215 (permutation inverse) You are given a list variable P of different items in $\square P$. Write a program for $P\ P' = [0;..\#P]$.

After trying the question, scroll down to the solution.

§ Linear time with 1 extra bit per item (plus a few other binary and natural variables) should be enough. Outer loop checks for uninverted items as indicated by the extra bit. When one is found, the inner loop inverts the cycle that includes that item.

This solution uses state variables P and L, which are lists, and n, which is natural. And if P and P' are inverses, then $P\,P' = P'\,P = [0;..\#P]$.

$$P'\,P = [0;..\#P] \;\Longleftarrow\; L{:=}\,P.\;\; n{:=}\,0.\;\; n{\le}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot P(L\,i){=}i) \;\Rightarrow\; P'\,L = [0;..\#P]$$

Proof: substitution law twice.

$$n{\le}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot P(L\,i){=}i) \;\Rightarrow\; P'\,L = [0;..\#P] \;\Longleftarrow$$
$$\quad \textbf{if } n{=}\#L \textbf{ then } ok$$
$$\quad \textbf{else } P{:=}\,L\,n \rightarrow n \mid P.\; n{:=}\,n{+}1.\; n{\le}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot P(L\,i){=}i) \Rightarrow P'\,L = [0;..\#P] \textbf{ fi}$$

Proof: by cases. First case:

$\quad (n{\le}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot P(L\,i){=}i) \;\Rightarrow\; P'\,L = [0;..\#P]) \;\Leftarrow\; n{=}\#L \wedge ok$ — portation

$=\quad n{=}\#L \wedge ok \wedge n{\le}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot P(L\,i){=}i) \;\Rightarrow\; P'\,L = [0;..\#P]$ — expand ok

$=\quad n{=}\#L{=}\#P \wedge P'{=}P \wedge L'{=}L \wedge n'{=}n \wedge (\forall i\colon 0,..n\cdot P(L\,i){=}i) \;\Rightarrow\; P'\,L = [0;..\#P]$ context

$=\quad n{=}\#L{=}\#P \wedge P'{=}P \wedge L'{=}L \wedge n'{=}n \wedge (\forall i\colon \square P\cdot P(L\,i){=}i) \;\Rightarrow\; P\,L = [0;..\#P]$

specialize

$\Leftarrow\quad \#L{=}\#P \wedge (\forall i\colon \square P\cdot P(L\,i){=}i) \;\Rightarrow\; P\,L = [0;..\#P]$ — HINT NEEDED

$=\quad \top$

Before tackling the other case, here is a lemma.

$\quad \forall i\colon 0,..n{+}1\cdot (L\,n \rightarrow n \mid P)(L\,i) = i$ — definition of $\forall$

$=\quad (\forall i\colon 0,..n\cdot (L\,n \rightarrow n \mid P)(L\,i) = i) \;\wedge\; (L\,n \rightarrow n \mid P)(L\,n) = n$ — definition of $\mid$

$=\quad (\forall i\colon 0,..n\cdot (L\,n \rightarrow n \mid P)(L\,i) = i) \;\wedge\; \top$ — identity

$=\quad \forall i\colon 0,..n\cdot (L\,n \rightarrow n \mid P)(L\,i) = i$

Now the other case.

$\quad (\quad n{\le}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot P(L\,i){=}i) \;\Rightarrow\; P'\,L = [0;..\#P]$

$\Leftarrow\quad n{\ne}\#L$

$\quad \wedge\, (P{:=}\,L\,n \rightarrow n \mid P.\;\; n{:=}\,n{+}1.\;\; n{\le}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot P(L\,i){=}i) \Rightarrow P'L{=}[0;..\#P]))$

portation

$=\quad n{\ne}\#L$

$\quad \wedge\, (P{:=}\,L\,n \rightarrow n \mid P.\;\; n{:=}\,n{+}1.\;\; n{\le}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot P(L\,i) = i) \Rightarrow P'L{=}[0;..\#P])$

$\quad \wedge\, n{\le}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot P(L\,i) = i)$

$\quad \Rightarrow\; P'\,L = [0;..\#P]$

substitution law twice

$=\quad n{\ne}\#L$

$\quad \wedge\; (\quad n{+}1{\le}\#L{=}\#(L\,n \rightarrow n \mid P) \wedge (\forall i\colon 0,..n{+}1\cdot (L\,n \rightarrow n \mid P)(L\,i) = i)$

$\quad\quad \Rightarrow P'L{=}[0;..\#(L\,n \rightarrow n \mid P)])$

$\quad \wedge\;\; n{\le}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot P(L\,i) = i)$

$\quad \Rightarrow\; P'\,L = [0;..\#P]$ — combine $n{\ne}\#L \wedge n{\le}\#L$

$=\quad n{<}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot P(L\,i) = i)$

$\quad \wedge\; (\quad n{+}1{\le}\#L{=}\#(L\,n \rightarrow n \mid P) \wedge (\forall i\colon 0,..n{+}1\cdot (L\,n \rightarrow n \mid P)(L\,i) = i)$

