

MODULE *SumSequence*

This module contains a trivial *PlusCal* algorithm to sum the elements of a sequence of integers, together with its non-trivial complete *TLAPS*-checked proof.

This algorithm is one of the examples in Section 7.3 of “Proving Safety Properties”, which is at <http://lamport.azurewebsites.net/tla/proving-safety.pdf>

EXTENDS *Integers*, *SequenceTheorems*, *NaturalsInduction*, *TLAPS*

To facilitate model checking, we assume that the sequence to be summed consists of integers in a set *Values* of integers.

CONSTANT *Values*

ASSUME $ValAssump \triangleq Values \subseteq Int$

In order to be able to express correctness of the algorithm, we define in TLA+ an operator *SeqSum* so that, if *s* is the sequence

*s*_1, ... , *s*_n

of integers, then *SumSeq(s)* equals

*s*_1 + ... + *s*_n

The obvious TLA+ definition of *SeqSum* is

RECURSIVE *SeqSum*(-)
SeqSum(*s*) \triangleq IF *s* = $\langle \rangle$ THEN 0 ELSE *s*[1] + *SeqSum*(*Tail*(*s*))

However, *TLAPS* does not yet handle recursive operator definitions, but it does handle recursive function definitions. So, we define *SeqSum* in terms of a recursively defined function.

SeqSum(*s*) \triangleq
 LET $SS[ss \in Seq(Int)] \triangleq$ IF *ss* = $\langle \rangle$ THEN 0 ELSE *ss*[1] + *SS*[*Tail*(*ss*)]
 IN *SS*[*s*]

Here's the algorithm. It initially sets *seq* to an arbitrary sequence of integers in *Values* and leaves its value unchanged. It terminates with the variable *sum* equal to the sum of the elements of *seq*.

```
--fair algorithm SumSequenceF
  variables seq ∈ Seq(Values), sum = 0, n = 1 ;
  f a: while ( n ≤ Len(seq) )
    f sum := sum + seq[n] ;
    n := n + 1 ;
  g
g
```

BEGIN TRANSLATION

VARIABLES *seq*, *sum*, *n*, *pc*

vars $\triangleq \langle seq, sum, n, pc \rangle$

Init \triangleq Global variables
 $\wedge seq \in Seq(Values)$

$$\begin{aligned}
& \wedge \textit{sum} = 0 \\
& \wedge \textit{n} = 1 \\
& \wedge \textit{pc} = \text{"a"}
\end{aligned}$$

$$\begin{aligned}
\textit{a} \triangleq & \wedge \textit{pc} = \text{"a"} \\
& \wedge \text{IF } \textit{n} \leq \textit{Len}(\textit{seq}) \\
& \quad \text{THEN } \wedge \textit{sum}' = \textit{sum} + \textit{seq}[\textit{n}] \\
& \quad \wedge \textit{n}' = \textit{n} + 1 \\
& \quad \wedge \textit{pc}' = \text{"a"} \\
& \quad \text{ELSE } \wedge \textit{pc}' = \text{"Done"} \\
& \quad \wedge \text{UNCHANGED } \langle \textit{sum}, \textit{n} \rangle \\
& \wedge \textit{seq}' = \textit{seq}
\end{aligned}$$

$$\begin{aligned}
\textit{Next} \triangleq & \textit{a} \\
& \vee \text{Disjunct to prevent deadlock on termination} \\
& (\textit{pc} = \text{"Done"} \wedge \text{UNCHANGED } \textit{vars})
\end{aligned}$$

$$\begin{aligned}
\textit{Spec} \triangleq & \wedge \textit{Init} \wedge \Box[\textit{Next}]_{\textit{vars}} \\
& \wedge \text{WF}_{\textit{vars}}(\textit{Next})
\end{aligned}$$

$$\wedge \text{sum} = \text{SeqSum}([i \in 1 \dots (n-1) \mapsto \text{seq}[i]]) \\ \wedge (\text{pc} = \text{"Done"}) \Rightarrow (n = \text{Len}(\text{seq}) + 1)$$

TLC can check that *Inv* is an inductive invariant on a large enough model to give us confidence in its correctness. We can therefore try to use it to prove the postcondition.

