

Homework 7

Homework Due: Monday, March 15 at 5 pm.

The second part of the course is to prove the correctness of OCaml programs by induction. The focus of Homework 7 will be on the binary trees we have seen in Homework 4.

Prepare Your Homework 7 Directory

```
cd ~/csci2041
./setup hw7
cd hw7
```

Problem 1 (30 points)

```
type 'a tree = Leaf | Branch of 'a tree * 'a * 'a tree

let rec flip_tree =
  fun t ->
  match t with
  | Leaf -> Leaf
  | Branch (left, root, right) ->
    Branch (flip_tree right, root, flip_tree left)
```

Prove the following theorem by induction. Use the lecture notes as a style guide.

Theorem 1. *For any type `a` and any value `t` of type `a tree`, we have*

$$\text{flip_tree (flip_tree } t) = t$$

Problem 2 (30 points)

```
let rec sum_tree =
  fun t ->
  match t with
  | Leaf -> 0
  | Branch (left, root, right) ->
    sum_tree left + root + sum_tree right
```

Prove the following theorem by induction. Use the lecture notes as a style guide.

Theorem 2. *For any value `t` of type `int tree`, we have*

$$\text{sum_tree (flip_tree } t) = \text{sum_tree } t$$

Bonus Problem (10 points)

```
let rec length l =
  match l with
  | [] -> 0
  | h :: t -> 1 + length t

let rec size_tree =
  fun t ->
  match t with
  | Leaf -> 0
  | Branch (left, _, right) ->
    size_tree left + 1 + size_tree right

let rec list_of_tree_helper =
  fun t l ->
  match t with
  | Leaf -> l
  | Branch (left, root, right) ->
    list_of_tree_helper left (root :: list_of_tree_helper right l)

let list_of_tree t = list_of_tree_helper t []
```

The main function `list_of_tree t` returns a list containing the elements inside the tree `t` in the infix order (for each node, visit its left subtree, root, and right subtree). The helper function `list_of_tree_helper t l` returns a list containing the elements in `t` in the same order, concatenated with the list `l`. For example,

```
list_of_tree_helper (Branch (Leaf, 1, Branch (Leaf, 2, Leaf))) [3; 4] = [1; 2; 3; 4]
```

Understanding how these two functions work for *concrete* inputs (with no variables) is not part of the problem. Please *immediately* ask a TA for help if you are not sure about their outputs for concrete inputs. On the other hand, how the functions work for *abstract* inputs (with variables) such as `Branch (left, root, right)` is part of the problem and you should make some efforts before asking for help. (Please do ask for help!)

Your task is to prove the following theorem:

Theorem 3. *For any type `a` and any value `t` of type `a tree`, we have*

```
length (list_of_tree t) = size_tree t
```

Remark 4. *You probably want a lemma about `list_of_tree_helper` that involves an equation like this:*

```
length (list_of_tree_helper t l) = size_tree t + length l
```

You probably do not need induction for the main theorem. Do not blindly use induction when it is not needed—though you will still earn full points with unnecessary induction if your proof is mathematically correct.

If you choose to prove lemmas and use them in the main proof, please pay attention to these two points:

- You have to carefully state the complete lemma with all the universal quantifiers, not just the equation itself.
- You have to clearly mark their use in the main proof. If in doubts, please review the rubric for Homework 6.

Submission

You need to typeset or scan your homework as one single file `hw7.pdf` within the directory `hw7`, and submit it to GitHub as in Homework 6. It is encouraged to use professional typesetting software such as \LaTeX and you can reuse the sources of handouts and lecture notes for your work. Otherwise, your hand-writing must be clear and

legible, and it must be scanned and consolidated as one single file `hw7.pdf`. Submissions consisting of multiple image files are not acceptable.

If you plan to hand-write your solutions, please practice scanning your documents *now*, and reserve enough time before the homework deadline for scanning your work. Claiming that you have done the homework before the deadline but somehow cannot scan or typeset your solution will not grant you a special deadline extension. You need to use late days to cover such mistakes.

Grading

We will use exactly the same rubric as the one for Homework 6.