```
1 ──────────── MODULE MVCC_Ledger ────────────
  Middle level specification of DLT Ledger, expressed as a single state machine with MVCC validation
6 EXTENDS Sequences, Integers, TLAPS

  Constants
11 CONSTANTS State, InitState  a set of states, and
12 ASSUME InitStateAxiom ≜ InitState ∈ State  the designated initial state.
13 NULL ≜ CHOOSE x : x ∉ BOOLEAN

  Read-write set, which is a result of a simulation
18 CONSTANTS RWSet

  State variables of this module
23 VARIABLES state,    current state of the ledger state machine.
24           chain,    blockchain, a list of received transactions.
25           index     unprocessed TX index at the blockchain.
26 vars ≜ ⟨state, chain, index⟩

  Datatype definition

32 TotalFunc(S1, S2) ≜ [S1 → S2 \ {{}}]  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.
38 Operation ≜ TotalFunc(State, SUBSET State)
39  Operation ≜ [State → (SUBSET State) \ {{}}]

41 TX ≜ [f : Operation]  a transaction. note that "f" is just a label

  Type invariant

  At this module, endorsement is just a RWSet, which will be extended at lower models.
51 Endorsement ≜ RWSet

53  each entry of blockchain now has a RWSet.
54 ChainEntry ≜ [tx : TX, endorsement : Endorsement, is_valid : BOOLEAN ∪ {NULL}]
55 Chain ≜ Seq(ChainEntry)
56 TypeInv ≜
57     ∧ state ∈ State
58     ∧ index ∈ Nat
59     ∧ index > 0
60      Each TX in the blockchain has a flag if it's valid or not. Before the TX
61      is processed, its value is NULL.
62     ∧ chain ∈ Chain

64 ──────────────────────────────────────────────

  Initial condition
```

```
68  Init ≜
69      ∧ state  = InitState      state is at the initial state, and
70      ∧ index = 1
71      ∧ chain = ⟨⟩              empty transaction queue.

73  ├──────────────────────────────────────────────────────────────────────┤
    Actions

    (non-deterministic) simulaton result for the operation f
81  simulate(f) ≜ {rwset ∈ RWSet : ∃ post_state ∈ f[state] : diff(state, post_state, rwset)}

    SubmitTx: A client appends a transaction and its simulation result to the transaction queue.
87  SubmitTx(tx) ≜
88      ∧ ∃ rs ∈ simulate(tx.f) :
89          transactions′ = Append(transactions, [tx ↦ tx, rwset ↦ rs, processed ↦ FALSE])
90      ∧ UNCHANGED state

    CommitTx: Ledger processes the oldest unprocessed TX and
95  CONSTANTS apply(_, _)  Applies rwset to the current state

97  commitSub(idx) ≜
98      changes ledger's state by the transaction at index idx
99      LET
100         cur_tx ≜ transactions[idx].tx
101         rwset ≜ transactions[idx].rwset
102         f ≜ cur_tx.f
103     IN
104         ∧ transactions′ = [transactions EXCEPT ![idx].processed = TRUE]  updates processed flag
105         ∧
106             ∨ state′ = apply(state, rwset)  perform state transition, which is non-deterministic
107             ∨ UNCHANGED state               or state does not change (by TX failure)

109 CommitTx ≜
110     ∃ idx ∈ 1 .. Len(transactions) :
111         idx is the smallest index where TX is not processed
112         ∧ ∀ j ∈ 1 .. idx − 1 : transactions[j].processed = TRUE
113         ∧ ∀ k ∈ idx .. Len(transactions) : transactions[k].processed = FALSE
114         ∧ commitSub(idx)  process idx-th item

116 Next ≜ (∃ tx ∈ TX : SubmitTx(tx)) ∨ CommitTx

    Specification
121 Spec ≜ Init ∧ □[Next]⟨state, transactions⟩

123 ├──────────────────────────────────────────────────────────────────────┤
    Invariants
```