In the course of writing the proof, *I* found that *I* needed two simple properties of sequences and *SeqSum*. The first essentially states that the definition of *SeqSum* is correct—that is, that it defines the operator we expect it to. TLA+ doesn't require you to prove anything when making a definition, and it allows you to write silly recursive definitions like

```

RECURSIVE NotFactorial(_)
  NotFactorial(i)  $\triangleq$  IF i = 0 THEN 1 ELSE i * NotFactorial(i + 1)

```

Writing this definition doesn't mean that *NonFactorial*(4) actually equals $4 * \text{NonFactorial}(5)$. I think it actually does, but I'm not sure. I do know that it doesn't imply that *NonFactorial*(4) is a natural number. But the recursive definition of *SeqSum* is sensible, and we can prove the following lemma, which implies that *SeqSum*(⟨1, 2, 3, 4⟩) equals $1 + \text{SeqSum}(\langle 2, 3, 4 \rangle)$.

LEMMA *Lemma1* \triangleq
 $\forall s \in \text{Seq}(\text{Int}) :$
 $\text{SeqSum}(s) = \text{IF } s = \langle \rangle \text{ THEN } 0 \text{ ELSE } s[1] + \text{SeqSum}(\text{Tail}(s))$

What makes a formal proof of the algorithm non-trivial is that the definition of *SeqSum* essentially computes *SeqSum*(*seq*) by summing the elements of *seq* from left to right, starting with *seq*[1]. However, the algorithm sums the elements from right to left, starting with *seq*[*Len*(*s*)]. Proving the correctness of the algorithm requires proving that the two ways of computing the sum produce the same result. To state that result, it's convenient to define the operator *Front* on sequences to be the mirror image of *Tail*:

$$\text{Front}(\langle 1, 2, 3, 4 \rangle) = \langle 2, 3, 4 \rangle$$

This operator is defined in the *SequenceTheorems* module. I find it more convenient to use the slightly different definition expressed by this theorem.

THEOREM *FrontDef* $\triangleq \forall S : \forall s \in \text{Seq}(S) :$
 $\text{Front}(s) = [i \in 1 \dots (\text{Len}(s) - 1) \mapsto s[i]]$

BY DEF *Front*, *SubSeq*

LEMMA *Lemma5* $\triangleq \forall s \in \text{Seq}(\text{Int}) :$
 $(\text{Len}(s) > 0) \Rightarrow$
 $(\text{SeqSum}(s) = \text{SeqSum}(\text{Front}(s)) + s[\text{Len}(s)])$

OBVIOUS
 $\langle 2 \rangle 1. \text{TypeOK}$
 BY *Lemma1*, *ValAssump* DEF *Init*, *Inv*, *TypeOK*
 $\langle 2 \rangle 2. \text{sum} = \text{SeqSum}([i \in 1 \dots (n-1) \mapsto \text{seq}[i]])$
 $\langle 3 \rangle 1. (n-1) = 0$
 BY DEF *Init*
 $\langle 3 \rangle 2. [i \in 1 \dots 0 \mapsto \text{seq}[i]] = \langle \rangle$
 OBVIOUS
 $\langle 3 \rangle 3. \langle \rangle \in \text{Seq}(\text{Int})$
 OBVIOUS
 $\langle 3 \rangle 4. \text{QED}$
 BY $\langle 3 \rangle 2$, $\langle 3 \rangle 1$, $\langle 3 \rangle 3$, *Lemma1* DEF *Init*
 $\langle 2 \rangle 3. (pc = \text{"Done"}) \Rightarrow (n = \text{Len}(\text{seq}) + 1)$
 BY *Lemma1*, *ValAssump* DEF *Init*, *Inv*, *TypeOK*
 $\langle 2 \rangle 4. \text{QED}$
 BY $\langle 2 \rangle 1$, $\langle 2 \rangle 2$, $\langle 2 \rangle 3$ DEF *Inv*
 $\langle 1 \rangle 2. \text{Inv} \wedge [\text{Next}]_{\text{vars}} \Rightarrow \text{Inv}'$
 $\langle 2 \rangle$ SUFFICES ASSUME *Inv*,
 $[\text{Next}]_{\text{vars}}$
 PROVE *Inv'*
 OBVIOUS
 $\langle 2 \rangle 1. \text{CASE } a$
 $\langle 3 \rangle 1. \text{TypeOK}'$
 BY $\langle 2 \rangle 1$ DEF *a*, *Inv*, *TypeOK*
 $\langle 3 \rangle 2. (\text{sum} = \text{SeqSum}([i \in 1 \dots (n-1) \mapsto \text{seq}[i]]))'$
 $\langle 4 \rangle 1. \text{CASE } n > \text{Len}(\text{seq})$
 $\langle 5 \rangle \neg(n \leq \text{Len}(\text{seq}))$
 BY $\langle 4 \rangle 1$ DEF *Inv*, *TypeOK*
 $\langle 5 \rangle \text{QED}$
 BY $\langle 2 \rangle 1$, $\langle 4 \rangle 1$ DEF *a*, *Inv*, *TypeOK*
 $\langle 4 \rangle 2. \text{CASE } n \in 1 \dots \text{Len}(\text{seq})$
 $\langle 5 \rangle \text{DEFINE } \text{curseq} \triangleq [i \in 1 \dots (n-1) \mapsto \text{seq}[i]]$
 $s \triangleq \text{curseq}'$
 $\langle 5 \rangle \text{SUFFICES } \text{sum}' = \text{SeqSum}(s)$
 OBVIOUS
 $\langle 5 \rangle 1. \wedge n' - 1 = n$
 $\wedge \text{Len}(s) = n$
 $\wedge s[\text{Len}(s)] = \text{seq}[n]$
 BY $\langle 2 \rangle 1$, $\langle 4 \rangle 2$ DEF *a*, *Inv*, *TypeOK*
 $\langle 5 \rangle 2. s = [i \in 1 \dots n \mapsto \text{seq}[i]]$
 BY $\langle 5 \rangle 1$, $\langle 2 \rangle 1$ DEF *a*
 $\langle 5 \rangle 3. \text{sum}' = \text{sum} + \text{seq}[n]$
 BY $\langle 2 \rangle 1$, $\langle 4 \rangle 2$ DEF *a*
 $\langle 5 \rangle \text{HIDE DEF } s$
 $\langle 5 \rangle 4. \text{SeqSum}(s) = \text{SeqSum}([i \in 1 \dots (\text{Len}(s) - 1) \mapsto s[i]]) + s[\text{Len}(s)]$

