MODULE *trash*

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

VARIABLES

capacity,
bins,
count,
items,
item,
pc

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

$Init \triangleq$

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

$PutIn(bin,\ it) \triangleq$

$\land\ bins' = [bins \text{ EXCEPT } ![bin] = Append(bins[bin],\ it)]$
$\land\ count' = [count \text{ EXCEPT } ![bin] = count[bin] + 1]$
$\land\ capacity' = [capacity \text{ EXCEPT } ![bin] = capacity[bin] - it.size]$

$Process \triangleq$

- If the item is labeled as “recycling” and it is under the remaining capacity for the recycling bin, the item goes into recycling.
- If the item is labeled as “trash” OR the item is labeled as “recycling” and there is not enough recycling capacity AND there is sufficient capacity in the trash bin, the item goes into trash.
- Otherwise, it's dropped on the floor and somebody else gets to sweep it up.

$\land$ LET $it \triangleq Head(items)$ IN
  $\land\ items' = Tail(items)$
  $\land$ IF $it.type = \text{“recycle”} \land it.size < capacity.recycle$
    THEN $\land\ PutIn(\text{“recycle”},\ it)$
    ELSE IF $it.size < capacity.trash$
      THEN $\land\ PutIn(\text{“trash”},\ it)$
      ELSE $\land$ UNCHANGED $\langle bins, count, capacity \rangle$

$ProcessAll \triangleq$

$\land\ pc = \text{“init”}$
$\land$ IF $items \neq \langle\rangle$
  THEN $\land\ Process$
    $\land$ UNCHANGED $\langle pc, item \rangle$
  ELSE $\land\ pc' = \text{“done”}$
    $\land$ UNCHANGED $\langle capacity, bins, count, items, item \rangle$

$Done \triangleq$
$\quad \land\ pc = \text{“done”}$
$\quad \land\ \text{UNCHANGED}\ vars$

$Next \triangleq ProcessAll \lor Done$

$NoOverflow \triangleq capacity.trash \geq 0 \land capacity.recycle \geq 0$
$RecycleCountOk \triangleq Len(bins.recycle) = count.recycle$
$RecycleTrashOk \triangleq Len(bins.trash) = count.trash$