Cálculo de Programas

Class T01 — 19-Sep

Introduction to the course.

Website: **https://haslab.github.io/CP**

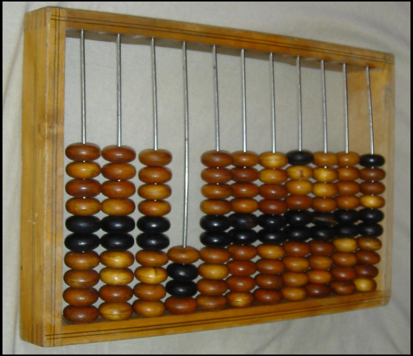
Other administrative details.

J.N. Oliveira

Programming by Calculation

Program versus Calculus

- **Calculus** — abacus **pebble**
- **Program** — *pro* ('before') + *graphein* ('write')

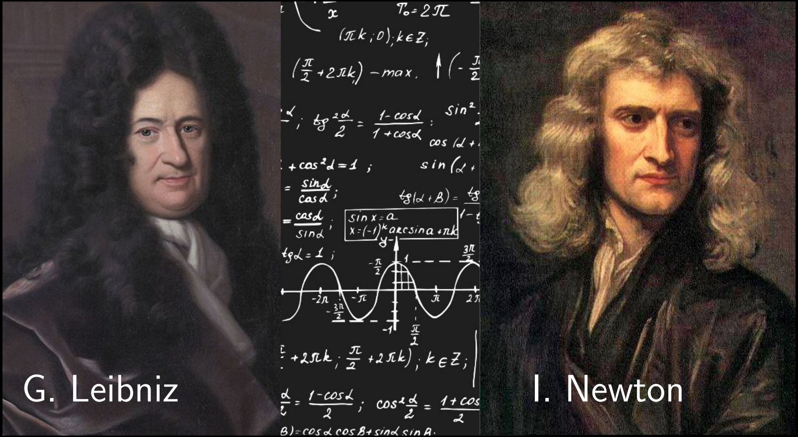


Programs...

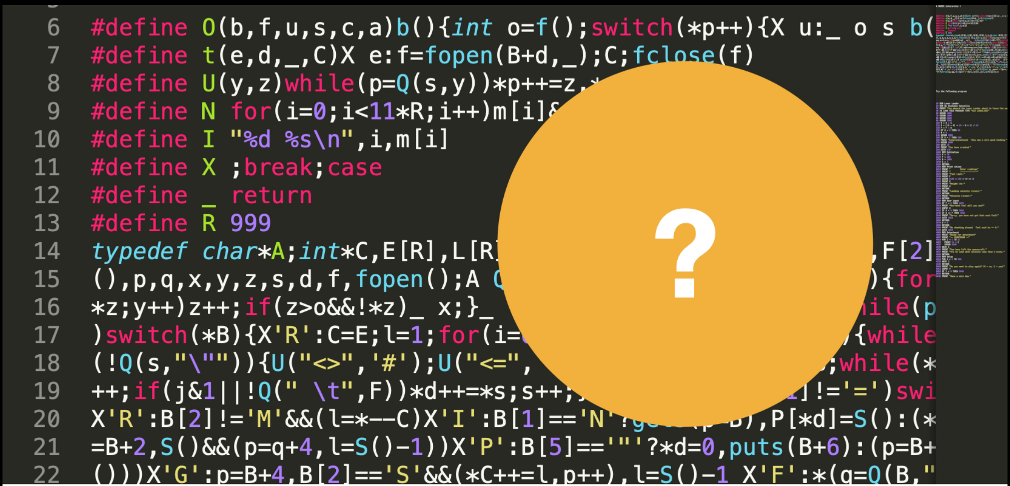
wc.c

```c
#define YES 1
#define NO 0
main()
{
int c, nl, nw, nc, inword ;
inword = NO ;
nl = 0;
nw = 0;
nc = 0;
c = getchar();
while ( c != EOF ) {
    nc = nc + 1;
    if ( c == '\n')
    nl = nl + 1;
    if ( c == ' ' || c == '\n' || c == '\t')
          inword = NO;
    else if (inword == NO) {
          inword = YES ;
          nw = nw + 1;
    }
    c = getchar();
}
printf("%d",nl);
printf("%d",nw);
printf("%d",nc);
}
```

Calculus



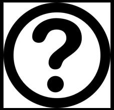
```
#define O(b,f,u,s,c,a)b(){int o=f();switch(*p++){X u:_ o s b(
#define t(e,d,_,C)X e:f=fopen(B+d,_);C;fclose(f)
#define U(y,z)while(p=Q(s,y))*p++=z,*p=' '
#define N for(i=0;i<11*R;i++)m[i]&&
#define I "%d %s\n",i,m[i]
#define X ;break;case
#define _ return
#define R 999
typedef char*A;int*C,E[R],L[R],M[R],P[R],l,i,j;char B[R],F[2]
(),p,q,x,y,z,s,d,f,fopen();A Q(s,o)A s,o;{for(x=s;*x;x++){for
*z;y++)z++;if(z>o&&!*z)_ x;}_   0;}main(){m[11*R]="E";while(p
)switch(*B){X'R':C=E;l=1;for(i=0;i<R;P[i++]=0);while(l){while
(!Q(s,"\"")){U("<>",'#');U("<=",'$');U(">=",'!');}d=B;while(*
++;if(j&1||!Q(" \t",F))*d++=*s;s++;}*d--=j=0;if(B[1]!='=')swi
X'R':B[2]!='M'&&(l=*--C)X'I':B[1]=='N'?gets(p=B),P[*d]=S():(*
=B+2,S()&&(p=q+4,l=S()-1))X'P':B[5]=='"'?*d=0,puts(B+6):(p=B+
()))X'G':p=B+4,B[2]=='S'&&(*C++=l,p++),l=S()-1 X'F':*(q=Q(B,"
```


#define O(b,f,u,s,c,a)b(){int o=f();switch(*p++){X u:_ o s b(
#define t(e,d,_,C)X e:f=fopen(B+d,_);C;fclose(f)
#define U(y,z)while(p=Q(s,y))*p++=z,
#define N for(i=0;i<11*R;i++)m[i]&
#define I "%d %s\n",i,m[i]
#define X ;break;case
#define _ return
#define R 999
typedef char*A;int*C,E[R],L[R
,F[2]
(),p,q,x,y,z,s,d,f,fopen();A
){for
*z;y++)z++;if(z>o&&!*z)_ x;}_
ile(p
)switch(*B){X'R':C=E;l=1;for(i=
{while
(!Q(s,"\"")){U("<>",'#');U("<=",
;while(*
++;if(j&1||!Q(" \t",F))*d++=*s;s++;
]!='=')swi
X'R':B[2]!='M'&&(l=*--C)X'I':B[1]=='N'
,P[*d]=S():(*
=B+2,S()&&(p=q+4,l=S()-1))X'P':B[5]=='"'?*d=0,puts(B+6):(p=B+
()))X'G':p=B+4,B[2]=='S'&&(*C++=l,p++),l=S()-1 X'F':*(q=Q(B,"

Software as a problem

Software

Process — 

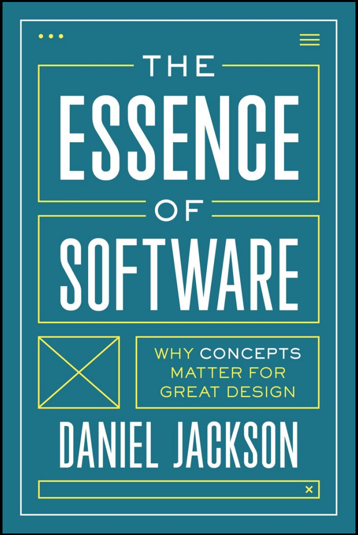
Product — 

THE ESSENCE OF SOFTWARE

WHY CONCEPTS MATTER FOR GREAT DESIGN

DANIEL JACKSON

"(...) The best services revolve around **a small number of concepts** that are **well designed and easy** (...) **to understand and use**, and their innovations often involve simple but compelling new concepts."



In

The Essence of Software

by

Prof. Daniel Jackson, MIT

(2021)

... small number

… small number

… well defined

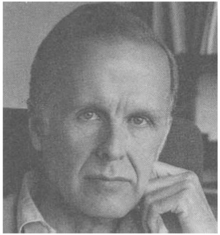
... small number
... well defined
... easy to understand

Urgently needed in software design

John Backus – Turing Award (1978)

Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs

John Backus
IBM Research Laboratory, San Jose



Conventional programming languages are growing ever more enormous, but not stronger. Inherent defects at the most basic level cause them to be both fat and weak: their primitive word-at-a-time style of programming inherited from their common ancestor—the von Neumann computer, their close coupling of semantics to state transitions, their division of programming into a world of expressions and a world of statements, their inability to effectively use powerful combining forms for building new programs from existing ones, and their lack of useful mathematical properties for reasoning about programs.

An alternative functional style of programming is

Functional Programming

FP = $$$...

Jose Pedro Magalhaes, *Functional Programming in Financial Markets* (Wednesday, April 3 2024, 11:38 -- Thursday, April 4 2024, 9:44)

> We present a case-study of using functional programming in the real world at a very large scale. At a large global financial institution, Haskell is used in a core software library supporting a business line with 5 billion USD operating income in 2023. Typed functional programming is used across the entire tech stack, including foundational APIs and CLIs for deal valuation and risk analysis, server-side components for long-running batches or sub-second

Part I

Motivation — back to the late 1990s

From a mobile phone manufacturer

(...) For each **list of calls** *stored in the mobile phone (e.g. numbers dialled, SMS messages, lost calls), the* **store** *operation should work in a way such that* **(a)** *the more recently a* **call** *is made the more accessible it is;* **(b)** *no number appears twice in a list;* **(c)** *only the last 10 entries in each list are stored.*

From a mobile phone manufacturer

```
store ::  Call -> [Call] -> [Call]
store c l = take 10 (nub (c:l))
```

From a mobile phone manufacturer

```
store :: Call -> [Call] -> [Call]
```

$$\texttt{store c l = }\underbrace{\texttt{take 10}}_{(c)}\ \texttt{(}\underbrace{\texttt{nub}}_{(b)}\ \underbrace{\texttt{(c:l)}}_{(a)}\texttt{)}$$

Compare with ...

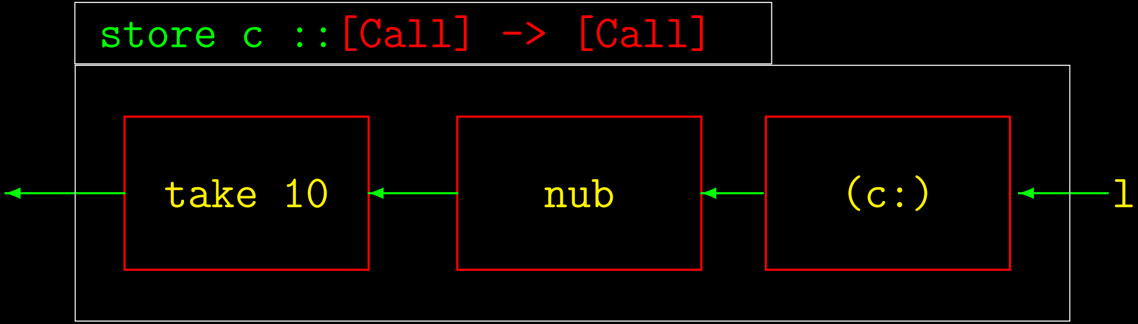
```
public void store10(string phoneNumber)
{
  System.Collections.ArrayList auxList =
      new System.Collections.ArrayList();
  auxList.Add(phoneNumber);
  auxList.AddRange(
      this.filteratmost9(phoneNumber) );
  this.callList = auxList;
}
```

+ `filteratmost9` (next slide)

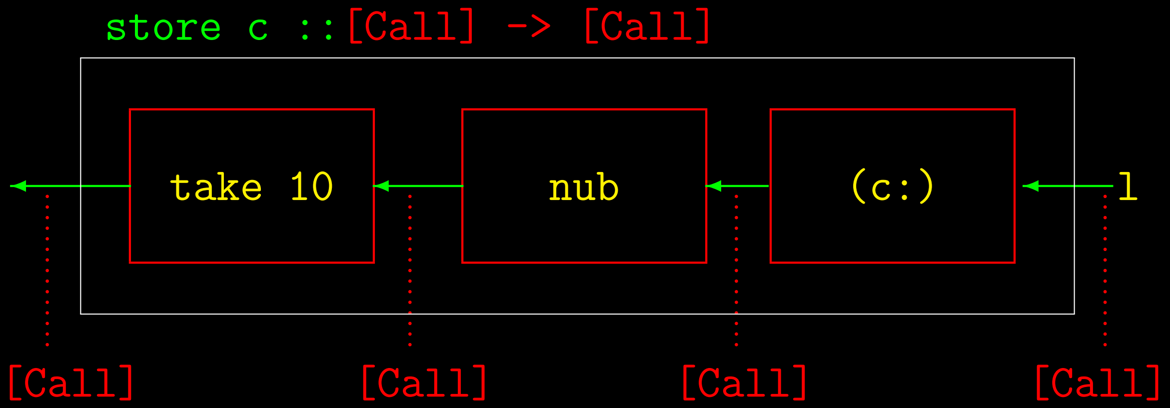
Compare with ...

```
public System.Collections.ArrayList filteratmost9(string n)
{
  System.Collections.ArrayList retList =
      new System.Collections.ArrayList();
      int i=0, m=0;
  while((i < this.callList.Count) && (m < 9))
  {
      if ((string)this.callList[i] != n)
      {
          retList.Add(this.callList[i]);
          m++;
      }
      i++;
  }
  return retList;
}
```

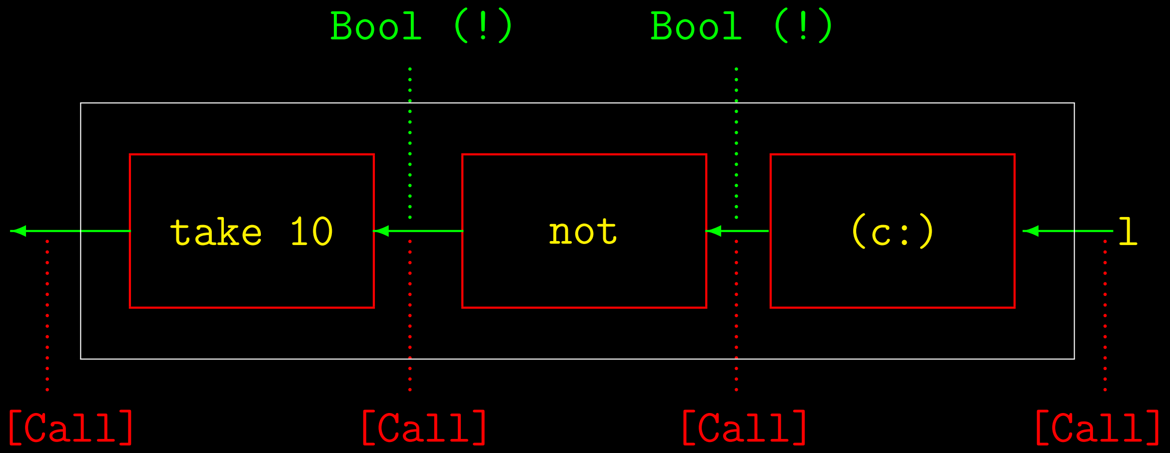
From a mobile phone manufacturer



From a mobile phone manufacturer

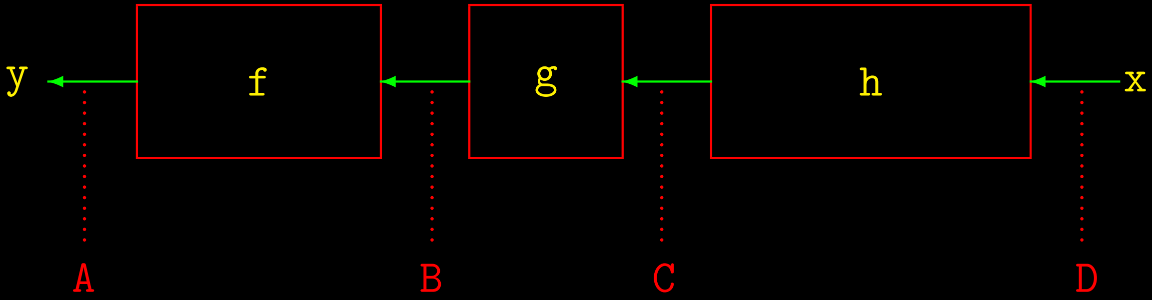


Uups!



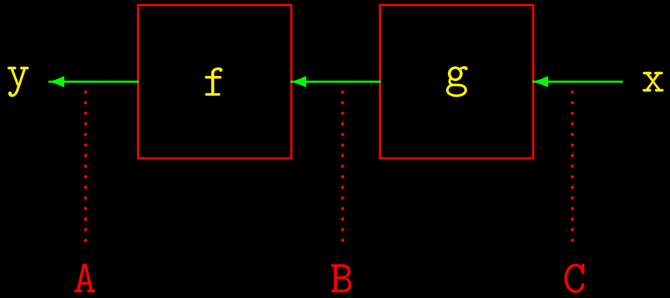
In general

```
y = f(g(h x))
```



In general

```
y = f(g x)
```

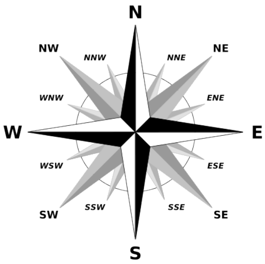


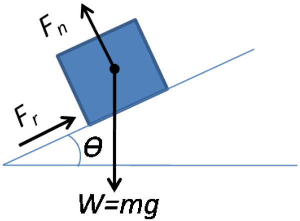
Simplification

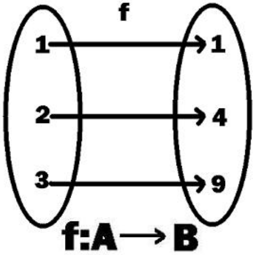
```
y = f(g x)
```

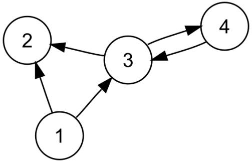
$$\begin{tikzcd} A & B \arrow[l, "f"'] & C \arrow[l, "g"'] \end{tikzcd}$$

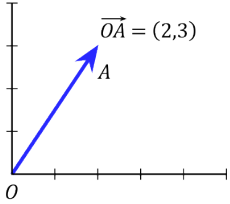
Check the pictures...



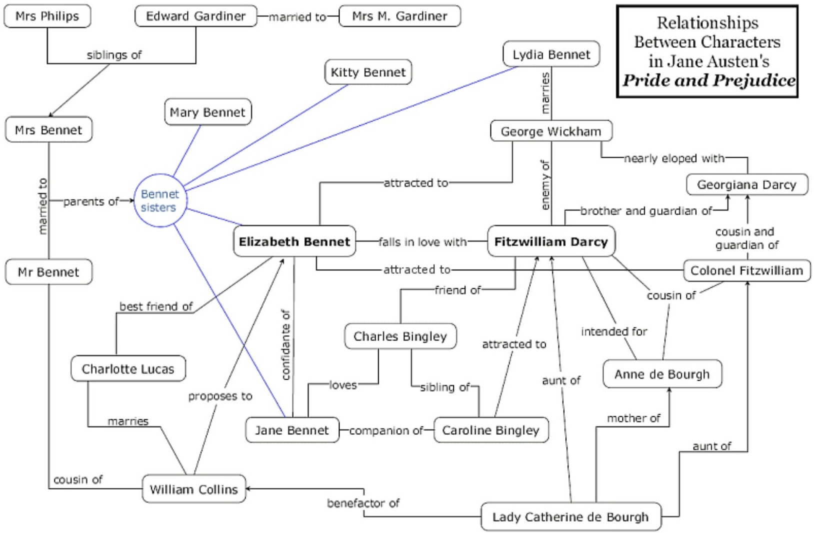






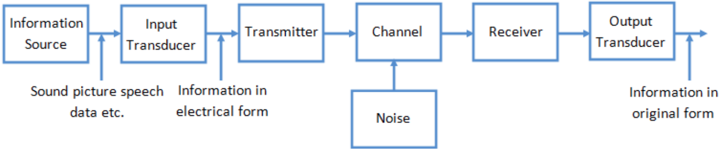


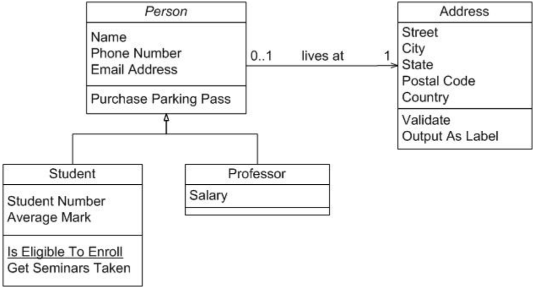
Arrows



Relationships Between Characters in Jane Austen's *Pride and Prejudice*
Mrs Philips
Edward Gardiner
married to
Mrs M. Gardiner
siblings of
Lydia Bennet
Kitty Bennet
Mary Bennet
marries
Mrs Bennet
George Wickham
nearly eloped with
married to
parents of
Bennet sisters
attracted to
enemy of
Georgiana Darcy
brother and guardian of
cousin and guardian of
Elizabeth Bennet
falls in love with
Fitzwilliam Darcy
Mr Bennet
attracted to
Colonel Fitzwilliam
friend of
cousin of
best friend of
confidante of
Charles Bingley
intended for
attracted to
Charlotte Lucas
Anne de Bourgh
loves
sibling of
aunt of
proposes to
marries
Jane Bennet
companion of
Caroline Bingley
mother of
aunt of
cousin of
William Collins
benefactor of
Lady Catherine de Bourgh

More arrows...


Information Source
Input Transducer
Transmitter
Channel
Receiver
Output Transducer
Noise
Sound picture speech data etc.
Information in electrical form
Information in original form


Person
Name
Phone Number
Email Address
Purchase Parking Pass
0..1
lives at
1
Address
Street
City
State
Postal Code
Country
Validate
Output As Label
Student
Student Number
Average Mark
Is Eligible To Enroll
Get Seminars Taken
Professor
Salary

Composition

$$y = f(g\ x)$$

$$A \xleftarrow{f} B \xleftarrow{g} C$$

$$A \xleftarrow{f \cdot g} C$$

$$y = (f \cdot g)\ x$$

$$A \xleftarrow{f} B \xleftarrow{g} C$$

$$C \xrightarrow{g} B \xrightarrow{f} A$$

$$
\begin{tikzcd}
A & & \\
& B \arrow[ul, "f"'] & \\
& & C \arrow[ul, "g"']
\end{tikzcd}
$$

Cf. Unix/Linux pipes

$$C \xrightarrow{g} B \xrightarrow{f} A$$

$$g \mid f$$

Composition

$$(f \cdot g) \cdot h = f \cdot (g \cdot h)$$

Composition

$$(f \cdot g) \cdot h = f \cdot (g \cdot h)$$

$$(a + b) + c = a + (b + c)$$

Composition

$$(f \cdot g) \cdot h = f \cdot (g \cdot h)$$

$$(a + b) + c = a + (b + c)$$

$$f \cdot g \cdot h$$

$$a + b + c$$

Composition

$$\mathit{store}\ c = \mathit{take}\ 10 \cdot \underbrace{\mathit{nub} \cdot (c:)}_{\mathit{store}'\ c}$$

Composition

$$store\ c = take\ 10 \cdot \underbrace{nub \cdot (c:)}_{store'\ c}$$

i.e.

$$take\ 10 \cdot (nub \cdot (c:))$$

Composition

$$\mathit{store}\ c = \mathit{take}\ 10 \cdot \underbrace{\mathit{nub} \cdot (c{:})}_{\mathit{store}'\ c}$$

i.e.

$$\mathit{take}\ 10 \cdot (\mathit{nub} \cdot (c{:}))$$

the same as

$$(\mathit{take}\ 10 \cdot \mathit{nub}) \cdot (c{:})$$

Composition

$$(f \cdot g) \cdot h \;=\; f \cdot (g \cdot h)$$
$$(a + b) + c \;=\; a + (b + c)$$

Composition

$$(f \cdot g) \cdot h \;=\; f \cdot (g \cdot h)$$
$$(a + b) + c \;=\; a + (b + c)$$

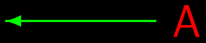
$$a + 0 = 0 + a = a$$

Composition

$$(f \cdot g) \cdot h \;=\; f \cdot (g \cdot h)$$
$$(a + b) + c \;=\; a + (b + c)$$

$$a + 0 = 0 + a = a$$
$$f \cdot ? = ? \cdot f = f$$

A


A

$$A \longleftarrow A$$

$$A \xleftarrow{id} A$$

$$\mathrm{A} \xleftarrow{id} \mathrm{A}$$

$$id\ a = a$$

Identity

$$\begin{tikzcd}
A \arrow[d, "f"'] & A \arrow[l, "id"'] \\
B &
\end{tikzcd}$$

Identity

$$\begin{tikzcd}
A \arrow[d, "f"'] & A \arrow[l, "id"'] \arrow[d, "f"] \\
B & B
\end{tikzcd}$$

Identity

$$\begin{tikzcd} A \arrow[d, "f"'] & A \arrow[l, "id"'] \arrow[d, "f"] \\ B & B \arrow[l, "id"] \end{tikzcd}$$

Identity

$$\begin{tikzcd}
A \arrow[d, "f"'] & A \arrow[l, "id"'] \arrow[dl, "f"'] \arrow[d, "f"] \\
B & B \arrow[l, "id"]
\end{tikzcd}$$

Identity

$$
\begin{tikzcd}
A \arrow[d, "f"'] & A \arrow[l, "id"'] \arrow[d, "f"] \arrow[dl, "f"] \\
B & B \arrow[l, "id"]
\end{tikzcd}
$$

$$f \cdot id = f = id \cdot f$$

Composition and identity

Associativity:

$$(f \cdot g) \cdot h = f \cdot (g \cdot h)$$

Composition and identity

Associativity:

$$(f \cdot g) \cdot h = f \cdot (g \cdot h)$$

" Natural-*id*" :

$$f \cdot id = f = id \cdot f$$

$$f \cdot g$$

$$f \cdot g$$

$$f \times g \ ?$$

$$f \cdot g$$

$$f \times g \ ?$$

$$f + g \ ?$$

$$C \xrightarrow{f} B \quad \text{and} \quad A \xrightarrow{g} C$$

$$C \xrightarrow{f} B \quad \text{and} \quad A \xrightarrow{g} C$$

$$\text{Composition:} \quad A \xrightarrow{f \cdot g} B$$

What about

$$\begin{tikzcd} D \arrow[r, "f"] & B \\ A \arrow[r, "g"] & C \end{tikzcd}$$

?