$\langle 6 \rangle 1. \forall S, T : S \subseteq T \Rightarrow Seq(S) \subseteq Seq(T)$
 OBVIOUS
 $\langle 6 \rangle 2. seq \in Seq(Int)$
 BY $\langle 6 \rangle 1, ValAssump$ DEF $Inv, TypeOK$
 $\langle 6 \rangle 3. \forall i \in 1 \dots n : seq[i] \in Int$
 BY $\langle 6 \rangle 2, \langle 4 \rangle 2$
 $\langle 6 \rangle 4. s \in Seq(Int)$
 BY $\langle 6 \rangle 3, \langle 5 \rangle 2, \langle 4 \rangle 2$
 $\langle 6 \rangle$ QED
 BY $\langle 6 \rangle 4, \langle 5 \rangle 1, \langle 4 \rangle 2, Lemma5$ DEF $Front$
 $\langle 5 \rangle 5. curseq = [i \in 1 \dots (Len(s) - 1) \mapsto s[i]]$
 BY $\langle 5 \rangle 1, \langle 5 \rangle 2$
 $\langle 5 \rangle 6. sum = SeqSum(curseq)$
 BY $\langle 2 \rangle 1, \langle 4 \rangle 2, \langle 5 \rangle 5$ DEF $Inv, TypeOK, s$
 $\langle 5 \rangle 7.$ QED
 BY $\langle 5 \rangle 1, \langle 5 \rangle 3, \langle 5 \rangle 4, \langle 5 \rangle 5, \langle 5 \rangle 6$ DEF $Inv, TypeOK, s$
 $\langle 4 \rangle 3.$ QED
 BY $\langle 4 \rangle 1, \langle 4 \rangle 2$ DEF $Inv, TypeOK$
 $\langle 3 \rangle 3. ((pc = \text{"Done"}) \Rightarrow (n = Len(seq) + 1))'$
 BY $\langle 2 \rangle 1$ DEF $a, Inv, TypeOK$
 $\langle 3 \rangle 4.$ QED
 BY $\langle 3 \rangle 1, \langle 3 \rangle 2, \langle 3 \rangle 3$ DEF Inv
 $\langle 2 \rangle 2.$ CASE UNCHANGED $vars$
 BY $\langle 2 \rangle 2$ DEF $Inv, TypeOK, vars$
 $\langle 2 \rangle 3.$ QED
 BY $\langle 2 \rangle 1, \langle 2 \rangle 2$ DEF $Next$
 $\langle 1 \rangle 3. Inv \Rightarrow PCorrect$
 $\langle 2 \rangle$ SUFFICES ASSUME $Inv,$
 $pc = \text{"Done"}$
 PROVE $sum = SeqSum(seq)$
 BY DEF $PCorrect$
 $\langle 2 \rangle 1. seq = [i \in 1 \dots Len(seq) \mapsto seq[i]]$
 BY DEF $Inv, TypeOK$
 $\langle 2 \rangle 2.$ QED
 BY $\langle 2 \rangle 1$ DEF $Inv, TypeOK$
 $\langle 1 \rangle 4.$ QED
 BY $\langle 1 \rangle 1, \langle 1 \rangle 2, \langle 1 \rangle 3, PTL$ DEF $Spec$

