217 (idempotent permutation) You are given a list variable L of items in $\square L$ (not necessarily all different). Write a program to permute the list so that finally $L'\,L' = L'$.

After trying the question, scroll down to the solution.

§ For each item in the list, one of its occurrences must be at its index position. All the duplicate items can go anywhere else. For example, the list $[3; 2; 0; 3; 0]$ might be permuted to $[0; 3; 2; 3; 0]$ because $[0; 3; 2; 3; 0]\,[0; 3; 2; 3; 0] = [0; 3; 2; 3; 0]$. Or to $[0; 0; 2; 3; 3]$ because $[0; 0; 2; 3; 3]\,[0; 0; 2; 3; 3] = [0; 0; 2; 3; 3]$.

Let n be a natural variable. All changes to L will be by *swap*, which swaps the items at positions n and $L\,n$, defined as

$$swap = L := n \rightarrow L(L\,n) \mid L\,n \rightarrow L\,n \mid L$$

which ensures that L' will be a permutation of L. And since we are given $L(\square L): \square L$ to begin, that will remain true always. Define specifications S and R as

$$\begin{aligned} S &= L'\,L' = L' \\ &= \forall m: 0,..\#L \cdot L'(L'm) = L'm \end{aligned}$$

$$R = (\forall m: 0,..n \cdot L(L\,m) = L\,m) \Rightarrow S$$

$$Q = n < \#L \Rightarrow R$$

Refine them as

$$S \Leftarrow n := 0.\ R$$

$$R \Leftarrow \textbf{if } n = \#L \textbf{ then } ok \textbf{ else } Q \textbf{ fi}$$

$$\begin{aligned} Q \Leftarrow\ & \textbf{if } L\,n = L(L\,n) \textbf{ then } n := n+1.\ R \\ & \textbf{else if } n > L\,n \textbf{ then } swap.\ n := n+1.\ R \\ & \quad \textbf{else } swap.\ Q \textbf{ fi fi} \end{aligned}$$

Timing: for S use $t' \le t + \#L + ¢\S m: 0,..\#L \cdot L\,m \neq m$
for R use $t' \le t + \#L - n + ¢\S m: n,..\#L \cdot L\,m \neq m$

solution thanks to Łukasz Jakimczuk