Try $D = A$ ...

$$\begin{tikzcd} B & & C \\ & A \arrow[ul, "f"] \arrow[ur, "g"'] & \end{tikzcd}$$

Try $D = A$ ...

$$\begin{tikzcd}
B & & C \\
& A \arrow[ul, "f"] \arrow[ur, "g"'] &
\end{tikzcd}$$

$f\ a$ ... $g\ a$

$$\begin{tikzcd}
B & & C \\
& A \arrow[ul, "f"] \arrow[ur, "g"'] &
\end{tikzcd}$$

$$(f\ a, g\ a)$$

Cartesian product

$$B \times C = \{(b, c) \mid b \in B \land c \in C\}$$

Cartesian product

$$B \times C = \{(b, c) \mid b \in B \land c \in C\}$$

$$f\ a \in B$$

Cartesian product

$$B \times C = \{(b, c) \mid b \in B \wedge c \in C\}$$

$$f\ a \in B$$
$$g\ a \in C$$

Cartesian product

$$B \times C = \{(b, c) \mid b \in B \land c \in C\}$$

$$\frac{f\ a \in B \qquad g\ a \in C}{(f\ a, g\ a) \in B \times C}$$

"Split"

$$\begin{tikzcd} B & B \times C & C \\ & A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"'] \arrow[ur, "g"'] & \end{tikzcd}$$

$$\langle f, g \rangle\ a = (f\ a, g\ a)$$

Product

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$$

Product

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$$

$$\pi_1 : A \times B \to A$$
$$\pi_1\,(a, b) = a$$

Product

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$$

$$\pi_1 : A \times B \to A \qquad \pi_2 : A \times B \to B$$

$$\pi_1 (a, b) = a \qquad \pi_2 (a, b) = b$$

Product

$$
\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"'] \arrow[ur, "g"'] &
\end{tikzcd}
$$

Product

$$\begin{tikzcd} B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\ & A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"'] \arrow[ur, "g"'] & \end{tikzcd}$$

$$\pi_1 \cdot \langle f, g \rangle = f$$

Product

$$
\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"'] \arrow[ur, "g"'] &
\end{tikzcd}
$$

$$\pi_1 \cdot \langle f, g \rangle = f \qquad \pi_2 \cdot \langle f, g \rangle = g$$

Product

$$\langle f, g \rangle$$

Product

$$\langle f, g \rangle$$

f and g in **parallel**

f “split” g

Product

$$\langle f, g \rangle$$

f and g in **parallel**

f "split" g

$$\langle f, g \rangle \; a = (f \; a, g \; a)$$

Cálculo de Programas

Class T02 — 26-Sep

Product

$$\begin{tikzcd} D \\ B \arrow[u, "f"] \end{tikzcd}$$

Product

$$
\begin{tikzcd}
D & & E \\
B \arrow[u, "f"] & & C \arrow[u, "g"']
\end{tikzcd}
$$

Product

$$\begin{tikzcd}
D & & & E \\
B \arrow[u, "f"] & B \times C \arrow[l, "\pi_1"'] \arrow[rr, "\pi_2"'] & & C \arrow[u, "g"']
\end{tikzcd}$$

Product

$$
\begin{tikzcd}
D & & E \\
B \arrow[u, "f"] & B \times C \arrow[l, "\pi_1"] \arrow[ul, "f \cdot \pi_1"'] \arrow[r, "\pi_2"'] & C \arrow[u, "g"']
\end{tikzcd}
$$

Product

$$
\begin{tikzcd}
D & & E \\
B \arrow[u, "f"] & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[ul, "f \cdot \pi_1"'] \arrow[ur, "g \cdot \pi_2"] & C \arrow[u, "g"']
\end{tikzcd}
$$

Product

$$
\begin{tikzcd}
D & D \times E & E \\
B \arrow[u, "f"] & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"'] \arrow[ul] \arrow[ur] \arrow[u, "\langle f \cdot \pi_1, g \cdot \pi_2 \rangle" description] & C \arrow[u, "g"']
\end{tikzcd}
$$

Product

$$\begin{tikzcd}
D & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E \\
B \arrow[u, "f"] & B \times C \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] \arrow[u, "{\langle f \cdot \pi_1, g \cdot \pi_2 \rangle}" description] \arrow[ul] \arrow[ur] & C \arrow[u, "g"']
\end{tikzcd}$$

Product

$$
\begin{tikzcd}
D & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E \\
B \arrow[u, "f"] & B \times C \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] \arrow[u, "f \times g"] \arrow[ul] \arrow[ur] & C \arrow[u, "g"']
\end{tikzcd}
$$

$$f \times g = \langle f \cdot \pi_1, g \cdot \pi_2 \rangle$$

Summing up

$f \cdot g$

Summing up

$f \cdot g$ Sequential composition

Summing up

$f \cdot g$ Sequential composition

$\langle f, g \rangle$

Summing up

$f \cdot g$ Sequential composition

$\langle f, g \rangle$ Parallel composition

Summing up

$f \cdot g$ Sequential composition

$\langle f, g \rangle$ Parallel composition (synchronous)

Summing up

$f \cdot g$ Sequential composition

$\langle f, g \rangle$ Parallel composition (synchronous)

$f \times g$

Summing up

$f \cdot g$	Sequential composition
$\langle f, g \rangle$	Parallel composition (synchronous)
$f \times g$	Parallel composition

Summing up

$f \cdot g$	Sequential composition
$\langle f, g \rangle$	Parallel composition (synchronous)
$f \times g$	Parallel composition (asynchronous)

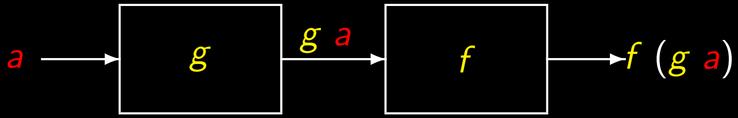
Summing up

$f \cdot g$	Sequential composition
$\langle f, g \rangle$	Parallel composition (synchronous)
$f \times g$	Parallel composition (asynchronous)

Compositional programming

In pictures

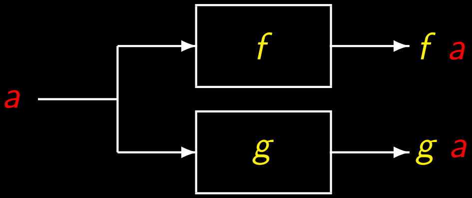
$$(f \cdot g)\ a = f\ (g\ a) \tag{2.6}$$



Function **composition**

In pictures

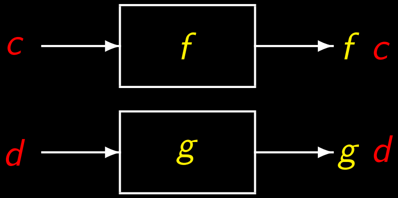
$$\langle f, g \rangle \; a = (f \; a, g \; a) \tag{2.20}$$



Functional "splits"

In pictures

$$f \times g = \langle f \cdot \pi_1, g \cdot \pi_2 \rangle \tag{2.24}$$



Functional **products**

$f \cdot g$	Sequential composition
$\langle f, g \rangle$	Parallel composition (synchronous)
$f \times g$	Parallel composition (asynchronous)

Compositional programming

Problem

Retrieve the address of a civil servant, knowing that she/he can be identified either by a citizen card number (CC) or a fiscal number (NIF).

Problem

Retrieve the address of a civil servant, knowing that she/he can be identified either by a citizen card number (CC) or a fiscal number (NIF).

$$address : Iden \to Address$$
$$Iden = CC \cup NIF$$

Problem!

$$CC = \mathbb{N}$$
$$NIF = \mathbb{N}$$

Problem!

$$CC = \mathbb{N}$$

$$NIF = \mathbb{N}$$

$$Iden = CC \cup NIF = \mathbb{N} \cup \mathbb{N} = \mathbb{N}$$

Problem!

$$CC = \mathbb{N}$$
$$NIF = \mathbb{N}$$

$$Iden = CC \cup NIF = \mathbb{N} \cup \mathbb{N} = \mathbb{N}$$

$$address : \mathbb{N} \to Address \ (!)$$

In general

We need to fix:

$$m : A \cup B \to C$$

In general

We need to fix:

$$m : A \cup B \to C$$

Let us start from

$$A \cup B = \{ a \mid a \in A \} \cup \{ b \mid b \in B \}$$

Disjoint union

In need of something like...

$$\{(1, a) \mid a \in A\} \cup \{(2, b) \mid b \in B\}$$

Disjoint union

In need of something like...

$$\{(1, a) \mid a \in A\} \cup \{(2, b) \mid b \in B\}$$

... we define:

$$A + B = \{(1, a) \mid a \in A\} \cup \{(2, b) \mid b \in B\}$$

Disjoint union

Clearly,

$$A + B = \{\, i_1\ a \mid a \in A \,\} \cup \{\, i_2\ b \mid b \in B \,\}$$

once we define:

$$i_1\ a = (1, a)$$
$$i_2\ b = (2, b)$$

Disjoint union

Types:

$$i_1 : A \rightarrow A + B$$

$$i_2 : B \rightarrow A + B$$

Disjoint union

Types:

$$i_1 : A \to A + B$$
$$i_2 : B \to A + B$$

Now think of:

$$m : A + B \to C$$

Disjoint union

Types:

$$i_1 : A \to A + B$$

$$i_2 : B \to A + B$$

Now think of:

$$m : A + B \to C$$

$$\begin{tikzcd} A + B \arrow[d, "m"] \\ C \end{tikzcd}$$

Disjoint union

Types:

$$i_1 : A \to A + B$$

$$i_2 : B \to A + B$$

Now think of:

$$m : A + B \to C$$

$$\begin{tikzcd} A \arrow[r, "i_1"] & A + B \arrow[d, "m"] \\ & C \end{tikzcd}$$

Disjoint union

Types:

$$i_1 : A \to A + B$$
$$i_2 : B \to A + B$$

Now think of:

$$m : A + B \to C$$

$$\begin{tikzcd} A \arrow[r, "i_1"] & A + B \arrow[d, "m"] & B \arrow[l, "i_2"'] \\ & C & \end{tikzcd}$$

Disjoint union

Types:

$$i_1 : A \to A + B$$
$$i_2 : B \to A + B$$

Now think of:

$$m : A + B \to C$$

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[dr, "f"'] & A + B \arrow[d, "m"] & B \arrow[l, "i_2"'] \\
 & C &
\end{tikzcd}$$

$$f = m \cdot i_1$$

Disjoint union

Types:

$$i_1 : A \to A + B$$
$$i_2 : B \to A + B$$

Now think of:

$$m : A + B \to C$$

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[dr, "f"'] & A + B \arrow[d, "m"] & B \arrow[l, "i_2"'] \arrow[dl, "g"] \\
& C &
\end{tikzcd}$$

$$f = m \cdot i_1 \qquad g = m \cdot i_2$$

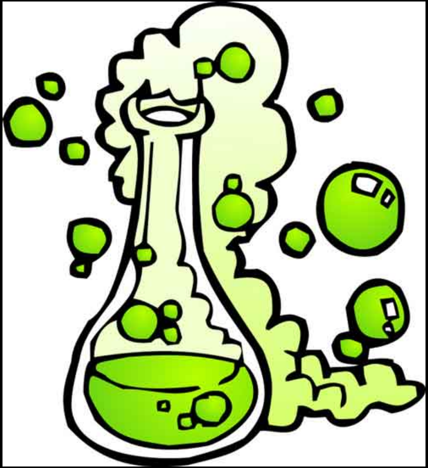
Compare...

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[dr, "f"'] & A + B \arrow[d, "{[f, g]}"] & B \arrow[l, "i_2"'] \arrow[dl, "g"] \\
& C &
\end{tikzcd}$$

Compare...

$$\begin{tikzcd}
A & A \times B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \\
& C \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"] \arrow[ur, "g"'] &
\end{tikzcd}$$

Calculus?



$$
\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"] \arrow[ur, "g"'] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle" description] \arrow[ur, "g"'] & \\
& D \arrow[u, "h"] &
\end{tikzcd}
$$

$$\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& D \arrow[u, "\langle f, g \rangle \cdot h"] &
\end{tikzcd}$$

$$
\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"] \arrow[ur, "g"'] & \\
& D \arrow[u, "h"] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"] \arrow[ur, "g"'] & \\
& D \arrow[u, "h"] \arrow[uul, "f \cdot h"] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
 & A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"] \arrow[ur, "g"'] & \\
 & D \arrow[u, "h"] \arrow[uul, "f \cdot h"] \arrow[uur, "g \cdot h"'] &
\end{tikzcd}
$$

$$\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& D \arrow[ul, "f \cdot h"] \arrow[u, "\langle f \cdot h, g \cdot h \rangle" description] \arrow[ur, "g \cdot h"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& D \arrow[u, "\langle f, g \rangle \cdot h = \langle f \cdot h, g \cdot h \rangle" description] &
\end{tikzcd}$$

$\times$-Fusion

$$\langle f, g \rangle \cdot h = \langle f \cdot h, g \cdot h \rangle \tag{2.26}$$

$$\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& A \arrow[ul, "f"] \arrow[u, "\langle f,g \rangle"] \arrow[ur, "g"'] & \\
& D \arrow[u, "h"] \arrow[uul, "f \cdot h"] \arrow[uur, "g \cdot h"'] &
\end{tikzcd}$$

$$
\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle" description] \arrow[ur, "g"'] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
D & & \\
B \arrow[u, "h"] & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"] \arrow[ur, "g"'] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
D & & E \\
B \arrow[u, "h"] & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \arrow[u, "k"'] \\
& A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"] \arrow[ur, "g"'] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
D & D \times E & E \\
B \arrow[u, "h"] & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[u, "h \times k"] & C \arrow[u, "k"'] \\
& A \arrow[ul, "f"] \arrow[u, "{\langle f, g \rangle}"] \arrow[ur, "g"'] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
D & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E \\
B \arrow[u, "h"] & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[u, "h \times k"] & C \arrow[u, "k"'] \\
& A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"] \arrow[ur, "g"'] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
D & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E \\
B \arrow[u, "h"] & & C \arrow[u, "k"'] \\
& A \arrow[ul, "f"] \arrow[ur, "g"'] \arrow[uu, "(h \times k) \cdot \langle f, g \rangle" description] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
D & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E \\
B \arrow[u, "h"] & & C \arrow[u, "k"'] \\
& A \arrow[ul, "f"] \arrow[ur, "g"'] \arrow[uul, "h \cdot f"'] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
D & & D \times E \arrow[ll, "\pi_1"'] \arrow[rr, "\pi_2"] & & E \\
B \arrow[u, "h"] & & & & C \arrow[u, "k"'] \\
& & A \arrow[ull, "f"] \arrow[urr, "g"'] \arrow[uull, "h \cdot f"'] \arrow[uurr, "k \cdot g"] & &
\end{tikzcd}
$$

$$\begin{tikzcd}
D & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E \\
B \arrow[u, "h"] & & C \arrow[u, "k"'] \\
& A \arrow[ul, "f"] \arrow[ur, "g"'] \arrow[uu, "{\langle h \cdot f, k \cdot g \rangle}" description] &
\end{tikzcd}$$

$$
\begin{tikzcd}
D & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E \\
B \arrow[u, "h"] & & C \arrow[u, "k"'] \\
& A \arrow[ul, "f"] \arrow[ur, "g"'] \arrow[uu, "(h \times k) \cdot \langle f, g \rangle" description] &
\end{tikzcd}
$$

$\times$-Absorption

$$(h \times k) \cdot \langle f, g \rangle = \langle h \cdot f, k \cdot g \rangle \tag{2.27}$$

$$\begin{tikzcd}
A & A \times B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \\
D \arrow[u, "h"] & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[u, "h \times k"'] & E \arrow[u, "k"'] \\
& C \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"] \arrow[ur, "g"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
D & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E \\
B \arrow[u, "h"] & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[u, "h \times k"] & C \arrow[u, "k"'] \\
& A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"] \arrow[ur, "g"'] &
\end{tikzcd}$$

$$
\begin{tikzcd}
D & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E \\
B \arrow[u, "h"] & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[u, "h \times k"] & C \arrow[u, "k"']
\end{tikzcd}
$$

$$
\begin{tikzcd}
D & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E \\
B \arrow[u, "h"] & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[u, "h \times k"] & C \arrow[u, "k"']
\end{tikzcd}
$$

$$\pi_1 \cdot (h \times k) = h \cdot \pi_1$$

$$
\begin{tikzcd}
D & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E \\
B \arrow[u, "h"] & B \times C \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] \arrow[u, "h \times k"] & C \arrow[u, "k"']
\end{tikzcd}
$$

$$\pi_1 \cdot (h \times k) = h \cdot \pi_1 \qquad\qquad \pi_2 \cdot (h \times k) = k \cdot \pi_2$$

Natural-π_1, natural-π_2

$$\pi_1 \cdot (h \times k) = h \cdot \pi_1 \tag{2.28}$$

$$\pi_2 \cdot (h \times k) = k \cdot \pi_2 \tag{2.29}$$

$$\begin{tikzcd}
D & D \times E \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E \\
B \arrow[u, "h"] & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[u, "h \times k"'] & C \arrow[u, "k"']
\end{tikzcd}$$

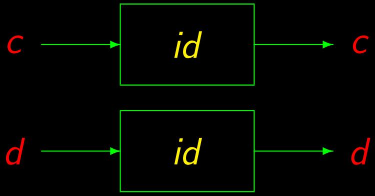
So far...

$f \cdot g$	Sequential composition
$\langle f, g \rangle$	Parallel composition
$f \times g$	Product composition
$[f, g]$	Alternative composition

Compositional programming 

Functor-id-$\times$

$$id \times id = id \tag{2.31}$$



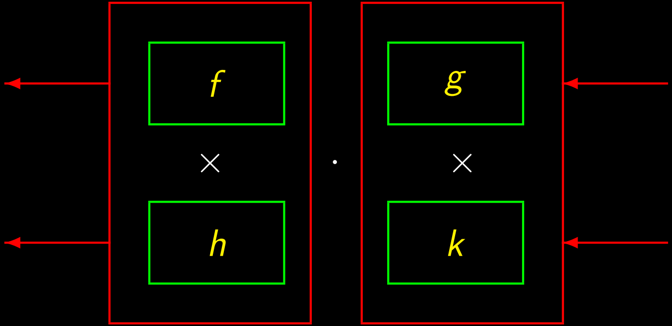
Product of two identities is an identity.

$\times$-Functor

$$(f \times h) \cdot (g \times k) = (f \cdot g) \times (h \cdot k) \tag{2.30}$$

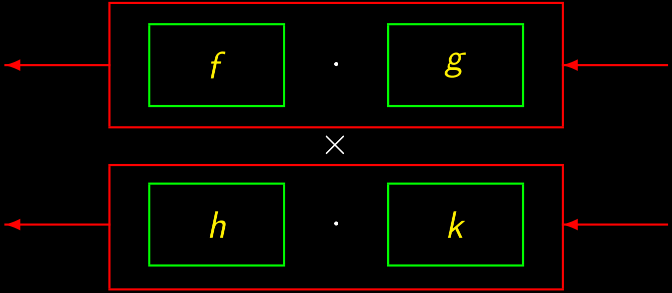
Composition of products is a **product** of compositions.

$\times$-Functor



$$(f \times h) \cdot (g \times k)$$

$\times$-Functor



$$(f \cdot g) \times (h \cdot k)$$

Two basic laws still missing

$\times$-**Reflexion** $$\langle \pi_1, \pi_2 \rangle = id \tag{2.32}$$

Two basic laws still missing

$\times$-Reflexion
$$\langle \pi_1, \pi_2 \rangle = id \tag{2.32}$$

$\times$-Eq
$$\langle i, j \rangle = \langle f, g \rangle \;\Leftrightarrow\; \left\{ \begin{array}{l} i = f \\ j = g \end{array} \right. \tag{2.64}$$

Before them, the most important...

Recall $\times$-**cancellation**:

$$\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& A \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"'] \arrow[ur, "g"'] &
\end{tikzcd}$$

$$\pi_1 \cdot \langle f, g \rangle = f \qquad \pi_2 \cdot \langle f, g \rangle = g$$

$$\begin{cases} \pi_1 \cdot \langle f, g \rangle = f \\ \pi_2 \cdot \langle f, g \rangle = g \end{cases}$$

$$\begin{cases} \pi_1 \cdot \langle f, g \rangle = f \\ \pi_2 \cdot \langle f, g \rangle = g \end{cases}$$

$$k = \langle f, g \rangle \Rightarrow \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases}$$

$$\begin{cases} \pi_1 \cdot \langle f, g \rangle = f \\ \pi_2 \cdot \langle f, g \rangle = g \end{cases}$$

$$k = \langle f, g \rangle \Leftarrow \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases}$$

$$\begin{cases} \pi_1 \cdot \langle f, g \rangle = f \\ \pi_2 \cdot \langle f, g \rangle = g \end{cases}$$

$$k = \langle f, g \rangle \Leftrightarrow \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases}$$

$\times$-Universal

$$k = \langle f , g \rangle \Leftrightarrow \left\{ \begin{array}{l} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{array} \right.$$

$\times$-Universal

Existence

$$k = \langle f, g \rangle \;\Rightarrow\; \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases}$$

"There exists a solution — $k = \langle f, g \rangle$ — for the equations on the right"

$\times$-Universal

Unicity

$$k = \langle f, g \rangle \;\Leftarrow\; \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases}$$

"Such a solution, $k = \langle f, g \rangle$, is unique"

Equations!

$$\begin{cases} x = 2\,y \\ z = \frac{y}{3} \\ x + y + z = 10 \end{cases}$$

Equations!

$$\begin{cases} x = 2\,y \\ z = \frac{y}{3} \\ x + y + z = 10 \end{cases} \Leftrightarrow \begin{cases} x = 6 \\ z = 1 \\ y = 3 \end{cases}$$

Equations!

Problem

Solve the equation

$$\langle f, g \rangle = id$$

for f *and* g.

Equations!

Calculation

In $$k = \langle f, g \rangle \Leftrightarrow \left\{ \begin{array}{l} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{array} \right.$$

Equations!

Calculation

In $\quad k = \langle f, g \rangle \Leftrightarrow \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases} \quad$ let $k = id$

Equations!

Calculation

In $\quad k = \langle f, g \rangle \Leftrightarrow \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases} \qquad$ let $k = id$

$$id = \langle f, g \rangle \Leftrightarrow \begin{cases} \pi_1 \cdot id = f \\ \pi_2 \cdot id = g \end{cases}$$

Equations!