Proofs of the Lemmas.

The LET definition at the heart of the definition of $SeqSum$ is a standard definition of a function on sequences by tail recursion. Theorem *TailInductiveDef* of module *SequenceTheorems* proves correctness of such a definition.

LEMMA $Lemma1_Proof \triangleq$
 $\forall s \in Seq(Int) :$

$SeqSum(s) = \text{IF } s = \langle \rangle \text{ THEN } 0 \text{ ELSE } s[1] + SeqSum(Tail(s))$
 $\langle 1 \rangle \text{ DEFINE } DefSS(ssOfTailss, ss) \triangleq ss[1] + ssOfTailss$
 $SS[ss \in Seq(Int)] \triangleq$
 $\text{IF } ss = \langle \rangle \text{ THEN } 0 \text{ ELSE } DefSS(SS[Tail(ss)], ss)$
 $\langle 1 \rangle 1. TailInductiveDefHypothesis(SS, Int, 0, DefSS)$
 $\text{BY DEF } TailInductiveDefHypothesis$
 $\langle 1 \rangle 2. TailInductiveDefConclusion(SS, Int, 0, DefSS)$
 $\text{BY } \langle 1 \rangle 1, TailInductiveDef$
 $\langle 1 \rangle 3. SS = [ss \in Seq(Int) \mapsto \text{IF } ss = \langle \rangle \text{ THEN } 0$
 $\text{ELSE } ss[1] + SS[Tail(ss)]]$
 $\text{BY } \langle 1 \rangle 2 \text{ DEF } TailInductiveDefConclusion$
 $\langle 1 \rangle \text{ QED}$
 $\text{BY } \langle 1 \rangle 3 \text{ DEF } SeqSum$

Lemmas 2 and 3 are simple properties of *Tail* and *Front* that are used in the proof of Lemma 5.

LEMMA *Lemma2* \triangleq
 $\forall S : \forall s \in Seq(S) :$
 $Len(s) > 0 \Rightarrow \wedge Tail(s) \in Seq(S)$
 $\wedge Front(s) \in Seq(S)$
 $\wedge Len(Tail(s)) = Len(s) - 1$
 $\wedge Len(Front(s)) = Len(s) - 1$
 $\langle 1 \rangle \text{ SUFFICES ASSUME NEW } S,$
 $\text{NEW } s \in Seq(S),$
 $Len(s) > 0$
 $\text{PROVE } \wedge Tail(s) \in Seq(S)$
 $\wedge Front(s) \in Seq(S)$
 $\wedge Len(Tail(s)) = Len(s) - 1$
 $\wedge Len(Front(s)) = Len(s) - 1$
 OBVIOUS
 $\langle 1 \rangle 1. Tail(s) \in Seq(S) \wedge Len(Tail(s)) = Len(s) - 1$
 OBVIOUS
 $\langle 1 \rangle 2. Front(s) \in Seq(S) \wedge Len(Front(s)) = Len(s) - 1$
 $\text{BY } FrontDef$
 $\langle 1 \rangle 3. \text{ QED}$
 $\text{BY } \langle 1 \rangle 1, \langle 1 \rangle 2$

