CSE 331: Software Design & Implementation Fall 2025

Homework 2

Due: Friday, October 10th, 6pm

Task 1 – Get'cha Head In the Game [14 pts]

The functions head : (List) $\to \mathbb{Z}$ and tail : (List) $\to$ List are defined only on non-nil lists, with the following definitions:

$$\text{head}(x :: L) = x$$
$$\text{tail}(x :: L) = L$$

You will use these functions in the problem below.

Let x and y be integers and L a list. Complete each of the following proofs **by calculation**.

Include an explanation on each step where a given fact is used. You can skip such an explanation only if the claim written in that step is itself, *literally*, a known fact. It is also fine to cite a fact that is equivalent (via simple algebra) to a known fact.

Include an explanation on each step where a definition is used. You do not need to include an explanation on lines that simply rewrite an expression in alternative but equivalent notation. For example, no explanation is needed when replacing "$[x]$" by "$x ::$ nil", which means the same thing.

a) $\text{head}(\text{nil} +\!\!+ [x]) = x$.

b) $\text{tail}(\text{nil} +\!\!+ [x]) = \text{nil}$.

c) $\text{head}((y :: L) +\!\!+ [x]) = y$.

d) $\text{tail}((y :: L) +\!\!+ [x]) = L +\!\!+ [x]$.

e) $\text{len}(\text{tail}([x, y] +\!\!+ L)) = 1 + \text{len}(L)$

Task 2 – Nil or Never [6 pts]

Let x be an integer and L a list. Complete the following proof **by cases**.

Your individual calculations should include explanations as in the previous problems. When citing a function definition that uses a side condition, your explanation must not only say which function's definition is being used, but also what side condition is known to hold so that the reader can see what line of the definition you are referring to.

Prove that $L +\!\!+ [x] \neq \text{nil}$.

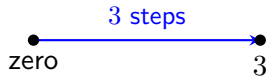
Note: in order to show that a list is not nil, you must show that it is a "cons", i.e., you must write it in the form "$(\ldots) :: (\ldots)$", where each "$\ldots$" can be filled in with any expression.

Task 3 – Succs to Succ [10 pts]

Recall the definition of the natural numbers $\mathbb{N}$:

$$\textbf{type } \mathbb{N} = \text{zero} \mid \text{succ}(\mathbb{N})$$

We can think of each natural number as counting out steps along the x axis. For example, the natural number succ(succ(succ(zero))) is taking three steps along the x axis, so it represents the number 3.



We can then define addition of natural numbers, add : $(\mathbb{N}, \mathbb{N}) \to \mathbb{N}$, by thinking of add$(n, m)$ as taking m steps and then n steps along the x axis. For example, adding 3 to 5 looks like this:

We can accomplish this with the following definition.

$$\begin{aligned} \text{add}(\text{zero}, m) &= m \\ \text{add}(\text{succ}(n), m) &= \text{succ}(\text{add}(n, m)) \end{aligned}$$

The first line of the definition says that m steps and then zero more steps is just m steps. The second line says that taking m steps and then $n + 1$ steps is the same as taking m steps and then n steps and then one more step. The resulting definition of add is structurally recursive on its first parameter.

Prove that add(succ$(n), m)$ = add$(n,$ succ$(m))$ holds for all n and m by structural induction on n.

Task 4 – Optional: If That Wasn't Add Enough [10 pts]

Recall the definitions of $\mathbb{N}$ and add : $(\mathbb{N}, \mathbb{N}) \to \mathbb{N}$ from Task 3.

a) Prove that add$(n, \text{zero}) = n$ by structural induction.

b) Prove that add$(n, m) =$ add(m, n) holds for all n and m by structural induction on n.

Task 5 – Extra Credit: If You Catch My Shift [14 pts]

The functions shift-left, shift-right : (List) $\to$ List rotate the elements in a list forward by one index and backward by one index, respectively, wrapping around at the beginning and end. They are defined formally as follows:

$$\begin{aligned} \text{shift-left}(\text{nil}) &= \text{nil} \\ \text{shift-left}(x :: L) &= L +\!\!+ [x] \end{aligned}$$

$$\begin{aligned} \text{shift-right}(\text{nil}) &= \text{nil} \\ \text{shift-right}(x :: L) &= \text{last}(x :: L) :: \text{init}(x :: L) \end{aligned}$$

You will use these functions in the problem below.

a) Let x be an integer. Prove $\text{last}(L +\!\!+ [x]) = x$ by structural induction on L.

b) Let x be an integer. Prove $\text{init}(L +\!\!+ [x]) = L$ by structural induction on L.

c) Prove that $\text{shift-right}(\text{shift-left}(L)) = L$ holds by cases.