MODULE *bank_account_assembly*

Joint bank account by husband and wife; Only assembly statements (not $C$)
are assumed atomic. This version models the code at assembly level,
and so the "if" statement is no longer atomic. It will assert an error
when $total \neq 120$, even though initially, $account = 100$, and
$cash[\text{"husband"}] = cash[\text{"wife"}] = 10$.
In the "Model" sub-window, try initializing the constant "$N$" to 1.

Note that if you remove the labels
$w0b$, $w0c$, $w1b$, $d0b$, $d0c$, $d1b$, then there will be no assertion error.

EXTENDS *Naturals*, *Sequences*, *TLC*  Sequences required for "procedure" stmt
CONSTANT $N$  $N$ is number of iterations. Assign to it in model overview.

```
--algorithm bank{
  variables $account = 100,\ cash = [i \in \{\text{"husband"}, \text{"wife"}\} \mapsto 10],$
            $iterations = [i \in \{\text{"husband"}, \text{"wife"}\} \mapsto N]$;
      Note that we need to define $iterations[\text{"husband"}]$ and $iterations[\text{"wife"}]$.
        We do _not_ want a single global variable, iterations, that is
        shared between "husband" and "wife".
      In model, replace "$N$" (a constant) by value for iterations

    The procedures withdraw and deposit have been translated here
      to pseudo-assembly language
    Note that "$register1$" and "$register2$" were declared as local variables
      inside the processes for husband and wife.
  procedure withdraw( amount1 )
    variable $register1$, $register2$;
  {
    withdraw_start: $register1 := amount1$;          lw $register1$, ($amount1$)
     w0b:                $register2 := account - register1$;
                lw $register2$, (account) ; sub $register2$, $register2$, $register1$
     w0c:                $account := register2$;      sw $register2$, (account)

     w1:             $register2 := cash[self] + register1$;
                lw $register2$, ($cash[self]$) ; add $register2$, $register2$, $register1$
     w1b:                $cash[self] := register2$;   sw $register2$, ($cash[self]$)

     w2:             return;
   }

  procedure deposit( amount1 )
    variable $register1$, $register2$;
  {
```

```
  deposit_start: register1 := amount1 ;              lw register1, (amount1)
    d0b:         register2 := account + register1 ;  lw register2, (account)
                                               add register2, register2, register1
    d0c:         account := register2 ;              sw register2, (account)

    d1:          register2 := cash[self] − register1 ;
                                             lw register2, (cash[self])
                                             sub register2, register2, register1
    d1b:         cash[self] := register2 ;         sw register2, (cash[self])

    d2:          return ;
  }

 process ( spouse ∈ {"husband", "wife"} )
   variable total ;
 { start: while ( iterations[self] > 0 ) {
        We hard-wire the max amount below, but this could have been a CONSTANT .
      s1: with ( amount ∈ 1 . . 2 )
             call withdraw(amount) ;
      s2: with ( amount ∈ 1 . . 2 )
             call deposit(amount) ;
      s3: iterations[self] := iterations[self] − 1 ;
          total := account + cash["husband"] + cash["wife"] ;
       } ;
      assert iterations[self] = 0 ;

      if ( iterations["husband"] = 0 ∧ iterations["wife"] = 0 ) {
        total := account + cash["husband"] + cash["wife"] ;
        print total ;
        assert total = 120 ;
       }
  }  end process block

}   \* end algorithm
 BEGIN TRANSLATION
 Procedure variable register1 of procedure withdraw at line 33 col 14 changed to register1_
 Procedure variable register2 of procedure withdraw at line 33 col 25 changed to register2_
 Parameter amount1 of procedure withdraw at line 32 col 22 changed to amount1_
CONSTANT defaultInitValue
VARIABLES account, cash, iterations, pc, stack, amount1_, register1_,
          register2_, amount1, register1, register2, total

vars ≜ ⟨account, cash, iterations, pc, stack, amount1_, register1_,
        register2_, amount1, register1, register2, total⟩

ProcSet ≜ ({"husband", "wife"})

Init ≜  Global variables
```

$\land\ account = 100$
$\land\ cash = [i \in \{\text{“husband”}, \text{“wife”}\} \mapsto 10]$
$\land\ iterations = [i \in \ \{\text{“husband”}, \text{“wife”}\} \mapsto N]$