LEMMA *Lemma3* \triangleq
 $\forall S : \forall s \in Seq(S) :$
 $(Len(s) > 1) \Rightarrow (Tail(Front(s)) = Front(Tail(s)))$
 $\langle 1 \rangle \text{ SUFFICES ASSUME NEW } S,$
 $\text{NEW } s \in Seq(S),$
 $Len(s) > 1$
 $\text{PROVE } Tail(Front(s)) = Front(Tail(s))$
 OBVIOUS

$$\langle 1 \rangle 1. \text{Tail}(\text{Front}(s)) = [i \in 1 \dots (\text{Len}(s) - 2) \mapsto s[i + 1]]$$
$$\langle 2 \rangle 1. \wedge \text{Front}(s) = [i \in 1 \dots (\text{Len}(s) - 1) \mapsto s[i]]$$
$$\wedge Len(Front(s)) = Len(s) - 1$$
$$\wedge Front(s) \in Seq(S)$$
$$\wedge Len(s) \in Nat$$
BY *FrontDef*

2. $Len(Front(s)) > 0$

BY $\langle 2 \rangle 1$

⟨2⟩3. $Front(s) \neq \langle \rangle$

BY $\langle 2 \rangle 1, \langle 2 \rangle 2$
$$\langle 2 \rangle 4. \text{Tail}(\text{Front}(s)) = [i \in 1 \dots (\text{Len}(\text{Front}(s)) - 1) \mapsto \text{Front}(s)[i + 1]]$$

⟨3⟩ DEFINE $t \triangleq Front(s)$

$$\langle 3 \rangle 1. t \in Seq(S) \wedge t \neq \langle \rangle$$
BY $\langle 2 \rangle 1, \langle 2 \rangle 3$

$\langle 3 \rangle$ $2.80T_{\text{H}}(t)TJ/F11.070T_{\text{H}}(T)77(a)TJ/F89.9626T_{\text{H}}6.6870T_{\text{H}}(t)TJ/F209.9626T_{\text{H}}8.8750T_{\text{H}}(s)TJ/F89.9626$

~~9~~ $\langle 2 \rangle 1$

4. Tail(Front(s)) = s

 $\langle 2 \rangle 1i$ $h2\rangle$

BY DEF P
 $\langle 2 \rangle 1. s = \langle \rangle$
 OBVIOUS
 $\langle 2 \rangle$ QED
 BY $\langle 2 \rangle 1$, *Lemma1*
 $\langle 1 \rangle 2.$ ASSUME NEW $N \in Nat$, $P(N)$
 PROVE $P(N + 1)$
 $\langle 2 \rangle$ SUFFICES ASSUME NEW $s \in Seq(Int)$,
 $Len(s) = (N + 1)$
 PROVE $SeqSum(s) \in Int$
 BY DEF P
 $\langle 2 \rangle 1. s \neq \langle \rangle$
 OBVIOUS
 $\langle 2 \rangle 2. SeqSum(s) = s[1] + SeqSum(Tail(s))$
 BY $\langle 2 \rangle 1$, *Lemma1*
 $\langle 2 \rangle 3. s[1] \in Int$
 BY $\langle 2 \rangle 1$
 $\langle 2 \rangle 4. \wedge Len(Tail(s)) = N$
 $\wedge Tail(s) \in Seq(Int)$
 BY $\langle 2 \rangle 2$, *Lemma2*
 $\langle 2 \rangle 5. SeqSum(Tail(s)) \in Int$
 BY $\langle 1 \rangle 2$, $\langle 2 \rangle 4$
 $\langle 2 \rangle 6.$ QED
 BY $\langle 2 \rangle 2$, $\langle 2 \rangle 3$, $\langle 2 \rangle 5$
 $\langle 1 \rangle$ HIDE DEF P
 $\langle 1 \rangle 3. \forall N \in Nat : P(N)$
 BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$, *NatInduction*
 $\langle 1 \rangle 4.$ QED
 BY $\langle 1 \rangle 3$ DEF P

