CIS505/705 Exercise  Sample solution

Here is the list-induction rule:

\[ \text{fun } f \text{ nil } = b \]
\[ \text{fun } f \text{ (x::xs) } = e(f(x)) \]

\[ \text{SUM}(ms, n) = (i) \text{ ms } = [\text{n0, n1, \ldots, nk}] \]
\[ \quad \text{and} \]
\[ \quad (ii) \text{ n } = 0 + n_0 + n_1 + \ldots + nk \]

Finish the proof that function sumit has the SUM correctness property:

1. \text{0 : SUM([], 0)}
   because (i) [] has no ints in it,
   (ii) 0 is the sum of zero ints.
   So, property SUM([], 0) holds true.
   So, 0 has "data type" SUM([], 0)

2. \text{(x::xs) : int list}
   assumption --- say the list is nonempty

3. \text{(sumit xs) : SUM(xs, sumit xs)}
   assumption --- say the recursive call,
   (sumit xs), returns
   the correct answer for arg xs

4. \text{x + (sumit xs) : SUM(x::xs, x+(sumit xs))}
   because, Line 3 says:
   (i) \text{xs } = [\text{n0, n1, \ldots, nk}],
   (ii) \text{(sumit xs) } = 0 + n_0 + n_1 + \ldots + nk
   So, we have that
   (i) \text{x::xs } = [\text{x, n0, n1, \ldots, nk}]
   (ii) \text{x + (sumit xs) } = x + 0 + n_0 + n_1 + \ldots + nk
   \text{SUM(x::xs, x+(sumit xs)) } = 0 + x + n_0 + n_1 + \ldots + nk
   So, \text{SUM(x::xs, x+(sumit xs)) holds true.}
   So, \text{x + (sumit xs) has "data type" SUM(x::xs, x+(sumit xs))}

5. \text{fun sumit nil } = 0
   \text{sumit (x::xs) } = x + (sumit xs)
   : Forall arg Exists ans : SUM(arg, ans)
   by list ind 1, 2-4