Calculation

In $\quad k = \langle f, g \rangle \Leftrightarrow \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases} \quad$ let $k = id$

$$id = \langle f, g \rangle \Leftrightarrow \begin{cases} \pi_1 = f \\ \pi_2 = g \end{cases}$$

Equations!

$$id = \langle f, g \rangle \iff \begin{cases} \pi_1 = f \\ \pi_2 = g \end{cases}$$

Equations!

$$id = \langle f, g \rangle \iff \begin{cases} \pi_1 = f \\ \pi_2 = g \end{cases}$$

Substituting:

$$id = \langle \pi_1, \pi_2 \rangle$$

Equations!

$$id = \langle f, g \rangle \;\Leftrightarrow\; \begin{cases} \pi_1 = f \\ \pi_2 = g \end{cases}$$

Substituting:

$$id = \langle \pi_1, \pi_2 \rangle$$

$\times$-Reflexion

Problem

Solve the equation

$$\langle h, k \rangle = \langle f, g \rangle$$

Problem

Solve the equation

$$\langle h, k \rangle = \langle f, g \rangle$$

(1 equation, 4 unknowns)

Calculation

$$\langle h, k \rangle = \langle f, g \rangle$$

Calculation

$$\langle h, k \rangle = \langle f, g \rangle$$

$$\Leftrightarrow \quad \{ \ \times\text{-universal} \ \}$$

$$\begin{cases} \pi_1 \cdot \langle h, k \rangle = f \\ \pi_2 \cdot \langle h, k \rangle = g \end{cases}$$

Calculation

$$
\begin{aligned}
& \langle h, k \rangle = \langle f, g \rangle \\
\Leftrightarrow \quad & \{ \ \times\text{-universal} \ \} \\
& \begin{cases} \pi_1 \cdot \langle h, k \rangle = f \\ \pi_2 \cdot \langle h, k \rangle = g \end{cases} \\
\Leftrightarrow \quad & \{ \ \times\text{-cancellation} \ \} \\
& \begin{cases} h = f \\ k = g \end{cases}
\end{aligned}
$$

Calculation

$$
\begin{aligned}
& \langle h, k \rangle = \langle f, g \rangle \\
\Leftrightarrow \quad & \{ \ \times\text{-universal} \ \} \\
& \begin{cases} \pi_1 \cdot \langle h, k \rangle = f \\ \pi_2 \cdot \langle h, k \rangle = g \end{cases} \\
\Leftrightarrow \quad & \{ \ \times\text{-cancellation} \ \} \\
& \begin{cases} h = f \\ k = g \end{cases}
\end{aligned}
$$

$\times$-Eq !

$$\langle \pi_1, \pi_2 \rangle = \mathit{id}$$

$\langle \pi_1, \pi_2 \rangle = \mathit{id}$

$\langle \pi_2, \pi_1 \rangle$?

$\langle \pi_1, \pi_2 \rangle = id$

$\langle \pi_2, \pi_1 \rangle$?

$$\begin{tikzcd} \cdot \\ \cdot \arrow[u, "\langle \pi_2, \pi_1 \rangle" description] \end{tikzcd}$$

$\langle \pi_1, \pi_2 \rangle = id$ $\langle \pi_2, \pi_1 \rangle$?

$$
\begin{tikzcd}
B & \\
& A \times B \arrow[ul, "\pi_2"] \arrow[u, "{\langle \pi_2, \pi_1 \rangle}"']
\end{tikzcd}
$$

$\langle \pi_1, \pi_2 \rangle = id$ $\langle \pi_2, \pi_1 \rangle$?

$$\begin{tikzcd} B & B \times A & A \\ & A \times B \arrow[ul, "\pi_2"] \arrow[u, "\langle \pi_2, \pi_1 \rangle"] \arrow[ur, "\pi_1"'] & \end{tikzcd}$$

$\langle \pi_1, \pi_2 \rangle = id$

$\langle \pi_2, \pi_1 \rangle$?

$$\begin{tikzcd}
B & B \times A & A \\
& A \times B \arrow[ul, "\pi_2"] \arrow[u, "\langle \pi_2, \pi_1 \rangle"] \arrow[ur, "\pi_1"'] &
\end{tikzcd}$$

$\langle \pi_1, \pi_2 \rangle = id$

$\langle \pi_2, \pi_1 \rangle$?

$$
\begin{tikzcd}
B & B \times A \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & A \\
& A \times B \arrow[ul, "\pi_2"] \arrow[u, "\langle \pi_2, \pi_1 \rangle"] \arrow[ur, "\pi_1"'] &
\end{tikzcd}
$$

Problem

Solve

$$\langle \pi_2, \pi_1 \rangle \cdot k = id$$

for k

Problem

Solve

$$\langle \pi_2, \pi_1 \rangle \cdot k = id$$

for k

$$\begin{tikzcd} A \times B \arrow[rr, bend left, "\langle \pi_2, \pi_1 \rangle"] & & B \times A \arrow[ll, bend left, "k"] \end{tikzcd}$$

Calculation

$$\langle \pi_2, \pi_1 \rangle \cdot k = id$$

Calculation

$$
\begin{aligned}
& \langle \pi_2, \pi_1 \rangle \cdot k = id \\
\Leftrightarrow \quad & \{ \ \times\text{-fusion} \ \} \\
& \langle \pi_2 \cdot k, \pi_1 \cdot k \rangle = id
\end{aligned}
$$

Calculation

$$\langle \pi_2, \pi_1 \rangle \cdot k = id$$

$$\Leftrightarrow \quad \{ \times\text{-fusion} \}$$

$$\langle \pi_2 \cdot k, \pi_1 \cdot k \rangle = id$$

$$\Leftrightarrow \quad \{ \times\text{-universal} \}$$

$$\begin{cases} \pi_2 \cdot k = \pi_1 \\ \pi_1 \cdot k = \pi_2 \end{cases}$$

Calculation

$$\langle \pi_2, \pi_1 \rangle \cdot k = id$$

$$\Leftrightarrow \quad \{ \ \times\text{-fusion} \ \}$$

$$\langle \pi_2 \cdot k, \pi_1 \cdot k \rangle = id$$

$$\Leftrightarrow \quad \{ \ \times\text{-universal} \ \}$$

$$\begin{cases} \pi_2 \cdot k = \pi_1 \\ \pi_1 \cdot k = \pi_2 \end{cases}$$

$$\Leftrightarrow \quad \{ \ \text{trivial} \ \}$$

$$\begin{cases} \pi_1 \cdot k = \pi_2 \\ \pi_2 \cdot k = \pi_1 \end{cases}$$

Calculation

$$\langle \pi_2, \pi_1 \rangle \cdot k = id$$

$$\Leftrightarrow \quad \{ \ \times\text{-fusion} \ \}$$

$$\langle \pi_2 \cdot k, \pi_1 \cdot k \rangle = id$$

$$\Leftrightarrow \quad \{ \ \times\text{-universal} \ \}$$

$$\begin{cases} \pi_2 \cdot k = \pi_1 \\ \pi_1 \cdot k = \pi_2 \end{cases}$$

$$\Leftrightarrow \quad \{ \ \text{trivial} \ \}$$

$$\begin{cases} \pi_1 \cdot k = \pi_2 \\ \pi_2 \cdot k = \pi_1 \end{cases}$$

$$\Leftrightarrow \quad \{ \ \times\text{-universal} \ \}$$

$$k = \langle \pi_2, \pi_1 \rangle$$

Swap

$$\begin{tikzcd} A \times B \arrow[rr, bend left, "\langle \pi_2, \pi_1 \rangle"] & & B \times A \arrow[ll, bend left, "\langle \pi_2, \pi_1 \rangle"] \end{tikzcd}$$

$$swap = \langle \pi_2, \pi_1 \rangle$$

Swap

$$\begin{tikzcd} A \times B \arrow[rr, bend left, "\langle \pi_2, \pi_1 \rangle"] & & B \times A \arrow[ll, bend left, "\langle \pi_2, \pi_1 \rangle"] \end{tikzcd}$$

$$swap = \langle \pi_2, \pi_1 \rangle$$

$$swap \cdot swap = id$$

So far

$f \cdot g$ Sequential composition

So far

$f \cdot g$ Sequential composition

$\langle f, g \rangle$ Parallel composition

So far

$f \cdot g$ Sequential composition

$\langle f, g \rangle$ Parallel composition

Associativity

$$(f \cdot g) \cdot h = f \cdot (g \cdot h)$$

So far

$f \cdot g$ Sequential composition

$\langle f, g \rangle$ Parallel composition

Associativity

$$(f \cdot g) \cdot h = f \cdot (g \cdot h)$$

Associativity?

$$\langle \langle f, g \rangle, h \rangle \stackrel{?}{=} \langle f, \langle g, h \rangle \rangle$$

So far

$f \cdot g$ Sequential composition

$\langle f, g \rangle$ Parallel composition

Associativity

$$(f \cdot g) \cdot h = f \cdot (g \cdot h)$$

Associativity?

$$\langle \langle f, g \rangle, h \rangle \stackrel{?}{=} \langle f, \langle g, h \rangle \rangle$$

No! but...

$\langle\langle f, g\rangle, h\rangle$

$$\begin{tikzcd}
A & \\
& D \arrow[ul, "f"]
\end{tikzcd}$$

$\langle\langle f, g\rangle, h\rangle$

$$\begin{tikzcd} A & & B \\ & D \arrow[ul, "f"] \arrow[ur, "g"'] & \end{tikzcd}$$

$\langle\langle f, g\rangle, h\rangle$

$$\begin{tikzcd} & D \arrow[ul, "f"'] \arrow[d, "\langle f, g\rangle"] \arrow[ur, "g"] & \\ A & A \times B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \end{tikzcd}$$

$\langle\langle f, g\rangle, h\rangle$

$$
\begin{tikzcd}
A \times B & & C \\
& D \arrow[ul, "{\langle f, g\rangle}"] \arrow[ur, "h"'] &
\end{tikzcd}
$$

$\langle\langle f, g\rangle, h\rangle$

$$\begin{tikzcd}
A \times B & {} & C \\
& D \arrow[ul, "{\langle f, g\rangle}"'] \arrow[u, "{\langle\langle f, g\rangle, h\rangle}" description] \arrow[ur, "h"'] &
\end{tikzcd}$$

$\langle\langle f, g\rangle, h\rangle$

$$\begin{tikzcd} A \times B & (A \times B) \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\ & D \arrow[ul, "\langle f, g\rangle"] \arrow[u, "\langle\langle f, g\rangle, h\rangle"] \arrow[ur, "h"'] & \end{tikzcd}$$

$\langle f, \langle g, h \rangle \rangle$

$$\begin{tikzcd} B & & C \\ & D \arrow[ul, "g"] \arrow[ur, "h"'] & \end{tikzcd}$$

$\langle f, \langle g, h \rangle \rangle$

$$\begin{tikzcd}
B & B \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& D \arrow[ul, "g"] \arrow[u, "\langle g, h \rangle"] \arrow[ur, "h"'] &
\end{tikzcd}$$

$\langle f, \langle g, h \rangle \rangle$

$$\begin{tikzcd} A & & B \times C \\ & D \arrow[ul, "f"] \arrow[ur, "\langle g, h \rangle"'] & \end{tikzcd}$$

$\langle f, \langle g, h \rangle \rangle$

$$
\begin{tikzcd}
A & {} & B \times C \\
& D \arrow[ul, "f"] \arrow[u, "{\langle f, \langle g, h \rangle \rangle}" description] \arrow[ur, "{\langle g, h \rangle}"'] &
\end{tikzcd}
$$

$\langle f, \langle g, h \rangle \rangle$

$$
\begin{tikzcd}
A & A \times (B \times C) \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \times C \\
& D \arrow[ul, "f"] \arrow[u, "{\langle f, \langle g, h \rangle \rangle}"] \arrow[ur, "{\langle g, h \rangle}"'] &
\end{tikzcd}
$$

$\langle\langle f, g\rangle, h\rangle$

$$\begin{tikzcd}
A \times B & (A \times B) \times C \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & C \\
& D \arrow[ul, "{\langle f, g\rangle}"] \arrow[u, "{\langle\langle f, g\rangle, h\rangle}"] \arrow[ur, "h"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
(A \times B) \times C \arrow[rr, bend left, "k"] & & A \times (B \times C) \\
& D \arrow[ul, "\langle\langle f,g\rangle,h\rangle"] \arrow[ur, "\langle f,\langle g,h\rangle\rangle"'] &
\end{tikzcd}$$

$$
\begin{tikzcd}
(A \times B) \times C \arrow[rr, bend left, "k"] & & A \times (B \times C) \\
& D \arrow[ul, "\langle\langle f, g\rangle, h\rangle"] \arrow[ur, "\langle f, \langle g, h\rangle\rangle"'] &
\end{tikzcd}
$$

$$k \cdot \langle\langle f, g\rangle, h\rangle = \langle f, \langle g, h\rangle\rangle$$

$$k \cdot \langle\langle f, g\rangle, h\rangle = \langle f, \langle g, h\rangle\rangle$$

$$k \cdot \langle \langle f, g \rangle, h \rangle = \langle f, \langle g, h \rangle \rangle$$

$$k \cdot \underbrace{\langle \langle f, g \rangle, h \rangle}_{id?} = \langle f, \langle g, h \rangle \rangle$$

$$k \cdot \langle \langle f, g \rangle, h \rangle = \langle f, \langle g, h \rangle \rangle$$

$$k \cdot \underbrace{\langle \langle f, g \rangle, h \rangle}_{id?} = \langle f, \langle g, h \rangle \rangle$$

$$\text{Solve } \langle \langle f, g \rangle, h \rangle = id$$

$$\langle \langle f, g \rangle, h \rangle = id$$

$$\langle\langle f, g\rangle, h\rangle = id$$

$$\Leftrightarrow \quad \{\ \times\text{-universal}\ \}$$

$$\begin{cases} \pi_1 = \langle f, g\rangle \\ \pi_2 = h \end{cases}$$

$$\langle\langle f, g\rangle, h\rangle = id$$

$$\Leftrightarrow \quad \{\ \times\text{-universal}\ \}$$

$$\begin{cases} \pi_1 = \langle f, g\rangle \\ \pi_2 = h \end{cases}$$

$$\Leftrightarrow \quad \{\ \times\text{-universal}\ \}$$

$$\begin{cases} \pi_1 \cdot \pi_1 = f \\ \pi_2 \cdot \pi_1 = g \\ \pi_2 = h \end{cases}$$

Substitute solutions

$$\begin{cases} \pi_1 \cdot \pi_1 = f \\ \pi_2 \cdot \pi_1 = g \\ \pi_2 = h \end{cases}$$

Substitute solutions

$$\begin{cases} \pi_1 \cdot \pi_1 = f \\ \pi_2 \cdot \pi_1 = g \\ \pi_2 = h \end{cases}$$

$$k \cdot \underbrace{\langle \langle f, g \rangle, h \rangle}_{id} = \langle f, \langle g, h \rangle \rangle$$

Substitute solutions

$$\begin{cases} \pi_1 \cdot \pi_1 = f \\ \pi_2 \cdot \pi_1 = g \\ \pi_2 = h \end{cases}$$

$$k = \langle \pi_1 \cdot \pi_1, \langle \pi_2 \cdot \pi_1, \pi_2 \rangle \rangle$$

Slight improvement...

$$k = \langle \pi_1 \cdot \pi_1, \langle \pi_2 \cdot \pi_1, \pi_2 \rangle \rangle$$

Slight improvement...

$$
\begin{array}{ll}
 & k = \langle \pi_1 \cdot \pi_1, \langle \pi_2 \cdot \pi_1, \pi_2 \rangle \rangle \\
\Leftrightarrow & \quad \{ \; \pi_2 = id \cdot \pi_2 \; \} \\
 & k = \langle \pi_1 \cdot \pi_1, \langle \pi_2 \cdot \pi_1, id \cdot \pi_2 \rangle \rangle
\end{array}
$$

Slight improvement...

$$k = \langle \pi_1 \cdot \pi_1, \langle \pi_2 \cdot \pi_1, \pi_2 \rangle\rangle$$

$$\Leftrightarrow \quad \{\ \pi_2 = id \cdot \pi_2\ \}$$

$$k = \langle \pi_1 \cdot \pi_1, \langle \pi_2 \cdot \pi_1, id \cdot \pi_2 \rangle\rangle$$

$$\Leftrightarrow \quad \{\ f \times g = \langle f \cdot \pi_1, g \cdot \pi_2 \rangle\ \}$$

$$k = \langle \pi_1 \cdot \pi_1, \pi_2 \times id \rangle$$

Analogy

In an **arithmetic** context, find k such that

$$k + (x - y) = x + 2$$

holds for any x and y.

Further assume that you **don't know** property:

$$a + b = c \Leftrightarrow a = c - b.$$

How can you find k?

Analogy

Try to cancel $x - y$, ie. solve $x - y = 0$ for x.

You get $x = y$.

Substitute x:

$$k + (y - y) = y + 2$$

You get $k = y + 2$.

$$
\begin{tikzcd}
(A \times B) \times C \arrow[rr, bend left, "\langle \pi_1 \cdot \pi_1, \pi_2 \times id \rangle"] & & A \times (B \times C) \\
& D \arrow[ul, "\langle \langle f, g \rangle, h \rangle"] \arrow[ur, "\langle f, \langle g, h \rangle \rangle"'] &
\end{tikzcd}
$$

Back to

$$
\begin{tikzcd}
(A \times B) \times C \arrow[rr, bend left, "\langle \pi_1 \cdot \pi_1, \pi_2 \times id \rangle"] & & A \times (B \times C) \arrow[ll, bend left, "j"] \\
& D \arrow[ul, "\langle \langle f, g \rangle, h \rangle"] \arrow[ur, "\langle f, \langle g, h \rangle \rangle"'] &
\end{tikzcd}
$$

$$\begin{tikzcd}[column sep=huge]
(A \times B) \times C \arrow[r, bend left, "\langle \pi_1 \cdot \pi_1, \pi_2 \times id \rangle"] & A \times (B \times C) \arrow[l, bend left, "\langle id \times \pi_1, \pi_2 \cdot \pi_2 \rangle"]
\end{tikzcd}$$

$$
\begin{tikzcd}
(A \times B) \times C \arrow[rr, bend left, "assocr"] & & A \times (B \times C) \arrow[ll, bend left, "assocl"]
\end{tikzcd}
$$

$$
\begin{aligned}
assocr &= \langle \pi_1 \cdot \pi_1, \pi_2 \times id \rangle \\
assocl &= \langle id \times \pi_1, \pi_2 \cdot \pi_2 \rangle
\end{aligned}
$$

$$
\begin{tikzcd}
(A \times B) \times C \arrow[r, bend left, "assocr"] & A \times (B \times C) \arrow[l, bend left, "assocl"]
\end{tikzcd}
$$

$$assocr \cdot assocl = id$$

$$\begin{tikzcd}
(A \times B) \times C \arrow[rr, bend left, "assocr"] & & A \times (B \times C) \arrow[ll, bend left, "assocl"]
\end{tikzcd}$$

$$assocr \cdot assocl = id$$

$$assocl \cdot assocr = id$$

Isomorphism

$$\begin{tikzcd}
(A \times B) \times C \arrow[rr, bend left, "assocr"] & \cong & A \times (B \times C) \arrow[ll, bend left, "assocl"]
\end{tikzcd}$$

$$assocr \cdot assocl = id$$

$$assocl \cdot assocr = id$$

Isomorphism

$$
\begin{tikzcd}
A \times B \arrow[r, bend left, "swap"] \arrow[r, phantom, "\cong"] & B \times A \arrow[l, bend left, "swap"]
\end{tikzcd}
$$

$$swap \cdot swap = id$$

Isomorphism

$$\begin{tikzcd} A \arrow[rr, bend left, "f"] \arrow[rr, phantom, "\cong"] & & B \arrow[ll, bend left, "g"] \end{tikzcd}$$

$$f \cdot g = id$$

$$g \cdot f = id$$

Isomorphism

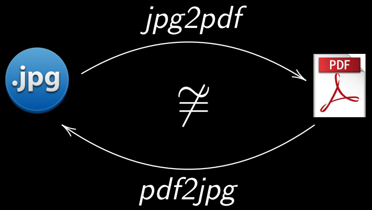
$$\underbrace{\textit{iso}}_{\textit{the same}}\ (\iota\sigma o)$$

Isomorphism

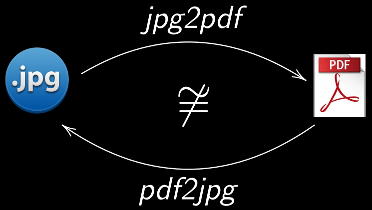
$$\underbrace{\textit{iso}}_{\textit{the same}} \ (\iota\sigma o) + \underbrace{\textit{morphism}}_{\textit{shape}} \ (\mu o\rho\phi\iota\sigma\mu o\zeta)$$

"Similar shape"

Practical problem!



Practical problem!



$$jpg2pdf \cdot pdf2jpg \neq id$$

$$pdf2jpg \cdot jpg2pdf \neq id$$

Cálculo de Programas

Class T03 — 03-Oct

Format conversion

Need

$$\begin{tikzcd} A \arrow[d, "f"'] \\ B \end{tikzcd}$$

Format conversion

Need	Reusable
$A \xrightarrow{f} B$	$C \xrightarrow{r} D$

Format conversion

Need | Reusable

$$\begin{tikzcd} A \arrow[r, "export"] \arrow[d, "f"'] & C \arrow[d, "r"] \\ B & D \end{tikzcd}$$

Format conversion

$$
\begin{tikzcd}
\text{Need} & & \text{Reusable} \\
A \arrow[rr, "export"] \arrow[d, "f"'] & & C \arrow[d, "r"] \\
B & & D \arrow[ll, "import"]
\end{tikzcd}
$$

Format conversion

$$\begin{tikzcd}[column sep=huge, row sep=large] \text{Need} & \text{Reusable} \\ A \arrow[r, "export"] \arrow[d, "f"'] & C \arrow[d, "r"] \\ B & D \arrow[l, "import"] \end{tikzcd}$$

$$f = import \cdot r \cdot export$$

Format conversion

Need   Reusable

$$\begin{tikzcd}
A \arrow[r, "export"] \arrow[d, "f"'] \arrow[dr, phantom, "(\cong)"] & C \arrow[d, "r"] \\
B & D \arrow[l, "import"]
\end{tikzcd}$$

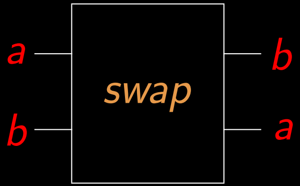
$$f = import \cdot r \cdot export$$

By the way — *swap*

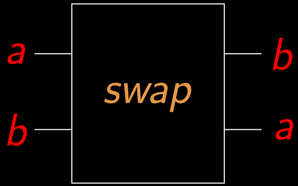
$$\begin{tikzcd} A \times B \arrow[rr, bend left, "swap"] \arrow[rr, phantom, "\cong"] & & B \times A \arrow[ll, bend left, "swap"] \end{tikzcd}$$

Isomorphisms are **reversible** computations

By the way — *swap*

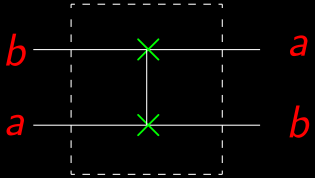


By the way — *swap*



swap is a basic gate in **quantum programming**.

By the way — *swap*



What about alternation $[f, g]$?...

$$[f, g] : A + B \to C$$

$$[f, g]\ x \;=\; \begin{cases} x = i_1\ a \Rightarrow f\ a \\ x = i_2\ b \Rightarrow g\ b \end{cases}$$

Recall...

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[dr, "f"'] & A + B \arrow[d, "{[f, g]}"] & B \arrow[l, "i_2"'] \arrow[dl, "g"] \\
& C &
\end{tikzcd}$$

Compare...

$$
\begin{tikzcd}
A & A \times B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \\
& C \arrow[ul, "f"] \arrow[u, "\langle f, g \rangle"] \arrow[ur, "g"'] &
\end{tikzcd}
$$

$+$-Universal

$$k = [f, g] \Leftrightarrow \begin{cases} k \cdot i_1 = f \\ k \cdot i_2 = g \end{cases}$$

$+$-Universal

$$k = [f, g] \iff \begin{cases} k \cdot i_1 = f \\ k \cdot i_2 = g \end{cases}$$

Compare with

$$k = \langle f, g \rangle \iff \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases}$$

From $[f, g]$ to $f + g$

$$[f, g] : A + B \to C$$

$$[f, g]\, x \;=\; \begin{cases} x = i_1\, a \Rightarrow f\, a \\ x = i_2\, b \Rightarrow g\, b \end{cases}$$

$f + g \qquad ?$

Sum of two functions

$$\begin{tikzcd} D \arrow[d, "f"'] \\ B \end{tikzcd}$$

Sum of two functions

$$\begin{tikzcd} D \arrow[d, "f"'] & & E \arrow[d, "g"] \\ B & & C \end{tikzcd}$$

Sum of two functions

$$\begin{tikzcd}
D \arrow[d, "f"'] & & E \arrow[d, "g"] \\
B \arrow[r, "i_1"'] & B + C & C \arrow[l, "i_2"]
\end{tikzcd}$$

Sum of two functions

$$
\begin{tikzcd}
D \arrow[d, "f"'] \arrow[dr] & & E \arrow[d, "g"] \\
B \arrow[r, "i_1"'] & B + C & C \arrow[l, "i_2"]
\end{tikzcd}
$$

Sum of two functions

$$\begin{tikzcd}
D \arrow[d, "f"'] \arrow[drr] & & & & E \arrow[dll] \arrow[d, "g"] \\
B \arrow[rr, "i_1"'] & & B + C & & C \arrow[ll, "i_2"]
\end{tikzcd}$$

Sum of two functions

$$\begin{tikzcd}
D \arrow[r, "i_1"] \arrow[d, "f"'] \arrow[dr] & D + E \arrow[d, "f + g"] & E \arrow[l, "i_2"'] \arrow[d, "g"] \arrow[dl] \\
B \arrow[r, "i_1"'] & B + C & C \arrow[l, "i_2"]
\end{tikzcd}$$

$$f + g = [i_1 \cdot f\,,\, i_2 \cdot g]$$

Coproduct laws

"Just reverse the arrows", cf.

