Parallelism + effects

```
fun print(s: string): unit =
  TextIO.output(TextIO.stdOut, s)
```

(print "10", print "23")

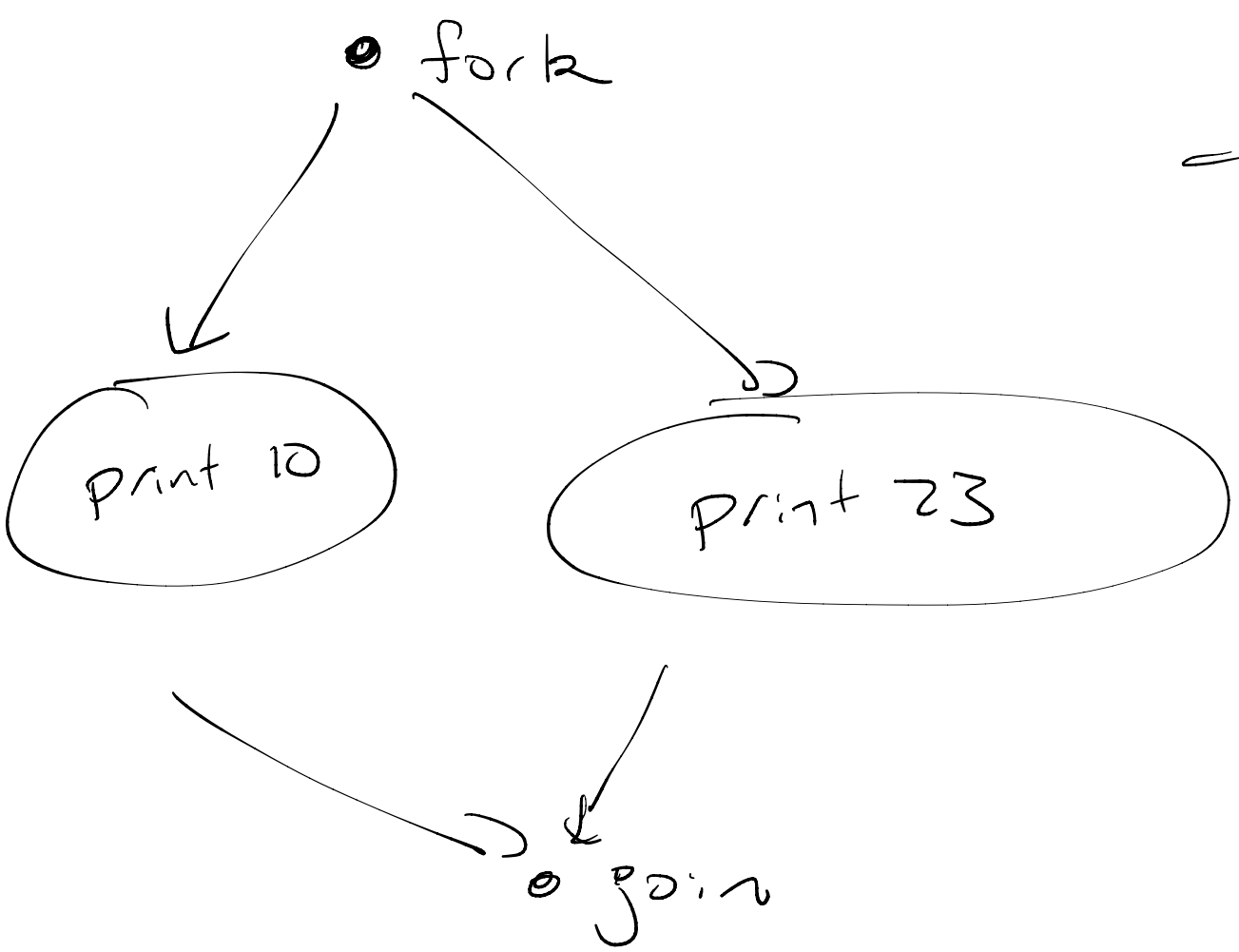
$\mapsto$ ((), print "23")

$\mapsto$ ((), ())

