MODULE *TransactionV1*

EXTENDS *Integers*, *Sequences*

CONSTANT *accounts*, *initialBalances*

VARIABLE *balances*, *msgs*

$$Init \triangleq balances = initialBalances \\ \land \quad msgs = \{\}$$

$$TransferMoney(from, to, amount) \triangleq balances[from] \geq amount$$

Account needs to have enough balance, from property testing

$$\land\ msgs' = msgs \cup \{ \\ [account \mapsto from,\ amount \mapsto balances[from] - amount], \\ [account \mapsto to,\ amount \mapsto balances[to] + amount] \\ \} \\ \land\ \text{UNCHANGED}\ \langle balances \rangle$$

$$DbUpdate \triangleq msgs \neq \{\} \\ \land\ \text{LET}\ msg \triangleq \text{CHOOSE}\ msg \in msgs : \text{TRUE} \\ \text{IN} \quad msgs' = msgs \setminus \{msg\} \\ \land \quad balances' = [balances\ \text{EXCEPT}\ ![msg.account] = msg.amount]$$

$$Next \triangleq DbUpdate \\ \lor \quad \land \exists\, from, to \in accounts : \\ from \neq to \land \exists\, amount \in 1 \,..\, balances[from] : \\ TransferMoney(from, to, amount)$$

Send only positive integers, from property testing

HELPERS

RECURSIVE $SumBalance(\_, \_, \_)$

$$SumBalance(accs, bal, total) \triangleq \text{IF}\ accs = \{\} \\ \text{THEN}\ total \\ \text{ELSE}\ \text{LET}\ acc \triangleq \text{CHOOSE}\ acc \in accs : \text{TRUE} \\ \text{IN} \quad SumBalance(accs \setminus \{acc\},\ bal,\ total + bal[acc])$$

INVARIANTS

$$TypeOK \triangleq msgs \subseteq [account : accounts,\ amount : Int]$$

Amount has to be an number, from static typing

$$BalancesAlwaysPositive \triangleq \forall\, acc \in accounts : balances[acc] \geq 0$$

$$TotalMoneyStable \triangleq SumBalance(accounts, initialBalances, 0) = SumBalance(accounts, balances, 0)$$

* Modification History
* Last modified *Fri Aug* 13 12:15:31 *CEST* 2021 by *rchaves*

\ * Created Sun *Aug* 08 21:06:07 *CEST* 2021 by *rchaves*