

Homework 6

Fletcher Gornick

March 8, 2021

Problem 1

```
let rec append l1 l2 =  
  match l1 with  
  | [] -> l2  
  | h :: t -> h :: append t l2
```

Theorem 1. For any type a and any value l of type a list, we have

$$\text{append } l \ [] = l$$

Domain All values of type a list.

Property $P(l)$: $\text{append } l \ [] = l$.

Inductive order $R(l1, l2)$: $l1$ is a proper suffix of $l2$.

1. $l = []$.

Left-hand side: $\text{append } [] \ [] = []$.

Right-hand side: $[]$.

Both sides agree.

2. $l = h :: t$.

Left-hand side:

$$\text{append } (h :: t) \ [] = h :: (\text{append } t \ [])$$

Right-hand side:

$$l = h :: t$$

It is thus sufficient to show that $\text{append } t \ [] = t$, which is exactly $P(t)$. $P(t)$ is implied by the inductive hypothesis because t is a proper suffix of l . That is, $R(t, l)$.

Problem 2 (30 points)

```
let rec sum l =  
  match l with  
  | [] -> 0  
  | h :: t -> h + sum t  
  
let rec append l1 l2 =  
  match l1 with  
  | [] -> l2  
  | h :: t -> h :: append t l2
```

Theorem 2. For any two values `l1` and `l2` of type `int list`, we have

$$\text{sum (append l1 l2)} = \text{sum l1} + \text{sum l2}$$

Domain All values of type `a list`.

Property $P(l1)$: for any value `l2` of type `a list`, $\text{sum (append l1 l2)} = \text{sum l1} + \text{sum l2}$.

Inductive order $R(l1, l2)$: `l1` is a proper suffix of `l2`.

1. `l1 = []`.

Left-hand side: $\text{sum (append [] l2)} = \text{sum l2}$.

Right-hand side: $\text{sum []} + \text{sum l2} = 0 + \text{sum l2} = \text{sum l2}$.

Both sides agree.

2. `l1 = h :: t`.

Left-hand side:

$$\begin{aligned} & \text{sum (append (h :: t) l2)} \\ &= \text{sum (h :: append t l2)} \\ &= h + \text{sum (append t l2)} \end{aligned}$$

Right-hand side:

$$\begin{aligned} & \text{sum (h :: t)} + \text{sum l2} \\ &= (h + \text{sum t}) + \text{sum l2} \\ &= h + (\text{sum t} + \text{sum l2}) \end{aligned}$$

It is thus sufficient to show that $\text{sum (append t l2)} = \text{sum t} + \text{sum l2}$, which is exactly $P(t)$. $P(t)$ is implied by the inductive hypothesis because `t` is a proper suffix of `l1`. That is, $R(t, l1)$.

Bonus Problem (10 points)

```
let rec filter p l =  
  match l with  
  | [] -> []  
  | h :: t -> if p h then h :: filter p t else filter p t
```

Theorem 3. For any type `a` and any value `p` of type `a -> bool`, any value `l` of type `a list`, we have

$$\text{filter p (filter p l)} = \text{filter p l}$$

Domain All values of type `a list`.

Property $P(l)$: $\text{filter p (filter p l)} = \text{filter p l}$.

Inductive order $R(l1, l2)$: `l1` is a proper suffix of `l2`.

1. `l = []`.

$$\text{filter p (filter p [])} = \text{filter p []}.$$

2. `l = h :: t`.

$$\begin{aligned} & \text{filter p (filter p (h :: t))} \\ &= \text{filter p (if p h then h :: filter p t else filter p t)} \end{aligned}$$

The result critically depends on the value of $p\ h$, which is of type `bool`. We can thus proceed by case analysis.

Domain All values of type `bool`. That is, `true` and `false`.

Property $Q(b)$: If $p\ h = b$, then $\text{filter } p\ (\text{filter } p\ (h :: t)) = \text{filter } p\ (h :: t)$

(a) $p\ h = b = \text{true}$.

```
filter p (filter p (h :: t))
= filter p (if p h then h :: filter p t else filter p t)
= filter p (if true then h :: filter p t else filter p t)
= filter p (h :: filter p t)
= if p h then h :: filter p (filter p t) else filter p (filter p t)
= if true then h :: filter p (filter p t) else filter p (filter p t)
= h :: filter p (filter p t)
```

(b) $p\ h = b = \text{false}$.

```
filter p (filter p (h :: t))
= filter p (if p h then h :: filter p t else filter p t)
= filter p (if false then h :: filter p t else filter p t)
= filter p (filter p t)
```

In both cases, it is sufficient to show that $\text{filter } p\ (\text{filter } p\ t) = \text{filter } p\ t$, which is exactly $P(t)$. $P(t)$ is implied by the inductive hypothesis because t is a proper suffix of l . That is, $R(t, l)$.