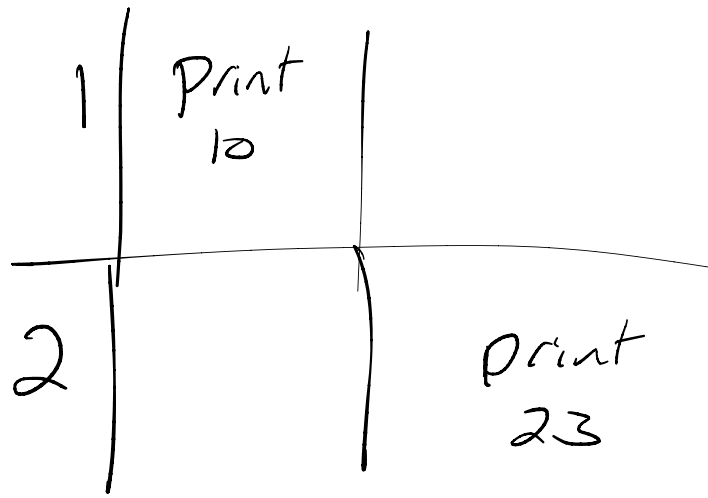
StdOut

In terminal:

1023

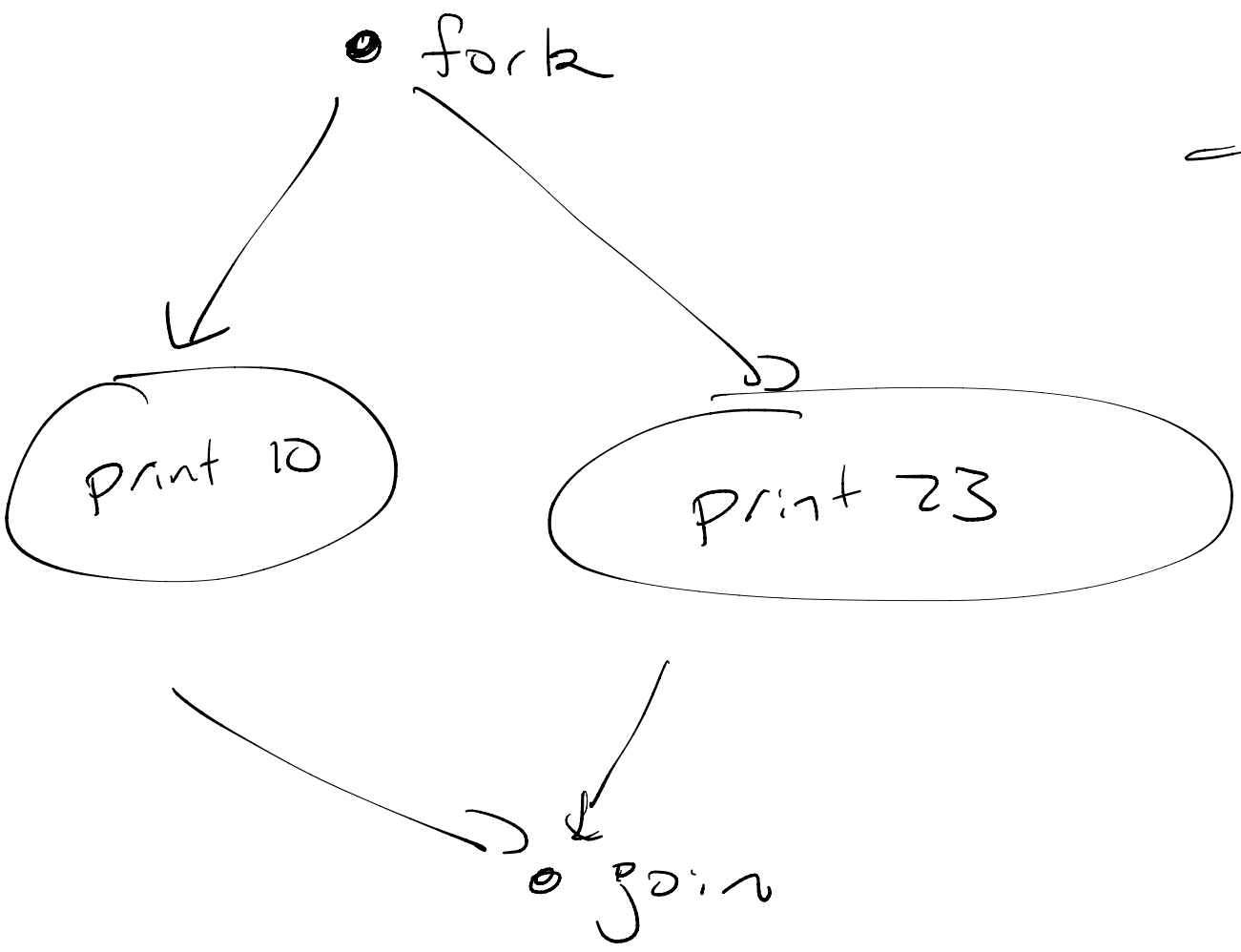


schedule

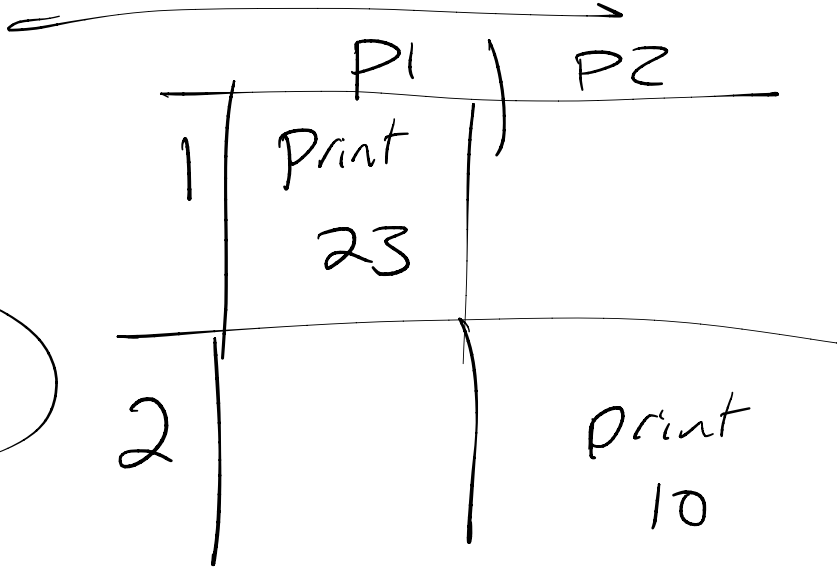


stdout

1023

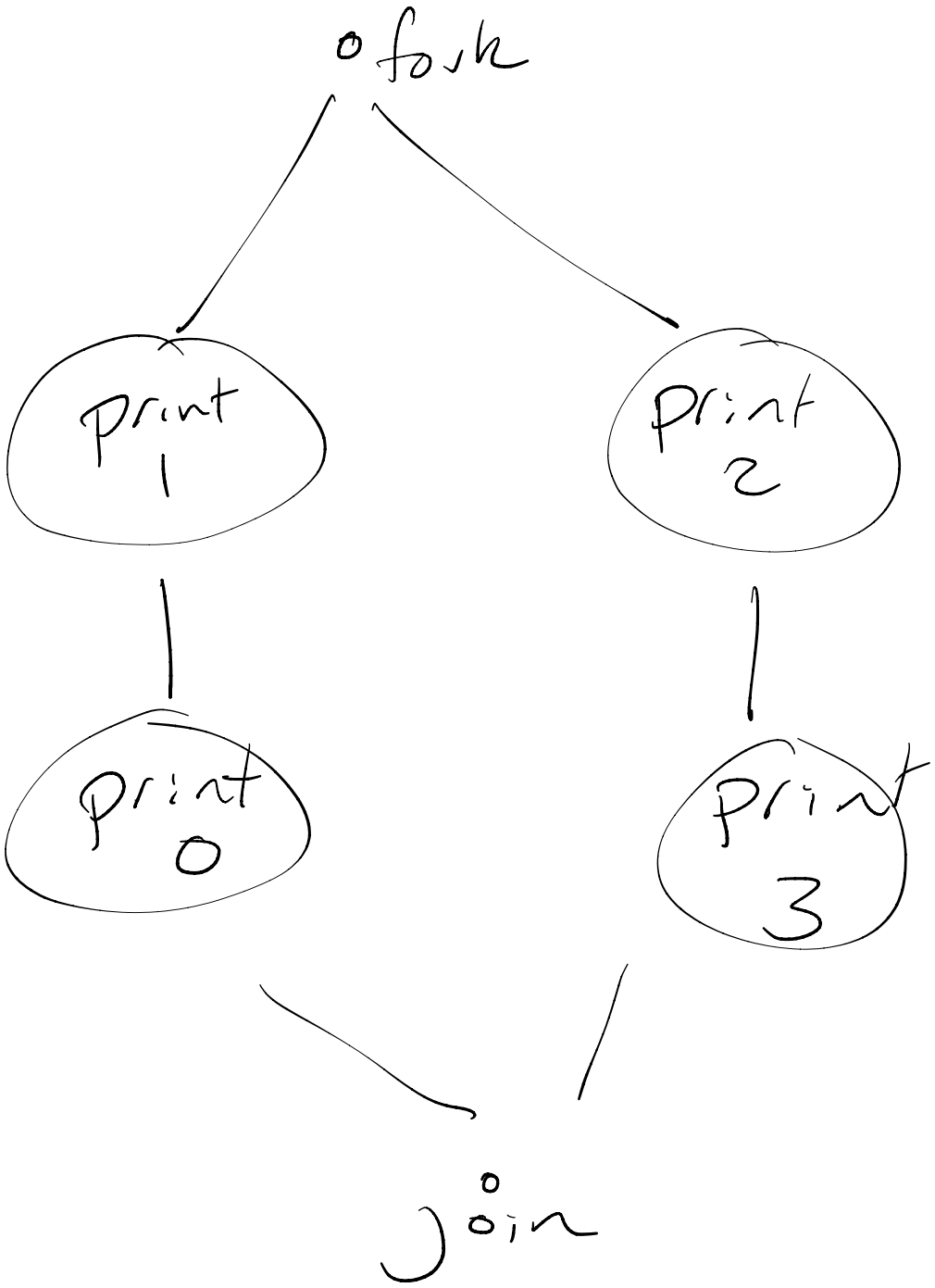


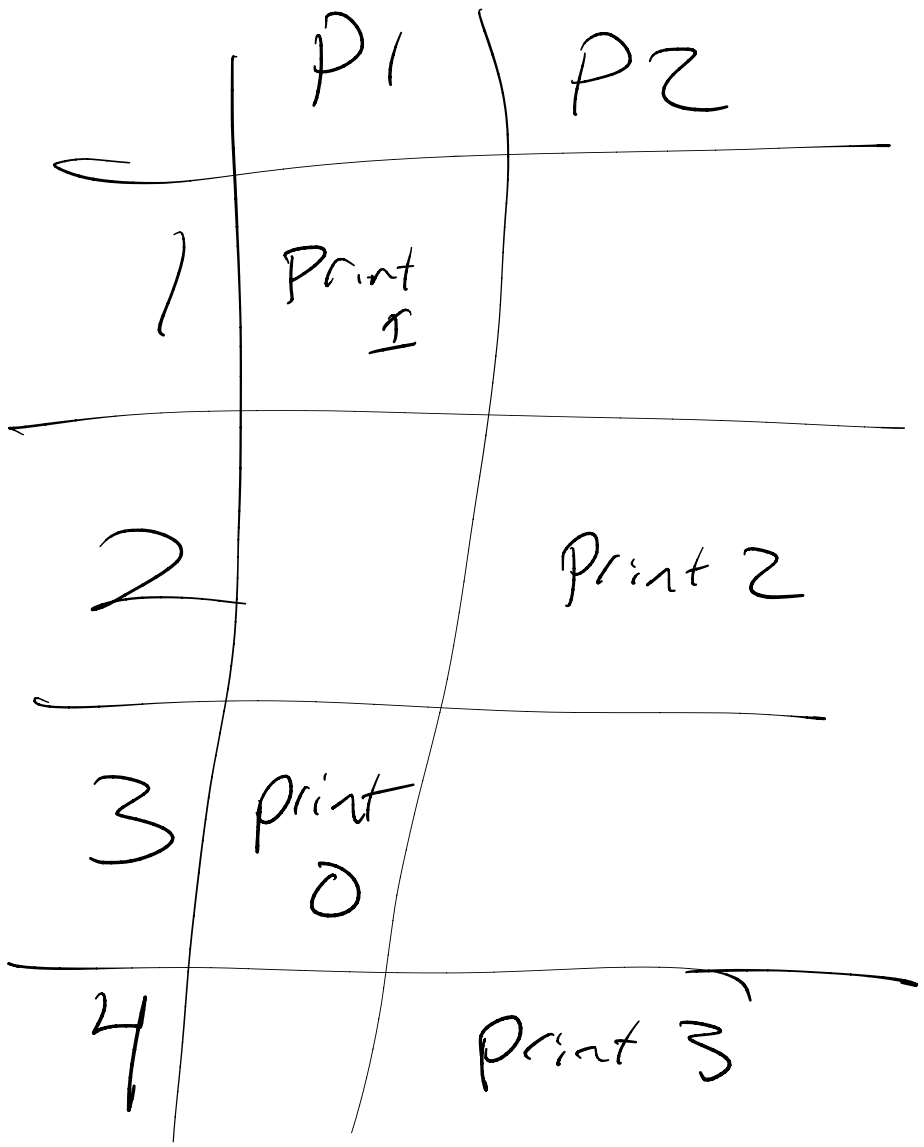
schedule



stdout

2310


fork
print 1
print 2
print 0
print 3
join

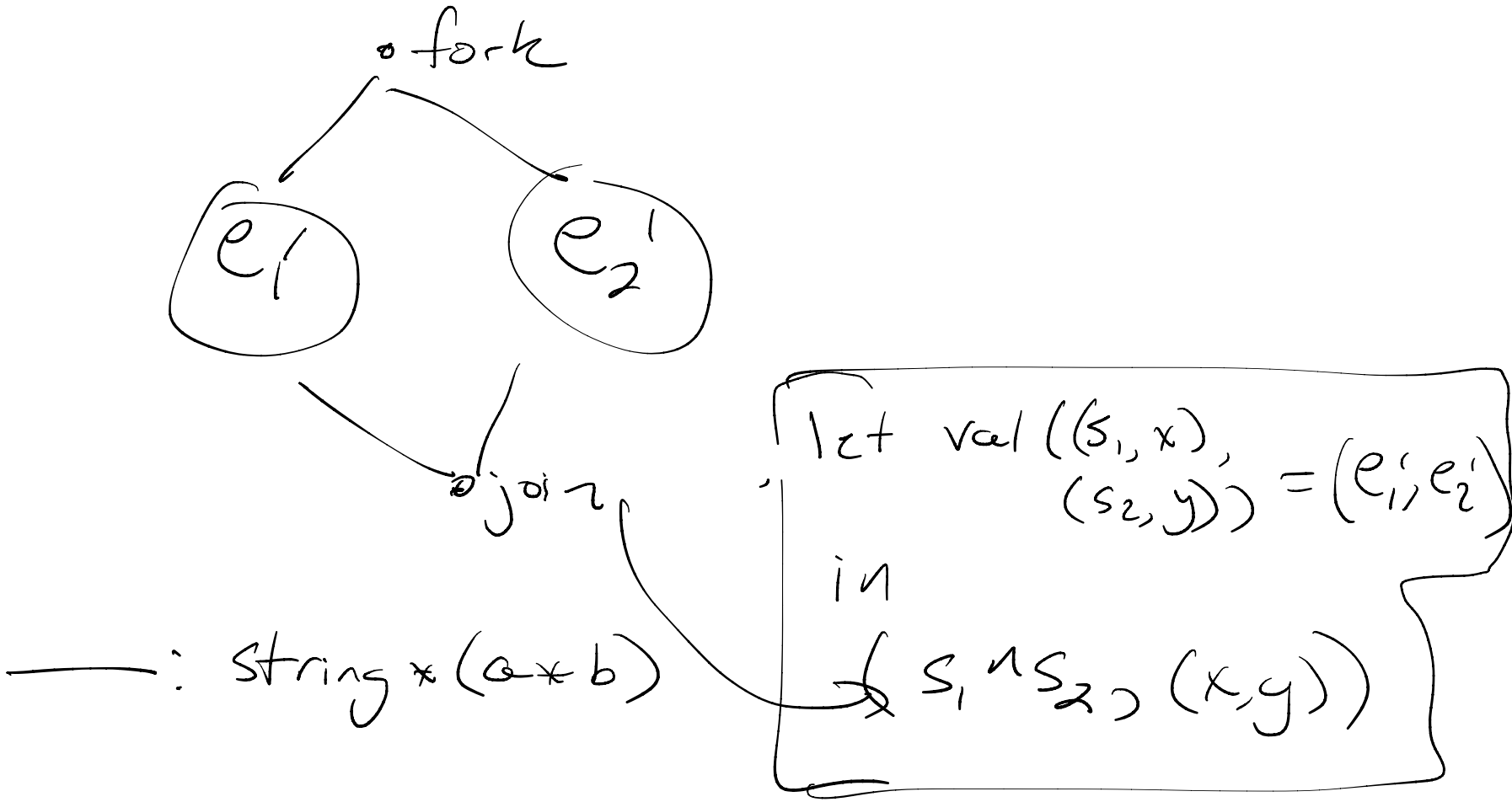

P1
P2
1