$\quad\quad \Rightarrow P'L{=}[0;..\#(Ln{\rightarrow}n \mid P)])$

$$\begin{array}{lll}
 & \Rightarrow\ P'\ L = [0;..\#P] & \text{lemma and } n{+}1 \le \#L = n < \#L \\
= & \qquad n{<}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot\ P(L\ i) = i) & \\
 & \quad \wedge\ \ (\quad n{<}\#L{=}\#(L\ n \rightarrow n \mid P) \wedge (\forall i\colon 0,..n\cdot\ (L\ n \rightarrow n \mid P)(L\ i) = i) & \\
 & \qquad\ \Rightarrow P'L{=}[0;..\#(L\ n \rightarrow n \mid P)]) & \\
 & \Rightarrow\ P'\ L = [0;..\#P] & \#(L\ n \rightarrow n \mid P) = \#P \\
= & \qquad n{<}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot\ P(L\ i) = i) & \\
 & \quad \wedge\ \ (\quad n{<}\#L{=}\#(Ln{\rightarrow}n \mid P) \wedge (\forall i\colon 0,..n\cdot\ (L\ n \rightarrow n \mid P)(L\ i) = i) & \\
 & \qquad\ \Rightarrow P'L{=}[0;..\#P]) & \\
 & \Rightarrow\ P'\ L = [0;..\#P] & \text{discharge} \\
= & \quad n{<}\#L{=}\#P \wedge (\forall i\colon 0,..n\cdot\ P(L\ i) = i) \wedge P'L{=}[0;..\#P] & \\
 & \Rightarrow\ P'\ L = [0;..\#P] & \text{specialize} \\
= & \top &
\end{array}$$

Now for the timing.

$$t' \le t{+}\#P \quad \Longleftarrow \quad L{:=}\ P.\ \ n{:=}\ 0.\ \ n{\le}\#L \Rightarrow t' \le t{+}\#L{-}n$$

Proof: substitution law twice.

$$\begin{array}{l}
n{\le}\#L \Rightarrow t' \le t{+}\#L{-}n \quad \Longleftarrow \\
\qquad \textbf{if } n{=}\#L \textbf{ then } ok \\
\qquad \textbf{else } P{:=}\ L\ n \rightarrow n \mid P.\ \ n{:=}\ n{+}1.\ \ t{:=}\ t{+}1.\ \ n{\le}\#L \Rightarrow t' \le t{+}\#L{-}n \textbf{ fi}
\end{array}$$

Proof: by cases. First case:

$$\begin{array}{lll}
 & (n{\le}\#L \Rightarrow t' \le t{+}\#L{-}n)\ \ \Leftarrow\ \ n{=}\#L \wedge ok & \text{expand } ok \\
= & (n{\le}\#L \Rightarrow t' \le t{+}\#L{-}n)\ \ \Leftarrow\ \ n{=}\#L \wedge P'{=}P \wedge L'{=}L \wedge n'{=}n \wedge t'{=}t & \text{context} \\
= & (\#L{\le}\#L \Rightarrow t \le t{+}\#L{-}\#L)\ \ \Leftarrow\ \ n{=}\#L \wedge P'{=}P \wedge L'{=}L \wedge n'{=}n \wedge t'{=}t & \text{arithmetic} \\
= & \top\ \ \Leftarrow\ \ n{=}\#L \wedge P'{=}P \wedge L'{=}L \wedge n'{=}n \wedge t'{=}t & \text{base} \\
= & \top &
\end{array}$$

Now the other case.

$$\begin{array}{lll}
 & \quad (n{\le}\#L \Rightarrow t' \le t{+}\#L{-}n) & \\
 & \Leftarrow\ \ n{\ne}\#L \wedge (P{:=}\ L\ n \rightarrow n \mid P.\ \ n{:=}\ n{+}1.\ \ t{:=}\ t{+}1.\ \ n{\le}\#L \Rightarrow t' \le t{+}\#L{-}n) & \\
 & & \text{substitution law three times} \\
= & (n{\le}\#L \Rightarrow t' \le t{+}\#L{-}n)\ \ \Leftarrow\ \ n{\ne}\#L \wedge (n{+}1{\le}\#L \Rightarrow t' \le t{+}1{+}\#L{-}n{-}1) & \text{arithmetic} \\
= & (n{\le}\#L \Rightarrow t' \le t{+}\#L{-}n)\ \ \Leftarrow\ \ n{\ne}\#L \wedge (n{+}1{\le}\#L \Rightarrow t' \le t{+}\#L{-}n) & \text{portation} \\
= & n{\ne}\#L \wedge (n{+}1{\le}\#L \Rightarrow t' \le t{+}\#L{-}n) \Rightarrow (n{\le}\#L \Rightarrow t' \le t{+}\#L{-}n) & \text{context } n{\ne}\#L \\
= & n{\ne}\#L \wedge (n{<}\#L \Rightarrow t' \le t{+}\#L{-}n)\ \Rightarrow\ (n{<}\#L \Rightarrow t' \le t{+}\#L{-}n) & \text{specialize} \\
= & \top &
\end{array}$$

In the specification $n{\le}\#L \Rightarrow t' \le t{+}\#L{-}n$, the antecedent is not needed for proof. The proof is easier without it. It is needed to make the specification implementable.