MODULE *recycler*

EXTENDS *Sequences*, *Integers*, *TLC*

```
  --algorithm recycler
variables
    $capacity \in [trash : 1 \,..\, 10,\ recycle : 1 \,..\, 10]$,
    $bins = [trash \mapsto \langle\rangle,\ recycle \mapsto \langle\rangle]$,
    $count = [trash \mapsto 0,\ recycle \mapsto 0]$,
    $item = [type : \{\text{"trash"}, \text{"recycle"}\},\ size : 1 \,..\, 6]$,
    $items \in item \times item \times item \times item$,
    $curr = \text{""}$ ;   helper: current item

macro add_item(type) begin
    $bins[type] := Append(bins[type], curr)$ ;
    $capacity[type] := capacity[type] - curr.size$ ;
    $count[type] := count[type] + 1$ ;
end macro

begin
    while $items \neq \langle\rangle$ do
        $curr := Head(items)$ ;
        $items := Tail(items)$ ;
        if $curr.type = \text{"recycle"} \land curr.size < capacity.recycle$ then
            add_item("recycle") ;
        elsif $curr.size < capacity.trash$ then
            add_item("trash") ;
        end if
    end while ;

    assert $capacity.trash \geq 0 \land capacity.recycle \geq 0$ ;
    assert $Len(bins.trash) = count.trash$ ;
    assert $Len(bins.recycle) = count.recycle$ ;
end algorithm ;
```

BEGIN TRANSLATION

VARIABLES *capacity*, *bins*, *count*, *item*, *items*, *curr*, *pc*

$vars \triangleq \langle capacity, bins, count, item, items, curr, pc\rangle$

$Init \triangleq$ Global variables

$\land\ capacity \in [trash : 1 \,..\, 10,\ recycle : 1 \,..\, 10]$
$\land\ bins = [trash \mapsto \langle\rangle,\ recycle \mapsto \langle\rangle]$
$\land\ count = [trash \mapsto 0,\ recycle \mapsto 0]$
$\land\ item = [type : \{\text{"trash"}, \text{"recycle"}\},\ size : 1 \,..\, 6]$
$\land\ items \in item \times item \times item \times item$
$\land\ curr = \text{""}$
$\land\ pc = \text{"Lbl\_1"}$

```
$Lbl\_1 \triangleq$ $\land\ pc = $ “Lbl_1”
          $\land$ IF $items \neq \langle\rangle$
                THEN $\land\ curr' = Head(items)$
                     $\land\ items' = Tail(items)$
                     $\land$ IF $curr'.type =$ “recycle” $\land\ curr'.size < capacity.recycle$
                           THEN $\land\ bins' = [bins$ EXCEPT $![$“recycle”$] = Append(bins[$“recycle”$], curr')]$
                                $\land\ capacity' = [capacity$ EXCEPT $![$“recycle”$] = capacity[$“recycle”$] - curr'.s$
                                $\land\ count' = [count$ EXCEPT $![$“recycle”$] = count[$“recycle”$] + 1]$
                           ELSE $\land$ IF $curr'.size < capacity.trash$
                                      THEN $\land\ bins' = [bins$ EXCEPT $![$“trash”$] = Append(bins[$“trash”$], cur$
                                           $\land\ capacity' = [capacity$ EXCEPT $![$“trash”$] = capacity[$“trash”$]$
                                           $\land\ count' = [count$ EXCEPT $![$“trash”$] = count[$“trash”$] + 1]$
                                      ELSE $\land$ TRUE
                                           $\land$ UNCHANGED $\langle capacity, bins,$
                                                              $count\rangle$
                     $\land\ pc' =$ “Lbl_1”
                ELSE $\land\ Assert(capacity.trash \geq 0 \land capacity.recycle \geq 0,$
                                “Failure of assertion at line 31, column 5.”$)$
                     $\land\ Assert(Len(bins.trash) = count.trash,$
                                “Failure of assertion at line 32, column 5.”$)$
                     $\land\ Assert(Len(bins.recycle) = count.recycle,$
                                “Failure of assertion at line 33, column 5.”$)$
                     $\land\ pc' =$ “Done”
                     $\land$ UNCHANGED $\langle capacity, bins, count, items, curr\rangle$
          $\land\ item' = item$

$Next \triangleq Lbl\_1$
            $\lor$  Disjunct to prevent deadlock on termination
              $(pc =$ “Done” $\land$ UNCHANGED $vars)$

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

$Termination \triangleq \Diamond(pc =$ “Done”$)$

END TRANSLATION
```

* Modification History
* Last modified *Tue Apr* 09 23:11:09 *CEST* 2019 by *jrediger*
* Created *Tue Apr* 09 22:53:18 *CEST* 2019 by *jrediger*