288 (squash) Let L be a list variable assigned a non-empty list. Reassign it so that any run of two or more identical items is collapsed to a single item.

After trying the question, scroll down to the solution.

§ Let i be a natural variable used to index L. The program is

$$P \Leftarrow i:= 1.\ Q$$
$$Q \Leftarrow \textbf{if } i{=}\#L \textbf{ then } ok \textbf{ else if } L\,i = L(i-1) \textbf{ then } L:= L((0;..i)\ ;\ (i+1;..\#L)) \textbf{ else } i:= i+1 \textbf{ fi}.\ Q \textbf{ fi}$$

Now we need to define specifications P and Q, and then prove the two refinements. Part of the specification says that L and L' have the same items in them.

$$L(0,..\#L) = L'(0,..\#L')$$

Another part of the specification is that in L' no two adjacent items are equal.

$$\neg\exists j\colon 1,..\#L'\cdot\ L'j{=}L'(j-1)$$

The rest of the specification is that the items of L' are in the same order as in L. I don't know how to formalize that. So I'll prove what I can.

Let's start with

$$P = Q = L(0,..\#L) = L'(0,..\#L')$$

Here's the proof of the first refinement, starting with the right side.

$$\begin{array}{lll} & i:= 1.\ Q & \text{expand } Q \\ = & i:= 1.\ L(0,..\#L) = L'(0,..\#L') & \text{substitution law} \\ = & L(0,..\#L) = L'(0,..\#L') & \\ = & P & \end{array}$$

Now the last refinement, by cases. First case:

$$\begin{array}{lll} & i{=}\#L \land ok & \text{expand } ok \\ = & i{=}\#L \land L'{=}L \land i'{=}i & \\ \Rightarrow & Q & \end{array}$$

Middle case:

$$\begin{array}{ll} & i{\ne}\#L \land L\,i = L(i-1) \land (L:= L((0;..i)\ ;\ (i+1;..\#L)).\ Q) \\ = & \text{UNFINISHED} \\ \Rightarrow & Q \end{array}$$

Last case:

$$\begin{array}{lll} & i{\ne}\#L \land L\,i \ne L(i-1) \land (i:= i+1.\ Q) & \text{expand } Q \\ = & i{\ne}\#L \land L\,i \ne L(i-1) \land (i:= i+1.\ L(0,..\#L) = L'(0,..\#L')) & \text{substitution law} \\ = & i{\ne}\#L \land L\,i \ne L(i-1) \land L(0,..\#L) = L'(0,..\#L') & \text{specialization} \\ \Rightarrow & Q & \end{array}$$

Now redefine

$$P = \neg\exists j\colon 1,..\#L'\cdot\ L'j{=}L'(j-1)$$
$$Q = \neg\exists j\colon i,..\#L'\cdot\ L'j{=}L'(j-1)$$

Here's the proof of the first refinement, starting with the right side.

$$\begin{array}{lll} & i:= 1.\ Q & \text{expand } Q \\ = & i:= 1.\ \neg\exists j\colon i,..\#L'\cdot\ L'j{=}L'(j-1) & \text{substitution law} \\ = & \neg\exists j\colon 1,..\#L'\cdot\ L'j{=}L'(j-1) & \\ = & P & \end{array}$$

Now the last refinement, by cases. First case:

$$\begin{array}{lll} & i{=}\#L \land ok & \text{expand } ok \\ = & i{=}\#L \land L'{=}L \land i'{=}i & \text{null domain} \\ \Rightarrow & Q & \end{array}$$

Middle case:

$$\begin{array}{ll} & i{\ne}\#L \land L\,i = L(i-1) \land (L:= L((0;..i)\ ;\ (i+1;..\#L)).\ Q) \\ = & \text{UNFINISHED} \\ \Rightarrow & Q \end{array}$$

Last case:

$i \ne \#L \wedge L\ i \ne L(i-1) \wedge (i := i+1.\ \ Q)$

$=$ UNFINISHED

$\Rightarrow$ Q

The recursive time is $\#L-1$. Redefine

$P\ \ =\ \ t' = t + \#L - 1$

$Q\ \ =\ \ t' = t + \#L - i$

and insert the time increment

$P\ \ \Leftarrow\ \ i := 1.\ \ Q$

$Q\ \ \Leftarrow\ \ \mathbf{if}\ i = \#L\ \mathbf{then}\ ok$

$\quad\quad \mathbf{else\ if}\ L\ i = L(i-1)\ \mathbf{then}\ L := L((0;..i)\ ;\ (i+1;..\#L))$

$\quad\quad\quad \mathbf{else}\ i := i+1\ \mathbf{fi}.\ \ t := t+1.\ \ Q\ \mathbf{fi}$

UNFINISHED