+-Absorption $$[h, k] \cdot (f + g) = [h \cdot f, k \cdot g] \qquad (2.43)$$

Coproduct laws

"Just reverse the arrows", cf.

+-Absorption $$[h, k] \cdot (f + g) = [h \cdot f, k \cdot g] \qquad (2.43)$$

+-Fusion $$f \cdot [h, k] = [f \cdot h, f \cdot k] \qquad (2.42)$$

Coproduct laws

"Just reverse the arrows", cf.

+-Absorption $$[h, k] \cdot (f + g) = [h \cdot f, k \cdot g] \qquad (2.43)$$

+-Fusion $$f \cdot [h, k] = [f \cdot h, f \cdot k] \qquad (2.42)$$

+-Reflexion $$[i_1, i_2] = id \qquad (2.41)$$

Coproduct laws

$+$-Equality
$$[h, k] = [f, g] \Leftrightarrow \begin{cases} h = f \\ k = g \end{cases} \qquad (2.66)$$

Coproduct laws

+-Equality $$[h, k] = [f, g] \Leftrightarrow \begin{cases} h = f \\ k = g \end{cases} \qquad (2.66)$$

+-Functor $$(h + k) \cdot (f + g) = h \cdot f + k \cdot g \qquad (2.44)$$

Coproduct laws

+-Equality
$$[h, k] = [f, g] \Leftrightarrow \left\{ \begin{array}{l} h = f \\ k = g \end{array} \right. \qquad (2.66)$$

+-Functor
$$(h + k) \cdot (f + g) = h \cdot f + k \cdot g \qquad (2.44)$$

+-Functor-*id*
$$id + id = id \qquad (2.45)$$

and so on

All these laws can be found in the reference sheet

(WWW $\rightarrow$ **Material**)

Summary

$f \cdot g$	Sequential composition
$\langle f, g \rangle$	Parallel composition
$f \times g$	Product composition

Summary

$f \cdot g$	Sequential composition
$\langle f, g \rangle$	Parallel composition
$f \times g$	Product composition
$[f, g]$	Alternative composition

Summary

$f \cdot g$	Sequential composition
$\langle f, g \rangle$	Parallel composition
$f \times g$	Product composition
$[f, g]$	Alternative composition
$f + g$	Structural alternation (coproduct)

Summary

$f \cdot g$	Sequential composition
$\langle f, g \rangle$	Parallel composition
$f \times g$	Product composition
$[f, g]$	Alternative composition
$f + g$	Structural alternation (coproduct)

Compositional programming

What about...

$$f \times (g + h) \stackrel{?}{=} f \times g + f \times h$$

What about...

$$f \times (g + h) \stackrel{?}{=} f \times g + f \times h$$

...?

$$A \times (B + C) \stackrel{?}{\cong} A \times B + A \times C$$

What about...

$$f \times (g + h) \stackrel{?}{=} f \times g + f \times h$$

...?

$$A \times (B + C) \stackrel{?}{\cong} A \times B + A \times C$$

Moreover, any "0" such that

What about...

$$f \times (g + h) \stackrel{?}{=} f \times g + f \times h$$

...?

$$A \times (B + C) \stackrel{?}{\cong} A \times B + A \times C$$

Moreover, any "0" such that

$$A \times 0 = 0 \quad ? \; A + 0 = A \quad ??$$

Recall

$$\begin{cases} id : A \to A \\ id\ a = a \end{cases}$$

$$\begin{tikzcd} A \arrow[rr, bend left, "id"] \arrow[rr, phantom, "\cong"] & & A \arrow[ll, bend left, "id"] \end{tikzcd}$$

Recall

$$\begin{cases} id : A \to A \\ id\ a = a \end{cases}$$

$$\begin{tikzcd} A \arrow[r, bend left, "id"] \arrow[r, phantom, "\cong"] & A \arrow[l, bend left, "id"] \end{tikzcd}$$

$$\begin{tikzcd} A \arrow[r, bend left, "f"] \arrow[r, phantom, "\cong"] & B \arrow[l, bend left, "g"] \end{tikzcd}$$

$$f \cdot g = id$$

$$g \cdot f = id$$

Recall

$$\begin{cases} swap : A \times B \to B \times A \\ swap\,(a, b) = (b, a) \end{cases}$$

Recall

$$\begin{cases} swap : A \times B \to B \times A \\ swap\,(a, b) = (b, a) \end{cases}$$

$$\begin{tikzcd} A \times B \arrow[rr, bend left, "swap"] \arrow[rr, phantom, "\cong"] & & B \times A \arrow[ll, bend left, "swap"] \end{tikzcd}$$

Recall

$$\begin{cases} swap : A \times B \to B \times A \\ swap\ (a, b) = (b, a) \end{cases}$$

$$\begin{tikzcd} A \times B \arrow[rr, bend left, "swap"] \arrow[rr, phantom, "\cong"] & & B \times A \arrow[ll, bend left, "swap"] \end{tikzcd}
\qquad
\begin{tikzcd} A \arrow[rr, bend left, "?"] \arrow[rr, phantom, "?"] & & A \times B \arrow[ll, bend left, "\pi_1"] \end{tikzcd}$$

Recall

$$\begin{cases} swap : A \times B \to B \times A \\ swap\ (a, b) = (b, a) \end{cases}$$

$$\begin{tikzcd} A \times B \arrow[rr, bend left, "swap"] \arrow[rr, phantom, "\cong"] & & B \times A \arrow[ll, bend left, "swap"] \end{tikzcd}$$

$$\begin{tikzcd} A \arrow[rr, bend left, "?"] \arrow[rr, phantom, "?"] & & A \times B \arrow[ll, bend left, "\pi_1"] \end{tikzcd}$$

$$\pi_1 \cdot \langle id, h \rangle = id$$

Recall

$$\begin{cases} swap : A \times B \to B \times A \\ swap\ (a, b) = (b, a) \end{cases}$$

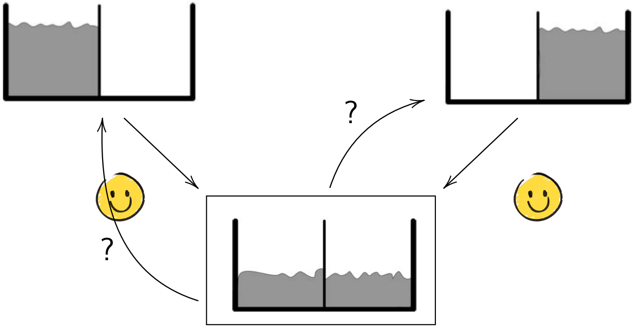
$$\begin{tikzcd} A \times B \arrow[rr, bend left, "swap"] \arrow[rr, phantom, "\cong"] & & B \times A \arrow[ll, bend left, "swap"] \end{tikzcd}$$

$$\begin{tikzcd} A \arrow[rr, bend left, "?"] \arrow[rr, phantom, "?"] & & A \times B \arrow[ll, bend left, "\pi_1"] \end{tikzcd}$$

$$\pi_1 \cdot \langle id, h \rangle = id$$

$$\langle id, h \rangle \cdot \pi_1 \neq id$$

Irreversibility...



Rolf Landauer (1927–99)

Even worse than π_1...

$$\begin{cases} \mathit{zero}\ x = 0 \\ \mathit{one}\ x = 1 \end{cases}$$

Even worse than π_1...

$$\begin{cases} \mathit{zero}\ x = 0 \\ \mathit{one}\ x = 1 \end{cases}$$

$$\mathit{one}\ 10 = 1$$

Even worse than π_1...

$$\begin{cases} \mathit{zero}\ x = 0 \\ \mathit{one}\ x = 1 \end{cases}$$

$$\mathit{one}\ 10 = 1$$

$$\mathit{one}\ \texttt{"string"} = 1$$

Even worse than π_1...

$$\begin{cases} \mathit{zero}\ x = 0 \\ \mathit{one}\ x = 1 \end{cases}$$

$$\mathit{one}\ 10 = 1$$

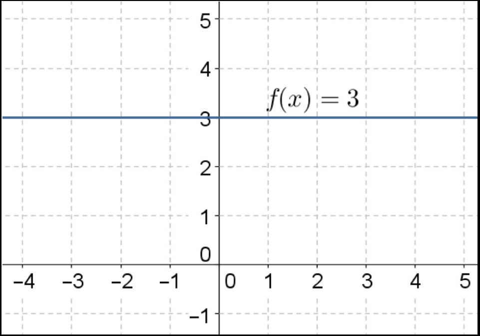
$$\mathit{one}\ \texttt{"string"} = 1$$

$$\mathit{zero}\ [50\,..\,1000] = 0$$

Even worse than π_1...

$$\begin{cases} zero\ x = 0 \\ one\ x = 1 \end{cases}$$

$$f\ x = 3$$



$$one\ 10 = 1$$

$$one\ \texttt{"string"} = 1$$

$$zero\ [50\,..\,1000] = 0$$

Constant functions

$3 \in \mathbb{N}_0$

$f\ x = 3$

Constant functions

$3 \in \mathbb{N}_0$

$f\ x = 3$

$A \xrightarrow{f} \mathbb{N}_0$

Constant functions

$3 \in \mathbb{N}_0$

$b_0 \in B$

$f\ x = 3$

$A \xrightarrow{f} \mathbb{N}_0$

Constant functions

$3 \in \mathbb{N}_0$

$f\ x = 3$

$A \xrightarrow{f} \mathbb{N}_0$

$b_0 \in B$

$$\begin{cases} \underline{b_0} : A \to B \\ \underline{b_0}\ a = b_0 \end{cases}$$

Constant functions

$3 \in \mathbb{N}_0$

$f\ x = 3$

$A \xrightarrow{f} \mathbb{N}_0$

$b_0 \in B$

$$\left\{ \begin{array}{l} \underline{b_0} : A \to B \\ \underline{b_0}\ a = b_0 \end{array} \right.$$

A

Constant functions

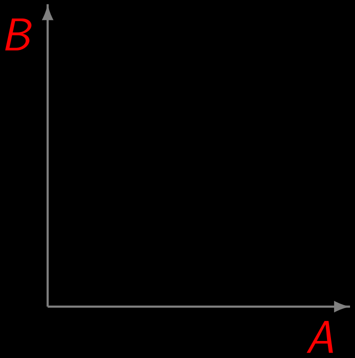
$3 \in \mathbb{N}_0$

$f\ x = 3$

$A \xrightarrow{f} \mathbb{N}_0$

$b_0 \in B$

$$\begin{cases} \underline{b_0} : A \to B \\ \underline{b_0}\ a = b_0 \end{cases}$$



Constant functions

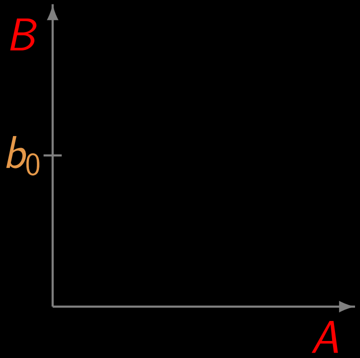
$3 \in \mathbb{N}_0$

$f\ x = 3$

$A \xrightarrow{f} \mathbb{N}_0$

$b_0 \in B$

$$\begin{cases} \underline{b_0} : A \to B \\ \underline{b_0}\ a = b_0 \end{cases}$$



Constant functions

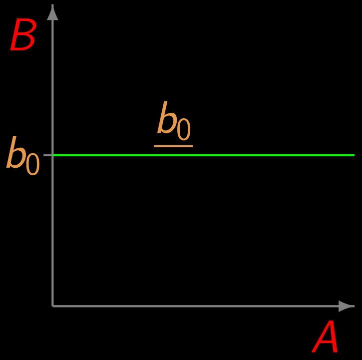
$3 \in \mathbb{N}_0$

$f\ x = 3$

$A \xrightarrow{f} \mathbb{N}_0$

$b_0 \in B$

$$\begin{cases} \underline{b_0} : A \to B \\ \underline{b_0}\ a = b_0 \end{cases}$$



Constant functions

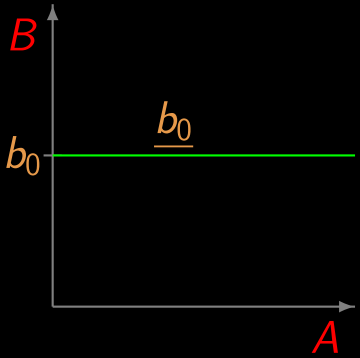
$3 \in \mathbb{N}_0$

$f\ x = 3$

$A \xrightarrow{f} \mathbb{N}_0$

$b_0 \in B$

$$\begin{cases} \underline{b_0} : A \to B \\ \underline{b_0}\ a = b_0 \end{cases}$$



(**Haskell**: `const` b_0)

Properties

$$\underline{b} \cdot f = \underline{b}$$

Properties

$$\underline{b} \cdot f = \underline{b}$$

$$g \cdot \underline{b} = \underline{g\ b}$$

Properties

$$\underline{b} \cdot f = \underline{b}$$

$$g \cdot \underline{b} = \underline{g \; b}$$

$$B \xleftarrow{\underline{b}} A$$

Properties

$$\underline{b} \cdot f = \underline{b}$$

$$g \cdot \underline{b} = \underline{g\ b}$$

$$\begin{tikzcd} & C \arrow[d, "f"] \\ B & A \arrow[l, "\underline{b}"'] \end{tikzcd}$$

Properties

$$\underline{b} \cdot f = \underline{b}$$

$$g \cdot \underline{b} = \underline{g\ b}$$

$$\begin{tikzcd}
& C \arrow[dl, "\underline{b}"'] \arrow[d, "f"] \\
B & A \arrow[l, "\underline{b}"']
\end{tikzcd}$$

Properties

$$\underline{b} \cdot f = \underline{b}$$

$$g \cdot \underline{b} = \underline{g\ b}$$

$$\begin{tikzcd}
 & C \arrow[dl, "\underline{b}"'] \arrow[d, "f"] \\
B \arrow[d, "g"'] & A \arrow[l, "\underline{b}"'] \\
D &
\end{tikzcd}$$

Properties

$$\underline{b} \cdot f = \underline{b}$$

$$g \cdot \underline{b} = \underline{g\ b}$$

$$\begin{tikzcd}
 & C \arrow[dl, "\underline{b}"'] \arrow[d, "f"] \\
B \arrow[d, "g"'] & A \arrow[l, "\underline{b}"'] \arrow[dl, "\underline{g\ b}"] \\
D &
\end{tikzcd}$$

No loss of information?

Isomorphisms shining...

Isomorphisms

$$A \times (B \times C) \cong (A \times B) \times C$$

$$A \times B \cong B \times A$$

Isomorphisms

$$A \times (B \times C) \;\cong\; (A \times B) \times C$$

$$A \times B \;\cong\; B \times A$$

$$A + (B + C) \;\cong\; (A + B) + C$$

$$A + B \;\cong\; B + A$$

Isomorphisms

$$A + A \cong 2 \times A \qquad ?$$

Isomorphisms

$$A + A \cong 2 \times A \qquad ?$$

$$A \times A \cong A^2 \qquad ?$$

Isomorphisms

$$A + A \cong 2 \times A \qquad ?$$

$$A \times A \cong A^2 \qquad ?$$

$$A \times 1 \cong A \qquad ?$$

Isomorphisms

$$A + A \cong 2 \times A \qquad ?$$

$$A \times A \cong A^2 \qquad ?$$

$$A \times 1 \cong A \qquad ?$$

$$A \times 0 \cong 0 \qquad ?$$

Isomorphisms

$$A + A \cong 2 \times A \qquad ?$$

$$A \times A \cong A^2 \qquad ?$$

$$A \times 1 \cong A \qquad ?$$

$$A \times 0 \cong 0 \qquad ?$$

$$A + 0 \cong A \qquad ?$$

Isomorphisms

$$A + A \cong 2 \times A \qquad ?$$

$$A \times A \cong A^2 \qquad ?$$

$$A \times 1 \cong A \qquad ?$$

$$A \times 0 \cong 0 \qquad ?$$

$$A + 0 \cong A \qquad ?$$

$$A \times (B + C) \cong A \times B + A \times C \qquad ?$$

Distributivity

$$\begin{tikzcd} A \times (B + C) \arrow[r, bend left, "distr"] \arrow[r, phantom, "\cong"] & A \times B + A \times C \arrow[l, bend left, "undistr"] \end{tikzcd}$$

Distributivity

$$\begin{tikzcd} A \times (B + C) \arrow[rr, bend left, "distr"] \arrow[r, phantom, "\cong"] & & A \times B + A \times C \arrow[ll, bend left, "undistr"] \end{tikzcd}$$

$$\begin{tikzcd} (B + C) \times A \arrow[rr, bend left, "distl"] \arrow[r, phantom, "\cong"] & & B \times A + C \times A \arrow[ll, bend left, "undistl"] \end{tikzcd}$$

Distributivity

$$\begin{tikzcd} A \times (B + C) \arrow[rr, bend left, "distr"] \arrow[r, phantom, "\cong"] & & A \times B + A \times C \arrow[ll, bend left, "undistr"] \end{tikzcd}$$

Distributivity

$$\begin{tikzcd} A \times (B + C) \arrow[rr, bend left, "distr"] \arrow[r, phantom, "\cong"] & & A \times B + A \times C \arrow[ll, bend left, "undistr"] \end{tikzcd}$$

$$undistr = [\langle f, g \rangle, \langle h, k \rangle]$$

Distributivity

$$\begin{tikzcd} A \times (B + C) \arrow[rr, bend left, "distr"] \arrow[r, phantom, "\cong"] & & A \times B + A \times C \arrow[ll, bend left, "undistr"] \end{tikzcd}$$

$$undistr = [\langle f, g \rangle, \langle h, k \rangle]$$

$$A \xleftarrow{f} A \times B$$

Distributivity

$$
\begin{tikzcd}
A \times (B + C) \arrow[rr, bend left, "distr"] \arrow[r, phantom, "\cong"] & & A \times B + A \times C \arrow[ll, bend left, "undistr"]
\end{tikzcd}
$$

$$undistr = [\langle f, g\rangle, \langle h, k\rangle]$$

$$A \xleftarrow{f} A \times B$$

$$B + C \xleftarrow{g} A \times B$$

Distributivity

$$
\begin{tikzcd}
A \times (B + C) \arrow[rr, bend left, "distr"] \arrow[rr, phantom, "\cong"] & & A \times B + A \times C \arrow[ll, bend left, "undistr"]
\end{tikzcd}
$$

$$undistr = [\langle f, g \rangle, \langle h, k \rangle]$$

$$A \xleftarrow{f} A \times B \qquad\qquad A \xleftarrow{h} A \times C$$

$$B + C \xleftarrow{g} A \times B$$

Distributivity

$$\begin{tikzcd} A \times (B + C) \arrow[rr, bend left, "distr"] \arrow[rr, phantom, "\cong"] & & A \times B + A \times C \arrow[ll, bend left, "undistr"] \end{tikzcd}$$

$$undistr = [\langle f, g \rangle, \langle h, k \rangle]$$

$$A \xleftarrow{f} A \times B \qquad A \xleftarrow{h} A \times C$$

$$B + C \xleftarrow{g} A \times B \qquad B + C \xleftarrow{k} A \times C$$

Distributivity

$$undistr = [\langle f, g \rangle, \langle h, k \rangle]$$

$$A \xleftarrow{f} A \times B \qquad\qquad A \xleftarrow{h} A \times C$$

$$B + C \xleftarrow{g} A \times B \qquad\qquad B + C \xleftarrow{k} A \times C$$

Distributivity

$$undistr = [\langle f, g\rangle, \langle h, k\rangle]$$

$$A \xleftarrow{f} A \times B$$

$$B + C \xleftarrow{g} A \times B$$

$$A \xleftarrow{h} A \times C$$

$$B + C \xleftarrow{k} A \times C$$

$$A \xleftarrow{\pi_1} A \times B$$

Distributivity

$$undistr = [\langle f, g\rangle, \langle h, k\rangle]$$

$$A \xleftarrow{f} A \times B \qquad A \xleftarrow{h} A \times C$$

$$B + C \xleftarrow{g} A \times B \qquad B + C \xleftarrow{k} A \times C$$

$$A \xleftarrow{\pi_1} A \times B$$

$$B + C \xleftarrow{i_1 \cdot \pi_2} A \times B$$

Distributivity

$$undistr = [\langle f, g\rangle, \langle h, k\rangle]$$

$$A \xleftarrow{f} A \times B \qquad A \xleftarrow{h} A \times C$$

$$B + C \xleftarrow{g} A \times B \qquad B + C \xleftarrow{k} A \times C$$

$$A \xleftarrow{\pi_1} A \times B \qquad A \xleftarrow{\pi_1} A \times C$$

$$B + C \xleftarrow{i_1 \cdot \pi_2} A \times B$$

Distributivity

$$undistr = [\langle f, g \rangle, \langle h, k \rangle]$$

$$A \xleftarrow{f} A \times B \qquad\qquad A \xleftarrow{h} A \times C$$

$$B + C \xleftarrow{g} A \times B \qquad\qquad B + C \xleftarrow{k} A \times C$$

$$A \xleftarrow{\pi_1} A \times B \qquad\qquad A \xleftarrow{\pi_1} A \times C$$

$$B + C \xleftarrow{i_1 \cdot \pi_2} A \times B \qquad\qquad B + C \xleftarrow{i_2 \cdot \pi_2} A \times C$$

$$undistr = [\langle \pi_1, i_1 \cdot \pi_2 \rangle, \langle \pi_1, i_2 \cdot \pi_2 \rangle]$$

Distributivity

$$\mathit{undistr} = [\langle \pi_1, i_1 \cdot \pi_2 \rangle, \langle \pi_1, i_2 \cdot \pi_2 \rangle]$$

$$\begin{tikzcd} A \times (B + C) \arrow[rr, bend left, "\mathit{distr}"] \arrow[r, phantom, "\cong"] & {} & A \times B + A \times C \arrow[ll, bend left, "\mathit{undistr}"] \end{tikzcd}$$

Distributivity

$$undistr = [\langle \pi_1, i_1 \cdot \pi_2 \rangle, \langle \pi_1, i_2 \cdot \pi_2 \rangle]$$

$$\begin{tikzcd} A \times (B + C) \arrow[rr, bend left, "distr"] \arrow[r, phantom, "\cong"] & & A \times B + A \times C \arrow[ll, bend left, "undistr"] \end{tikzcd}$$

By definition of $f \times g$

$$undistr = [id \times i_1, id \times i_2]$$

The exchange law

$$\downarrow \alpha$$

The exchange law

$$\begin{tikzcd} A + B \arrow[d, "\alpha"] \\ {} \end{tikzcd}$$

The exchange law

$$\begin{tikzcd} A + B \arrow[d, "\alpha"] \\ C \times D \end{tikzcd}$$

Exchange law

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[dr, "i"'] & A + B \arrow[d, "{[i, j]}"] & B \arrow[l, "i_2"'] \arrow[dl, "j"] \\
& C \times D &
\end{tikzcd}$$

Exchange law

$$\begin{tikzcd}
 & A + B \arrow[dl, "m"'] \arrow[d, "\langle m, n \rangle"] \arrow[dr, "n"] & \\
C & C \times D \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & D
\end{tikzcd}$$

Exchange law

$$\langle m, n \rangle = [i, j]$$

One equation, four unknowns

Exchange law

$$\begin{tikzcd}
 & A + B \arrow[dl, "m"'] \arrow[d, "\langle m, n \rangle"] \arrow[dr, "n"] & \\
C & C \times D \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & D
\end{tikzcd}$$

Exchange law

$$
\begin{tikzcd}
A \arrow[r, "i_1"] & A + B \arrow[dl, "m"'] \arrow[d, "\langle m, n\rangle"] \arrow[dr, "n"] & B \arrow[l, "i_2"'] \\
C & C \times D \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & D
\end{tikzcd}
$$

Exchange law

$$
\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[d, "f"] & A + B & B \arrow[l, "i_2"'] \arrow[dll, "g"'] \\
C & C \times D \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & D
\end{tikzcd}
$$

Exchange law

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[drr, "h"'] & A + B & B \arrow[l, "i_2"'] \arrow[d, "k"] \\
C & C \times D \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & D
\end{tikzcd}$$

Exchange law

$$\begin{tikzcd}[column sep=huge, row sep=huge]
A \arrow[r, "i_1"] & A + B \arrow[dl, "{m = [f, g]}"'] \arrow[d, "{\langle m, n \rangle}"] \arrow[dr, "{n = [h, k]}"] & B \arrow[l, "i_2"'] \\
C & C \times D \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & D
\end{tikzcd}$$

Exchange law

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[dr, "i = \langle f, h \rangle"'] & A + B \arrow[d, "{[i, j]}"] & B \arrow[l, "i_2"'] \arrow[dl, "j = \langle g, k \rangle"] \\
C & C \times D \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & D
\end{tikzcd}$$

Summing up

Exchange law:

$$\langle [f, g], [h, k] \rangle = [\langle f, h \rangle, \langle g, k \rangle]$$

"I have a functional **split** but need an **alternative**..."

