MODULE *Demo1*

EXTENDS *Integers*, *TLC*

People $\triangleq$ {"alice", "bob"}
Money $\triangleq$ $1 \,..\, 10$
NumTransfers $\triangleq$ 2

```
--algorithm wire
{
    variables acct ∈ [People → Money] ;   [People → Money] Set

    define
    {
        NoOverdrafts ≜ ∀ p ∈ People : acct[p] ≥ 0
     }

    process ( wire ∈ 1 .. NumTransfers )   1∼NumTransfers
        variables
            amnt ∈ 1 .. 5 ;
            from  ∈ People ;
            to ∈ People ;
    {
        Check:
        {
            if ( acct[from] ≥ amnt )
            {
                Withdraw:
                {
                    acct[from] := acct[from] − amnt ;
                 } ;
                Desposit:
                {
                    acct[to] := acct[to] + amnt ;
                 }
             }
         }
     } ;
}
```

algorithm

BEGIN TRANSLATION ($chksum(pcal) =$ "48b19a81" $\wedge\ chksum(tla) =$ "58d7d06a")
VARIABLES *acct*, *pc*

define statement
NoOverdrafts $\triangleq$ $\forall\, p \in People : acct[p] \geq 0$

VARIABLES *amnt*, *from*, *to*

$vars \triangleq \langle acct,\ pc,\ amnt,\ from,\ to \rangle$

$ProcSet \triangleq (1 \,..\, NumTransfers)$

$Init \triangleq$ Global variables
$\land\ acct \in [People \to Money]$
Process wire
$\land\ amnt \in [1 \,..\, NumTransfers \to 1 \,..\, 5]$
$\land\ from\ \in [1 \,..\, NumTransfers \to People]$
$\land\ to\ \in [1 \,..\, NumTransfers \to People]$
$\land\ pc = [self \in ProcSet \mapsto \text{“Check”}]$

$Check(self) \triangleq \land\ pc[self] = \text{“Check”}$
$\land$ IF $acct[from[self]] \geq amnt[self]$
THEN $\land\ pc' = [pc$ EXCEPT $![self] = \text{“Withdraw”}]$
ELSE $\land\ pc' = [pc$ EXCEPT $![self] = \text{“Done”}]$
$\land$ UNCHANGED $\langle acct,\ amnt,\ from,\ to \rangle$

$Withdraw(self) \triangleq \land\ pc[self] = \text{“Withdraw”}$
$\land\ acct' = [acct$ EXCEPT $![from[self]] = acct[from[self]] - amnt[self]]$
$\land\ pc' = [pc$ EXCEPT $![self] = \text{“Desposit”}]$
$\land$ UNCHANGED $\langle amnt,\ from,\ to \rangle$

$Desposit(self) \triangleq \land\ pc[self] = \text{“Desposit”}$
$\land\ acct' = [acct$ EXCEPT $![to[self]] = acct[to[self]] + amnt[self]]$
$\land\ pc' = [pc$ EXCEPT $![self] = \text{“Done”}]$
$\land$ UNCHANGED $\langle amnt,\ from,\ to \rangle$

$wire(self) \triangleq Check(self) \lor Withdraw(self) \lor Desposit(self)$

Allow infinite stuttering to prevent deadlock on termination.
$Terminating \triangleq \land \forall\, self \in ProcSet : pc[self] = \text{“Done”}$
$\land$ UNCHANGED $vars$

$Next \triangleq (\exists\, self \in 1 \,..\, NumTransfers : wire(self))$
$\lor\ Terminating$

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

$Termination \triangleq \Diamond(\forall\, self \in ProcSet : pc[self] = \text{“Done”})$

END TRANSLATION

* Modification History
* Last modified *Tue Jul* 23 17:40:28 *CST* 2024 by *liubang*01
* Created *Tue Jul* 23 17:29:55 *CST* 2024 by *liubang*01