Quiz Section 5: Reasoning

The problems that follow make use of the following inductive type, representing lists of integers

$$\textbf{type}\ \text{List} := \quad \text{nil} \quad | \quad \text{cons}(\text{hd} : \mathbb{Z},\ \text{tl} : \text{List})$$

Below, we will also use the function sum, which returns the sum of the integers in the list:

$$\text{sum} : \text{List} \to \mathbb{Z}$$

$$\begin{aligned} \text{sum}(\text{nil}) &:= 0 \\ \text{sum}(a :: L) &:= a + \text{sum}(L) \end{aligned}$$

the function twice, which doubles each number in the list:

$$\text{twice} : \text{List} \to \text{List}$$

$$\begin{aligned} \text{twice}(\text{nil}) &:= \text{nil} \\ \text{twice}(\text{a} :: \text{L}) &:= \text{2a} :: \text{twice}(\text{L}) \end{aligned}$$

the functions twice-evens and twice-odds, which double the integers at even and odd indexes in the list:

$$\text{twice-evens} : \text{List} \to \text{List}$$

$$\begin{aligned} \text{twice-evens}(\text{nil}) &:= \text{nil} \\ \text{twice-evens}(a :: L) &:= 2a :: \text{twice-odds}(L) \end{aligned}$$

$$\text{twice-odds} : \text{List} \to \text{List}$$

$$\begin{aligned} \text{twice-odds}(\text{nil}) &:= \text{nil} \\ \text{twice-odds}(a :: L) &:= a :: \text{twice-evens}(L) \end{aligned}$$

and the function swap, which swaps adjacent integers in the list:

$$\text{swap} : \text{List} \to \text{List}$$

$$\begin{aligned} \text{swap}(\text{nil}) &:= \text{nil} \\ \text{swap}(a :: nil) &:= a :: \text{nil} \\ \text{swap}(a :: b :: L) &:= b :: a :: \text{swap}(L) \end{aligned}$$

and the function len, which finds the length of the list:

$$\text{len} : \text{List} \to \mathbb{Z}$$

$$\begin{aligned} \text{len}(\text{nil}) &:= 0 \\ \text{len}(a :: L) &:= 1 + \text{len}(L) \end{aligned}$$

Task 1 – Twice Things Up

You see the following snippet in some TypeScript code. It uses `cons` and `nil`, which are TypeScript implementations of "cons" and "nil", and also `equal`, which is a TypeScript implementation of "=" on lists.

```
if (equal(L, cons(1, cons(2, nil)))) {
  const R = cons(2, cons(4, nil));  // = twice(L)
  return cons(0, R);                // = twice(cons(0, L))
}
```

The comments show the definition of what *should* be returned (the specification), but the code is *not* a direct translation of those. Below, we will use reasoning to prove that the code is correct.

(a) Using the fact that $L = 1::2::\text{nil}$, prove by calculation that $\text{twice}(L) = R$, where R is the constant list defined in the code. I.e., prove that

$$\text{twice}(L) = 2::4::\text{nil}$$

(b) Using the facts that $L = 1::2::\text{nil}$ and $R = 2::4::\text{nil}$, prove by calculation that the code above returns the correct value, i.e., prove that

$$\text{twice}(0 :: L) = 0 :: R$$

Feel free to cite part (a) in your calculation.

Task 2 – It's Raining Len

You see the following snippet in some TypeScript code. It uses `twice_evens`, which is a TypeScript implementation of twice-evens from the previous problem, as well as `len` from before.

```
return 2 + len(twice_evens(L));  // = len(twice-evens(cons(3, cons(4, L))))
```

The comment shows the definition of what should be returned (the specification), but the code is not a direct translation of that. Below, we will use reasoning to prove that the code is correct.

(a) Let a and b be any integers. Prove by calculation that

$$\text{len}(\text{twice-evens}(a :: b :: L)) = 2 + \text{len}(\text{twice-evens}(L))$$

(b) Explain why the calculation from part (a) shows that the code is correct according to the specification (written in the comment).

Task 3 – Swapaholic

Prove by cases that swap$(a :: L) \neq$ nil for any integer $a : \mathbb{Z}$ and list L.

Task 4 – Here Comes the Sum

You see following snippet in some TypeScript code:

```
const s = sum(L);
...
return 2 * s;  // = sum(twice(L))
```

This code claims to calculate the answer $\mathsf{sum}(\mathsf{twice}(L))$, but it actually returns $2\,\mathsf{sum}(L)$. Prove this code is correct by showing that $\mathsf{sum}(\mathsf{twice}(S)) = 2\,\mathsf{sum}(S)$ holds for any list S by structural induction.

Task 5 – Can You Sum a Few Bars?

Prove that

$$\mathsf{sum}(\mathsf{twice\text{-}evens}(L)) + \mathsf{sum}(\mathsf{twice\text{-}odds}(L)) = 3\,\mathsf{sum}(L)$$

holds for any list S by structural induction.