161 (running total) Given list variable L and any other variables you need, write a program to convert L into a list of cumulative sums. Formally,

(a) $\forall n\colon \Box L\cdot L'n = \Sigma L\,[0;..n]$

(b) $\forall n\colon \Box L\cdot L'n = \Sigma L\,[0;..n+1]$

After trying the question, scroll down to the solution.

(a) $\forall n\colon \square L\cdot L'n = \Sigma L[0;..n]$

§ Let $Q = (\forall n\colon 0,..m\cdot L'n = L\,n) \land (\forall n\colon m,..\#L\cdot L'n = s + \Sigma L[m;..n])$

Then

$\forall n\colon \square L\cdot L'n = \Sigma L[0;..n] \Leftarrow s{:=}\,0.\ m{:=}\,0.\ Q$

$Q \Leftarrow$ **if** $m{=}\#L$ **then** ok **else** $s{:=}\,s + L\,m.\ L{:=}\,m \rightarrow s - L\,m \mid L.\ m{:=}\,m{+}1.\ Q$ **fi**

Proof of the first refinement, starting with the right side.

$s{:=}\,0.\ m{:=}\,0.\ Q$ replace Q

$= s{:=}\,0.\ m{:=}\,0.\ (\forall n\colon 0,..m\cdot L'n = L\,n) \land (\forall n\colon m,..\#L\cdot L'n = s + \Sigma L[m;..n])$

substitution law twice

$= (\forall n\colon 0,..0\cdot L'n = L\,n) \land (\forall n\colon \square L\cdot L'n = 0 + \Sigma L[0;..n])$

arithmetic (base), quantifier law ($null$ domain)

$= \top \land (\forall n\colon \square L\cdot L'n = \Sigma L[0;..n])$ base

$= (\forall n\colon \square L\cdot L'n = \Sigma L[0;..n])$

Proof of the last refinement by cases. First case:

$m{=}\#L \land ok \Rightarrow Q$ expand ok and Q

$= \quad m{=}\#L \land L'{=}L \land s'{=}s \land m'{=}m$

$\Rightarrow (\forall n\colon 0,..m\cdot L'n = L\,n) \land (\forall n\colon m,..\#L\cdot L'n = s + \Sigma L[m;..n])$

use the antecedent as context in the consequent

$= \quad m{=}\#L \land L'{=}L \land s'{=}s \land m'{=}m$

$\Rightarrow (\forall n\colon \square L\cdot L\,n = L\,n) \land (\forall n\colon \#L,..\#L\cdot L\,n = s + \Sigma L[m;..n])$

in the consequent, the left conjunct says $L{=}L$,
and the right conjunct has a $null$ domain

$= m{=}\#L \land L'{=}L \land s'{=}s \land m'{=}m \Rightarrow \top \land \top$ idempotence and base

$= \top$

Proof of the last refinement, last case, starting with the right side

$m{\ne}\#L \land (s{:=}\,s + L\,m.\ L{:=}\,m \rightarrow (s - L\,m) \mid L.\ m{:=}\,m{+}1.\ Q)$ expand Q

$= m{\ne}\#L \land (s{:=}\,s + L\,m.\ L{:=}\,m \rightarrow (s - L\,m) \mid L.\ m{:=}\,m{+}1.$
$(\forall n\colon 0,..m\cdot L'n = L\,n) \land (\forall n\colon m,..\#L\cdot L'n = s + \Sigma L[m;..n]))$

Substitution Law three times

$= \quad m{\ne}\#L$
$\land (\forall n\colon 0,..m{+}1\cdot L'n = (m{\rightarrow}s \mid L)\,n)$
$\land (\forall n\colon m{+}1,..\#(m{\rightarrow}s \mid L)\cdot L'n = s + L\,m + \Sigma\,(m{\rightarrow}s \mid L)\,[m{+}1;..n]))$

a quantifier law allows us to break the domain of the first quantification

$= \quad m{\ne}\#L$
$\land (\forall n\colon 0,..m\cdot L'n = (m{\rightarrow}s \mid L)\,n) \land L'm = (m{\rightarrow}s \mid L)\,m$
$\land (\forall n\colon m{+}1,..\#(m{\rightarrow}s \mid L)\cdot L'n = s + L\,m + \Sigma\,(m{\rightarrow}s \mid L)\,[m{+}1;..n])$

In the first quantification, n is not in $0,..m$. In the last one, m is not in $m{+}1,\ n$.

