MODULE *Transfer*

EXTENDS *Naturals*, *TLC*

```
--algorithm transfer{

variables alice_account = 10, bob_account = 10,
          account_total = alice_account + bob_account ;


process ( Transfer ∈ 1 .. 2 )
    variable money ∈ 1 .. 20 ;
{
Transfer:
    if ( alice_account ≥ money ) {
       alice_account := alice_account − money ;
       bob_account := bob_account + money ;
     } ;

C: assert alice_account ≥ 0 ;
 }

 }  algorithm
```

BEGIN TRANSLATION (*chksum*(*pcal*) = "4621a15e" ∧ *chksum*(*tla*) = "7e946c47")
Label *Transfer* of process *Transfer* at line 14 col 5 changed to *Transfer_*

VARIABLES *alice_account*, *bob_account*, *account_total*, *pc*, *money*

$vars \triangleq \langle alice\_account,\ bob\_account,\ account\_total,\ pc,\ money \rangle$

$ProcSet \triangleq (1 \,..\, 2)$

$Init \triangleq$ Global variables
$\land\ alice\_account = 10$
$\land\ bob\_account = 10$
$\land\ account\_total = alice\_account + bob\_account$
Process *Transfer*
$\land\ money \in [1 \,..\, 2 \to 1 \,..\, 20]$
$\land\ pc = [self \in ProcSet \mapsto \text{"Transfer\_"}]$

$Transfer\_(self) \triangleq$
$\land\ pc[self] = \text{"Transfer\_"}$
$\land$ IF $alice\_account \geq money[self]$
THEN $\land\ alice\_account' = alice\_account - money[self]$
$\land\ bob\_account' = bob\_account + money[self]$
ELSE $\land$ TRUE
$\land$ UNCHANGED $\langle alice\_account,\ bob\_account \rangle$
$\land\ pc' = [pc$ EXCEPT $![self] = \text{"C"}]$
$\land$ UNCHANGED $\langle account\_total,\ money \rangle$

$C(self) \triangleq\ \land\ pc[self] = \text{"C"}$

$\land\ Assert(alice\_account \geq 0,$
"Failure of assertion at line 19, column 4.")
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{"Done"}]$
$\land$ UNCHANGED $\langle alice\_account,\ bob\_account,\ account\_total,\ money\rangle$

$Transfer(self) \triangleq Transfer\_(self) \lor C(self)$

Allow infinite stuttering to prevent deadlock on termination.

$Terminating \triangleq \land \forall\, self \in ProcSet : pc[self] = \text{"Done"}$
$\land$ UNCHANGED $vars$

$Next \triangleq (\exists\, self \in 1 \,..\, 2 : Transfer(self))$
$\lor\ Terminating$

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

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

END TRANSLATION

$MoneyNotNegative \triangleq money \geq 0$
$MoneyInvariant \triangleq alice\_account + bob\_account = account\_total$

* Modification History
* Last modified *Wed Jul* 24 17:24:41 *CST* 2024 by *liubang*01
* Created *Wed Jul* 24 15:08:38 *CST* 2024 by *liubang*01