$Finality \triangleq$ TRUE *TODO*

$Safety \triangleq Finality$

$Invariant \triangleq$
- $\land\ TypeInvariant$
- $\land\ Len(transactions) > 0 \Rightarrow \exists\, idx \in 1\,..\,Len(transactions)+1 :$
  - $\land \forall\, j \in 1\,..\,idx-1 : transactions[j].processed = \text{TRUE}$
  - $\land \forall\, k \in idx\,..\,Len(transactions) : transactions[k].processed = \text{FALSE}$

THEOREM $Spec \Rightarrow \Box Invariant$

$LedgerSpec(\_state, \_transactions) \triangleq$ INSTANCE $Ledger$ WITH $state \leftarrow \_state,\ transactions \leftarrow \_transactions$

ideal form

THEOREM $Spec \Rightarrow \exists\, ex\_state,\ ex\_txs : LedgerSpec(ex\_state,\ ex\_txs)!Spec$

OMITTED

Refinement Mapping

$h\_state \triangleq state$ ad hoc impl

$h\_txs \triangleq$ CHOOSE $seq \in Seq([tx : TX,\ processed : \text{BOOLEAN}\ ])$:
- $\land\ Len(seq) = Len(transactions)$
- $\land \forall\, i \in 1\,..\,Len(seq)$:
  - $\land\ seq[i].tx = transactions[i].tx$
  - $\land\ seq[i].processed = transactions[i].processed$

$h\_txs \triangleq transactions$ ad hoc impl

LEMMA $L1 \triangleq transactions = \langle\rangle \Rightarrow h\_txs = \langle\rangle$

BY DEF h_txs

Refinement Theorem

THEOREM $Spec \Rightarrow LedgerSpec(h\_state,\ h\_txs)!Spec$

$\langle 1\rangle$ USE DEF $Spec,\ LedgerSpec!Spec,\ h\_state,\ h\_txs$

$\langle 1\rangle 1.\ Init \Rightarrow LedgerSpec(h\_state,\ h\_txs)!Init$

BY DEF $Init,\ LedgerSpec!Init$

$\langle 1\rangle 2.\ Next \Rightarrow$
- $\lor\ LedgerSpec(h\_state,\ h\_txs)!Next$
- $\lor$ UNCHANGED $\langle h\_state,\ h\_txs\rangle$
- $\langle 2\rangle 1.\ (\exists\, tx \in TX : SubmitTx(tx)) \Rightarrow$
  - $\lor\ LedgerSpec(h\_state,\ h\_txs)!Next$
  - $\langle 3\rangle 1.\ (\exists\, tx \in TX : SubmitTx(tx)) \Rightarrow$
    - $(\exists\, tx \in LedgerSpec(h\_state,\ h\_txs)!TX : LedgerSpec(h\_state,\ h\_txs)!SubmitTx(tx))$

    OMITTED

    $(\exists\, tx \in LedgerSpec(h\_state,\ h\_txs)!TX : LedgerSpec(h\_state,\ h\_txs)!SubmitTx(tx))$
  - $\langle 3\rangle 2.$ QED

    BY $\langle 3\rangle 1$ DEF $LedgerSpec!Next$
- $\langle 2\rangle 2.\ CommitTx \Rightarrow$
  - $\lor\ LedgerSpec(h\_state,\ h\_txs)!Next$

```
180             ∨ UNCHANGED ⟨h_state, h_txs⟩
181             ⟨3⟩1. CommitTx ⇒
182                 ∨ LedgerSpec(h_state, h_txs)!CommitTx
183                 ∨ UNCHANGED ⟨h_state, h_txs⟩
184                 OMITTED
185             ⟨3⟩2. QED
186                 BY ⟨3⟩1 DEF LedgerSpec!Next
187         ⟨2⟩3. QED
188             BY ⟨2⟩1, ⟨2⟩2 DEF Next
189     ⟨1⟩3. UNCHANGED ⟨state, transactions⟩ ⇒
190         ∨ UNCHANGED ⟨h_state, h_txs⟩
191         OBVIOUS
192     ⟨1⟩4. QED
193         BY PTL, ⟨1⟩1, ⟨1⟩2

195 └──────────────────────────────────────────────────────────────────────────┘

    \* Modification History
    \* Last modified Fri Jul 19 13:11:18 JST 2019 by shinsa
    \* Created Tue Jul 02 01:10:01 JST 2019 by shinsa
```