Procedure withdraw

$\land\ amount1\_ = [self\ \in ProcSet \mapsto defaultInitValue]$
$\land\ register1\_ = [self\ \in ProcSet \mapsto defaultInitValue]$
$\land\ register2\_ = [self\ \in ProcSet \mapsto defaultInitValue]$

Procedure deposit

$\land\ amount1 = [self\ \in ProcSet \mapsto defaultInitValue]$
$\land\ register1 = [self\ \in ProcSet \mapsto defaultInitValue]$
$\land\ register2 = [self\ \in ProcSet \mapsto defaultInitValue]$

Process spouse

$\land\ total\ = [self\ \in \{\text{“husband”}, \text{“wife”}\} \mapsto defaultInitValue]$
$\land\ stack = [self\ \in ProcSet \mapsto \langle\rangle]$
$\land\ pc = [self\ \in ProcSet \mapsto \text{“start”}]$

$$\begin{array}{ll} withdraw\_start(self) \triangleq & \land\ pc[self] = \text{“withdraw\_start”} \\ & \land\ register1\_' = [register1\_ \text{ EXCEPT } ![self] = amount1\_[self]] \\ & \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“w0b”}] \\ & \land\ \text{UNCHANGED } \langle account, cash, iterations, stack, \\ & \qquad amount1\_, register2\_, amount1, \\ & \qquad register1, register2, total\rangle \end{array}$$

$$\begin{array}{ll} w0b(self) \triangleq & \land\ pc[self] = \text{“w0b”} \\ & \land\ register2\_' = [register2\_ \text{ EXCEPT } ![self] = account - register1\_[self]] \\ & \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“w0c”}] \\ & \land\ \text{UNCHANGED } \langle account, cash, iterations, stack, amount1\_, \\ & \qquad register1\_, amount1, register1, register2, total\rangle \end{array}$$

$$\begin{array}{ll} w0c(self) \triangleq & \land\ pc[self]\ = \text{“w0c”} \\ & \land\ account' = register2\_[self] \\ & \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“w1”}] \\ & \land\ \text{UNCHANGED } \langle cash, iterations, stack, amount1\_, register1\_, \\ & \qquad register2\_, amount1, register1, register2, total\rangle \end{array}$$

$$\begin{array}{ll} w1(self) \triangleq & \land\ pc[self] = \text{“w1”} \\ & \land\ register2\_' = [register2\_ \text{ EXCEPT } ![self] = cash[self] + register1\_[self]] \\ & \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“w1b”}] \\ & \land\ \text{UNCHANGED } \langle account, cash, iterations, stack, amount1\_, \\ & \qquad register1\_, amount1, register1, register2, total\rangle \end{array}$$

$$\begin{array}{ll} w1b(self) \triangleq & \land\ pc[self] = \text{“w1b”} \\ & \land\ cash' = [cash \text{ EXCEPT } ![self] = register2\_[self]] \\ & \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“w2”}] \\ & \land\ \text{UNCHANGED } \langle account, iterations, stack, amount1\_, register1\_, \\ & \qquad register2\_, amount1, register1, register2, total\rangle \end{array}$$

$$
\begin{array}{lll}
w2(\mathit{self}) & \triangleq & \land\ \mathit{pc}[\mathit{self}] = \text{“w2”} \\
 & & \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![\mathit{self}] = \mathit{Head}(\mathit{stack}[\mathit{self}]).\mathit{pc}] \\
 & & \land\ \mathit{register1\_}' = [\mathit{register1\_}\ \text{EXCEPT}\ ![\mathit{self}] = \mathit{Head}(\mathit{stack}[\mathit{self}]).\mathit{register1\_}] \\
 & & \land\ \mathit{register2\_}' = [\mathit{register2\_}\ \text{EXCEPT}\ ![\mathit{self}] = \mathit{Head}(\mathit{stack}[\mathit{self}]).\mathit{register2\_}] \\
 & & \land\ \mathit{amount1\_}' = [\mathit{amount1\_}\ \text{EXCEPT}\ ![\mathit{self}] = \mathit{Head}(\mathit{stack}[\mathit{self}]).\mathit{amount1\_}] \\
 & & \land\ \mathit{stack}' = [\mathit{stack}\ \text{EXCEPT}\ ![\mathit{self}] = \mathit{Tail}(\mathit{stack}[\mathit{self}])] \\
 & & \land\ \text{UNCHANGED}\ \langle \mathit{account}, \mathit{cash}, \mathit{iterations}, \mathit{amount1}, \mathit{register1}, \\
 & & \qquad\qquad \mathit{register2}, \mathit{total} \rangle
\end{array}
$$