LEMMA *Lemma5_Proof* \triangleq
 $\forall s \in Seq(Int) :$
 $(Len(s) > 0) \Rightarrow$
 $SeqSum(s) = SeqSum(Front(s)) + s[Len(s)]$
 $\langle 1 \rangle$ DEFINE $P(N) \triangleq \forall s \in Seq(Int) :$
 $(Len(s) = N) \Rightarrow$
 $(SeqSum(s) = \text{IF } Len(s) = 0$
 $\text{THEN } 0$
 $\text{ELSE } SeqSum(Front(s)) + s[Len(s)])$
 $\langle 1 \rangle 1. P(0)$
 $\langle 2 \rangle$ SUFFICES ASSUME NEW $s \in Seq(Int)$,
 $Len(s) = 0$
 PROVE $SeqSum(s) = \text{IF } Len(s) = 0$
 $\text{THEN } 0$


```

ELSE SeqSum(Front(s)) + s[Len(s)]

BY DEF P
⟨2⟩ QED
BY  $s = \langle \rangle$ , Lemma1
⟨1⟩2. ASSUME NEW  $N \in Nat$ ,  $P(N)$ 
PROVE  $P(N + 1)$ 
⟨2⟩ SUFFICES ASSUME NEW  $s \in Seq(Int)$ ,
       $Len(s) = (N + 1)$ 
PROVE  $SeqSum(s) = \text{IF } Len(s) = 0$ 
      THEN 0
      ELSE SeqSum(Front(s)) + s[Len(s)]

BY DEF P
⟨2⟩ SUFFICES  $SeqSum(s) = SeqSum(Front(s)) + s[Len(s)]$ 
OBVIOUS
⟨2⟩1.  $\wedge Front(s) \in Seq(Int)$ 
       $\wedge Len(Front(s)) = N$ 
BY Lemma2
⟨2⟩ DEFINE  $t \triangleq Tail(s)$ 
⟨2⟩ USE FrontDef
⟨2⟩2.  $\wedge t \in Seq(Int)$ 
       $\wedge Len(t) = N$ 
       $\wedge SeqSum(s) = s[1] + SeqSum(t)$ 
BY HeadTailProperties, Lemma1,  $s \neq \langle \rangle$ 
⟨2⟩3.CASE  $N = 0$ 
⟨3⟩ USE ⟨2⟩3
⟨3⟩ HIDE FrontDef DEF Front
⟨3⟩1.  $SeqSum(Front(s)) = 0$ 
BY Lemma1, ⟨2⟩1,  $Front(s) = \langle \rangle$ 
⟨3⟩2.  $Len(Tail(s)) = 0$ 
BY HeadTailProperties
⟨3⟩3.  $SeqSum(Tail(s)) =$ 
      IF  $Tail(s) = \langle \rangle$  THEN 0 ELSE  $Tail(s)[1] + SeqSum(Tail(Tail(s)))$ 
BY ⟨2⟩2, Lemma1
⟨3⟩4.  $SeqSum(Tail(s)) = 0$ 
BY ⟨3⟩2, ⟨2⟩2, EmptySeq,  $Tail(s) = \langle \rangle$ , ⟨3⟩3
⟨3⟩5. QED
BY ⟨2⟩2, ⟨3⟩1, ⟨3⟩4
⟨2⟩4.CASE  $N > 0$ 
⟨3⟩  $\wedge Front(s) \in Seq(Int)$ 
       $\wedge Front(t) \in Seq(Int)$ 
       $\wedge Tail(Front(s)) \in Seq(Int)$ 
⟨4⟩1.  $Front(s) \in Seq(Int)$ 
BY ⟨2⟩4, ⟨2⟩2, Lemma2
⟨4⟩2.  $Front(t) \in Seq(Int)$ 
BY ⟨2⟩4, ⟨2⟩2, Lemma2

