Lecture 6: Lists

+ List Induction

A boolean is either

- true
- false

→ and that's it!

A nat is either

- 0, or
- 1+k, where k nat

→ and that's it!

0 ✓

1 = 1+0 ✓

2 = 1+1+0 ✓

3 = ...

int_list
↳ space

A $\boxed{\text{list of numbers}}$ is

either

- [ ], or "nil" "empty"

- x :: xs "cons"

  where x : int

  xs is a list of numbers

→ and that's it!

$[\,]$: int list

$\underbrace{1}_{:\text{int}} :: \underbrace{[\,]}_{:\text{int list}}$: int list

$2 :: (1 :: [\,])$: int list

$3 :: (2 :: (1 :: [\,]))$: int list

⋮

nice notation

[1]

[2, 1]

[3, 2, 1]

fun f (l : int list) : $\underline{T}$ =

case l of

[] => $\boxed{\underline{\quad\quad}}$: T

| x :: xs => $\boxed{\text{use x, xs, } \underline{f(xs)} : T}$: T

x : int

xs : int list

recursive call

new

bound variables

that stand for the first/rest

head/tail

To step

1. case [] of [] => e_0 | x::xs => e_1

$\longmapsto e_0$

2. case $\overbrace{(v::vs)}^{\text{value}}$ of [] => e_0

| x::xs => e_1

$\longmapsto e_1$ with v plugged in for x

vs plugged in for xs

3. otherwise step the thing you're casing on

(* Goal: compute the length of a list

E.g.

length (2 :: (1 :: [])) = 2

length (3 :: (1 :: [])) = 2

length [] = 0

length (4 :: (2 :: (1 :: []))) = 3

*)

fun length(l : int list) : int =
case l of
[ ] => $\underline{\quad 0 \quad}$

| x::xs => 1 + length(xs)
[$\underline{\qquad\qquad\qquad}$

can use x and xs and length(xs)

$$\text{length } (4 :: (2 :: (1 :: [\,])))$$

$$\longmapsto \text{case } 4 :: (2 :: (1 :: [\,])) \text{ of}$$
$$[\,] \Rightarrow 0$$
$$\mid x :: xs \Rightarrow 1 + \text{length}(xs)$$

$$\longmapsto 1 + \text{length } (\underline{2 :: (1 :: [\,])})$$

$$\longmapsto 1 + (\text{case } 2 :: 1 :: [\,] \text{ of}$$
$$[\,] \Rightarrow 0$$
$$\mid \underline{x :: xs} \Rightarrow 1 + \text{length } xs)$$

$$\longmapsto 1 + (1 + \text{length } (\underline{(1 :: [\,])}))$$

$$\longmapsto 1 + (1 + \text{case } 1 :: [\,] \text{ of } [\,] \Rightarrow 0$$
$$\mid x :: xs \Rightarrow 1 + \text{length } xs)$$

$$\longmapsto 1 + (1 + (1 + \text{length } [\,]))$$

$$\longmapsto 1 + (1 + (1 + \text{case } \underline{[\,]} \text{ of}$$
$$\underline{[\,]} \Rightarrow 0$$
$$\mid \underline{x :: xs} \Rightarrow 1 + \text{length}(xs)))$$

$$\longmapsto 1 + (1 + (1 + 0))$$

$$\longmapsto^* 3$$

(* Goal: add up all numbers
in a list *)

(* E.g.
sum (5 :: (1 :: (2 :: []))) = 8
sum (1 :: (2 :: [])) = 3
*)

```
fun sum(l: int list): int =
  case l of
    [] => 0
  | f :: r => f + sum(r)
```

names of the variables can change

but do it consistently

$$\mathrm{sum}(\underbrace{[5,1,2]}_{5::[1,2]})$$

$$\mapsto \mathrm{case}\ 5::[1,2]\ \mathrm{of}\ [\,] \Rightarrow 0$$
$$\mid f::r \Rightarrow f + \mathrm{sum}(r)$$

$$\mapsto 5 + \mathrm{sum}([1,2])$$

$$\mapsto 5 + \mathrm{case}\ 1::2::[\,]\ \mathrm{of}\ [\,] \Rightarrow 0$$
$$\mid f::r \Rightarrow f + \mathrm{sum}(r)$$

$$\mapsto 5 + (1 + \mathrm{sum}(2::[\,]))$$

$$\mapsto 5 + (1 + (2 + \mathrm{sum}[\,]))$$

$$\mapsto 5 + (1 + (2 + \mathrm{case}\ [\,]\ \mathrm{of}\ [\,] \Rightarrow 0 \mid \ldots))$$

$$\mapsto 5 + (1 + (2 + 0)) \mapsto^* 8$$

(* **Purpose:** given a list of salaries, give everyone a raise by a fixed amount *)

(* E.g.

raiseSalaries([11, 14, 16], 4) =

[15, 18, 20] *)

(l, a) : int list * int

```
fun raiseSalaries (l : int list, a : int) : int list =
    case l of
        [] => [] 
      | x :: xs => (x + a) :: raiseSalaries(xs, a)
```

x: first salary

xs: the other salaries

:: "cons"

$rs([11, 14, 16], 4)$

$\mapsto (11+4) :: rs([14, 16], 4)$

$\mapsto 15 :: rs([14, 16], 4)$

$\mapsto 15 :: (14+4) :: rs([16], 4)$

$\mapsto 15 :: 19 :: rs([16], 4)$

$\vdots$

$\mapsto (15 :: 19 :: 20 :: [\,]$

$($

To prove something about **all** lists

List induction

1. Prove it for []

2. Prove it for a list x::xs, assuming it is true for xs

Theorem: for all lists

$$rs(rs(l, a), b)$$

$$\cong$$ "same behavior"

$$rs(l, a+b)$$

Proof:

Case for [ ]:

To show: $rS(rS([\,], a), b) \cong rS([\,], a+b)$

$\mapsto [\,]$

$rS(rS([\,], a), b)$

$\mapsto rS([\,], b)$

$\mapsto [\,]$

$\leftarrow rS([\,], a+b)$

Case for $x::xs$:

Ind hyp: $rs(rS(xs, a), b) \overset{?}{=} rS(xs, a+b)$

To Show: $rs(rS(x::xs, a), b) \overset{?}{=} rS(x::xs, a+b)$

$\mapsto rS(\boxed{(x+a) :: rS(xs, a)}, b)$ $\qquad \mapsto x+(a+b) :: rS(xs, a+b)$

$\mapsto ((x+a)+b) :: \boxed{rS(rs(xs, a), b)}$

by associativity

by Ind. hyp. these are equal