... and vice-versa

Exchange law — Example

$$\mathit{undistr} = [\langle \pi_1, i_1 \cdot \pi_2 \rangle, \langle \pi_1, i_2 \cdot \pi_2 \rangle]$$

$$\begin{tikzcd} A \times (B + C) \arrow[rr, bend left, "\mathit{distr}"] \arrow[r, phantom, "\cong"] & {} & A \times B + A \times C \arrow[ll, bend left, "\mathit{undistr}"] \end{tikzcd}$$

Exchange law — Example

$$undistr = [\langle \pi_1, i_1 \cdot \pi_2 \rangle, \langle \pi_1, i_2 \cdot \pi_2 \rangle]$$

$$\begin{tikzcd} A \times (B + C) \arrow[rr, bend left, "distr"] \arrow[r, phantom, "\cong"] & {} & A \times B + A \times C \arrow[ll, bend left, "undistr"] \end{tikzcd}$$

$$undistr = \langle [\pi_1, \pi_1], [i_1 \cdot \pi_2, i_2 \cdot \pi_2] \rangle$$

The types 0, 1 and 2...

The type 1

In Haskell:

$$() \; :: \; ()$$

Then

$$1 = \{()\}$$

Isomorphism

$$\begin{tikzcd} A \times 1 \arrow[rr, bend left, "\pi_1"] \arrow[rr, phantom, "\cong"] & & A \arrow[ll, bend left, "\langle id, \mathbf{!} \rangle"] \end{tikzcd}$$

The type 1

In Haskell:

$$() \; :: \; ()$$

Then

$$1 = \{()\}$$

Isomorphism

$$\begin{tikzcd} A \times 1 \arrow[rr, bend left, "\pi_1"] \arrow[rr, phantom, "\cong"] & & A \arrow[ll, bend left, "\langle id, \mathbf{!} \rangle"] \end{tikzcd}$$

... "$\mathbf{!}$" ?

The type 1

Fact functions involving type 1
are necessarily **constant**

The type 1

Fact functions involving type 1 are necessarily **constant**

"Bang":

$$A \xrightarrow{!} 1$$

$$! = \underline{()}$$

The type 1

Fact functions involving type 1 are necessarily **constant**

"Bang":

$$A \xrightarrow{\;!\;} 1$$

$$! = \underline{()}$$

"Points":

$$1 \xrightarrow{\underline{a}} A$$

$$\underline{a}\,() = a$$

(provided $a \in A$)

The type 0

$$0 = \{\,\}$$

The type 0

$$0 = \{\,\}$$

Immediate in e.g. Haskell:

```
data Zero
```

The type 0

$$0 = \{ \}$$

Immediate in e.g. Haskell:

```
data Zero
```

0 is not inhabited: $\neg\,(a \in 0)$
for all a

The type 0

$$0 = \{\,\}$$

Immediate in e.g. Haskell:

```
data Zero
```

0 is not inhabited: $\neg\,(a \in 0)$ for all a

Fact impossible $f : A \to 0$ whenever $A \neq 0$

The type 0

$$\begin{tikzcd} A \times 0 \arrow[rr, bend left, "id"] & = & 0 \arrow[ll, bend left, "id"] \end{tikzcd}$$

The type 0

$$\begin{tikzcd} A \times 0 \arrow[rr, bend left, "id"] \arrow[rr, phantom, "="] & & 0 \arrow[ll, bend left, "id"] \end{tikzcd}$$

In fact

$$A \times 0 = \{(a, b) \mid a \in A \land b \in 0\}$$

The type 0

$$\begin{tikzcd} A \times 0 \arrow[rr, bend left, "id"] \arrow[rr, phantom, "="] & & 0 \arrow[ll, bend left, "id"] \end{tikzcd}$$

In fact

$$A \times 0 = \{(a, b) \mid a \in A \wedge b \in 0\}$$
$$A \times 0 = \{(a, b) \mid a \in A \wedge F\}$$

The type 0

$$\begin{tikzcd} A \times 0 \arrow[rr, bend left, "id"] \arrow[rr, phantom, "="] & & 0 \arrow[ll, bend left, "id"] \end{tikzcd}$$

In fact

$$A \times 0 = \{(a, b) \mid a \in A \wedge b \in 0\}$$

$$A \times 0 = \{(a, b) \mid a \in A \wedge F\}$$

$$A \times 0 = \{\,\}$$

The type $1 + A$

$$1 + A$$

The type $1 + A$

$$\begin{tikzcd} 1 \arrow[r, "i_1"] & 1 + A & A \arrow[l, "i_2"'] \end{tikzcd}$$

The type $1 + A$

$$\begin{tikzcd} 1 \arrow[r, "i_1"] & 1 + A & A \arrow[l, "i_2"'] \\ & B & \end{tikzcd}$$

The type $1 + A$

$$\begin{tikzcd}
1 \arrow[r, "i_1"] & 1 + A & A \arrow[l, "i_2"'] \arrow[dl, "f"] \\
 & B &
\end{tikzcd}$$

The type $1 + A$

$$
\begin{tikzcd}
1 \arrow[r, "i_1"] \arrow[dr, "\underline{b_0}"'] & 1 + A & A \arrow[l, "i_2"'] \arrow[dl, "f"] \\
& B &
\end{tikzcd}
$$

The type $1 + A$

$$
\begin{tikzcd}
1 \arrow[r, "i_1"] \arrow[dr, "\underline{b_0}"'] & 1 + A \arrow[d, "{[\underline{b_0}, f]}"] & A \arrow[l, "i_2"'] \arrow[dl, "f"] \\
& B &
\end{tikzcd}
$$

The type $1 + A$

$$
\begin{tikzcd}
1 \arrow[r, "i_1"] \arrow[dr, "\underline{b_0}"'] & 1 + A \arrow[d, "{[\underline{b_0}, f]}"] & A \arrow[l, "i_2"'] \arrow[dl, "f"] \\
 & B &
\end{tikzcd}
$$

$1 + A$ ”pointer to A”

More isomorphisms!

More isomorphisms!

Back to

> *Retrieve the address of a civil servant, knowing that she/he can be identified either by a citizen card number (CC) or a fiscal number (NIF).*

More isomorphisms!

Back to

> *Retrieve the address of a civil servant, knowing that she/he can be identified either by a citizen card number (CC) or a fiscal number (NIF).*

In Haskell:

```
data Iden = CC Int | NIF Int
```

More isomorphisms!

> *Obter a morada of a funcionário público, knowing that este se can identificar através d its number of the cartão of cidadão (CC) ou d its number of identificação fiscal (NIF).*

In Haskell (Portugal):

```
data Iden = CC Int | NIF Int
```

In Haskell (Germany):

```
data Iden = IDK Int | SZN Int
```

More isomorphisms!

$$
\begin{tikzcd}
\mathit{Int} \arrow[r, "i_1"] \arrow[dr, "\mathtt{CC}"'] & \mathit{Int} + \mathit{Int} \arrow[d, "{[\mathtt{CC}, \mathtt{NIF}]}"] & \mathit{Int} \arrow[l, "i_2"'] \arrow[dl, "\mathtt{NIF}"] \\
& \mathtt{Iden} &
\end{tikzcd}
$$

```
data Iden = CC Int | NIF Int
```

More isomorphisms!

$$\begin{tikzcd}
\mathtt{Iden} \arrow[rr, bend left, "\mathsf{out}=\mathsf{in}^{\circ}"] \arrow[rr, phantom, "\cong"] & & \mathit{Int} + \mathit{Int} \arrow[ll, bend left, "\mathsf{in}=[\mathtt{CC},\mathtt{NIF}]"]
\end{tikzcd}$$

More isomorphisms!

$$\begin{tikzcd} \mathtt{Iden} \arrow[rr, bend left, "\mathbf{out}=\mathbf{in}^\circ"] \arrow[rr, phantom, "\cong"] & & Int + Int \arrow[ll, bend left, "\mathbf{in}=[\mathtt{CC},\mathtt{NIF}]"] \end{tikzcd}$$

Concrete syntax

More isomorphisms!

$$\begin{tikzcd} \mathtt{Iden} \arrow[rr, bend left, "\mathbf{out}=\mathbf{in}^{\circ}"] \arrow[rr, phantom, "\cong"] & & \mathit{Int} + \mathit{Int} \arrow[ll, bend left, "\mathbf{in}={[\mathtt{CC},\mathtt{NIF}]}"] \end{tikzcd}$$

Concrete syntax

Abstract syntax

More isomorphisms!

$$\begin{tikzcd} \mathtt{Iden} \arrow[rr, bend left, "\mathbf{out}=\mathbf{in}^{\circ}"] \arrow[rr, phantom, "\cong"] & & \mathit{Int} + \mathit{Int} \arrow[ll, bend left, "\mathbf{in}=[\mathtt{CC},\mathtt{NIF}]"] \end{tikzcd}$$

Concrete syntax

Abstract syntax

in ("get in")

More isomorphisms!

$$\begin{tikzcd} \mathtt{Iden} \arrow[rr, bend left, "\mathbf{out} = \mathbf{in}^{\circ}"] \arrow[rr, phantom, "\cong"] & & \mathit{Int} + \mathit{Int} \arrow[ll, bend left, "\mathbf{in} = {[\mathtt{CC}, \mathtt{NIF}]}"] \end{tikzcd}$$

Concrete syntax

Abstract syntax

in ("get in")

out ("get out")

Maybe

$$
\begin{tikzcd}
1 \arrow[r, "i_1"] \arrow[dr, swap, "\underline{\texttt{Nothing}}"] & 1 + A \arrow[d, "{[\underline{\texttt{Nothing}}, \texttt{Just}]}"] & A \arrow[l, swap, "i_2"] \arrow[dl, "\texttt{Just}"] \\
& \texttt{Maybe}\ A &
\end{tikzcd}
$$

```
data Maybe a = Nothing | Just a
```

Maybe

$$\begin{tikzcd} \mathtt{Maybe}\ A \arrow[rr, bend left, "\mathsf{out}=\mathsf{in}^{\circ}"] & \cong & 1 + A \arrow[ll, bend left, "\mathsf{in}\ =\ {[\underline{\mathtt{Nothing}},\mathtt{Just}]}"] \end{tikzcd}$$

Maybe

$$\begin{tikzcd} \mathtt{Maybe}\ A \arrow[rr, bend left, "\mathsf{out}=\mathsf{in}^{\circ}"] \arrow[rr, phantom, "\cong"] & & 1+A \arrow[ll, bend left, "\mathsf{in}\ =\ {[\underline{\mathtt{Nothing}},\mathtt{Just}]}"] \end{tikzcd}$$

$$\mathsf{out} \cdot \mathsf{in} = id$$

Maybe — calculate out

$$\mathsf{out} \cdot \mathsf{in} = id$$

Maybe — calculate out

$$\mathsf{out} \cdot \mathsf{in} = id$$

$$\Leftrightarrow \qquad \{ \ \mathsf{in} = [\underline{\mathtt{Nothing}}, \mathtt{Just}] \ \}$$

$$\mathsf{out} \cdot [\underline{\mathtt{Nothing}}, \mathtt{Just}] = id$$

Maybe — calculate out

$$
\begin{aligned}
& \mathsf{out} \cdot \mathsf{in} = \mathit{id} \\
\Leftrightarrow \quad & \{ \ \mathsf{in} = [\underline{\mathtt{Nothing}}, \mathtt{Just}] \ \} \\
& \mathsf{out} \cdot [\underline{\mathtt{Nothing}}, \mathtt{Just}] = \mathit{id} \\
\Leftrightarrow \quad & \{ \ +\text{-fusion} \ \} \\
& [\mathsf{out} \cdot \underline{\mathtt{Nothing}}, \mathsf{out} \cdot \mathtt{Just}] = \mathit{id}
\end{aligned}
$$

Maybe — calculate out

$$\mathsf{out} \cdot \mathsf{in} = id$$

$$\Leftrightarrow \quad \{ \ \mathsf{in} = [\mathtt{Nothing}, \mathtt{Just}] \ \}$$

$$\mathsf{out} \cdot [\underline{\mathtt{Nothing}}, \mathtt{Just}] = id$$

$$\Leftrightarrow \quad \{ \ +\text{-fusion} \ \}$$

$$[\mathsf{out} \cdot \underline{\mathtt{Nothing}}, \mathsf{out} \cdot \mathtt{Just}] = id$$

$$\Leftrightarrow \quad \{ \ \text{universal-}+\text{; constant function} \ \}$$

$$\left\{ \begin{array}{l} \underline{\mathsf{out}\ \mathtt{Nothing}} = i_1 \\ \mathsf{out} \cdot \mathtt{Just} = i_2 \end{array} \right.$$

Maybe

$$\Leftrightarrow \qquad \{\ \text{variables}\ ;\ \underline{\texttt{Nothing}} : 1 \to \texttt{Maybe}\ A\ \}$$

$$\begin{cases} \underline{\textbf{out}\ \texttt{Nothing}}\ () = i_1\ () \\ (\textbf{out} \cdot \texttt{Just})\ a = i_2\ a \end{cases}$$

Maybe

$$\Leftrightarrow \quad \{\ \text{variables} \ ;\ \underline{\mathtt{Nothing}} : 1 \rightarrow \mathtt{Maybe}\ A\ \}$$

$$\begin{cases} \underline{\mathbf{out}\ \mathtt{Nothing}}\ () = i_1\ () \\ (\mathbf{out} \cdot \mathtt{Just})\ a = i_2\ a \end{cases}$$

$$\Leftrightarrow \quad \{\ \text{composition } (\textit{pointwise})\ \}$$

$$\begin{cases} \mathbf{out}\ \mathtt{Nothing} = i_1\ () \\ \mathbf{out}\ (\mathtt{Just}\ a) = i_2\ a \end{cases}$$

Maybe

$$\Leftrightarrow \qquad \{ \text{ variables ; } \underline{\mathtt{Nothing}} : 1 \to \mathtt{Maybe}\ A \ \}$$

$$\begin{cases} \underline{\mathbf{out}\ \mathtt{Nothing}}\ () = i_1\ () \\ (\mathbf{out} \cdot \mathtt{Just})\ a = i_2\ a \end{cases}$$

$$\Leftrightarrow \qquad \{ \text{ composition } (\textit{pointwise}) \ \}$$

$$\begin{cases} \mathbf{out}\ \mathtt{Nothing} = i_1\ () \\ \mathbf{out}\ (\mathtt{Just}\ a) = i_2\ a \end{cases}$$

$$\begin{tikzcd} \mathtt{Maybe}\ A \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] & & 1 + A \arrow[ll, bend left, "\mathbf{in}\ =\ {[\underline{\mathtt{Nothing}}, \mathtt{Just}]}"] \end{tikzcd}$$

Either

$$
\begin{tikzcd}
\texttt{Either}\ A\ B \arrow[rr, bend left, "\mathsf{out}"] \arrow[rr, phantom, "\cong"] & & A + B \arrow[ll, bend left, "{\mathsf{in} = [\texttt{Left},\texttt{Right}]}"]
\end{tikzcd}
$$

Either

$$
\begin{tikzcd}
\mathtt{Either}\ A\ B \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] & & A + B \arrow[ll, bend left, "\mathbf{in} = {[\mathtt{Left}, \mathtt{Right}]}"]
\end{tikzcd}
$$

$$
\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[dr, "\mathtt{Left}"'] & A + B \arrow[d, "{[\mathtt{Left}, \mathtt{Right}]}"] & B \arrow[l, "i_2"'] \arrow[dl, "\mathtt{Right}"] \\
& \mathtt{Either}\ A\ B &
\end{tikzcd}
$$

$[f, g]$

$\mathtt{either}\ f\ g$

Cálculo de Programas

Class T04 — 10-Oct

Polymorphism
(and why it matters!)

Polymorphism

$$reverse :: [a] \to [a]$$

Polymorphism

$$reverse :: [a] \rightarrow [a]$$

$$reverse :: [Int] \rightarrow [Int]$$

Polymorphism

$$reverse :: [a] \to [a]$$

$$reverse :: [Int] \to [Int]$$
$$reverse :: String \to String$$

Polymorphism

$$reverse :: [a] \to [a]$$

$$reverse :: [Int] \to [Int]$$
$$reverse :: String \to String$$
$$\ldots$$

Polymorphism

$$reverse :: [a] \to [a]$$

$$reverse :: [Int] \to [Int]$$
$$reverse :: String \to String$$
$$\ldots$$

"From the **type** of a **polymorphic function** we can derive a **theorem** that it satisfies. (...) How useful are the theorems so generated? Only time and experience will tell (...)" [**Philip Wadler** 1989]

Natural properties

$$[2, 5, 0, 5, 1]$$

Natural properties

$$[1, 5, 0, 5, 2] \xleftarrow{\textit{reverse}} [2, 5, 0, 5, 1]$$

Natural properties

$$
\begin{tikzcd}
{[1, 5, 0, 5, 2]} \arrow[d, mapsto, "succs"'] & {[2, 5, 0, 5, 1]} \arrow[l, mapsto, "reverse"'] \\
{[2, 6, 1, 6, 3]} &
\end{tikzcd}
$$

Natural properties

$$
\begin{tikzcd}
{[1,5,0,5,2]} \arrow[d, mapsto, "succs"'] & & {[2,5,0,5,1]} \arrow[ll, mapsto, "reverse"'] \arrow[d, mapsto, "succs"] \\
{[2,6,1,6,3]} & & {[3,6,1,6,2]}
\end{tikzcd}
$$

Natural properties

$$\begin{tikzcd}
{[1,5,0,5,2]} \arrow[d, mapsto, "succs"'] & & {[2,5,0,5,1]} \arrow[ll, mapsto, "reverse"'] \arrow[d, mapsto, "succs"] \\
{[2,6,1,6,3]} & & {[3,6,1,6,2]} \arrow[ll, mapsto, "reverse"]
\end{tikzcd}$$

Natural properties

$$\begin{tikzcd}[column sep=huge, row sep=huge]
{[1,5,0,5,2]} \arrow[d, mapsto, "\mathit{map}\ \mathit{succ}"'] & {[2,5,0,5,1]} \arrow[l, mapsto, "\mathit{reverse}"'] \arrow[d, mapsto, "\mathit{map}\ \mathit{succ}"] \\
{[2,6,1,6,3]} & {[3,6,1,6,2]} \arrow[l, mapsto, "\mathit{reverse}"]
\end{tikzcd}$$

Natural properties

$$[1, 5, 0, 5, 2] \xleftarrow{\textit{reverse}} [2, 5, 0, 5, 1]$$

Natural properties

$$\begin{tikzcd}
{[1,5,0,5,2]} \arrow[d, mapsto, "map\ even"'] & & {[2,5,0,5,1]} \arrow[ll, mapsto, "reverse"'] \\
{[F,F,T,F,T]} & &
\end{tikzcd}$$

Natural properties

$$\begin{tikzcd}[1,5,0,5,2] \arrow[d, mapsto, "map\ even"'] & & {[2,5,0,5,1]} \arrow[ll, mapsto, "reverse"'] \arrow[d, mapsto, "map\ even"] \\ {[F,F,T,F,T]} & & {[T,F,T,F,F]}\end{tikzcd}$$

Natural properties

$$\begin{tikzcd}
{[1, 5, 0, 5, 2]} \arrow[d, mapsto, "map\ even"'] & & {[2, 5, 0, 5, 1]} \arrow[ll, mapsto, "reverse"'] \arrow[d, mapsto, "map\ even"] \\
{[F, F, T, F, T]} & & {[T, F, T, F, F]} \arrow[ll, mapsto, "reverse"]
\end{tikzcd}$$

Natural properties

$$[2, 5, 0] \xleftarrow{\textit{take}\ 3} [2, 5, 0, 5, 1]$$

Natural properties

$$\begin{tikzcd}
{[2,5,0]} \arrow[d, mapsto, "map\ (2*)"'] & & {[2,5,0,5,1]} \arrow[ll, mapsto, "take\ 3"'] \\
{[4,10,0]}
\end{tikzcd}$$

Natural properties

$$\begin{tikzcd}
{[2,5,0]} \arrow[d, mapsto, "map\ (2*)"'] & & {[2,5,0,5,1]} \arrow[ll, mapsto, "take\ 3"'] \arrow[d, mapsto, "map\ (2*)"] \\
{[4,10,0]} & & {[4,10,0,10,2]}
\end{tikzcd}$$

Natural properties

$$
\begin{tikzcd}
{[2,5,0]} \arrow[d, mapsto, "map\ (2*)"'] & {[2,5,0,5,1]} \arrow[l, mapsto, "take\ 3"'] \arrow[d, mapsto, "map\ (2*)"] \\
{[4,10,0]} & {[4,10,0,10,2]} \arrow[l, mapsto, "take\ 3"]
\end{tikzcd}
$$

Natural properties

$$\texttt{"ogramas"} \xleftarrow{\mathit{drop}\ 2} \texttt{"Programas"}$$

Natural properties

$$\begin{tikzcd}
\texttt{"ogramas"} \arrow[d, mapsto, "\mathit{map}\ \mathit{upper}"'] & & \texttt{"Programas"} \arrow[ll, mapsto, "\mathit{drop}\ 2"'] \\
\texttt{"OGRAMAS"} & &
\end{tikzcd}$$

Natural properties

$$
\begin{tikzcd}
\texttt{"ogramas"} \arrow[d, mapsto, "\mathit{map}\ \mathit{upper}"'] & & \texttt{"Programas"} \arrow[ll, mapsto, "\mathit{drop}\ 2"'] \arrow[d, mapsto, "\mathit{map}\ \mathit{upper}"] \\
\texttt{"OGRAMAS"} & & \texttt{"PROGRAMAS"}
\end{tikzcd}
$$

Natural properties

$$\begin{tikzcd}
\texttt{"ogramas"} \arrow[d, mapsto, "map\ upper"'] & & \texttt{"Programas"} \arrow[ll, mapsto, "drop\ 2"'] \arrow[d, mapsto, "map\ upper"] \\
\texttt{"OGRAMAS"} & & \texttt{"PROGRAMAS"} \arrow[ll, mapsto, "drop\ 2"]
\end{tikzcd}$$

Back to...

$$
\begin{tikzcd}[column sep=huge]
(A \times B) \times C \arrow[r, bend left, "assocr"] \arrow[r, phantom, "\cong"] & A \times (B \times C) \arrow[l, bend left, "assocl"]
\end{tikzcd}
$$

$$assocr \cdot assocl = id$$

$$assocl \cdot assocr = id$$

“Natural” property

$$((2, T), \texttt{"CP"}) \xleftarrow{\quad assocl \quad} (2, (T, \texttt{"CP"}))$$

“Natural” property

$$\begin{tikzcd}
((2, T), \texttt{"CP"}) \arrow[d, mapsto, "(succ \times \neg) \times !"'] & & (2, (T, \texttt{"CP"})) \arrow[ll, mapsto, "assocl"'] \\
((3, F), ()) & &
\end{tikzcd}$$

“Natural” property

$$
\begin{tikzcd}
((2, T), \texttt{"CP"}) \arrow[d, mapsto, "(succ \times \neg) \times !"'] & (2, (T, \texttt{"CP"})) \arrow[l, mapsto, "assocl"'] \arrow[d, mapsto, "succ \times (\neg \times !)"] \\
((3, F), ()) & (3, (F, ()))
\end{tikzcd}
$$

“Natural” property

$$
\begin{tikzcd}
((2, T), \texttt{"CP"}) \arrow[d, mapsto, "(succ \times \neg) \times !"'] & (2, (T, \texttt{"CP"})) \arrow[l, mapsto, "assocl"'] \arrow[d, mapsto, "succ \times (\neg \times !)"] \\
((3, F), ()) & (3, (F, ())) \arrow[l, mapsto, "assocl"]
\end{tikzcd}
$$

“Natural” property

$$((a, b), c) \xleftarrow{\textit{assocl}} (a, (b, c))$$

“Natural” property

$$\begin{tikzcd}
((a, b), c) \arrow[d, mapsto, "(f \times g) \times h"'] & & (a, (b, c)) \arrow[ll, mapsto, "assocl"'] \\
((f\ a, g\ b), h\ c) & &
\end{tikzcd}$$

"Natural" property

$$
\begin{tikzcd}
((a, b), c) \arrow[d, mapsto, "(f \times g) \times h"'] & & (a, (b, c)) \arrow[ll, mapsto, "assocl"'] \arrow[d, mapsto, "f \times (g \times h)"] \\
((f\ a, g\ b), h\ c) & & (f\ a, (g\ b, h\ c))
\end{tikzcd}
$$

“Natural” property

$$
\begin{tikzcd}
((a, b), c) \arrow[d, mapsto, "(f \times g) \times h"'] & & (a, (b, c)) \arrow[ll, mapsto, "assocl"'] \arrow[d, mapsto, "f \times (g \times h)"] \\
((f\ a, g\ b), h\ c) & & (f\ a, (g\ b, h\ c)) \arrow[ll, mapsto, "assocl"]
\end{tikzcd}
$$

Natural properties

$$A \times (B \times C)$$

Natural properties

$$(A \times B) \times C \xleftarrow{\mathit{assocl}} A \times (B \times C)$$

Natural properties

$$
\begin{tikzcd}
(A \times B) \times C \arrow[d, "(f \times g) \times h"'] & & A \times (B \times C) \arrow[ll, "assocl"'] \\
(D \times E) \times F
\end{tikzcd}
$$

Natural properties

$$\begin{tikzcd}
(A \times B) \times C \arrow[d, "(f \times g) \times h"'] & & A \times (B \times C) \arrow[ll, "assocl"'] \arrow[d, "f \times (g \times h)"] \\
(D \times E) \times F & & D \times (E \times F)
\end{tikzcd}$$

Natural properties

$$
\begin{tikzcd}
(A \times B) \times C \arrow[d, "(f \times g) \times h"'] & & A \times (B \times C) \arrow[ll, "assocl"'] \arrow[d, "f \times (g \times h)"] \\
(D \times E) \times F & & D \times (E \times F) \arrow[ll, "assocl"]
\end{tikzcd}
$$

$$((f \times g) \times h) \cdot assocl = assocl \cdot (f \times (g \times h))$$

Practical rule

Back to...

$$\begin{tikzcd} A \times B \arrow[rr, bend left, "swap"] \arrow[rr, phantom, "\cong"] & & B \times A \arrow[ll, bend left, "swap"] \end{tikzcd}$$

$$swap \cdot swap = id$$

Practical rule

$$B \times A \xleftarrow{\;swap\;} A \times B$$

Practical rule

$$B \times A \xleftarrow{swap} A \times B$$

$$B' \times A' \xleftarrow[swap]{} A' \times B'$$

Practical rule

$$\begin{tikzcd}
B \times A \arrow[d, "\text{"}B \times A\text{"}"'] & A \times B \arrow[l, "swap"'] \arrow[d, "\text{"}A \times B\text{"}"] \\
B' \times A' & A' \times B' \arrow[l, "swap"]
\end{tikzcd}$$

Practical rule

$$\begin{tikzcd}
B \times A \arrow[d, "g \times f"'] & & A \times B \arrow[ll, "swap"'] \arrow[d, "f \times g"] \\
B' \times A' & & A' \times B' \arrow[ll, "swap"]
\end{tikzcd}$$

$$(g \times f) \cdot swap = swap \cdot (f \times g)$$

Practical rule

In the vertical arrows,

- Substitute $A := f$, $B := g$, etc

Practical rule

In the vertical arrows,

- Substitute $A := f$, $B := g$, etc
- In case of concrete types, replace by id, e.g. $2 := id$

Practical rule

In the vertical arrows,

- Substitute $A := f$, $B := g$, etc
- In case of concrete types, replace by id, e.g. $2 := id$
- Remove the quotes.