Print 1
2
Print 2
3
print 0
4
Print 3

StdOut: 1203

$e_1 : a$ $e_2 : b$ might print

want: $(e_1, e_2) : a * b$

$e_1' : string * a$ $e_2' : string * b$



e_1' e_2'

("10", ()) ("23", ())

overall: ("1023", ((), ()))

- raising exceptions
- output
- reading input

→ determinism

↓

schedule doesn't change the behavior!

Currying

Haskell Curry

```
fun add(x:int, y:int): int = x+y
```

int * int ⟶ int

uncurried

$int * int \longrightarrow int$

$int \longrightarrow (int \longrightarrow int)$

```
fun cadd (x:int) : int -> int =
    fn y:int => x + y
```

int

Curried

add (3, 4)

$\mapsto$ 3+4

$\mapsto$ 7

(add 3) 4

int → int

$\mapsto$ (fn y => 3+y) 4

$\mapsto$ 3+4

$\mapsto$ 7

```
fun addp (x:int):int->int =
    fn y => let val () = print "hello"
            in x+y
            end
```

same as `fun add(x,y) = let val () = print "hello" in (x+y) end`

`add : int * int -> int`

```
fun addq(x:int):int->int =
    let val () = print "hello"
    in
        fn y => x+y
    end
```

addp 3 4

$\mapsto$ (fn y => let val () = print "hi"
in
3+y
end) 3+y 4

$\mapsto$ let val () = print "hi"
in
3+7
end

$\mapsto$ 3+7

$\mapsto$ 7

stdOut

"hi"

addg 3 4

$\mapsto$ (let val () = print "hi"
in
fn y => 3 + y
end) 4

$\mapsto$ (fn y => 3 + y) 4

$\mapsto$ 3 + 4

$\mapsto$ 7

stdout

hi

map(addp 3, [1,2,3])

stdout

hi hi hi

⟼ map(fn y => let val () = print "hi"
in
3+y,
[1,2,3])

⟼ [let val () = print "hi"
in 3+1
,

let val () = print "hi"
in 3+2
,

let val () = print "hi"
in
3+3]

⟼ [4,
5,
6]

map(addg 3, [1,2,3])