```

$\langle 4 \rangle 3. \text{Tail}(\text{Front}(s)) \in \text{Seq}(\text{Int})$
 $\langle 5 \rangle \text{Len}(s) > 1$
 BY $\langle 2 \rangle 4$
 $\langle 5 \rangle \text{Len}(\text{Front}(s)) > 0$
 BY *Lemma2*
 $\langle 5 \rangle \text{Front}(s) \in \text{Seq}(\text{Int})$
 BY *Lemma2*
 $\langle 5 \rangle \text{Tail}(\text{Front}(s)) \in \text{Seq}(\text{Int})$
 BY *Lemma2*
 $\langle 5 \rangle \text{QED}$
 BY *Lemma2*
 $\langle 4 \rangle 4. \text{QED}$
 BY $\langle 4 \rangle 1, \langle 4 \rangle 2, \langle 4 \rangle 3$
 $\langle 3 \rangle 1. \text{SeqSum}(t) = \text{SeqSum}(\text{Front}(t)) + t[N]$
 BY $\langle 1 \rangle 2, \langle 2 \rangle 2, \langle 2 \rangle 4$
 $\langle 3 \rangle 2. \text{SeqSum}(t) = \text{SeqSum}(\text{Tail}(\text{Front}(s))) + t[N]$
 BY $\langle 3 \rangle 1, \langle 2 \rangle 4, \text{Len}(s) > 1, \text{Lemma3}$
 $\langle 3 \rangle 3. t[N] = s[N + 1]$
 BY $\langle 2 \rangle 2, \langle 2 \rangle 4$
 $\langle 3 \rangle \text{HIDE DEF Front}$
 $\langle 3 \rangle 4. \wedge \text{SeqSum}(s) \in \text{Int}$
 $\wedge \text{SeqSum}(t) \in \text{Int}$
 $\wedge \text{SeqSum}(\text{Tail}(\text{Front}(s))) \in \text{Int}$
 $\wedge t[N] \in \text{Int}$
 $\wedge s[1] \in \text{Int}$
 $\langle 4 \rangle 1. \text{SeqSum}(s) \in \text{Int}$
 BY $\langle 2 \rangle 4, \langle 2 \rangle 2, \langle 2 \rangle 1, \text{Lemma4}$
 $\langle 4 \rangle 2. \text{SeqSum}(t) \in \text{Int}$
 BY $\langle 2 \rangle 4, \langle 2 \rangle 2, \langle 2 \rangle 1, \text{Lemma4}$
 $\langle 4 \rangle 3. \text{SeqSum}(\text{Tail}(\text{Front}(s))) \in \text{Int}$
 $\langle 5 \rangle 1. \text{Len}(s) > 1$
 BY $\langle 2 \rangle 4$
 $\langle 5 \rangle 2. \text{Len}(\text{Front}(s)) > 0$
 BY $\langle 5 \rangle 1, \text{FrontDef}$ DEF Front
 $\langle 5 \rangle 3. \text{Front}(s) \neq \langle \rangle$
 BY $\langle 5 \rangle 2$
 $\langle 5 \rangle 4. \text{Tail}(\text{Front}(s)) \in \text{Seq}(\text{Int})$
 BY $\langle 5 \rangle 3$
 $\langle 5 \rangle 5. \text{QED}$
 BY $\langle 2 \rangle 4, \langle 2 \rangle 2, \langle 2 \rangle 1, \langle 5 \rangle 3, \text{Lemma4}$
 $\langle 4 \rangle 4. t[N] \in \text{Int}$
 BY $\langle 2 \rangle 4, \langle 2 \rangle 2, \langle 2 \rangle 1$
 $\langle 4 \rangle 4a. s[1] \in \text{Int}$
 BY $\langle 2 \rangle 4$
 $\langle 4 \rangle 5. \text{QED}$

BY $\langle 4 \rangle 1, \langle 4 \rangle 2, \langle 4 \rangle 3, \langle 4 \rangle 4$
 $\langle 3 \rangle 5. SeqSum(s) = s[1] + SeqSum(Tail(Front(s))) + t[N]$
 $\langle 4 \rangle 1. SeqSum(s) = s[1] + SeqSum(t)$
 BY $\langle 2 \rangle 2$
 $\langle 4 \rangle 2. QED$
 BY $\langle 4 \rangle 1, \langle 3 \rangle 2, \langle 3 \rangle 4, Lemma4, Z3$
 $\langle 3 \rangle 6. t[N] = s[N + 1]$
 BY $\langle 2 \rangle 4$
 $\langle 3 \rangle 7. s[1] = Front(s)[1]$
 BY $\langle 2 \rangle 4$ DEF *Front*
 $\langle 3 \rangle 8. SeqSum(Front(s)) = Front(s)[1] + SeqSum(Tail(Front(s)))$
 BY $\langle 2 \rangle 4, Lemma1$
 $\langle 3 \rangle 9. QED$
 BY $\langle 3 \rangle 5, \langle 3 \rangle 6, \langle 3 \rangle 7, \langle 3 \rangle 8$
 $\langle 2 \rangle 5. QED$
 BY $\langle 2 \rangle 3, \langle 2 \rangle 4$
 $\langle 1 \rangle 3. \forall N \in Nat : P(N)$
 BY $\langle 1 \rangle 1, \langle 1 \rangle 2, NatInduction$
 $\langle 1 \rangle 4. QED$
 BY $\langle 1 \rangle 3$

\ * Modification History
 \ * Last modified *Fri* May 03 16:40:42 *PDT* 2019 by *lamport*
 \ * Created *Fri* Apr 19 14:13:06 *PDT* 2019 by *lamport*