

CS131: Programming Languages

Lun Liu

04.14.17


OCaml: Folding

```
let rec sumList l = match l with  
  [] -> 0  
| h::t -> h + (sumList t)
```


```
let rec concatList l = match l with  
  [] -> ""  
| h::t -> h ^ (concatList t)
```

OCaml: Folding

```
let rec sumList l acc = match l with  
  [] -> acc  
| h::t -> h + (sumList t acc)
```



```
let rec concatList l acc = match l with  
  [] -> acc  
| h::t -> h ^ (concatList t acc)
```



OCaml: Folding

```
let rec sumList l acc = match l with  
  [] -> acc  
| h::t -> (+) h (sumList t acc)
```

```
let rec concatList l acc = match l with  
  [] -> acc  
| h::t -> (^) h (concatList t acc)
```

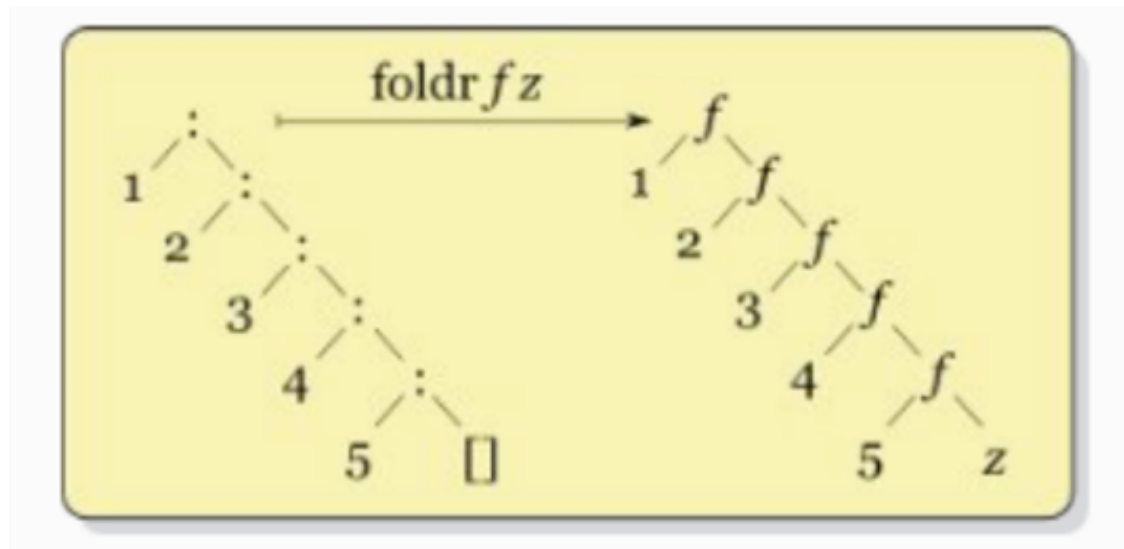
OCaml: `fold_right`

```
let rec fold_right (f : 'a -> 'b -> 'b) (l : 'a
list) (acc : 'b) =
    match l with
    | [] -> acc
    | h::t -> f h (fold_right f t acc);;

val fold_right : ('a -> 'b -> 'b) -> 'a list ->
'b -> 'b = <fun>
```

OCaml: `fold_right`

- `List.fold_right f [a1; ...; an] b` is
`f a1 (f a2 (... (f an b) ...))`.



OCaml: `fold_right`

```
# let length l = List.fold_right (fun e acc ->  
acc + 1) l 0;;
```

```
val length : 'a list -> int = <fun>
```

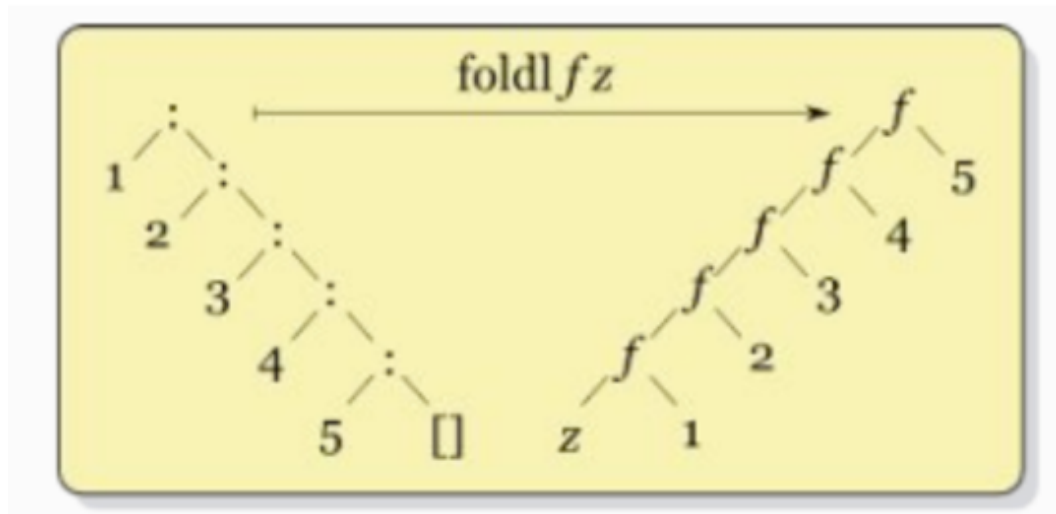
```
# length [2;3;4;5];;
```

```
- : int = 4
```

Exercise

OCaml: `fold_left`

- `List.fold_left f a [b1; ...; bn]` is
 $f \ (... \ (f \ (f \ a \ b1) \ b2) \ ...) \ b_n$.



Exercise: `fold_left`

OCaml: `fold_left` vs `fold_right`

- `fold_left`
 - Elements are folded from left to right
 - `f` takes the previous value first and then list element
- `fold_right`
 - Elements are folded from left to right
 - `f` takes the list element first and then the previous value

OCaml: `fold_left` vs `fold_right`

```
# List.fold_left (-) 0 [1;2;3;4];;
```

```
- : int = ?
```

```
# List.fold_right (-) [1;2;3;4] 0;;
```

```
- : int = ?
```

OCaml: `fold_left` vs `fold_right`

```
# List.fold_left (-) 0 [1;2;3;4];;
```

```
- : int = -10
```

```
# List.fold_right (-) [1;2;3;4] 0;;
```

```
- : int = -2
```

OCaml: `fold_left` vs `fold_right`

- `fold_left`
 - Tail-recursive.
- `fold_right`
 - Not tail-recursive.
- `fold_left` is typically faster

Exercise

Backup

SEASnet Server

- `/usr/local/cs/bin/ocaml`
- Add path to `~/.profile`
 - `PATH=/usr/local/cs/bin:$PATH` //add this line to `~/.profile`