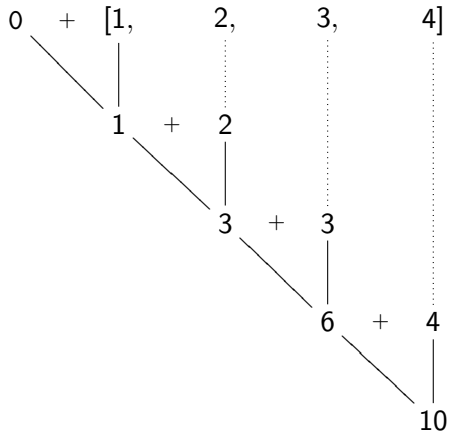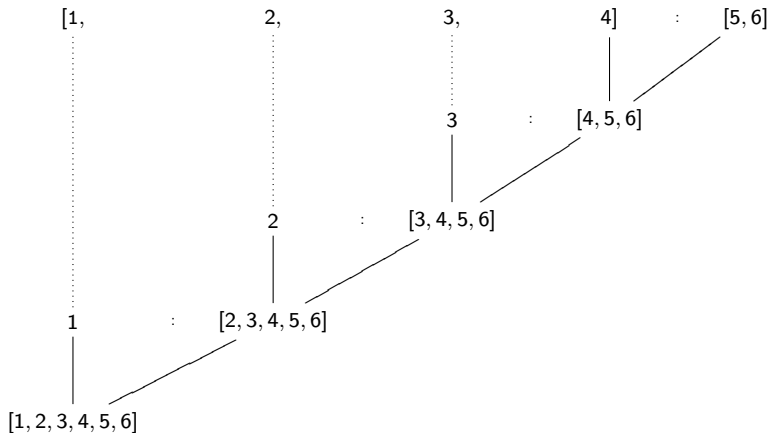# foldl, foldr

```
foldl (+) 0 [1, 2, 3, 4]
```

```
foldr (:) [5,6] [1,2,3,4]
```

Lambda abstrakce, částečná aplikace

```
f x y =           x + y
f     = (\x y -> x + y)

a ⊕ b = ...

(⊕) = (\a b -> ...)
(a ⊕) = (\b -> a ⊕ b) =      (⊕) a
(⊕ b) = (\a -> a ⊕ b) = flip (⊕) b

flip :: (a -> b -> c) -> b -> a -> c
flip f x y = f y x
```

Pointfree

```
(.) :: (b -> c) -> (a -> b) -> a -> c
(f . g) x = f (g x)
(.) f g x = f (g x)
```

```
h x =   f   (g   x)
h x = (f  .  g) x
h   =  f . g
```

η-redukce x

Pointfree – příklady

```
h x =   f   (g  x)
h x = (f . g) x
h   =   f . g
```

```
f u v w x = u v (w x)


f u v w x =   u v   (w  x)
f u v w x = (u v . w) x
f u v w x = (u v .) w x
f u v     = (u v .)          po η-redukci w a x
```
`(\a -> b ⊕ a) = (b ⊕)`

```
f u v     =   (.)  (u  v)
f u v     = ((.) . u) v
f u v     = ((.) .) u v      η-redukujeme u a v
f         = ((.) .)          (f ⊕) = ((⊕) f)
f         = ((.)(.))
```

```
f         =  (.)(.)
```

Pointfree – příklady

```
h x   =   f   (g   x)
h x   = (f . g) x
h     =   f . g
```

```
h x y = q y . q x
```

```
h x y =               q y .   q  x
h x y =          (.) (q y)   (q  x)
h x y =   flip (.) (q x)   (q  y)
h x y = (flip (.) (q x) . q) y
h x   =  flip (.) (q x) . q
```
`f b a = flip f a b`

`po η-redukci y`

```
h x   =   (. q)  ( flip (.)  (q   x))
h x   =   (. q)  ((flip (.) . q)  x)
```

```
h x   =   (. q)   ((flip (.) . q)  x)
h x   = ((. q) . (flip (.) . q)) x
```
`f.(g.h) = f.g.h`

```
h     =   (. q) .  flip (.) . q
```
`po η-redukci x`

Pointfree – příklady

```
h x =   f   (g   x)
h x = (f . g) x
h   =  f . g
```

```
f x y = q y (q x)

f x y =      q y      (q x)        f b a = flip f a b
f x y = flip q (q x) y
f x   = flip q (q x)          po η-redukci y

f x   =  flip q  (q  x)
f x   = (flip q . q) x
f     =  flip q . q       po η-redukci x
```