$$
\begin{array}{lll}
\mathit{withdraw}(\mathit{self}) & \triangleq & \mathit{withdraw\_start}(\mathit{self}) \lor w0b(\mathit{self}) \lor w0c(\mathit{self}) \\
 & & \quad \lor\ w1(\mathit{self}) \lor w1b(\mathit{self}) \lor w2(\mathit{self})
\end{array}
$$

$$
\begin{array}{lll}
\mathit{deposit\_start}(\mathit{self}) & \triangleq & \land\ \mathit{pc}[\mathit{self}] = \text{“deposit\_start”} \\
 & & \land\ \mathit{register1}' = [\mathit{register1}\ \text{EXCEPT}\ ![\mathit{self}] = \mathit{amount1}[\mathit{self}]] \\
 & & \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![\mathit{self}] = \text{“d0b”}] \\
 & & \land\ \text{UNCHANGED}\ \langle \mathit{account}, \mathit{cash}, \mathit{iterations}, \mathit{stack}, \\
 & & \qquad\qquad \mathit{amount1\_}, \mathit{register1\_}, \mathit{register2\_}, \\
 & & \qquad\qquad \mathit{amount1}, \mathit{register2}, \mathit{total} \rangle
\end{array}
$$

$$
\begin{array}{lll}
d0b(\mathit{self}) & \triangleq & \land\ \mathit{pc}[\mathit{self}] = \text{“d0b”} \\
 & & \land\ \mathit{register2}' = [\mathit{register2}\ \text{EXCEPT}\ ![\mathit{self}] = \mathit{account} + \mathit{register1}[\mathit{self}]] \\
 & & \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![\mathit{self}] = \text{“d0c”}] \\
 & & \land\ \text{UNCHANGED}\ \langle \mathit{account}, \mathit{cash}, \mathit{iterations}, \mathit{stack}, \mathit{amount1\_}, \\
 & & \qquad\qquad \mathit{register1\_}, \mathit{register2\_}, \mathit{amount1}, \mathit{register1}, \mathit{total} \rangle
\end{array}
$$

$$
\begin{array}{lll}
d0c(\mathit{self}) & \triangleq & \land\ \mathit{pc}[\mathit{self}] \ = \text{“d0c”} \\
 & & \land\ \mathit{account}' = \mathit{register2}[\mathit{self}] \\
 & & \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![\mathit{self}] = \text{“d1”}] \\
 & & \land\ \text{UNCHANGED}\ \langle \mathit{cash}, \mathit{iterations}, \mathit{stack}, \mathit{amount1\_}, \mathit{register1\_}, \\
 & & \qquad\qquad \mathit{register2\_}, \mathit{amount1}, \mathit{register1}, \mathit{register2}, \mathit{total} \rangle
\end{array}
$$

$$
\begin{array}{lll}
d1(\mathit{self}) & \triangleq & \land\ \mathit{pc}[\mathit{self}] = \text{“d1”} \\
 & & \land\ \mathit{register2}' = [\mathit{register2}\ \text{EXCEPT}\ ![\mathit{self}] = \mathit{cash}[\mathit{self}] - \mathit{register1}[\mathit{self}]] \\
 & & \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![\mathit{self}] = \text{“d1b”}] \\
 & & \land\ \text{UNCHANGED}\ \langle \mathit{account}, \mathit{cash}, \mathit{iterations}, \mathit{stack}, \mathit{amount1\_}, \\
 & & \qquad\qquad \mathit{register1\_}, \mathit{register2\_}, \mathit{amount1}, \mathit{register1}, \mathit{total} \rangle
\end{array}
$$

