

# Recursion

98-317: Hype for Types

Due: 16 October 2018 at 6:30 PM

## Introduction

This week we explored recursion and how a recursive function can be implemented as the fixed point of a non-recursive function. We also saw how the same approach can be used to implement mutual recursion, and how to use a technique called open recursion to incrementally modify mutually recursive functions. This homework has three tasks, all of which are required.

**Turning in the Homework** Submit `06-recursion.sml` to Autolab.

## Cool Fixed Points

In class we implemented a cool fixed point function which produces a memoized recursive function. There are many other ways in which fixed point functions can be cool. One of them is by limiting the depth of recursion, which can prevent infinite looping.

**Required Task** Implement the function

```
fix_limited : int -> (('a -> 'b) -> ('a -> 'b)) -> ('a -> 'b)
```

such that `fix_limited n f` (where `n` is nonnegative) produces a function similar to the fixed point of `f`, except that it is not allowed to recur deeper than `n` levels. When it tries to recur deeper than `n` levels, it should raise `RecursionDepthExceeded`.

Note that being prohibited from recurring deeper than `n` levels is different than being prohibited from making `n` recursive calls.

## Fixed Point for Mutual Recursion

Recall the `tree_counter` type from class:

```
type tree_counter = {  
  counter : tree -> int,  
  leaf_counter : char -> int,  
  branch_counter : tree * tree -> int  
}
```

**Required Task** Implement the function

```
fix_counter : (tree_counter -> tree_counter) -> tree_counter
```

such that `fix_counter f` evaluates to a fixed point of `f`.

## Open Recursion

In class we implemented a function to count the leaves of a tree by only overriding one field of the `pre_super` function.

**Required Task** Using this philosophy, implement

```
pre_left_spine_length : tree_counter -> tree_counter
```

by only overriding one field of the `pre_super` function.

For any tree `T`, `#counter (fix_counter pre_left_spine_length) T` should evaluate to the number of edges in the tree's left spine.