Simplest case...

$$A \xleftarrow{id} A$$

Simplest case...

$$\begin{tikzcd} A & & A \arrow[ll, "id"'] \\ \\ A' & & A' \arrow[ll, "id"] \end{tikzcd}$$

Simplest case...

$$
\begin{tikzcd}
A \arrow[d, "``A\text{''}"'] & A \arrow[l, "id"'] \arrow[d, "``A\text{''}"] \\
A' & A' \arrow[l, "id"]
\end{tikzcd}
$$

Natural-id

$$\begin{tikzcd}
A \arrow[d, "f"'] & A \arrow[l, "id"'] \arrow[d, "f"] \\
A' & A' \arrow[l, "id"]
\end{tikzcd}$$

$$f \cdot id = id \cdot f$$

Natural-π_1

$$A \xleftarrow{\pi_1} A \times B$$

Natural-π_1

$$\begin{tikzcd}
A & A \times B \arrow[l, "\pi_1"'] \\
 & \\
A' & A' \times B' \arrow[l, "\pi_1"]
\end{tikzcd}$$

Natural-π_1

$$\begin{tikzcd}
A \arrow[d, "``A\text{''}"'] & A \times B \arrow[l, "\pi_1"'] \arrow[d, "``A \times B\text{''}"] \\
A' & A' \times B' \arrow[l, "\pi_1"]
\end{tikzcd}$$

Natural-π_1

$$
\begin{tikzcd}
A \arrow[d, "f"'] & A \times B \arrow[l, "\pi_1"'] \arrow[d, "f \times g"] \\
A' & A' \times B' \arrow[l, "\pi_1"]
\end{tikzcd}
$$

$$(f) \cdot \pi_1 = \pi_1 \cdot (f \times g)$$

If ... then ... else...?

if-then-else ?

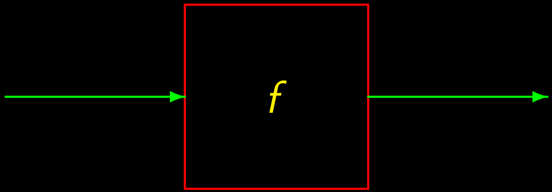
$$y = \textbf{if}\ p\ x\ \textbf{then}\ f\ x\ \textbf{else}\ g\ x$$

if-then-else ?

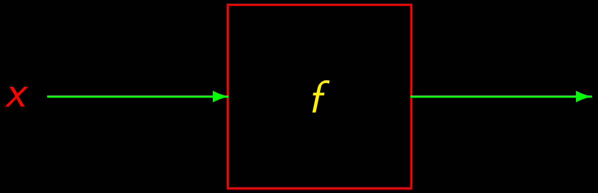
$$y = \textbf{if } p\ x \textbf{ then } f\ x \textbf{ else } g\ x$$

x should **flow** to f **or** g depending on p...

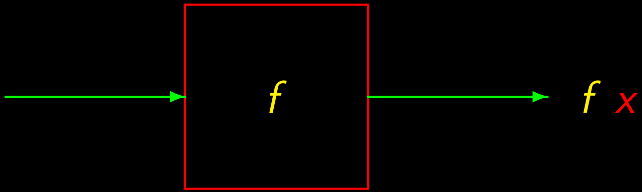
Data flow



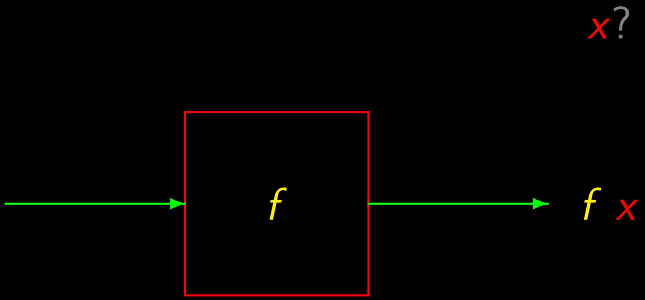
Data flow



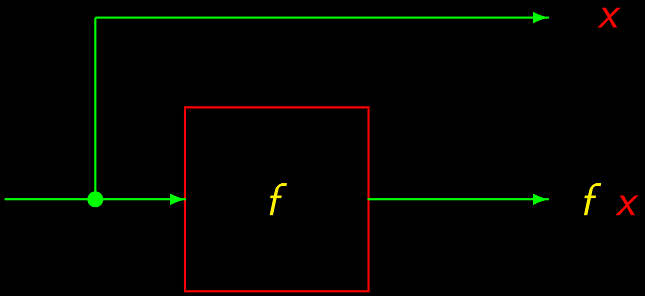
Data flow



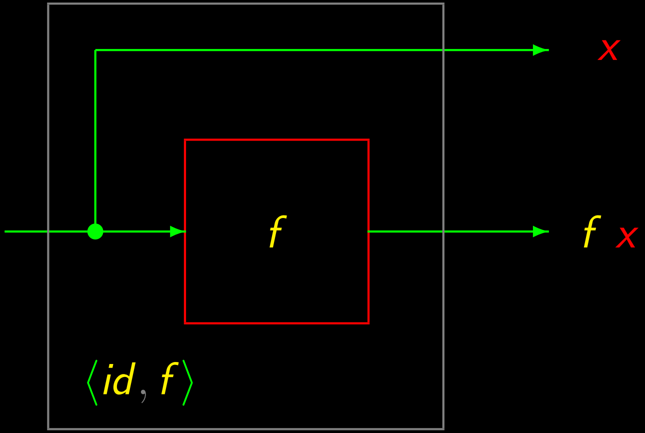
Data flow



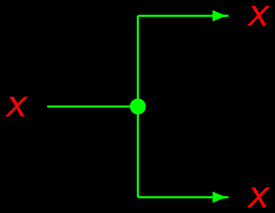
Data flow



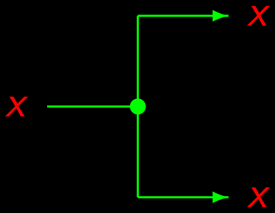
Data flow



Data duplication

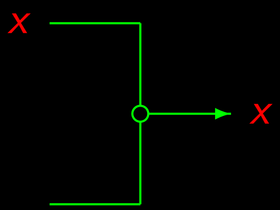


Data duplication



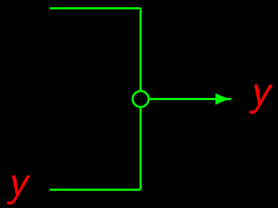
$$dup = \langle id, id \rangle$$

Data joins



$$join = [id, id]$$

Data joins

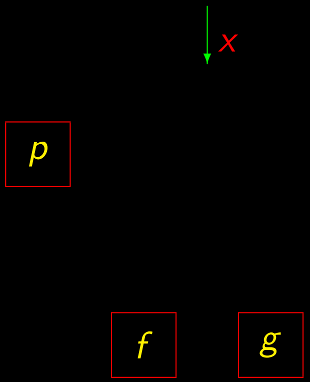


$$join = [id, id]$$

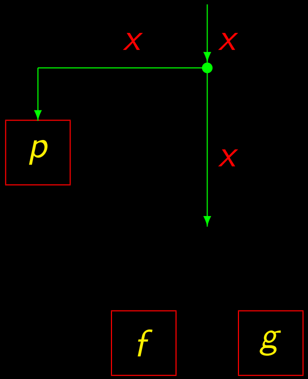
if-then-else

$$y = \textbf{if } p\ x \textbf{ then } f\ x \textbf{ else } g\ x$$

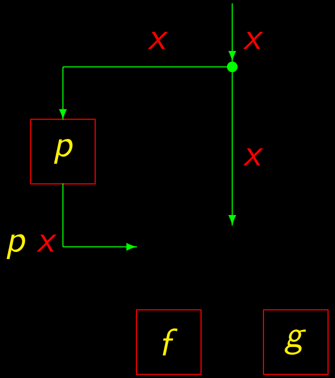
if-then-else



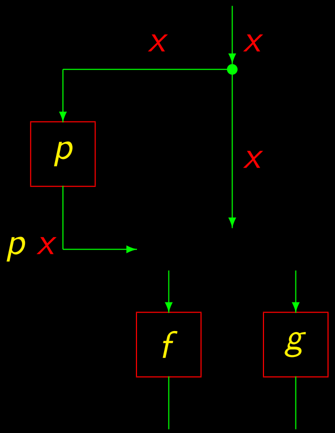
if-then-else



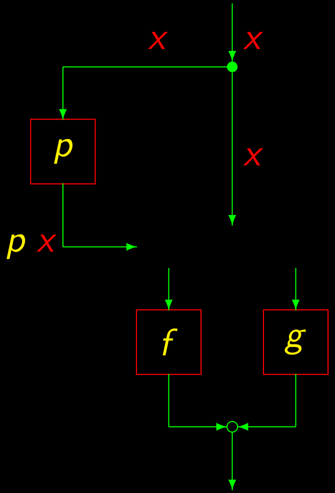
if-then-else



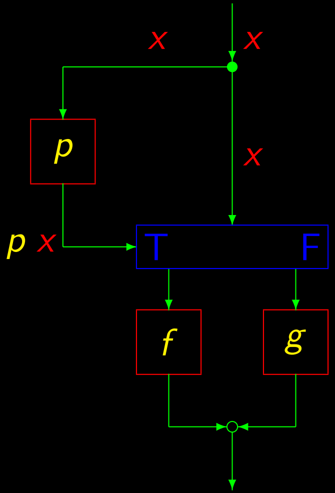
if-then-else



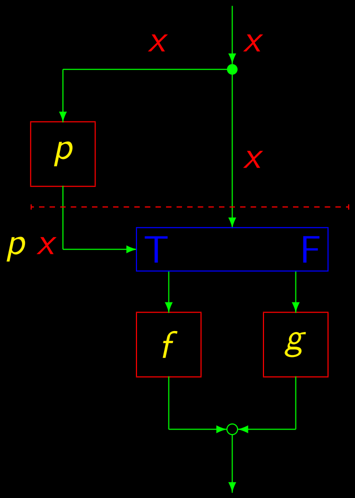
if-then-else



if-then-else



if-then-else



if-then-else ("point-free")

$x \in A$

if-then-else (“point-free”)

$x \in A$

$(p\ x, x) \in \mathbb{B} \times A$

if-then-else ("point-free")

$$x \in A$$

$$(p\ x, x) \in \mathbb{B} \times A$$

$$\mathbb{B} = \{F, T\}$$

if-then-else ("point-free")

$x \in A$

$(p\ x, x) \in \mathbb{B} \times A$

$\mathbb{B} = \{F, T\}$

$\mathbb{B}$ in Haskell:

```
data Bool = False | True
```

if-then-else ("point-free")

$x \in A$

$(p\ x, x) \in \mathbb{B} \times A$

$\mathbb{B} = \{F, T\}$

$\mathbb{B}$ in Haskell:

```
data Bool = False | True
```

T represented by `True`

F represented by `False`

The type 2

$$\mathbf{B} = \{\mathit{F}, \mathit{T}\} \cong \{\texttt{False}, \texttt{True}\}$$

The type 2

$$\mathbb{B} = \{F, T\} \cong \{\texttt{False}, \texttt{True}\} \cong \{0, 1\} \cong \cdots \cong 2$$

$$\begin{tikzcd} \mathbb{B} \arrow[rr, bend left, "g"] \arrow[rr, phantom, "\cong"] & & \{0,1\} \arrow[ll, bend left, "f"] \end{tikzcd}$$

The type 2

$$\mathbb{B} = \{F, T\} \cong \{\texttt{False}, \texttt{True}\} \cong \{0, 1\} \cong \cdots \cong 2$$

$$\begin{tikzcd} \mathbb{B} \arrow[rr, bend left, "g"] \arrow[rr, phantom, "\cong"] & & \{0,1\} \arrow[ll, bend left, "f"] \end{tikzcd}$$

$$\begin{cases} f\ 0 = F \\ f\ 1 = T \end{cases}$$

The type 2

$$\mathbb{B} = \{F, T\} \cong \{\texttt{False}, \texttt{True}\} \cong \{0, 1\} \cong \cdots \cong 2$$

$$\begin{tikzcd} \mathbb{B} \arrow[rr, bend left, "g"] \arrow[rr, phantom, "\cong"] & & \{0,1\} \arrow[ll, bend left, "f"] \end{tikzcd}$$

$$\begin{cases} f\ 0 = F \\ f\ 1 = T \end{cases} \qquad \begin{cases} g\ F = 0 \\ g\ T = 1 \end{cases}$$

$$a + a = 2\,a$$

$$A + A \xrightarrow{\textit{join}} A$$

$$a + a = 2\,a$$

$$A + A \xrightarrow{join} A$$

$$A \xrightarrow{\langle p, id \rangle} 2 \times A$$

$a + a = 2\ a$

$$A + A \xrightarrow{join} A$$

$$A \xrightarrow{\langle p, id \rangle} 2 \times A$$

Is it the case that...?

$$\begin{tikzcd} 2 \times A \arrow[rr, bend left, "g"] \arrow[rr, phantom, "\cong"] & & A + A \arrow[ll, bend left, "f"] \end{tikzcd}$$

$$
\begin{tikzcd}
2 \times A \arrow[rr, bend left, "g"] \arrow[rr, phantom, "\cong"] & & A + A \arrow[ll, bend left, "f"]
\end{tikzcd}
$$

$$
\begin{aligned}
f &= [\langle \underline{T}, id \rangle, \langle \underline{F}, id \rangle] \\
g &= \cdots (?)
\end{aligned}
$$

Notation

$$\begin{tikzcd} 2 \times A \arrow[rr, bend left, "\alpha^{\circ}"] \arrow[rr, phantom, "\cong"] && A + A \arrow[ll, bend left, "\alpha"] \end{tikzcd}$$

$$\alpha = [\langle \underline{T}, id \rangle, \langle \underline{F}, id \rangle]$$

$$\alpha^{\circ} = \cdots$$

Notation

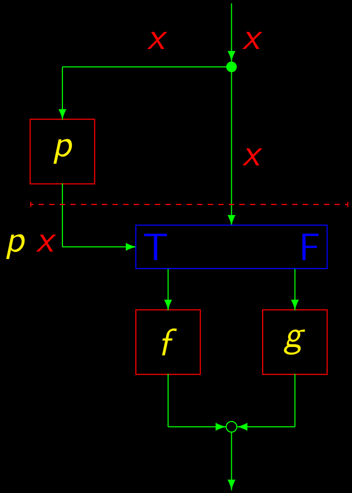
$$\begin{tikzcd}[column sep=huge] 2 \times A \arrow[r, bend left, "\alpha^{\circ}"] \arrow[r, phantom, "\cong"] & A + A \arrow[l, bend left, "\alpha"] \end{tikzcd}$$

$$\alpha \cdot \alpha^{\circ} = id$$

$$\alpha^{\circ} \cdot \alpha = id$$

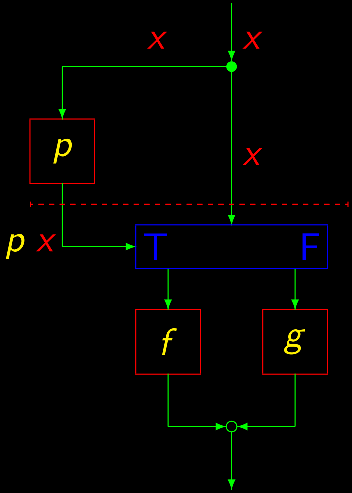
α determines α°

if-then-else



if-then-else

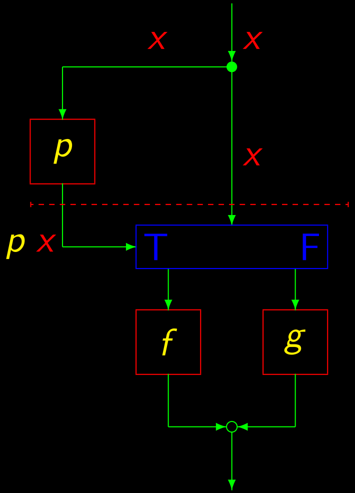
$$\begin{tikzcd} A \arrow[d, "\langle p, id \rangle"] \\ 2 \times A \end{tikzcd}$$



if-then-else

$$A \xrightarrow{\langle p,\, id \rangle} 2 \times A$$

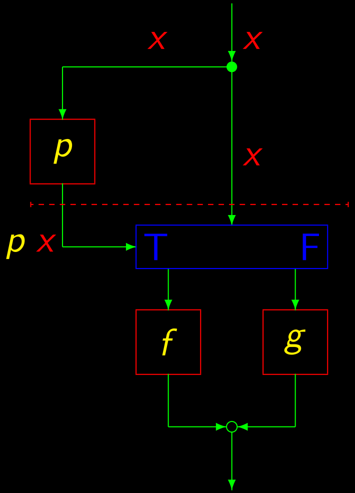
$$A \xrightarrow{f} B \qquad A \xrightarrow{g} B$$



if-then-else

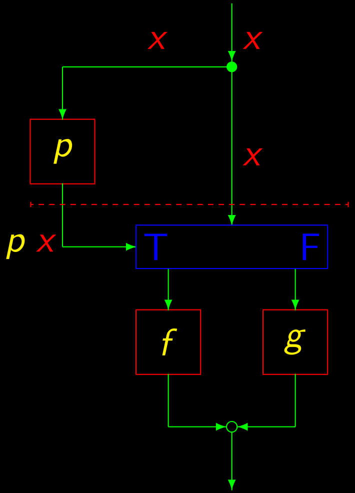
$$\begin{tikzcd} A \arrow[d, "\langle p, id \rangle"] \\ 2 \times A \end{tikzcd}$$

$$\begin{tikzcd} A \arrow[r, "i_1"] \arrow[d, "f"'] & A + A \arrow[d, "f + g"] & A \arrow[l, "i_2"'] \arrow[d, "g"] \\ B \arrow[r, "i_1"'] & B + B & B \arrow[l, "i_2"] \end{tikzcd}$$



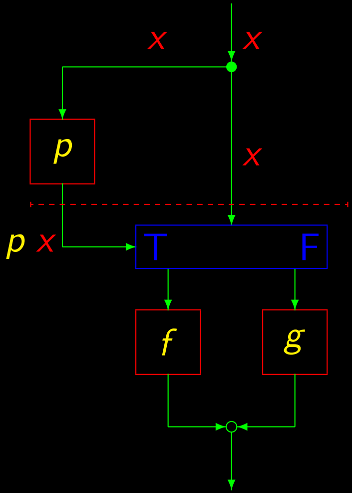
if-then-else

$$
\begin{tikzcd}
& A \arrow[d, "\langle p, id \rangle"] & \\
& 2 \times A & \\
A \arrow[r, "i_1"] \arrow[d, "f"'] & A + A \arrow[d, dashed, "f+g"] & A \arrow[l, "i_2"'] \arrow[d, "g"] \\
B \arrow[r, "i_1"'] & B + B \arrow[d, "join"] & B \arrow[l, "i_2"] \\
& B &
\end{tikzcd}
$$



if-then-else

$$\begin{tikzcd}
 & A \arrow[d, "\langle p, id\rangle"] & \\
 & 2 \times A \arrow[d, "\alpha^\circ"] & \\
A \arrow[r, "i_1"] \arrow[d, "f"'] & A + A \arrow[d, "f + g"] & A \arrow[l, "i_2"'] \arrow[d, "g"] \\
B \arrow[r, "i_1"'] & B + B \arrow[d, "join"] & B \arrow[l, "i_2"] \\
 & B &
\end{tikzcd}$$



McCarthy Conditional $p \rightarrow f\ ,\ g$

From the diagram:

$$p \rightarrow f\ ,\ g \quad = \quad join \cdot (f + g) \cdot \alpha^{\circ} \cdot \langle p, id \rangle$$

McCarthy Conditional $p \rightarrow f\ ,\ g$

From the diagram:

$$p \rightarrow f\ ,\ g \quad = \quad join \cdot (f + g) \cdot \alpha^{\circ} \cdot \langle p, id \rangle$$

Since

$$join \cdot (f + g) = [id, id] \cdot (f + g) = [f, g]$$

McCarthy Conditional $p \rightarrow f\ ,\ g$

From the diagram:

$$p \rightarrow f\ ,\ g \quad = \quad join \cdot (f + g) \cdot \alpha^{\circ} \cdot \langle p, id \rangle$$

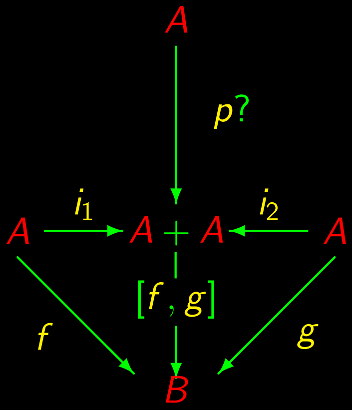
Since

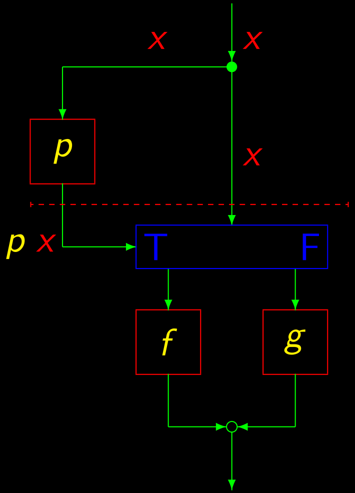
$$join \cdot (f + g) = [id\,, id] \cdot (f + g) = [f\,, g]$$

we have:

$$p \rightarrow f\ ,\ g \quad = \quad [f\,, g] \cdot \underbrace{\alpha^{\circ} \cdot \langle p, id \rangle}_{p?}$$

McCarthy Conditional $p \to f\ ,\ g$





McCarthy Conditional $p \rightarrow f\ ,\ g$

p-guard:

$$\begin{tikzcd} A \arrow[r, "\langle p, id \rangle"] \arrow[rr, bend right, "p?"'] & 2 \times A \arrow[r, "\alpha^\circ"] & A + A \end{tikzcd}$$

Conditional:

$$p \rightarrow f\ ,\ g \;=\; [f\,, g] \cdot p?$$

John McCarthy (1927-2011)

Turing Award

Founding father of **Artificial Intelligence**

PhD in mathematics (Princeton, 1951)

Inventor of **LISP**

Conditionals pointwise

$$p? : A \to A + A$$

$$p?\ a = \begin{cases} p\ a \Rightarrow i_1\ a \\ \neg\,(p\ a) \Rightarrow i_2\ a \end{cases}$$

Conditionals pointwise

$$p? : A \to A + A$$

$$p?\ a = \begin{cases} p\ a \Rightarrow i_1\ a \\ \neg\ (p\ a) \Rightarrow i_2\ a \end{cases}$$

wherefrom

$$(p \to f\ ,\ g)\ a = \begin{cases} p\ a \Rightarrow f\ a \\ \neg\ (p\ a) \Rightarrow g\ a \end{cases}$$

Conditionals pointwise

$$p? : A \to A + A$$

$$p?\ a = \begin{cases} p\ a \Rightarrow i_1\ a \\ \neg\ (p\ a) \Rightarrow i_2\ a \end{cases}$$

wherefrom

$$(p \to f\ ,\ g)\ a = \begin{cases} p\ a \Rightarrow f\ a \\ \neg\ (p\ a) \Rightarrow g\ a \end{cases}$$

Case analysis?

