```
let a  = (fun a b -> (+) b)
```

a : 'a -> int -> int -> int

```
let b  = (fun a b -> List.fold_left b 1 (List.map ( * ) a))
```

b : (int list) -> (int -> (int -> int) -> int)
   -> int

```
let c  = (fun a b c -> c (a + b)) 3
```

c : int -> (int -> 'a) -> 'a

```
let d  = (fun a b c -> b (c a) :: [a]) "x"
```

d : ('a -> string) -> (string -> 'a)
 -> string list

```
let e  = (let x = List.map in x (<))
```

e : 'a list -> ('a -> bool) list