$= \quad m{\ne}\#L$
$\land (\forall n\colon 0,..m\cdot L'n = L\,n) \land L'm = s$
$\land (\forall n\colon m{+}1,..\#L\cdot L'n = s + L\,m + \Sigma L[m{+}1;..n])$

$= \quad m{\ne}\#L$
$\land (\forall n\colon 0,..m\cdot L'n = L\,n) \land L'm = s$
$\land (\forall n\colon m{+}1,..\#L\cdot L'n = s + \Sigma L[m;..n])$

$= \quad m{\ne}\#L$
$\land (\forall n\colon 0,..m\cdot L'n = L\,n)$
$\land (\forall n\colon m,..\#L\cdot L'n = s{+}\Sigma L[m;..n])$

$\Rightarrow Q$

The timing refinements are

$t' = t{+}\#L \Leftarrow s{:=}\,0.\ m{:=}\,0.\ t' = t{+}\#L{-}m$

$t' = t{+}\#L{-}m \Leftarrow$ **if** $m{=}\#L$ **then** ok
else $s{:=}\,s + L\,m.\ L{:=}\,m \rightarrow (s - L\,m) \mid L.\ m{:=}\,m{+}1.\ t{:=}\,t{+}1.$

$t' = t+\#L-m$ **fi**

And here are the timing proofs. First refinement, starting with the right side:

$$s:= 0.\ \ m:= 0.\ \ t' = t+\#L-m \qquad \text{Substitution Law twice, and arithmetic}$$
$$= \quad t' = t+\#L$$

Last refinement, first case:

$$m=\#L \land ok \ \Rightarrow\ t' = t+\#L-m \qquad \text{context and arithmetic}$$
$$= \quad m=\#L \land ok \ \Rightarrow\ t' = t \qquad \text{specialization and definition of } ok$$
$$\Longrightarrow \quad \top$$

Last refinement, last case, starting with the right side:

$$m\ne\#L \land (s:= s+L\,m.\ \ L:= m \to (s - L\,m) \mid L.\ \ m:= m+1.\ \ t:= t+1.\ \ t' = t+\#L-m)$$
$$\text{specialization and four uses of Substitution Law}$$
$$\Longrightarrow \quad t' = t+1+\#L-(m+1) \qquad \text{arithmetic}$$
$$= \quad t' = t+\#L-m$$

(b) $\forall n\colon \square L\cdot\ L'n \ =\ \Sigma\, L\,[0;..n+1]$

§ This is similar to part (a), but a little uglier. Let

$$Q \ = \ \#L\ge1 \ \Rightarrow\ (\forall n\colon 0,..m\cdot\ L'n \ =\ L\,n)$$
$$\land\ \ (\forall n\colon m,..\#L\cdot\ L'n \ =\ L(m-1) + \Sigma\, L\,[m;..n+1])$$

The antecedent $\#L\ge1$ is needed to make it “work” for empty lists.

$$\forall n\colon \square L\cdot\ L'n \ =\ \Sigma\, L\,[0;..n+1] \ \Leftarrow\ m:= 1.\ \ Q$$
$$Q \ \Leftarrow\ \textbf{if } m\ge\#L \textbf{ then } ok \textbf{ else } L:= m \to L\,m + L\,(m-1) \mid L.\ \ m:= m+1.\ \ Q \textbf{ fi}$$

Now here's a **for**-loop solution. Let P be the specification we were given, and define

$$F\,i \ = \ (\forall n\colon (0,..i)\cdot\ L'n \ =\ L\,n)\ \land\ (\forall n\colon i,..\#L\cdot\ L'n \ =\ L(i-1) + \Sigma\, L\,[i;..n+1])$$

Then the solution is

$$P \ \Leftarrow\ \textbf{if } \#L=0 \textbf{ then } ok \textbf{ else } \#L>0 \Rightarrow P \textbf{ fi}$$
$$\#L>0 \Rightarrow P \ \Leftarrow\ F\,1$$
$$F\,1 \ \Leftarrow\ \textbf{for } i:= 1;..\#L \textbf{ do } L\,i:= L\,(i-1) + L\,i \textbf{ od}$$

We prove this last refinement by proving

$$F\,i \ \Leftarrow\ i\colon 1,..\#L \ \land\ (L\,i:= L\,(i-1) + L\,i.\ \ F(i+1))$$
$$F(\#L) \ \Leftarrow\ ok$$

Here is a parallel solution that takes $log\,(\#L)$ time.

for $j:= 0;..ceil(log(\#L))$ in sequence
do for $k:= 2^j,..\#L$ in parallel
do $L\,k:= L\,(k-2^j) + L\,k$ **od od**