Conditionals pointwise

$$p? : A \to A + A$$

$$p?\ a = \begin{cases} p\ a \Rightarrow i_1\ a \\ \neg\ (p\ a) \Rightarrow i_2\ a \end{cases}$$

wherefrom

$$(p \to f\ ,\ g)\ a = \begin{cases} p\ a \Rightarrow f\ a \\ \neg\ (p\ a) \Rightarrow g\ a \end{cases}$$

Case analysis? No!

Between conditional and composition

Any "chemistry"?

$$h \cdot (p \to f \; , g) = \, ?$$

Between conditional and composition

Any "chemistry"?

$$h \cdot (p \to f\ ,\ g) = ?$$

$$(p \to f\ ,\ g) \cdot h = ?$$

"Calculemus"

$$h \cdot (p \rightarrow f \;,\; g)$$

"Calculemus"

$$h \cdot (p \rightarrow f\ ,\ g)$$

$$\Leftrightarrow \qquad \{ \text{ definition of conditional } \}$$

$$h \cdot [f\,, g] \cdot p?$$

"Calculemus"

$$
\begin{aligned}
& h \cdot (p \rightarrow f \,,\, g) \\
\Leftrightarrow \quad & \{ \text{ definition of conditional } \} \\
& h \cdot [f \,,\, g] \cdot p? \\
\Leftrightarrow \quad & \{ \text{ } + \text{-fusion } \} \\
& [h \cdot f \,,\, h \cdot g] \cdot p?
\end{aligned}
$$

“Calculemus”

$$h \cdot (p \rightarrow f\ ,\ g)$$

$\Leftrightarrow$ { definition of conditional }

$$h \cdot [f\,, g] \cdot p?$$

$\Leftrightarrow$ { $+$-fusion }

$$[h \cdot f\,, h \cdot g] \cdot p?$$

$\Leftrightarrow$ { definition of conditional }

$$p \rightarrow (h \cdot f)\ ,\ (h \cdot g)$$

Conditional — 1st fusion law

Altogether:

$$h \cdot (p \rightarrow f\ ,\ g) \ = \ p \rightarrow (h \cdot f)\ ,\ (h \cdot g)$$

Conditional — 1st fusion law

Altogether:

$$h \cdot (p \rightarrow f\ ,\ g) \;=\; p \rightarrow (h \cdot f)\ ,\ (h \cdot g)$$

What about running h **before** the conditional?

$$(p \rightarrow f\ ,\ g) \cdot h = ?$$

Conditional — 2nd fusion law

Intuitively, one has:

$$(p \rightarrow f \,,\, g) \cdot h \;=\; (p \cdot h) \rightarrow (f \cdot h) \,,\, (g \cdot h)$$

Conditional — 2nd fusion law

Intuitively, one has:

$$(p \rightarrow f\ ,\ g) \cdot h \;=\; (p \cdot h) \rightarrow (f \cdot h)\ ,\ (g \cdot h)$$

Intuitively?

Conditional — 2nd fusion law

Intuitively, one has:

$$(p \rightarrow f\ ,\ g) \cdot h \;=\; (p \cdot h) \rightarrow (f \cdot h)\ ,\ (g \cdot h)$$

Intuitively?

Proof ?

2nd fusion law of conditionals

It can be shown that

$$(p \to f \, , g) \cdot h \; = \; p \cdot h \to f \cdot h \, , g \cdot h$$

follows from

$$p? \cdot f \; = \; (f + f) \cdot (p \cdot f)?$$

2nd fusion law of conditionals

It can be shown that

$$(p \rightarrow f\ ,\ g) \cdot h \;=\; p \cdot h \rightarrow f \cdot h\ ,\ g \cdot h$$

follows from

$$\boxed{\;p? \cdot f \;=\; (f + f) \cdot (p \cdot f)?\;}$$

So we are left with the **proof** of the equality above...

Let us do it now

Prove

$$p? \cdot f \;=\; (f + f) \cdot (p \cdot f)?$$

knowing:

$$\begin{tikzcd} A \arrow[r, "\langle p, id \rangle"] \arrow[rr, bend right, "p?"'] & 2 \times A \arrow[r, "\alpha^{\circ}"] & A + A \end{tikzcd}$$

Recall

$$\begin{tikzcd} 2 \times A \arrow[rr, bend left, "\alpha^\circ"] \arrow[rr, phantom, "\cong"] & & A + A \arrow[ll, bend left, "\alpha"] \end{tikzcd}$$

$$\alpha = [\langle \underline{T}, id \rangle, \langle \underline{F}, id \rangle]$$

$$\alpha^\circ = \cdots ?$$

Let us try it

$p? \cdot f$

Let us try it

$$
\begin{aligned}
& p? \cdot f \\
= & \quad \{ \text{ substitution } p? = \alpha^\circ \cdot \langle p, id \rangle \ \} \\
& \alpha^\circ \cdot \langle p, id \rangle \cdot f
\end{aligned}
$$

Let us try it

$$
\begin{aligned}
& p? \cdot f \\
= & \quad \{ \text{ substitution } p? = \alpha^\circ \cdot \langle p, id \rangle \ \} \\
& \alpha^\circ \cdot \langle p, id \rangle \cdot f \\
= & \quad \{ \ \times\text{-fusion } \} \\
& \alpha^\circ \cdot \langle p \cdot f, id \cdot f \rangle
\end{aligned}
$$

Let us try it

$$
\begin{aligned}
& p? \cdot f \\
= \quad & \{ \text{ substitution } p? = \alpha^\circ \cdot \langle p, id \rangle \ \} \\
& \alpha^\circ \cdot \langle p, id \rangle \cdot f \\
= \quad & \{ \ \times\text{-fusion } \} \\
& \alpha^\circ \cdot \langle p \cdot f, id \cdot f \rangle \\
= \quad & \{ \ ? \ \} \\
& \ldots
\end{aligned}
$$

Let us try it

$$
\begin{array}{cl}
& p? \cdot f \\
= & \{ \text{ substitution } p? = \alpha^\circ \cdot \langle p, id \rangle \ \} \\
& \alpha^\circ \cdot \langle p, id \rangle \cdot f \\
= & \{ \times\text{-fusion } \} \\
& \alpha^\circ \cdot \langle p \cdot f, id \cdot f \rangle \\
= & \{ \ ? \ \} \\
& \ldots
\end{array}
$$

$$
\begin{array}{cl}
& \vdots \\
= & \{ \ ? \ \} \\
& (f + f) \cdot (p \cdot f)?
\end{array}
$$

Recall

$$\begin{tikzcd} 2 \times A \arrow[rr, bend left, "\alpha^{\circ}"] \arrow[rr, phantom, "\cong"] & & A + A \arrow[ll, bend left, "\alpha"] \end{tikzcd}$$

$$\alpha = [\langle \underline{T}, id \rangle, \langle \underline{F}, id \rangle]$$

$$\alpha^{\circ} = \cdots ?$$

Natural-α°

$$A + A \xleftarrow{\alpha^\circ} 2 \times A$$

Natural-α°

$$A + A \xleftarrow{\alpha^{\circ}} 2 \times A$$

$$A' + A' \xleftarrow[\alpha^{\circ}]{} 2 \times A'$$

Natural-α°

$$\begin{tikzcd}
A + A \arrow[d, "{``A + A"}"'] & 2 \times A \arrow[l, "\alpha^\circ"'] \arrow[d, "{``2 \times A"}"] \\
A' + A' & 2 \times A' \arrow[l, "\alpha^\circ"]
\end{tikzcd}$$

Natural-α°

$$\begin{tikzcd}
A + A & & 2 \times A \arrow[ll, "\alpha^\circ"'] \arrow[d, "id \times f"] \\
A' + A' \arrow[from=u, "f + f"'] & & 2 \times A' \arrow[ll, "\alpha^\circ"]
\end{tikzcd}$$

$$(f + f) \cdot \alpha^\circ = \alpha^\circ \cdot (id \times f)$$

… and this finishes the proof!

$$p? \cdot f$$

$= \quad \{ \; p? = \alpha^{\circ} \cdot \langle p, id \rangle \; \}$

$$\alpha^{\circ} \cdot \langle p, id \rangle \cdot f$$

$= \quad \{ \; \times\text{-fusion} \; \}$

$$\alpha^{\circ} \cdot \langle p \cdot f, id \cdot f \rangle$$

$= \quad \{ \; \text{natural-}id \text{ twice} \; \}$

$$\alpha^{\circ} \cdot \langle id \cdot p \cdot f, f \cdot id \rangle$$

$= \quad \{ \; \times\text{-absorption} \; \}$

$$\alpha^{\circ} \cdot (id \times f) \cdot \langle p \cdot f, id \rangle$$

… and this finishes the proof!

$$
\begin{array}{ll}
& p? \cdot f \\
= & \{\ p? = \alpha^{\circ} \cdot \langle p, id \rangle\ \} \\
& \alpha^{\circ} \cdot \langle p, id \rangle \cdot f \\
= & \{\ \times\text{-fusion}\ \} \\
& \alpha^{\circ} \cdot \langle p \cdot f, id \cdot f \rangle \\
= & \{\ \text{natural-}id\text{ twice}\ \} \\
& \alpha^{\circ} \cdot \langle id \cdot p \cdot f, f \cdot id \rangle \\
= & \{\ \times\text{-absorption}\ \} \\
& \alpha^{\circ} \cdot (id \times f) \cdot \langle p \cdot f, id \rangle \\
= & \{\ \text{free thef } \alpha^{\circ}\ \} \\
& (f + f) \cdot \alpha^{\circ} \cdot \langle p \cdot f, id \rangle
\end{array}
$$

... and this finishes the proof!

$$
\begin{array}{ll}
& p? \cdot f \\
= & \quad \{\ p? = \alpha^\circ \cdot \langle p, id \rangle\ \} \\
& \alpha^\circ \cdot \langle p, id \rangle \cdot f \\
= & \quad \{\ \times\text{-fusion}\ \} \\
& \alpha^\circ \cdot \langle p \cdot f, id \cdot f \rangle \\
= & \quad \{\ \text{natural-}id \text{ twice}\ \} \\
& \alpha^\circ \cdot \langle id \cdot p \cdot f, f \cdot id \rangle
\end{array}
$$

$$
\begin{array}{ll}
= & \quad \{\ \times\text{-absorption}\ \} \\
& \alpha^\circ \cdot (id \times f) \cdot \langle p \cdot f, id \rangle \\
= & \quad \{\ \text{free thef } \alpha^\circ\ \} \\
& (f + f) \cdot \alpha^\circ \cdot \langle p \cdot f, id \rangle \\
= & \quad \{\ p? = \alpha^\circ \cdot \langle p, id \rangle\ \} \\
& (f + f) \cdot (p \cdot f)?
\end{array}
$$

Cálculo de Programas

Class T05

Higher-order functions
(and why they matter!)

Notation variants

$2 + 3$	**infix** notation	operator between

Notation variants

$2 + 3$	**infix** notation	operator between
$+\ 2\ 3$	**prefix** notation	operator before

Notation variants

$2 + 3$	**infix** notation	operator between
$+ \; 2 \; 3$	**prefix** notation	operator before
$2 \; 3 \; +$	**posfix** notation	operator after

Notation variants

$2 + 3$	**infix** notation	operator between
$+\ 2\ 3$	**prefix** notation	operator before
$2\ 3\ +$	**posfix** notation	operator after
$+\ (2, 3)$	**prefix** "uncurried"	grouped inputs

Notation variants

$f\ a\ b$ **prefix** notation "curried" separated inputs

Notation variants

$f\ a\ b$	**prefix** notation "curried"	separated inputs
$f\ (a, b)$	**prefix** "uncurried"	grouped inputs

“Curried”? “uncurried”

Terminology in honour of mathematician **Haskell Curry** (1900-1982) who developed the theory behind **functional programming** (1936).

Example

$$\pi_1 : A \times B \to A$$

Example

$$\pi_1 : A \times B \to A$$

$$\pi_1 \left(a, b \right) = a$$

Example

$$\pi_1 : A \times B \to A$$

$$\pi_1 \left(a, b\right) = a$$

(notation: prefix "uncurried")

Exponentials

```
Prelude> p1(a,b)=a
Prelude> p1' a b = a
Prelude> :t p1
p1 :: (a, b) -> a
Prelude> :t p1'
p1' :: a -> b -> a
Prelude>
```

In Haskell

$$\pi_1{}' = \texttt{curry}\ \pi_1$$

In Haskell

$$\pi_1{}' = \texttt{curry}\ \pi_1$$

$$\pi_1 = \texttt{uncurry}\ \pi_1{}'$$

In Haskell

$$\pi_1{}' = \texttt{curry}\ \pi_1$$

$$\pi_1 = \texttt{uncurry}\ \pi_1{}'$$

Important:

- `curry` (e `uncurry`) accept functions as **arguments**
- `curry` (e `uncurry`) yield functions as **outputs**

In Haskell

$$\pi_1{}' = \texttt{curry}\ \pi_1$$

$$\pi_1 = \texttt{uncurry}\ \pi_1{}'$$

Important:

- `curry` (e `uncurry`) accept functions as **arguments**
- `curry` (e `uncurry`) yield functions as **outputs**

They are said to be

higher order functions.

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to (a \to b \to c)$$

$$\texttt{uncurry} :: (a \to b \to c) \to ((a, b) \to c)$$

Curry | uncurry

$$\texttt{curry} :: \boxed{((a, b) \to c)} \to (a \to b \to c)$$

$$\texttt{uncurry} :: (a \to b \to c) \to \boxed{((a, b) \to c)}$$

Curry | uncurry

$$\texttt{curry} :: \boxed{((a, b) \to c)} \to \boxed{(a \to b \to c)}$$

$$\texttt{uncurry} :: \boxed{(a \to b \to c)} \to \boxed{((a, b) \to c)}$$

Curry | uncurry

$$\texttt{curry} :: \boxed{((a, b) \to c)} \to \boxed{(a \to b \to c)}$$

$$\texttt{uncurry} :: \boxed{(a \to b \to c)} \to \boxed{((a, b) \to c)}$$

Each other inverses?

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to a \to b \to c$$

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to a \to b \to c$$

`curry`

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to a \to b \to c$$

$$\texttt{curry}\ f$$

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to a \to b \to c$$

$$\texttt{curry}\ f\ a$$

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to a \to b \to c$$

$$\texttt{curry}\ f\ a\ b$$

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to a \to b \to c$$

$$\texttt{curry}\ f\ a\ b\ =$$

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to a \to b \to c$$

$$\texttt{curry}\ f\ a\ b = f\ (a, b)$$

Curry | uncurry

$$\texttt{curry} :: ((a, b) \rightarrow c) \rightarrow a \rightarrow b \rightarrow c$$
$$\texttt{curry}\ f\ a\ b = f\ (a, b)$$

$$\texttt{uncurry} :: (a \rightarrow b \rightarrow c) \rightarrow (a, b) \rightarrow c$$

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to a \to b \to c$$
$$\texttt{curry}\ f\ a\ b = f\ (a, b)$$

$$\texttt{uncurry} :: (a \to b \to c) \to (a, b) \to c$$
$$\texttt{uncurry}$$

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to a \to b \to c$$

$$\texttt{curry}\ f\ a\ b = f\ (a, b)$$

$$\texttt{uncurry} :: (a \to b \to c) \to (a, b) \to c$$

$$\texttt{uncurry}\ g$$

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to a \to b \to c$$
$$\texttt{curry}\ f\ a\ b = f\ (a, b)$$

$$\texttt{uncurry} :: (a \to b \to c) \to (a, b) \to c$$
$$\texttt{uncurry}\ g\ (a, b)$$

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to a \to b \to c$$
$$\texttt{curry}\ f\ a\ b = f\ (a, b)$$

$$\texttt{uncurry} :: (a \to b \to c) \to (a, b) \to c$$
$$\texttt{uncurry}\ g\ (a, b) =$$

Curry | uncurry

$$\texttt{curry} :: ((a, b) \to c) \to a \to b \to c$$
$$\texttt{curry}\ f\ a\ b = f\ (a, b)$$

$$\texttt{uncurry} :: (a \to b \to c) \to (a, b) \to c$$
$$\texttt{uncurry}\ g\ (a, b) = g\ a\ b$$

Curry | uncurry

$$\begin{cases} \texttt{curry}\ (\texttt{uncurry}\ f) = f? \\ \texttt{uncurry}\ (\texttt{curry}\ f) = f? \end{cases}$$

Curry | uncurry

$$\texttt{uncurry}\ g\ (a, b) = g\ a\ b$$

Curry | uncurry

$$\mathtt{uncurry}\ (\mathtt{curry}\ f)\ (a, b) = \mathtt{curry}\ f\ a\ b$$

$$\mathtt{curry}\ f\ a\ b = f\ (a, b)$$

Curry | uncurry

$$\texttt{uncurry}\ (\texttt{curry}\ f)\ (a, b) = f\ (a, b)$$

$$f\ x = g\ x \Leftrightarrow f = g$$

Curry | uncurry

$$\texttt{uncurry}\ (\texttt{curry}\ f) = f$$

$$f\ (g\ x) = (f \cdot g)\ x$$

Curry | uncurry

$$(\texttt{uncurry} \cdot \texttt{curry})\ f = f$$

$$id\ x = x$$

Curry | uncurry

$$(\texttt{uncurry} \cdot \texttt{curry})\ f = id\ f$$

$$f\ x = g\ x \Leftrightarrow f = g$$

Curry | uncurry

$$\texttt{uncurry} \cdot \texttt{curry} = \mathit{id}$$

$\square$

Curry | uncurry

$$\texttt{curry}\ (\texttt{uncurry}\ f) = f\ ?$$

Curry | uncurry

$$\texttt{curry}\ f\ a\ b = f\ (a, b)$$

Curry | uncurry

$$\texttt{curry}\ (\texttt{uncurry}\ g)\ a\ b = \texttt{uncurry}\ g\ (a, b)$$

$$\texttt{uncurry}\ g\ (a, b) = g\ a\ b$$

Curry | uncurry

$$\texttt{curry}\ (\texttt{uncurry}\ g)\ a\ b = g\ a\ b$$

$$f\ x = g\ x \Leftrightarrow f = g$$

Curry | uncurry

$$\texttt{curry}\ (\texttt{uncurry}\ g)\ a = g\ a$$

$$f\ x = g\ x \Leftrightarrow f = g$$

Curry | uncurry

$$\texttt{curry}\ (\texttt{uncurry}\ g) = g$$

$$f\ (g\ x) = (f \cdot g)\ x$$

Curry | uncurry

$$(\texttt{curry} \cdot \texttt{uncurry})\ g = g$$

$$\mathit{id}\ x = x$$

Curry | uncurry

$$(\texttt{curry} \cdot \texttt{uncurry})\ g = id\ g$$

$$f\ x = g\ x \Leftrightarrow f = g$$

Curry | uncurry

$$\texttt{curry} \cdot \texttt{uncurry} = \mathit{id}$$

$\square$

Isomorphism

$$\begin{tikzcd}
(a, b) \to c \arrow[rr, bend left, "\texttt{curry}"] \arrow[r, phantom, "\cong"] & {} & a \to b \to c \arrow[ll, bend left, "\texttt{uncurry}"]
\end{tikzcd}$$

The type B^A

The type B^A

$$B^A = \{ f \mid f : A \rightarrow B \}$$

NB: functions that map A to B

Isomorphism

$$\begin{tikzcd} A \times B \to C \arrow[rr, bend left, "\texttt{curry}"] \arrow[r, phantom, "\cong"] & {} & A \to (B \to C) \arrow[ll, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

Isomorphism

$$\begin{tikzcd} A \times B \to C \arrow[rr, bend left, "\texttt{curry}"] \arrow[r, phantom, "\cong"] & & A \to C^B \arrow[ll, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

Isomorphism

$$\begin{tikzcd} C^{A\times B} \arrow[rr, bend left, "\texttt{curry}"] \arrow[rr, phantom, "\cong"] & & (C^{B})^{A} \arrow[ll, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

Cf. arithmetic exponentials: $c^{a\,b} = (c^{b})^{a}$

Curry | uncurry

$$\begin{tikzcd} A \times B \to C \arrow[rr, bend left, "\texttt{curry}"] \arrow[r, phantom, "\cong"] & & A \to C^B \arrow[ll, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

Curry | uncurry

f

$$\begin{tikzcd} A \times B \to C \arrow[rr, bend left, "\texttt{curry}"] \arrow[rr, phantom, "\cong"] & & A \to C^B \arrow[ll, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

Curry | uncurry

$$f \xmapsto{\texttt{curry}} k$$

$$\begin{tikzcd} A \times B \to C \arrow[rr, bend left, "\texttt{curry}"] \arrow[r, phantom, "\cong"] & & A \to C^B \arrow[ll, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

Curry | uncurry

$$f \xmapsto{\texttt{curry}} k$$

$$\begin{tikzcd} A \times B \to C \arrow[rr, bend left, "\texttt{curry}"] \arrow[rr, phantom, "\cong"] & & A \to C^B \arrow[ll, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

k

Curry | uncurry

$$f \xmapsto{\texttt{curry}} k$$

$$\begin{tikzcd} A \times B \to C \arrow[rr, bend left, "\texttt{curry}"] \arrow[rr, phantom, "\cong"] & & A \to C^B \arrow[ll, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

$$f \xleftarrow[\quad]{\texttt{uncurry}} k$$

Curry | uncurry

$$k = \mathtt{curry}\ f$$

$$\begin{tikzcd} A \times B \to C \arrow[rr, bend left, "\mathtt{curry}"] \arrow[rr, phantom, "\cong"] & & A \to C^B \arrow[ll, bend left, "\mathtt{uncurry}"] \end{tikzcd}$$

$$\mathtt{uncurry}\ k = f$$

Curry | uncurry

$$\begin{tikzcd} A \times B \to C \arrow[rr, bend left, "\texttt{curry}"] & \cong & A \to C^B \arrow[ll, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

$$k = \texttt{curry}\ f \iff \texttt{uncurry}\ k = f$$

Curry | uncurry

$$k = \texttt{curry}\ f$$

$$\Leftrightarrow \quad \{\ \text{isomorphism (last slide)}\ \}$$

$$\texttt{uncurry}\ k = f$$

Curry | uncurry

$$
\begin{aligned}
& k = \mathtt{curry}\ f \\
\Leftrightarrow \quad & \quad \{ \text{ isomorphism (last slide) } \} \\
& \mathtt{uncurry}\ k = f \\
\Leftrightarrow \quad & \quad \{ \text{ add variables } \} \\
& \mathtt{uncurry}\ k\ (a, b) = f\ (a, b)
\end{aligned}
$$

Curry | uncurry

$$(k\ a)\ b = f\ (a, b)$$

$$
\begin{array}{ll}
 & k = \texttt{curry}\ f \\
\Leftrightarrow & \quad \{\ \text{isomorphism (last slide)}\ \} \\
 & \texttt{uncurry}\ k = f \\
\Leftrightarrow & \quad \{\ \text{add variables}\ \} \\
 & \texttt{uncurry}\ k\ (a, b) = f\ (a, b) \\
\Leftrightarrow & \quad \{\ \text{definition of } \texttt{uncurry}\ k\ \} \\
 & k\ a\ b = f\ (a, b)
\end{array}
$$

