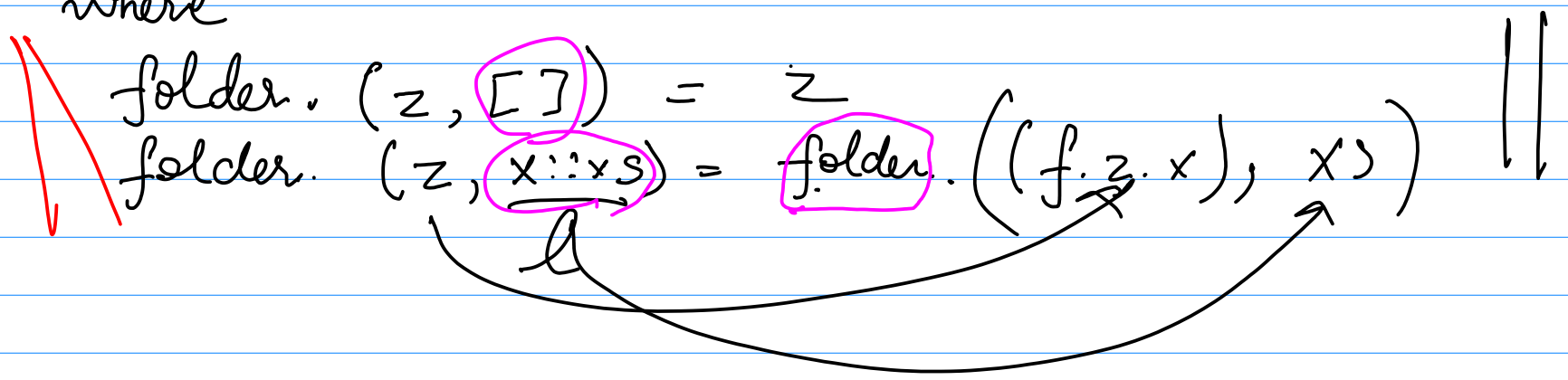
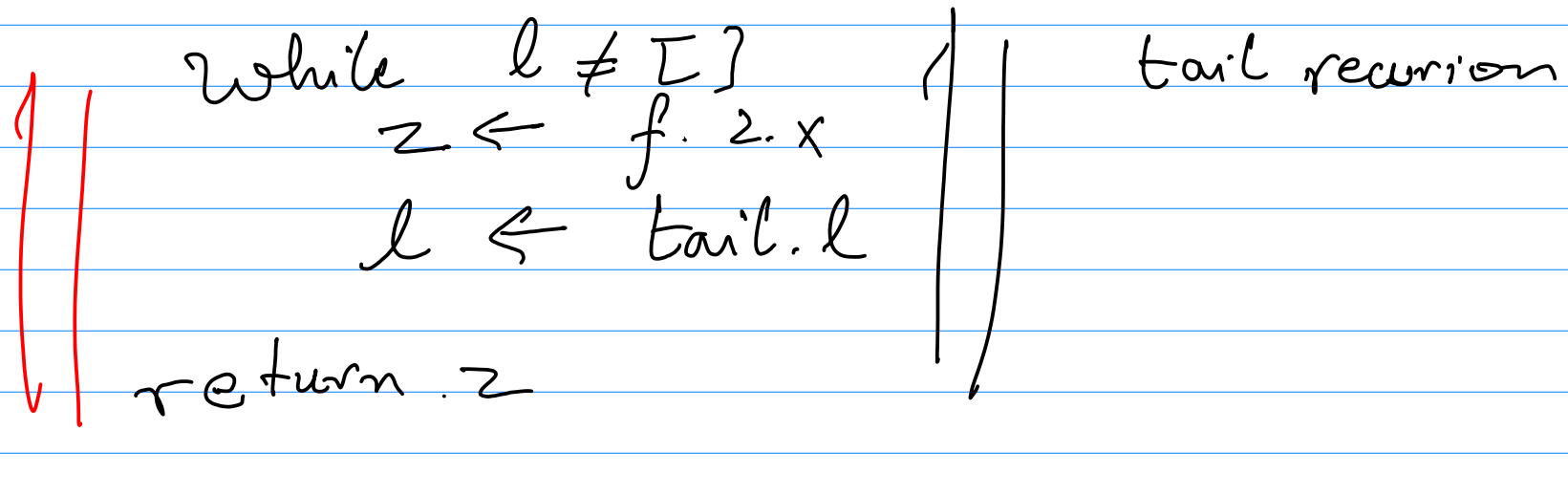
foldl . f = folder

where

folder . (z, []) = z

folder . (z, x:xs) = folder . ((f . z . x), xs)





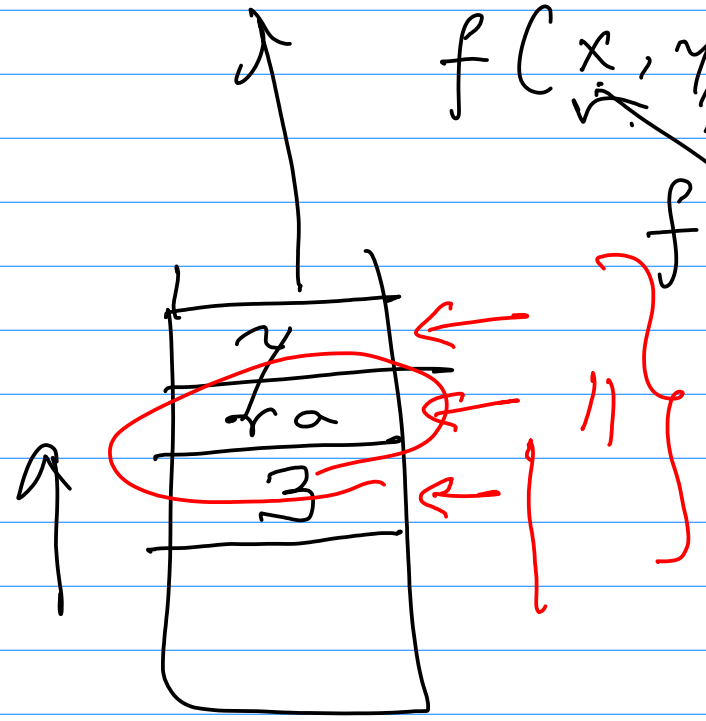
```
int f(int x)
{
    int y;
    .
    .
    .
}

main()
{
    f(3)
```

bool

f → f

f(x, y, z)

f(a, b, c)



Pauli

"It is not even wrong"

CPython	Pypy
foldl —	tail recursive
foldr —	?? recursive

etype [a] where

1. (:) : a → [a] → [a]
2. [] : [a]

```
data Tree a where
  Lf : a -> Tree a
  Br : a -> Tree a -> Tree a -> Tree a
```

fold Tree