The following is an STM version of the bank account example including waiting if the balance isn't enough (which is hard with locks/monitors).

```plaintext
let (!!) = readTVar   // Two simple abbreviations
let (==) = writeTVar

let retryIf cond = if cond then retry () else stm.Return()

let printing = newTVar false   // A "locking" version of printing
let tprintf f = Printf.kprintf (fun s ->
  stm { let! p = !!printing
    do! retryIf p
    do! printing == true } |> atomically
  printf "%s" s  // Not atomic: we can't undo printing
  printing == false |> atomically
) f

let sleepMaybe() = if (random 4) = 1 then Thread.Sleep (5*random 10)

type account(name:string) =
  let balance = newTVar 100
  member private this.adjust amount = stm { let! b = !!balance
    do! balance == b + (sleepMaybe(); amount) }
  member this.Balance = !!balance
  member this.Name = name
  member this.Withdraw amount = atomically (this.adjust -amount)
  member this.Deposit amount = atomically (this.adjust amount)
  member this.Transfer (toAcc:account) (amount:int) =
    stm { let! b = !!balance
      do! retryIf (b < amount)
      do! this.adjust -amount
      do! toAcc.adjust amount } |> atomically

let doTransfers (acc:account) toAcc () =
  for i in 1..20 do acc.Transfer toAcc 20
  sleepMaybe()
  let b1, b2 = stm { let! b1 = acc.Balance
    do! sleepMaybe()
    let! b2 = toAcc.Balance
    return (b1, b2) } |> atomically
  tprintf "%s balance: %d Other balance: %d\n" acc.Name b1 b2
  if b1+b2 <> 200 then raise (Exception "wrong sum")

let q2main() =
  let acc1=account("Account1")
  let acc2=account("Account2")

  startThread (doTransfers acc1 acc2)
  startThread (doTransfers acc2 acc1)

q2main()
```
Output:

Account2 balance: 80  Other balance: 120
Account1 balance: 100  Other balance: 100
Account1 balance: 80  Other balance: 120
Account1 balance: 60  Other balance: 140
Account1 balance: 40  Other balance: 160
Account1 balance: 20  Other balance: 180
Account1 balance: 0  Other balance: 200
Account1 balance: 0  Other balance: 200
Account2 balance: 200  Other balance: 0
Account2 balance: 180  Other balance: 20
Account1 balance: 0  Other balance: 200
Account2 balance: 180  Other balance: 20
Account2 balance: 180  Other balance: 20
Account1 balance: 20  Other balance: 180
Account1 balance: 0  Other balance: 200
Account2 balance: 180  Other balance: 20
Account1 balance: 20  Other balance: 180
Account2 balance: 180  Other balance: 20
Account1 balance: 20  Other balance: 180
Account1 balance: 0  Other balance: 200
Account2 balance: 200  Other balance: 0
Account2 balance: 180  Other balance: 20
Account2 balance: 180  Other balance: 20
Account1 balance: 40  Other balance: 160
Account1 balance: 20  Other balance: 180
Account1 balance: 0  Other balance: 200
Account2 balance: 200  Other balance: 0
Account1 balance: 0  Other balance: 200
Account2 balance: 200  Other balance: 0
Account2 balance: 180  Other balance: 20
Account1 balance: 0  Other balance: 200
Account2 balance: 200  Other balance: 0
Account2 balance: 180  Other balance: 20
Account2 balance: 180  Other balance: 20
Account2 balance: 120  Other balance: 80
Account1 balance: 100  Other balance: 100
Account1 balance: 80  Other balance: 120
Account2 balance: 120  Other balance: 80
Account2 balance: 100  Other balance: 100
Press any key to continue . . .