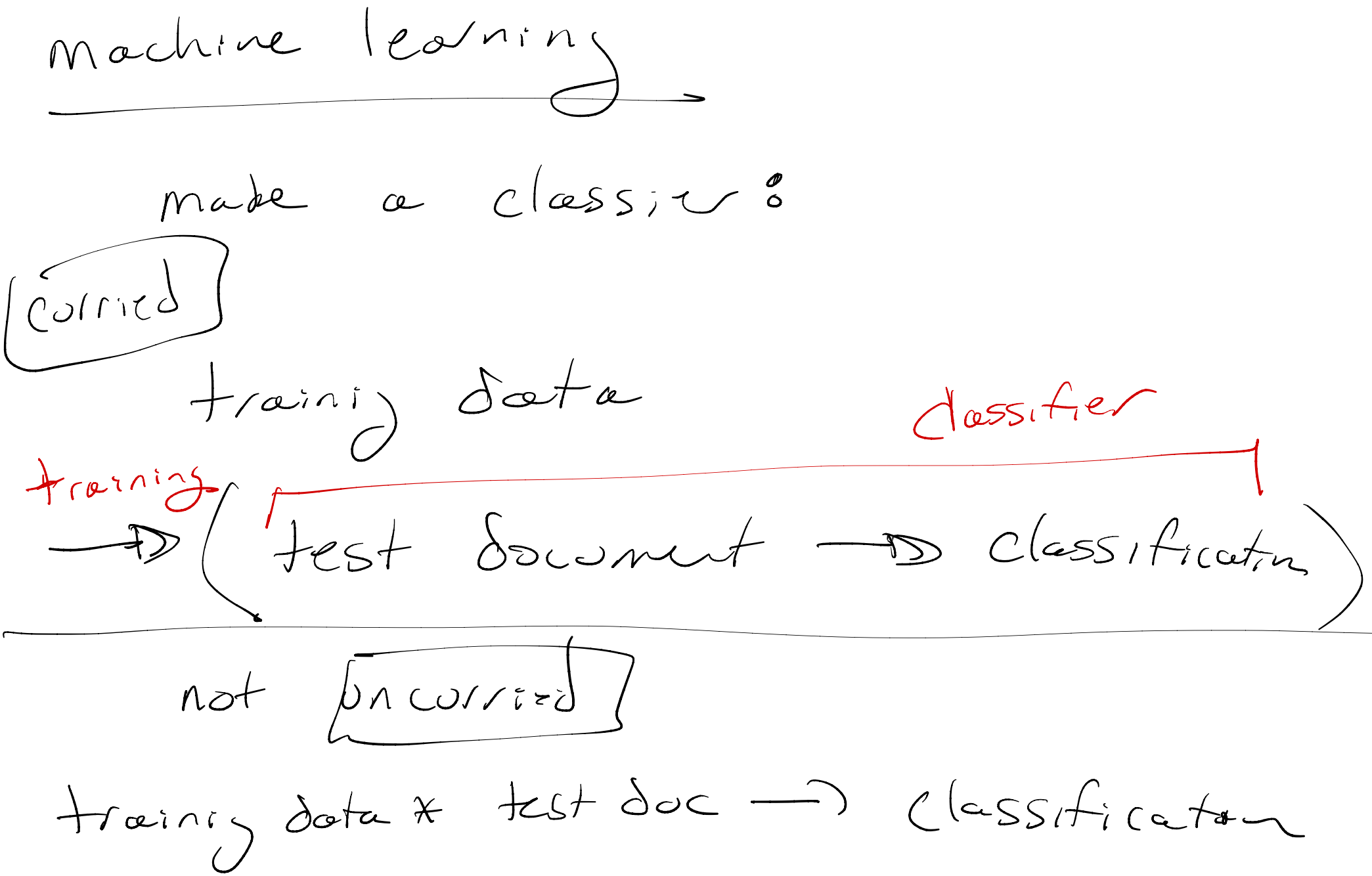
↦ map(let val () = print "hi"
in fn y => 3+y , [1,2,3])

↦ map(fn y => 3+y , [1,2,3])

↦* [4,5,6]

stdout

hi


machine learning
make a classifier
corrried
training data
training
classifier
test document
classification
not
uncorrrized
training data x test doc
classification