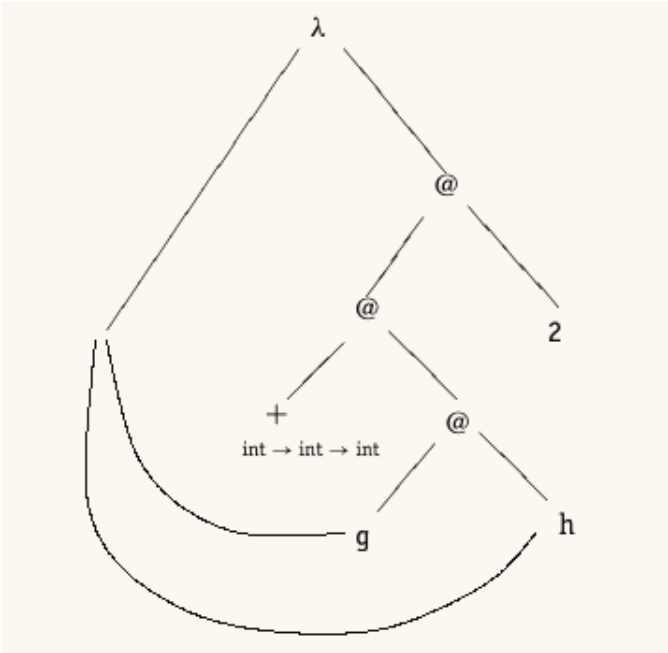
1 Type inference

Derive the types for the following expressions by:

- *Showing all the constraints that are generated*
- *Solving the constraints*

1. `fun f (g,h) = g (h 0)`
2. `fun apply (f,x) = f x`
3. ```
   fun reverse nil       = nil
     |   reverse (x::xs)   = reverse xs
   ```
4. `fun ff f x y = if (f x y) then (f 3 y) else (f x "zero")`
5. `fun gg f x y = if (f x y) then (f 3 y) else (f y "zero")`
6. `fun hh f x y = if (f x y) then (f x y) else (f x "zero")`
7. ```
   fun sort(less, nil) = nil
       | sort(less, a :: l) =
         let fun insert(a, nil) = a :: nil
               | insert(a, b :: l) = if less(a,b) then a :: (b :: l)
                                     else b :: insert(a, l)
         in
             insert(a, sort(less, l))
         end
   ```
8. ```
   fun append(nil, l)    = l
       | append(x :: l, m) = append(l, m)
   ```
   ```

9. `fun f(g,h) = g h + 2`



10. `fun f g = g(g) + 2`