I2I-2 Functional Verification Example Excercises

Dimitri Tabatadze

July 2, 2023

1 Dacentrili

The followign OCaml functions are given:

```
let rec damcentravi a lst = match lst with
  | [] -> a
  | h::t -> damcentravi (h::(List.rev a)) t

let rec get_nth n lst = match lst with
  | [] -> 0
  | h::t -> if n = 0 then h else get_nth (n-1) t

let pick_middle lst = get_nth ((List.length lst) / 2) lst;;
```

Prove that the equality

```
a = pick_middle (damcentravi [] (a::b))
```

holds.

2 Even list

The followign OCaml functions are given:

```
let rec even_list a lst = match lst with
  | h::_::t -> even_list (h::a) t
  | [h] -> even_list (h::a) []
  | _ -> a

let rec deal i a lst = match lst with
  | h::t -> if i mod 2 = 0 then
      deal (i+1) (h::a) t
    else
      deal (i+1) a t
  | _ -> a
```

Prove that the equality

```
deal 0 [] l = even_list [] l
```

holds.

3 Bigotree

The following OCaml functions are given:

```
type node = Empty | Inner of node * int * node

let rec insert_in_tree v t = match t with
  | Empty -> Inner (Empty , v, Empty)
  | Inner (l, u, r) -> if v > u then
      Inner (l, u, insert_in_tree v r)
    else
      Inner (insert_in_tree v l, u, r)

let rec to_tree a lst = match lst with
  | [] -> a
  | h::t -> insert_in_tree h (to_tree a t)

let rec to_list t = match t with
  | Empty -> []
  | Inner (l, v, r) -> to_list l @ [v] @ to_list r

let rec insert n lst = match lst with
  | [] -> [n]
  | h::t -> if n > h then
      h::(insert n t)
    else
      n::h::t

let rec sort lst = match lst with
  | [] -> []
  | h::t -> insert h (sort t)
```

Prove that the equality

```
to_list (to_tree Empty a) = sort a
```

holds.