MODULE *wire*

EXTENDS *Integers*

```
 --algorithm wire
   variables
       $people = \{\text{"alice"}, \text{"bob"}\}$,
       $acc = [p \in people \mapsto 5]$,

define
   $NoOverdrafts \triangleq \forall p \in people : acc[p] \geq 0$
end define ;

process $Wire \in 1 \,..\, 2$
   variables
       $sender = \text{"alice"}$,
       $receiver = \text{"bob"}$,
       $amount \in 1 \,..\, acc[sender]$ ;

begin
   CheckFunds:
       if $amount \leq acc[sender]$ then
           Withdraw:
               $acc[sender] := acc[sender] - amount$ ;
           Deposit:
               $acc[receiver] := acc[receiver] - amount$ ;
       end if ;
end process ;
end algorithm ;
```

BEGIN TRANSLATION

VARIABLES *people*, *acc*, *pc*

define statement

$NoOverdrafts \triangleq \forall p \in people : acc[p] \geq 0$

VARIABLES *sender*, *receiver*, *amount*

$vars \triangleq \langle people, acc, pc, sender, receiver, amount \rangle$

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

```
Init ≜  Global variables
        $\land\ people = \{\text{"alice"}, \text{"bob"}\}$
        $\land\ acc = [p \in people \mapsto 5]$
        Process Wire
        $\land\ sender = [self \in 1 \,..\, 2 \mapsto \text{"alice"}]$
        $\land\ receiver = [self \in 1 \,..\, 2 \mapsto \text{"bob"}]$
        $\land\ amount \in [1 \,..\, 2 \rightarrow 1 \,..\, acc[sender[\text{CHOOSE } self \in 1 \,..\, 2 : \text{TRUE}]]]$
```

```
          $\land\ pc = [self \in ProcSet \mapsto \text{"CheckFunds"}]$

$CheckFunds(self) \triangleq$ $\land\ pc[self] = \text{"CheckFunds"}$
                      $\land$ IF $amount[self] \leq acc[sender[self]]$
                            THEN $\land\ pc' = [pc$ EXCEPT $![self] = \text{"Withdraw"}]$
                            ELSE $\land\ pc' = [pc$ EXCEPT $![self] = \text{"Done"}]$
                      $\land$ UNCHANGED $\langle people, acc, sender, receiver, amount\rangle$

$Withdraw(self) \triangleq$ $\land\ pc[self] = \text{"Withdraw"}$
                    $\land\ acc' = [acc$ EXCEPT $![sender[self]] = acc[sender[self]] - amount[self]]$
                    $\land\ pc' = [pc$ EXCEPT $![self] = \text{"Deposit"}]$
                    $\land$ UNCHANGED $\langle people, sender, receiver, amount\rangle$

$Deposit(self) \triangleq$ $\land\ pc[self] = \text{"Deposit"}$
                   $\land\ acc' = [acc$ EXCEPT $![receiver[self]] = acc[receiver[self]] - amount[self]]$
                   $\land\ pc' = [pc$ EXCEPT $![self] = \text{"Done"}]$
                   $\land$ UNCHANGED $\langle people, sender, receiver, amount\rangle$

$Wire(self) \triangleq CheckFunds(self) \lor Withdraw(self) \lor Deposit(self)$

$Next \triangleq (\exists\, self \in 1 \,..\, 2 : Wire(self))$
           $\lor$  Disjunct to prevent deadlock on termination
             $((\forall\, self \in ProcSet : pc[self] = \text{"Done"}) \land$ UNCHANGED $vars)$

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

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

 END TRANSLATION
```

* Modification History
* Last modified Sun *Mar* 24 12:50:13 ART 2019 by *jrediger*
* Created Sun *Mar* 24 12:31:13 ART 2019 by *jrediger*