$$
\begin{array}{lll}
d1b(\mathit{self}) & \triangleq & \land\ \mathit{pc}[\mathit{self}] = \text{“d1b”} \\
 & & \land\ \mathit{cash}' = [\mathit{cash}\ \text{EXCEPT}\ ![\mathit{self}] = \mathit{register2}[\mathit{self}]] \\
 & & \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![\mathit{self}] = \text{“d2”}] \\
 & & \land\ \text{UNCHANGED}\ \langle \mathit{account}, \mathit{iterations}, \mathit{stack}, \mathit{amount1\_}, \mathit{register1\_}, \\
 & & \qquad\qquad \mathit{register2\_}, \mathit{amount1}, \mathit{register1}, \mathit{register2}, \mathit{total} \rangle
\end{array}
$$

$$
\begin{array}{lll}
d2(\mathit{self}) & \triangleq & \land\ \mathit{pc}[\mathit{self}] = \text{“d2”} \\
 & & \land\ \mathit{pc}' = [\mathit{pc}\ \text{EXCEPT}\ ![\mathit{self}] = \mathit{Head}(\mathit{stack}[\mathit{self}]).\mathit{pc}] \\
 & & \land\ \mathit{register1}' = [\mathit{register1}\ \text{EXCEPT}\ ![\mathit{self}] = \mathit{Head}(\mathit{stack}[\mathit{self}]).\mathit{register1}] \\
 & & \land\ \mathit{register2}' = [\mathit{register2}\ \text{EXCEPT}\ ![\mathit{self}] = \mathit{Head}(\mathit{stack}[\mathit{self}]).\mathit{register2}]
\end{array}
$$

```
                 $\land amount1' = [amount1 \text{ EXCEPT } ![self] = Head(stack[self]).amount1]$
                 $\land stack' = [stack \text{ EXCEPT } ![self] = Tail(stack[self])]$
                 $\land \text{UNCHANGED } \langle account, cash, iterations, amount1\_, register1\_,$
                                        $register2\_, total \rangle$

$deposit(self) \triangleq deposit\_start(self) \lor d0b(self) \lor d0c(self) \lor d1(self)$
                 $\lor d1b(self) \lor d2(self)$

$start(self) \triangleq \land pc[self] = \text{“start”}$
                  $\land \text{IF } iterations[self] > 0$
                      $\text{THEN } \land pc' = [pc \text{ EXCEPT } ![self] = \text{“s1”}]$
                             $\land total' = total$
                      $\text{ELSE } \land Assert(iterations[self] = 0,$
                                       $\text{“Failure of assertion at line 74, column 7.”})$
                             $\land \text{IF } iterations[\text{“husband”}] = 0 \land iterations[\text{“wife”}] = 0$
                                 $\text{THEN } \land total' = [total \text{ EXCEPT } ![self] = account + cash[\text{“husband”}] + cash[\text{“}$
                                        $\land PrintT(total'[self])$
                                        $\land Assert(total'[self] = 120,$
                                                  $\text{“Failure of assertion at line 79, column 9.”})$
                                 $\text{ELSE } \land \text{TRUE}$
                                        $\land total' = total$
                             $\land pc' = [pc \text{ EXCEPT } ![self] = \text{“Done”}]$
                  $\land \text{UNCHANGED } \langle account, cash, iterations, stack, amount1\_,$
                                         $register1\_, register2\_, amount1, register1,$
                                         $register2 \rangle$

$s1(self) \triangleq \land pc[self] = \text{“s1”}$
               $\land \exists amount \in 1 \,..\, 2 :$
                   $\land \land amount1\_' = [amount1\_ \text{ EXCEPT } ![self] = amount]$
                     $\land stack' = [stack \text{ EXCEPT } ![self] = \langle [procedure \mapsto \text{“withdraw”},$
                                                               $pc \mapsto \text{“s2”},$
                                                               $register1\_ \mapsto register1\_[self],$
                                                               $register2\_ \mapsto register2\_[self],$
                                                               $amount1\_ \mapsto amount1\_[self]] \rangle$
                                                               $\circ stack[self]]$
                   $\land register1\_' = [register1\_ \text{ EXCEPT } ![self] = defaultInitValue]$
                   $\land register2\_' = [register2\_ \text{ EXCEPT } ![self] = defaultInitValue]$
                   $\land pc' = [pc \text{ EXCEPT } ![self] = \text{“withdraw\_start”}]$
               $\land \text{UNCHANGED } \langle account, cash, iterations, amount1, register1,$
                                      $register2, total \rangle$

$s2(self) \triangleq \land pc[self] = \text{“s2”}$
               $\land \exists amount \in 1 \,..\, 2 :$
                   $\land \land amount1' = [amount1 \text{ EXCEPT } ![self] = amount]$
                     $\land stack' = [stack \text{ EXCEPT } ![self] = \langle [procedure \mapsto \text{“deposit”},$
                                                               $pc \mapsto \text{“s3”},$
```

