 | Operation  $\triangleq$  TotalFunc(State, SUBSET State)
34 | Operation  $\triangleq$  [State  $\rightarrow$  (SUBSET State)  $\setminus$   $\{\{\}\}$ ]

36 | TX  $\triangleq$  [f : Operation] a transaction. note that “f” is just a label

  Type invariant
41 | ChainEntry  $\triangleq$  [tx : TX, is_valid : BOOLEAN  $\cup$   $\{NULL\}$ ]
42 | Chain  $\triangleq$  Seq(ChainEntry)
43 | TypeInv  $\triangleq$ 
44 |    $\wedge$  state  $\in$  State
45 |    $\wedge$  index  $\in$  Nat
46 |    $\wedge$  index  $> 0$ 
47 |   Each TX in the blockchain has a flag if it's valid or not. Before the TX is processed, its value is NULL.
48 |    $\wedge$  chain  $\in$  Chain
49 |-----|

  Initial condition
53 | Init  $\triangleq$ 
54 |    $\wedge$  state = InitState state is at the initial state, and
55 |    $\wedge$  index = 1
56 |    $\wedge$  chain =  $\langle$  $\rangle$  empty transaction queue.

58 |-----|

  Actions

  SubmitTX: A client appends a transaction to the transaction queue.

```

```

66  $SubmitTX(tx) \triangleq$ 
67    $\wedge chain' = Append(chain, [tx \mapsto tx, is\_valid \mapsto NULL])$ 
68    $\wedge UNCHANGED \langle state, index \rangle$ 

CommitTx: Ledger processes the oldest unprocessed TX and

73  $ProcessTX\_OK \triangleq$ 
74   LET
75      $f \triangleq chain[index].tx.f$ 
76   IN
77      $\wedge Len(chain) \geq index$ 
78      $\wedge index \in DOMAIN\ chain$ 
79      $\wedge chain' = [chain\ EXCEPT\ ![index].is\_valid = TRUE]$  update validity flag
80      $\wedge index' = index + 1$  increment the index.
81      $\wedge state' \in f[state]$  perform non-deterministic state transition by f.

83  $ProcessTX\_ERR \triangleq$ 
84   LET
85      $f \triangleq chain[index].tx.f$ 
86   IN
87      $\wedge Len(chain) \geq index$ 
88      $\wedge index \in DOMAIN\ chain$ 
89      $\wedge chain' = [chain\ EXCEPT\ ![index].is\_valid = FALSE]$  see above.
90      $\wedge index' = index + 1$  see above.
91      $\wedge UNCHANGED\ state$  state does not change due to invalid TX.

93  $Next \triangleq (\exists tx \in TX : SubmitTX(tx)) \vee ProcessTX\_OK \vee ProcessTX\_ERR$ 

Specification

98  $Spec \triangleq Init \wedge \Box [Next]_{vars}$ 

100 |-----|
Invariants

104  $Finality \triangleq TRUE$  TODO
105  $Safety \triangleq Finality$ 

107 Invariant (safety) on the blockchain
108  $ChainInv \triangleq$ 
109    $chain = (processed\ part) + (unprocessed\ part)$ 
110    $\wedge \forall i \in 1 .. index - 1 : chain[i].is\_valid \in BOOLEAN$ 
111    $\wedge \forall i \in \{i \in Nat : index \leq i\} \cap DOMAIN\ chain : chain[i].is\_valid = NULL$ 

113  $Inv \triangleq TypeInv \wedge ChainInv$ 

115 THEOREM  $LedgerInv \triangleq Spec \Rightarrow \Box Inv$ 
116 PROOF
117    $\langle 1 \rangle 1\ Init \Rightarrow Inv$ 
118   BY InitStateAxiom DEF Init, Inv, TypeInv, ChainInv, Chain

```

```

119  <1>2  $Inv \wedge [Next]_{vars} \Rightarrow Inv'$ 
120    <2>1 SUFFICES ASSUME  $TypeInv, ChainInv, [Next]_{vars}$  PROVE  $Inv'$  BY DEF  $Inv$ 
121    <2>2 CASE  $Next$ 
122      <3> USE DEF  $Inv, Next$ 
123      <3> USE DEF  $TypeInv, ChainInv, Chain, ChainEntry$ 
124      <3>1 CASE  $(\exists tx \in TX : SubmitTX(tx))$ 
125        <4> USE DEF  $SubmitTX$ 
126        <4>a  $\forall i \in \text{DOMAIN } chain : chain[i] = chain'[i]$  BY <3>1
127        <4>1  $TypeInv'$  BY <2>1, <3>1
128        <4>2  $ChainInv'$ 
129          <5>1  $ChainInv!1'$  OBVIOUS
130          <5>2  $ChainInv!2'$ 
131            <6>a  $\text{DOMAIN } chain' = \text{DOMAIN } chain \cup \{Len(chain) + 1\}$  BY  $TypeInv, <3>1$ 
132            <6>1 PICK  $tx \in TX : SubmitTX(tx)$  BY <3>1
133            <6>2 TAKE  $i \in (\{i \in Nat : index \leq i\} \cap \text{DOMAIN } chain)'$ 
134            <6>3 CASE  $i \in (\{j \in Nat : index \leq j\} \cap \{Len(chain) + 1\})$  BY <2>1, <4>a, <6>1, <6>3
135            <6>4 CASE  $i \in (\{j \in Nat : index \leq j\} \cap \text{DOMAIN } chain)$  BY <2>1, <4>a, <6>1, <6>4
136            <6> QED BY <2>1, <6>a, <6>1, <6>2, <6>3, <6>4
137          <5> QED BY <5>1, <5>2
138        <4> QED BY <4>1, <4>2
139      <3>2 CASE  $ProcessTX\_OK$ 
140        <4> USE DEF  $ProcessTX\_OK$ 
141        <4>1  $TypeInv'$  BY <2>1, <3>2 DEF  $TX, Operation, TotalFunc$ 
142        <4>2  $ChainInv'$ 
143          <5>  $ChainInv!1'$  OBVIOUS
144          <5>  $ChainInv!2'$  BY <2>1, <3>2
145          <5> QED OBVIOUS
146        <4> QED BY <4>1, <4>2
147      <3>3 CASE  $ProcessTX\_ERR$ 
148        <4> USE DEF  $ProcessTX\_ERR$ 
149        <4>1  $TypeInv'$  BY <2>1, <2>2, <3>3 DEF  $TX, Operation, TotalFunc$ 
150        <4>2  $ChainInv'$ 
151          <5>  $ChainInv!1'$  OBVIOUS
152          <5>  $ChainInv!2'$  BY <2>1, <3>3
153          <5> QED OBVIOUS
154        <4> QED BY <4>1, <4>2
155      <3> QED
156    BY <2>1, <2>2, <3>1, <3>2, <3>3
157  <2>3 CASE UNCHANGED  $vars$ 
158    BY <2>1, <2>3 DEF  $Inv, TypeInv, ChainInv, vars$ 
159  <2> QED BY <2>1, <2>2, <2>3
160  <1> QED BY  $PTL, <1>1, <1>2$  DEF  $Spec$ 

```

162 \ * Modification History

* Last modified *Wed Jul 10 01:23:37 JST 2019* by *shinsa*
* Created *Fri Jun 07 01:51:28 JST 2019* by *shinsa*