

$$\frac{\vdash a \rightarrow (b \rightarrow c) \rightarrow b}{\vdash a \rightarrow (b \rightarrow c)}$$

$\vdash \forall x \forall y \exists z$   
 $(x \wedge y \rightarrow z)$

$\vdash \text{True} \wedge \dots$

$$\frac{\vdash a \rightarrow b \rightarrow c}{\vdash a \rightarrow b \rightarrow c}$$

$\vdash$

$g :: b \rightarrow c$

$f :: (a, b) \rightarrow a$

sum(LIST):

$a = 0$

$i = 0$

while ( $i < \text{List.length}$ ) {

$a += \text{LIST}[i];$

$i++$

}

return a

sum l = foldr (+) 0 l

foldr ::  $(d \rightarrow a \rightarrow a) \rightarrow a \rightarrow [d] \rightarrow a$

(in Python: "reduce")  
and Lisp

$a :: \text{Int}$

$d :: \text{Maybe Int}$

case  $d$  of

Nothing  $\rightarrow$  "False"

Just  $n$   $\rightarrow$  show  $n$

data DayMessage = Day-Regis Int  
| Told Player String

data Maybe a = Nothing | Just a

Anyway  $a = \{1, 2, 3\}$

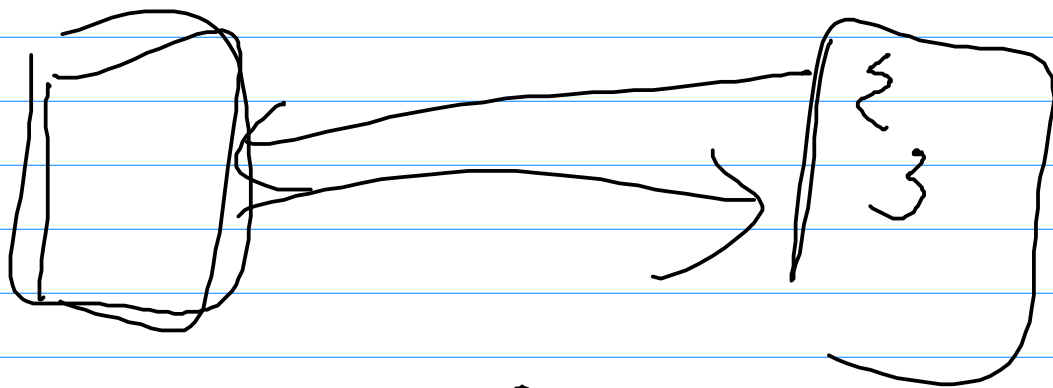
foo ( $\{ \}$ );

```

sub create_candle {
  my $candle = 0;
  return sub { $candle
}
}

```

---



```
for (1..10)
```

```

{
  (closure.make(),)
}

```

```

return; break;
}

```

Fib = 1: 1: ZipWith (+)  
fib (tail fib)

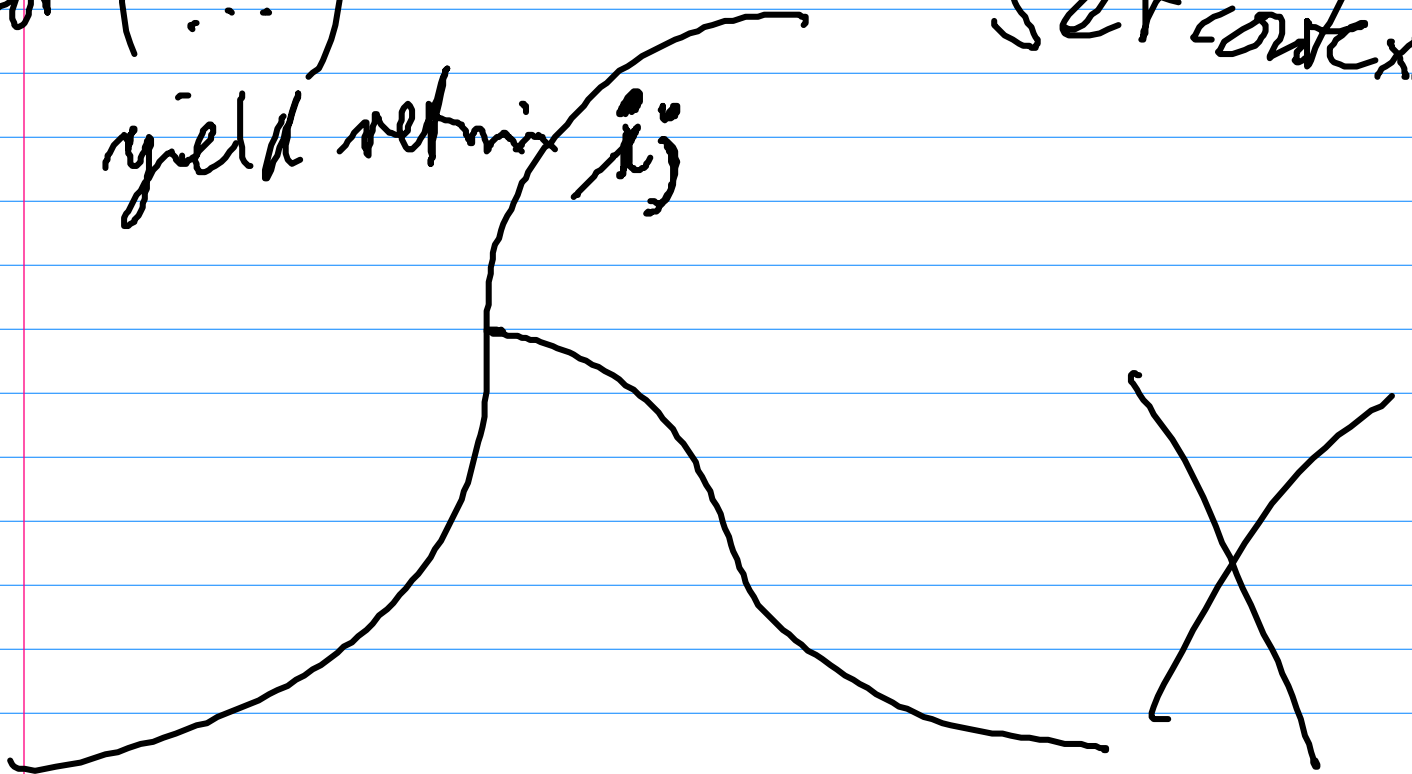
fib n = fib (n-1) + fib (n-2)

---

for (...)

yield return ij

set context



(x..x) / (x..x)

Haskell

Lisp Python

Java

Perl++

Perl

C# PHP

SVG

Lua