

## FUNCTIONAL PART

DATA TYPES      FUNCTION      "PROGRAMS"

functions → first class  
                                citizens

no side effects  
strict typing

LISP

integer string char  
float

list array

let add x y = x + y ;;

:int → int → int

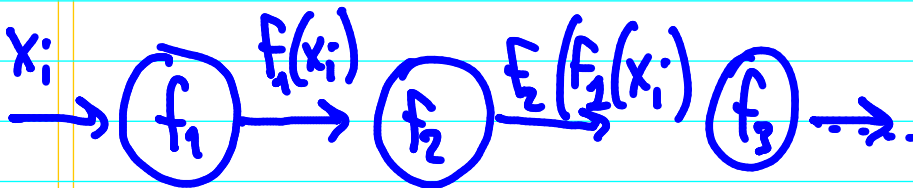
(add 3) =: int → int

# STREAM PARALLEL SKELETONS

pipeline

farm

pipeline( $s_1, \dots, s_n$ ): stream  $x_i \rightarrow$  stream  
 $f_1 \quad f_n \quad f_n(\dots f_2(f_1(x_i)))$



$f_1(x_i) \parallel f_2(f_1(x_{i-1}))$

Form

$x_1 x_2 \dots x_n \rightarrow \text{Form}(f) \rightarrow$

$f(x_1) f(x_2) \dots f(x_n)$

$f(x_1) \parallel f(x_2) \parallel \dots \parallel f(x_n)$