197 (all present) Given a natural number and a list of natural numbers, write a program to determine if every natural number up to the given number is an item in the list.

After trying the question, scroll down to the solution.

§ Let the natural number be n, and let the list be L. We will look through L from beginning to end, remembering which numbers in $0,..n$ have been seen. Let B: $[n*bin]$ be a list variable to say what has been found. To start, $B = [n*\bot]$ saying that no numbers in $0,..n$ have been found yet. In the end, if $B' = [n*\top]$ then all n numbers are present in list L. The specification is S, defined as

$$S \;=\; \forall i\colon 0,..n\cdot\ B'i = \exists j\colon 0,..\#L\cdot\ L\,j = i$$

Introduce natural variable k and define

$$R \;=\; 0{\le}k{\le}\#L \ \wedge\ (\forall i\colon 0,..n\cdot\ B\,i = \exists j\colon 0,..k\cdot\ L\,j = i) \ \Rightarrow\ S$$

Now refine

$$S \;\Leftarrow\; B{:=}\,[n*\bot].\ \ k{:=}\,0.\ \ R$$

$$R \;\Leftarrow\; \textbf{if } k{=}\#L \textbf{ then } ok \textbf{ else if } L\,k < n \textbf{ then } B{:=}\,L\,k \rightarrow \top \mid B \textbf{ else } ok \textbf{ fi}.\ k{:=}\,k{+}1.\ R \textbf{ fi}$$

Proof of the S refinement:

$$\begin{array}{ll}
& B{:=}\,[n*\bot].\ \ k{:=}\,0.\ \ R \hfill \text{expand } R\\
= & B{:=}\,[n*\bot].\ \ k{:=}\,0.\ \ 0{\le}k{\le}\#L \ \wedge\ (\forall i\colon 0,..n\cdot\ B\,i = \exists j\colon 0,..k\cdot\ L\,j = i) \ \Rightarrow\ S\\
& \hfill \text{substitution law twice. Note: } S \text{ does not mention } B \text{ nor } k\\
= & 0{\le}0{\le}\#L \ \wedge\ (\forall i\colon 0,..n\cdot\ [n*\bot]i = \exists j\colon 0,..0\cdot\ L\,j = i) \ \Rightarrow\ S\\
= & 0{\le}0{\le}\#L \ \wedge\ (\forall i\colon 0,..n\cdot\ \bot = \exists j\colon 0,..0\cdot\ L\,j = i) \ \Rightarrow\ S\\
= & \top \ \wedge\ \top \ \Rightarrow\ S\\
= & S
\end{array}$$

Proof of the R refinement, by cases. First case

$$\begin{array}{ll}
& k{=}\#L \wedge ok \ \Rightarrow\ R \hfill \text{expand } ok \text{ and } R\\
= & k{=}\#L \wedge k'{=}k \wedge B'{=}B \ \Rightarrow\ (0{\le}k{\le}\#L \ \wedge\ (\forall i\colon 0,..n\cdot\ B\,i = \exists j\colon 0,..k\cdot\ L\,j = i) \ \Rightarrow\ S)\\
& \hfill \text{portation}\\
= & k{=}\#L \wedge k'{=}k \wedge B'{=}B \wedge 0{\le}k{\le}\#L \ \wedge\ (\forall i\colon 0,..n\cdot\ B\,i = \exists j\colon 0,..k\cdot\ L\,j = i) \ \Rightarrow\ S \quad \text{expand } S\\
= & \quad k{=}\#L \wedge k'{=}k \wedge B'{=}B \wedge 0{\le}k{\le}\#L \ \wedge\ (\forall i\colon 0,..n\cdot\ B\,i = \exists j\colon 0,..k\cdot\ L\,j = i)\\
& \Rightarrow\ (\forall i\colon 0,..n\cdot\ B'i = \exists j\colon 0,..\#L\cdot\ L\,j = i) \hfill \text{context then specialization}\\
= & \top
\end{array}$$

For the second case of the R refinement, I start with a subexpression of the right side.

$$\begin{array}{ll}
& \textbf{if } L\,k < n \textbf{ then } B{:=}\,L\,k \rightarrow \top \mid B \textbf{ else } ok \textbf{ fi}.\ \ k{:=}\,k{+}1.\ \ R \hfill \text{distribute}\\
= & \textbf{if } L\,k < n \textbf{ then } B{:=}\,L\,k \rightarrow \top \mid B.\ \ k{:=}\,k{+}1.\ \ R \textbf{ else } ok.\ \ k{:=}\,k{+}1.\ \ R \textbf{ fi} \ \ \text{expand } R \text{ twice}\\
& \hfill \text{and substitution law and } ok \text{ is identity}\\
= & \textbf{if } L\,k < n \textbf{ then } 0{\le}k{+}1{\le}\#L \ \wedge\ (\forall i\colon 0,..n\cdot\ (L\,k \rightarrow \top \mid B)i = \exists j\colon 0,..k{+}1\cdot\ L\,j = i) \ \Rightarrow\ S\\
& \textbf{else } 0{\le}k{+}1{\le}\#L \ \wedge\ (\forall i\colon 0,..n\cdot\ B\,i = \exists j\colon 0,..k{+}1\cdot\ L\,j = i) \ \Rightarrow\ S \textbf{ fi}\\
= & \text{UNFINISHED}
\end{array}$$

Now the second case of the R refinement, using the previous simplification:

$$\begin{array}{ll}
& k{\ne}\#L \wedge (\textbf{if } L\,k < n \textbf{ then } B{:=}\,L\,k \rightarrow \top \mid B \textbf{ else } ok \textbf{ fi}.\ k{:=}\,k{+}1.\ R) \ \Rightarrow\ R\\
= & \text{UNFINISHED}\\
= & \top
\end{array}$$

The recursive time is $\#L$.