Quiz Section 5: Reasoning – Solutions

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

$$\textbf{type } \mathsf{List} := \ \mathsf{nil} \ \mid \ \mathsf{cons}(\mathsf{hd} : \mathbb{Z},\ \mathsf{tl} : \mathsf{List})$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned} &\text{twice}(L) \\ &\quad = \text{twice}(1{::}2{::}\text{nil}) && \text{Def of } L \\ &\quad = 2{::}\text{twice}(2{::}\text{nil}) && \text{Def of twice} \\ &\quad = 2{::}4{::}\text{twice}(\text{nil}) && \text{Def of twice} \\ &\quad = 2{::}4{::}\text{nil} && \text{Def of twice} \end{aligned}$$

(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.

$$\begin{aligned} &\text{twice}(0 :: L) \\ &\quad = 0 :: \text{twice}(L) && \text{Def of twice} \\ &\quad = 0 :: 2 :: 4 :: \text{nil} && \text{Part (a)} \\ &\quad = 0 :: R && \text{Def of } R \end{aligned}$$

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

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

$$\begin{aligned}
&\text{len}(\text{twice-evens}(a :: b :: L)) \\
&\quad = \text{len}(2a :: \text{twice-odds}(b :: L)) && \text{Def of twice-evens} \\
&\quad = \text{len}(2a :: b :: \text{twice-evens}(L)) && \text{Def of twice-odds} \\
&\quad = 1 + \text{len}(b :: \text{twice-evens}(L)) && \text{Def of len} \\
&\quad = 1 + 1 + \text{len}(\text{twice-evens}(L)) && \text{Def of len} \\
&\quad = 2 + \text{len}(\text{twice-evens}(L))
\end{aligned}$$

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

Applying part (a) with $a = 3$ and $b = 4$ gives us a proof that

$$\text{len}(\text{twice-evens}(3 :: 4 :: L)) = 2 + \text{len}(\text{twice-evens}(L))$$

which says that the code is correct.

Task 3 – Swapaholic

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

Let a be any integer and L be any list. We argue by cases on L.

First, suppose that $L = \text{nil}$. Then, we can see that

$$\begin{aligned} &\text{swap}(a :: L) \\ &\quad = \text{swap}(a :: \text{nil}) && \text{Def of } L \\ &\quad = a :: \text{nil} && \text{Def of swap} \\ &\quad \neq \text{nil} \end{aligned}$$

Next, suppose that $L \neq \text{nil}$. That means that $L = b :: R$ for some $b : \mathbb{Z}$ and $R : \text{List}$. Thus, we have

$$\begin{aligned} &\text{swap}(a :: L) \\ &\quad = \text{swap}(a :: b :: R) && \text{Def of } L \\ &\quad = b :: a :: \text{swap}(R) && \text{Def of swap} \\ &\quad \neq \text{nil} \end{aligned}$$

We have proven the claim in both cases. Since those cases are exhaustive, we have proven it in general.

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

Define $P(S)$ to be the claim that $\text{sum}(\text{twice}(S)) = 2\,\text{sum}(S)$. We will prove the claim by structural induction.

Base Case (nil). We can calculate

$$\begin{aligned}
&\text{sum}(\text{twice}(\text{nil})) \\
&\quad = \text{sum}(\text{nil}) && \text{Def of twice} \\
&\quad = 0 && \text{Def of sum} \\
&\quad = 2 \cdot 0 \\
&\quad = 2 \cdot \text{sum}(\text{nil}) && \text{Def of sum}
\end{aligned}$$

Inductive Hypothesis. Suppose that $P(L)$ holds for a list L. I.e., suppose that $\text{sum}(\text{twice}(L)) = 2\,\text{sum}(L)$.

Inductive Step. We need to show $P(a :: L)$ for any integer $a : \mathbb{Z}$.