$$\begin{array}{l}
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{register}1 \;\mapsto\; \mathit{register}1[\mathit{self}], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{register}2 \;\mapsto\; \mathit{register}2[\mathit{self}], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{amount}1 \;\mapsto\; \mathit{amount}1[\mathit{self}]]\rangle \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \circ\, \mathit{stack}[\mathit{self}]] \\
\qquad\qquad\quad \land\; \mathit{register}1' = [\mathit{register}1 \text{ EXCEPT } ![\mathit{self}] = \mathit{defaultInitValue}] \\
\qquad\qquad\quad \land\; \mathit{register}2' = [\mathit{register}2 \text{ EXCEPT } ![\mathit{self}] = \mathit{defaultInitValue}] \\
\qquad\qquad\quad \land\; \mathit{pc}' = [\mathit{pc} \text{ EXCEPT } ![\mathit{self}] = \text{“deposit\_start”}] \\
\qquad\qquad \land\; \text{UNCHANGED } \langle \mathit{account}, \mathit{cash}, \mathit{iterations}, \mathit{amount}1\_, \mathit{register}1\_, \\
\qquad\qquad\qquad\qquad\quad \mathit{register}2\_, \mathit{total}\rangle
\end{array}$$

$$\begin{array}{l}
s3(\mathit{self}) \;\triangleq\; \land\; \mathit{pc}[\mathit{self}] = \text{“s3”} \\
\qquad\qquad\quad \land\; \mathit{iterations}' = [\mathit{iterations} \text{ EXCEPT } ![\mathit{self}] = \mathit{iterations}[\mathit{self}] - 1] \\
\qquad\qquad\quad \land\; \mathit{total}' = [\mathit{total} \text{ EXCEPT } ![\mathit{self}] = \mathit{account} + \mathit{cash}[\text{“husband”}] + \mathit{cash}[\text{“wife”}]] \\
\qquad\qquad\quad \land\; \mathit{pc}' = [\mathit{pc} \text{ EXCEPT } ![\mathit{self}] = \text{“start”}] \\
\qquad\qquad\quad \land\; \text{UNCHANGED } \langle \mathit{account}, \mathit{cash}, \mathit{stack}, \mathit{amount}1\_, \mathit{register}1\_, \\
\qquad\qquad\qquad\qquad\qquad\quad \mathit{register}2\_, \mathit{amount}1, \mathit{register}1, \mathit{register}2\rangle
\end{array}$$

$$\mathit{spouse}(\mathit{self}) \;\triangleq\; \mathit{start}(\mathit{self}) \lor s1(\mathit{self}) \lor s2(\mathit{self}) \lor s3(\mathit{self})$$

$$\begin{array}{l}
\mathit{Next} \;\triangleq\; (\exists\, \mathit{self} \in \mathit{ProcSet} : \mathit{withdraw}(\mathit{self}) \lor \mathit{deposit}(\mathit{self})) \\
\qquad\qquad \lor\; (\exists\, \mathit{self} \in \{\text{“husband”}, \text{“wife”}\} : \mathit{spouse}(\mathit{self})) \\
\qquad\qquad \lor\; \text{Disjunct to prevent deadlock on termination} \\
\qquad\qquad\quad ((\forall\, \mathit{self} \in \mathit{ProcSet} : \mathit{pc}[\mathit{self}] = \text{“Done”}) \land \text{UNCHANGED } \mathit{vars})
\end{array}$$

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

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

END TRANSLATION