Let a be any integer. Then, we can calculate

$$\begin{aligned}
&\text{sum}(\text{twice}(a :: L)) \\
&\quad = \text{sum}(2a :: \text{twice}(L)) && \text{Def of twice} \\
&\quad = 2a + \text{sum}(\text{twice}(L)) && \text{Def of sum} \\
&\quad = 2a + 2\,\text{sum}(L) && \text{Inductive Hypothesis} \\
&\quad = 2(a + \text{sum}(L)) \\
&\quad = 2\,\text{sum}(a :: L) && \text{Def of sum}
\end{aligned}$$

Conclusion. $P(S)$ 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.

Define $P(S)$ to be the claim that $\mathsf{sum}(\mathsf{twice\text{-}evens}(S)) + \mathsf{sum}(\mathsf{twice\text{-}odds}(S)) = 3\,\mathsf{sum}(S)$. We will prove the claim by structural induction.

Base Case (nil). We can calculate

$$\begin{aligned}
&\mathsf{sum}(\mathsf{twice\text{-}evens}(\mathsf{nil})) + \mathsf{sum}(\mathsf{twice\text{-}odds}(\mathsf{nil})) && \\
&\quad = \mathsf{sum}(\mathsf{nil}) + \mathsf{sum}(\mathsf{twice\text{-}odds}(\mathsf{nil})) && \text{Def of twice-evens} \\
&\quad = \mathsf{sum}(\mathsf{nil}) + \mathsf{sum}(\mathsf{nil}) && \text{Def of twice-odds} \\
&\quad = \mathsf{sum}(\mathsf{nil}) + 0 && \text{Def of sum} \\
&\quad = 0 && \text{Def of sum} \\
&\quad = 3 \cdot 0 && \\
&\quad = 3\,\mathsf{sum}(\mathsf{nil}) && \text{Def of sum}
\end{aligned}$$

Inductive Hypothesis. Suppose that $P(L)$ holds for a list L. I.e., suppose that $\mathsf{sum}(\mathsf{twice\text{-}evens}(L)) + \mathsf{sum}(\mathsf{twice\text{-}odds}(L)) = 3\,\mathsf{sum}(L)$

Inductive Step. We need to show $P(a :: L)$ for any integer $a : \mathbb{Z}$.

Let a be any integer. Then, we can calculate

$$\begin{aligned}
&\mathsf{sum}(\mathsf{twice\text{-}evens}(a :: L)) + \mathsf{sum}(\mathsf{twice\text{-}odds}(a :: L)) && \\
&\quad = \mathsf{sum}(2a :: \mathsf{twice\text{-}odds}(L)) + \mathsf{sum}(\mathsf{twice\text{-}odds}(a :: L)) && \text{Def of twice-evens} \\
&\quad = 2a + \mathsf{sum}(\mathsf{twice\text{-}odds}(L)) + \mathsf{sum}(\mathsf{twice\text{-}odds}(a :: L)) && \text{Def of sum} \\
&\quad = 2a + \mathsf{sum}(\mathsf{twice\text{-}odds}(L)) + \mathsf{sum}(a :: \mathsf{twice\text{-}evens}(L)) && \text{Def of twice-odds} \\
&\quad = 2a + \mathsf{sum}(\mathsf{twice\text{-}odds}(L)) + a + \mathsf{sum}(\mathsf{twice\text{-}evens}(L)) && \text{Def of sum} \\
&\quad = 3a + \mathsf{sum}(\mathsf{twice\text{-}evens}(L)) + \mathsf{sum}(\mathsf{twice\text{-}odds}(L)) && \\
&\quad = 3a + 3\,\mathsf{sum}(L) && \text{Inductive Hypothesis} \\
&\quad = 3(a + \mathsf{sum}(L)) && \\
&\quad = 3\,\mathsf{sum}(a :: L) && \text{Def of sum}
\end{aligned}$$

Conclusion. $P(S)$ holds for any list S by structural induction.