Curry | uncurry

$$k = \mathtt{curry}\ f$$

$\Leftrightarrow$ { isomorphism (last slide) }

$$\mathtt{uncurry}\ k = f$$

$\Leftrightarrow$ { add variables }

$$\mathtt{uncurry}\ k\ (a, b) = f\ (a, b)$$

$\Leftrightarrow$ { definition of $\mathtt{uncurry}\ k$ }

$$k\ a\ b = f\ (a, b)$$

$$(k\ a)\ b = f\ (a, b)$$

$\Leftrightarrow$ { introduce $ap\ (f, x) = f\ x$ }

$$ap\ (k\ a, b) = f\ (a, b)$$

Curry | uncurry

$$
\begin{aligned}
& k = \mathtt{curry}\ f \\
\Leftrightarrow \quad & \{ \ \text{isomorphism (last slide)} \ \} \\
& \mathtt{uncurry}\ k = f \\
\Leftrightarrow \quad & \{ \ \text{add variables} \ \} \\
& \mathtt{uncurry}\ k\ (a, b) = f\ (a, b) \\
\Leftrightarrow \quad & \{ \ \text{definition of } \mathtt{uncurry}\ k \ \} \\
& k\ a\ b = f\ (a, b)
\end{aligned}
$$

$$
\begin{aligned}
& (k\ a)\ b = f\ (a, b) \\
\Leftrightarrow \quad & \{ \ \text{introduce } ap\ (f, x) = f\ x \ \} \\
& ap\ (k\ a, b) = f\ (a, b) \\
\Leftrightarrow \quad & \{ \ \text{add } id \ \} \\
& ap\ (k\ a, id\ b) = f\ (a, b)
\end{aligned}
$$

Curry | uncurry

$$k = \mathtt{curry}\ f$$

$\Leftrightarrow$ { isomorphism (last slide) }

$$\mathtt{uncurry}\ k = f$$

$\Leftrightarrow$ { add variables }

$$\mathtt{uncurry}\ k\ (a, b) = f\ (a, b)$$

$\Leftrightarrow$ { definition of $\mathtt{uncurry}\ k$ }

$$k\ a\ b = f\ (a, b)$$

$$(k\ a)\ b = f\ (a, b)$$

$\Leftrightarrow$ { introduce $ap\ (f, x) = f\ x$ }

$$ap\ (k\ a, b) = f\ (a, b)$$

$\Leftrightarrow$ { add id }

$$ap\ (k\ a, id\ b) = f\ (a, b)$$

$\Leftrightarrow$ { composition and $\times$ }

$$(ap \cdot (k \times id))\ (a, b) = f\ (a, b)$$

Curry | uncurry

$$k = \texttt{curry}\ f$$

$\Leftrightarrow$ { isomorphism (last slide) }

$$\texttt{uncurry}\ k = f$$

$\Leftrightarrow$ { add variables }

$$\texttt{uncurry}\ k\ (a, b) = f\ (a, b)$$

$\Leftrightarrow$ { definition of $\texttt{uncurry}\ k$ }

$$k\ a\ b = f\ (a, b)$$

$$(k\ a)\ b = f\ (a, b)$$

$\Leftrightarrow$ { introduce $ap\ (f, x) = f\ x$ }

$$ap\ (k\ a, b) = f\ (a, b)$$

$\Leftrightarrow$ { add id }

$$ap\ (k\ a, id\ b) = f\ (a, b)$$

$\Leftrightarrow$ { composition and $\times$ }

$$(ap \cdot (k \times id))\ (a, b) = f\ (a, b)$$

$\Leftrightarrow$ { drop variables }

$$ap \cdot (k \times id) = f$$

Universal property

$$\begin{tikzcd}
A \times B \to C \arrow[rr, bend left, "\mathtt{curry}"] \arrow[r, phantom, "\cong"] & & A \to C^B \arrow[ll, bend left, "\mathtt{uncurry}"]
\end{tikzcd}$$

$$k = \mathtt{curry}\ f \quad \Leftrightarrow \quad ap \cdot (k \times id) = f$$

Universal property

$$
\begin{tikzcd}
A \times B \to C \arrow[rr, bend left, "\overline{(\_)}"] \arrow[rr, phantom, "\cong"] & & A \to C^B \arrow[ll, bend left, "\widehat{(\_)}"]
\end{tikzcd}
$$

$$k = \overline{f} \;\Leftrightarrow\; ap \cdot (k \times id) = f$$

Diagrams

$$\begin{tikzcd} & C \\ A \times B \arrow[ur, "f"'] & \end{tikzcd}$$

Diagrams

$$\begin{tikzcd} C^B & & C \\ A \arrow[u, "k = \overline{f}"] & A \times B \arrow[ur, "f"'] & \end{tikzcd}$$

$\overline{f}$ abbreviates `curry` f

Diagrams

$$\begin{tikzcd} C^B & \\ A \arrow[u, "k = \overline{f}"] & \end{tikzcd} \qquad \begin{tikzcd} C^B & & C \\ A \times B \arrow[u, "k"] \arrow[urr, "f"'] & & \end{tikzcd}$$

$\overline{f}$ abbreviates `curry` f

Diagrams

$$
\begin{tikzcd}
C^B \\
A \arrow[u, "k = \overline{f}"]
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
C^B & & B & & C \\
A \arrow[u, "k"] & \times & B \arrow[u, "id"'] \arrow[urr, "f"']
\end{tikzcd}
$$

$\overline{f}$ abbreviates `curry` f

Universal property

$$\begin{tikzcd}
C^B & C^B \times B & C \\
A \arrow[u, "k = \overline{f}"] & A \times B \arrow[u, "k \times id"] \arrow[ur, "f"']
\end{tikzcd}$$

Universal property

$$\begin{tikzcd} C^B & C^B \times B \arrow[r, "ap"] & C \\ A \arrow[u, "k = \overline{f}"] & A \times B \arrow[u, "k \times id"] \arrow[ur, "f"'] & \end{tikzcd}$$

$$k = \overline{f} \;\Leftrightarrow\; ap \cdot (k \times id) = f$$

Reflexion $(k := id)$

$$\begin{tikzcd} C^B \\ C^B \arrow[u, "id = \overline{f}"] \end{tikzcd} \qquad \begin{tikzcd} C^B \times B \arrow[r, "ap"] & C \\ C^B \times B \arrow[u, "id \times id"] \arrow[ur, "f"'] & \end{tikzcd}$$

$$id = \overline{f} \Leftrightarrow ap = f$$

Fusion

$$\begin{tikzcd} C^B \\ A \arrow[u, "\overline{f}"] \end{tikzcd} \qquad \begin{tikzcd} C^B \times B \arrow[r, "ap"] & C \\ A \times B \arrow[u, "\overline{f} \times id"] \arrow[ur, "f"'] & \end{tikzcd}$$

Fusion

$$
\begin{tikzcd}
C^B \\
A \arrow[u, "\overline{f}"] \\
D \arrow[u, "g"]
\end{tikzcd}
\qquad
\begin{tikzcd}
C^B \times B \arrow[r, "ap"] & C \\
A \times B \arrow[u, "\overline{f} \times id"] \arrow[ur, "f"'] &
\end{tikzcd}
$$

Fusion

$$
\begin{tikzcd}
C^B \\
A \arrow[u, "\overline{f}"] \\
D \arrow[u, "g"]
\end{tikzcd}
\qquad
\begin{tikzcd}
C^B \times B \arrow[r, "ap"] & C \\
A \times B \arrow[u, "\overline{f} \times id"] \arrow[ur, "f"'] & \\
D \times B \arrow[u, "g \times id"] &
\end{tikzcd}
$$

Fusion

$$\begin{tikzcd}
C^B \\
A \arrow[u, "\overline{f}"] \\
D \arrow[u, "g"]
\end{tikzcd}
\qquad
\begin{tikzcd}
C^B \times B \arrow[rr, "ap"] & & C \\
A \times B \arrow[u, "\overline{f} \times id"] \arrow[urr, "f"'] & & \\
D \times B \arrow[u, "g \times id"] \arrow[uurr, "f \cdot (g \times id)"'] & &
\end{tikzcd}$$

Fusion

$$
\begin{tikzcd}
C^B & C^B \times B \arrow[r, "ap"] & C \\
 & A \times B \arrow[u, "\overline{f} \times id"] \arrow[ur, "f"'] & \\
D \arrow[uu, "\overline{f} \cdot g"] & D \times B \arrow[u, "g \times id"] \arrow[uur, "f \cdot (g \times id)"'] &
\end{tikzcd}
$$

Fusion

$$
\begin{tikzcd}
C^B \\
\\
D \arrow[uu, "\overline{f \cdot (g \times id)}"]
\end{tikzcd}
\qquad
\begin{tikzcd}
C^B \times B \arrow[r, "ap"] & C \\
A \times B \arrow[u, "\overline{f} \times id"] \arrow[ur, "f"'] & \\
D \times B \arrow[u, "g \times id"] \arrow[uur, "f \cdot (g \times id)"'] &
\end{tikzcd}
$$

$$\overline{f} \cdot g = \overline{f \cdot (g \times id)}$$

Finally

$$\begin{tikzcd} C^B \\ A \arrow[u, "\overline{f}"] \end{tikzcd} \qquad \begin{tikzcd} C^B \times B \arrow[r, "ap"] & C \\ A \times B \arrow[u, "\overline{f} \times id"] \arrow[ur, "f"'] \end{tikzcd}$$

Finally

$$
\begin{tikzcd}
C^B \\
A \arrow[u, "\overline{f}"]
\end{tikzcd}
\qquad
\begin{tikzcd}
 & D \\
C^B \times B \arrow[r, "ap"] \arrow[ur, "g \cdot ap"] & C \arrow[u, "g"'] \\
A \times B \arrow[u, "\overline{f} \times id"] \arrow[ur, "f"'] &
\end{tikzcd}
$$

Finally

$$
\begin{tikzcd}
D^B \\
C^B \arrow[u, "g^B"] \\
A \arrow[u, "\overline{f}"]
\end{tikzcd}
\qquad
\begin{tikzcd}
D^B \times B & D \\
C^B \times B \arrow[u, "g^B \times id"] \arrow[ur, "g \cdot ap"] \arrow[r, "ap"] & C \arrow[u, "g"'] \\
A \times B \arrow[u, "\overline{f} \times id"] \arrow[ur, "f"']
\end{tikzcd}
$$

$$g^B = \overline{g \cdot ap}$$

Finally

$$
\begin{tikzcd}
D^B & D^B \times B \arrow[r, "ap"] & D \\
C^B \arrow[u, "g^B"] & C^B \times B \arrow[u, "g^B \times id"] \arrow[r, "ap"] \arrow[ur, "g \cdot ap"] & C \arrow[u, "g"'] \\
A \arrow[u, "\overline{f}"] & A \times B \arrow[u, "\overline{f} \times id"] \arrow[ur, "f"'] &
\end{tikzcd}
$$

$$g^B = \overline{g \cdot ap}$$

Finally

$$\begin{tikzcd}
D^B \\
C^B \arrow[u, "g^B"] \\
A \arrow[u, "\overline{f}"]
\end{tikzcd}
\qquad
\begin{tikzcd}
D^B \times B \arrow[r, "ap"] & D \\
C^B \times B \arrow[u, "g^B \times id"] \arrow[r, "ap"] \arrow[ur, "g \cdot ap"] & C \arrow[u, "g"'] \\
A \times B \arrow[u, "\overline{f} \times id"] \arrow[ur, "f"']
\end{tikzcd}$$

$$g^B = \overline{g \cdot ap}$$

$$g^B\ \overline{f} = \overline{g \cdot f}$$

Summary

$$f \xmapsto{\texttt{curry}} k$$

$$\begin{tikzcd} A \times B \to C \arrow[r, bend left, "\texttt{curry}"] \arrow[r, phantom, "\cong"] & A \to C^B \arrow[l, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

$$f \xleftarrow{\texttt{uncurry}} k$$

Summary

$$f \overset{\texttt{curry}}{\longmapsto} k$$

$$\begin{tikzcd} A \times B \to C \arrow[rr, bend left, "\texttt{curry}"] & \cong & A \to C^B \arrow[ll, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

$$f \overset{\texttt{uncurry}}{\longleftarrow\!\!\!\!\dashv} k$$

Abbreviations:

$$\texttt{curry}\ f = \overline{f}$$

Summary

$$f \xmapsto{\texttt{curry}} k$$

$$\begin{tikzcd} A \times B \to C \arrow[r, bend left, "\texttt{curry}"] \arrow[r, phantom, "\cong"] & A \to C^B \arrow[l, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

$$f \xleftarrow[]{\texttt{uncurry}} k$$

Abbreviations:

$$\texttt{curry}\ f = \overline{f}$$
$$\texttt{uncurry}\ f = \widehat{f}$$

Summary

$$f \xmapsto{\texttt{curry}} k$$

$$\begin{tikzcd} A \times B \to C \arrow[rr, bend left, "\texttt{curry}"] \arrow[rr, phantom, "\cong"] & & A \to C^B \arrow[ll, bend left, "\texttt{uncurry}"] \end{tikzcd}$$

$$f \xleftarrow{\texttt{uncurry}} k$$

Abbreviations:

$$\texttt{curry}\ f = \overline{f}$$
$$\texttt{uncurry}\ f = \widehat{f}$$

Isomorphisms:

$$f = g \Leftrightarrow \overline{f} = \overline{g}$$
$$k = h \Leftrightarrow \widehat{k} = \widehat{h}$$

More isomorphisms

$$a^1 = a \qquad 1^a = 1 \qquad a^0 = 1$$

$$\begin{tikzcd} A^1 \arrow[r, bend left, "g"] \arrow[r, phantom, "\cong"] & A \arrow[l, bend left, "f"] \end{tikzcd} \quad \begin{tikzcd} 1^A \arrow[r, bend left, "g"] \arrow[r, phantom, "\cong"] & 1 \arrow[l, bend left, "f"] \end{tikzcd} \quad \begin{tikzcd} A^0 \arrow[r, bend left, "g"] \arrow[r, phantom, "\cong"] & 1 \arrow[l, bend left, "f"] \end{tikzcd}$$

$$\begin{cases} f\ a = \underline{a} \\ g\ \underline{a} = a \end{cases} \qquad \begin{cases} f\ () = ! \\ g = ! \end{cases} \qquad \begin{cases} f\ () = \bot \\ g = ! \end{cases}$$

(Even more) isomorphisms

$$\begin{tikzcd} A \times A \arrow[rr, bend left, "g"] \arrow[rr, phantom, "\cong"] & & A^2 \arrow[ll, bend left, "f"] \end{tikzcd}$$

$$f\ k = (k\ F, k\ T)$$

(Even more) isomorphisms

$$\begin{tikzcd} A \times A \arrow[rr, bend left, "g"] \arrow[rr, phantom, "\cong"] & & A^2 \arrow[ll, bend left, "f"] \end{tikzcd}$$

$$f\ k = (k\ F, k\ T)$$

$$\begin{cases} g\ (a_1, a_2)\ F = a_1 \\ g\ (a_1, a_2)\ T = a_2 \end{cases}$$

cf. $a \times a = a^2$ 

(Even more) isomorphisms

(Even more) isomorphisms

$$c^{a+b} = c^a \times c^b$$

(Even more) isomorphisms

$$c^{a+b} = c^a \times c^b \qquad (a \times b)^c = a^c \times b^c$$

(Even more) isomorphisms

$$c^{a+b} = c^a \times c^b \qquad (a \times b)^c = a^c \times b^c$$

$$\begin{tikzcd} C^{A+B} \arrow[r, bend left, "g"] \arrow[r, phantom, "\cong"] & C^A \times C^B \arrow[l, bend left, "f"] \end{tikzcd}$$

(Even more) isomorphisms

$$c^{a+b} = c^a \times c^b \qquad (a \times b)^c = a^c \times b^c$$

$$\begin{tikzcd} C^{A+B} \arrow[r, bend left, "g"] \arrow[r, phantom, "\cong"] & C^A \times C^B \arrow[l, bend left, "f"] \end{tikzcd} \qquad \begin{tikzcd} (A \times B)^C \arrow[r, bend left, "g"] \arrow[r, phantom, "\cong"] & A^C \times B^C \arrow[l, bend left, "f"] \end{tikzcd}$$

(Even more) isomorphisms

$$c^{a+b} = c^a \times c^b \qquad (a \times b)^c = a^c \times b^c$$

$$\begin{tikzcd} C^{A+B} \arrow[r, phantom, "\cong"] \arrow[r, bend left, "g"] & C^A \times C^B \arrow[l, bend left, "f"] \end{tikzcd} \qquad \begin{tikzcd} (A \times B)^C \arrow[r, phantom, "\cong"] \arrow[r, bend left, "g"] & A^C \times B^C \arrow[l, bend left, "f"] \end{tikzcd}$$

$$\begin{cases} f\ (h, k) = [h, k] \\ g\ k = (k \cdot i_1, k \cdot i_2) \end{cases}$$

(Even more) isomorphisms

$$c^{a+b} = c^a \times c^b \qquad (a \times b)^c = a^c \times b^c$$

$$\begin{tikzcd} C^{A+B} \arrow[r, bend left, "g"] \arrow[r, phantom, "\cong"] & C^A \times C^B \arrow[l, bend left, "f"] \end{tikzcd} \qquad \begin{tikzcd} (A \times B)^C \arrow[r, bend left, "g"] \arrow[r, phantom, "\cong"] & A^C \times B^C \arrow[l, bend left, "f"] \end{tikzcd}$$

$$\begin{cases} f\ (h, k) = [h, k] \\ g\ k = (k \cdot i_1, k \cdot i_2) \end{cases} \qquad \begin{cases} f\ (h, k) = \langle h, k \rangle \\ g\ k = (\pi_1 \cdot k, \pi_2 \cdot k) \end{cases}$$

Summary

$f \cdot g$ sequential

Summary

$f \cdot g$ sequential

$\langle f, g \rangle$

Summary

$f \cdot g$ sequential

$\langle f, g \rangle, \ f \times g$ parallel

Summary

$f \cdot g$ sequential

$\langle f, g \rangle$, $f \times g$ parallel

$[f, g]$

Summary

$f \cdot g$ sequential

$\langle f, g \rangle$, $f \times g$ parallel

$[f, g]$, $f + g$

Summary

$f \cdot g$ sequential

$\langle f, g \rangle$, $f \times g$ parallel

$[f, g]$, $f + g$, $p \to f\ ,\ g$ alternation

Summary

$f \cdot g$ sequential

$\langle f, g \rangle$, $f \times g$ parallel

$[f, g]$, $f + g$, $p \to f\ ,\ g$ alternation

$\overline{f}$

Summary

$f \cdot g$ sequential

$\langle f, g\rangle$, $f \times g$ parallel

$[f, g]$, $f + g$, $p \to f , g$ alternation

$\overline{f}$ higher order

Summary

$f \cdot g$ sequential

$\langle f, g \rangle$, $f \times g$ parallel

$[f, g]$, $f + g$, $p \rightarrow f \, , \, g$ alternation

$\overline{f}$ higher order

Compositional programming

Summary

$A \times B$ records ("structs")

Summary

$A \times B$ records ("structs")

$A + B$ variant records ("unions")

Summary

$A \times B$	records (“structs”)
$A + B$	variant records (“unions”)
$1 + A$	“pointers”

Summary

$A \times B$ records ("structs")

$A + B$ variant records ("unions")

$1 + A$ "pointers"

B^A "arrays", "streams"

Summary

$A \times B$	records ("structs")
$A + B$	variant records ("unions")
$1 + A$	"pointers"
B^A	"arrays", "streams"

data structuring

Summary

$A \times B$ records (“structs”)

$A + B$ variant records (“unions”)

$1 + A$ “pointers”

B^A “arrays” , “streams”

data structuring

$1 + A \times X^2$ polynomial structures (coming up soon)

Precedence rules

In order to save parentheses in pointfree expressions:

- **unary** operators take precedence over binary ones ;
- **composition** binds tighter than any other binary operator;
- **product** $(f \times g)$ binds tighter than **coproduct** $(f + g)$.

Analogy...

Our small set of **combinators** resembles

> **RISC** — *Reduced instruction-set computer*

an idea championed by Prof. David Patterson (UCB, Google)

Analogy...

[... and since you are here, don't miss reading Prof. Patterson's "16 life lessons" (+ "9 magic words"), a recent post in CACM, 10-Oct.]

Part II

Where do programs come from?

What is a program?

Where do programs come from?

What is an **algorithm**?

Where do programs come from?

What is an **algorithm**?

> *"Detailed* **instructions** *defining a* **computational process** *(which is then said to be* **algorithmic***), which begins with an arbitrary* **input** *(...), and with instructions aimed at obtaining a result (or* **output***) which is fully determined by the input."*

(https://encyclopediaofmath.org/index.php?title=Algorithm)

Where do programs come from?

The algorithms we know were born from what?

Where do programs come from?

The algorithms we know were born from what?

For instance, **quicksort**?

(See video 05b, t=3.25)

Where do programs come from?

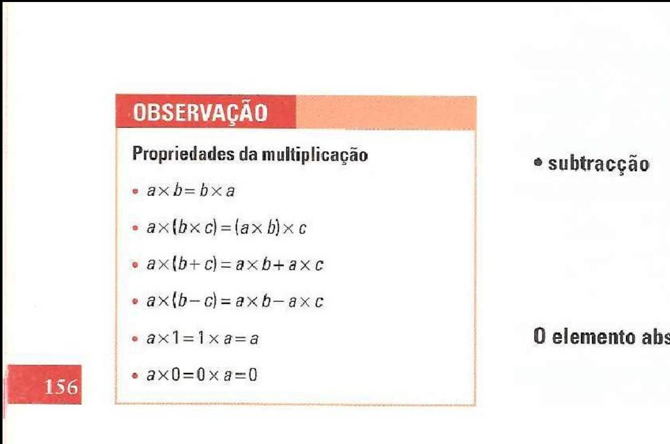
What is an **algorithm**?

> *"(...) For instance, the rules taught in elementary schools for column-wise* **addition**, **subtraction**, **multiplication** *and* **division** *are algorithms; (...)*

(`https://encyclopediaofmath.org/index.php?title=Algorithm`)

Where do programs come from?

From a Y7 maths textbook:



OBSERVAÇÃO

Propriedades da multiplicação

- $a \times b = b \times a$
- $a \times (b \times c) = (a \times b) \times c$
- $a \times (b + c) = a \times b + a \times c$
- $a \times (b - c) = a \times b - a \times c$
- $a \times 1 = 1 \times a = a$
- $a \times 0 = 0 \times a = 0$

156

- subtracção

O elemento abs

Where do programs come from?

$$\begin{cases} a \times 0 = 0 \\ a \times 1 = a \\ a \times (b + c) = a \times b + a \times c \end{cases}$$

Where do programs come from?

$$\begin{cases} a \times 0 = 0 \\ a \times 1 = a \\ a \times (b + 1) = a \times b + a \times 1 \end{cases}$$

Where do programs come from?

$$\begin{cases} a \times 0 = 0 \\ a \times 1 = a \\ a \times (b+1) = a \times b + a \end{cases}$$

Where do programs come from?

$$\begin{cases} a \times 0 = 0 \\ a \times 1 = a \\ a \times (0 + 1) = a \times 0 + a \end{cases}$$

Where do programs come from?

$$\begin{cases} a \times 0 = 0 \\ a \times 1 = a \\ a \times 1 = a \times 0 + a \end{cases}$$

Where do programs come from?

$$\begin{cases} a \times 0 = 0 \\ a \times 1 = a \\ a \times 1 = 0 + a \end{cases}$$

Where do programs come from?

$$\begin{cases} a \times 0 = 0 \\ a \times 1 = a \\ a \times 1 = a \end{cases}$$

Where do programs come from?

$$\begin{cases} a \times 0 = 0 \\ a \times 1 = a \\ a \times (b + 1) = a \times b + a \end{cases}$$

Where do programs come from?

$$\begin{cases} a \times 0 = 0 \\ \\ a \times (b+1) = a \times b + a \end{cases}$$

What **is** a program?

$$\begin{cases} a \times 0 = 0 \\ a \times (b + 1) = a + a \times b \end{cases}$$

What **is** a program?

$$\begin{cases} (a\times)\ 0 = 0 \\ \\ (a\times)\ (b+1) = a + (a\times)\ b \end{cases}$$

What **is** a program?

$$\begin{cases} (a\times)\;0 = 0 \\ \\ (a\times)\;(b+1) = (a+)\;((a\times)\;b) \end{cases}$$

What **is** a program?

$$\begin{cases} (a\times)\ (\underline{0}\ x) = \underline{0}\ x \\ \\ (a\times)\ (b+1) = (a+)\ ((a\times)\ b) \end{cases}$$

What **is** a program?

$$\begin{cases} (a\times)\ (\underline{0}\ x) = \underline{0}\ x \\ \\ (a\times)\ (\mathit{succ}\ b) = (a+)\ ((a\times)\ b) \end{cases}$$

What **is** a program?

$$\begin{cases} ((a\times)\cdot\underline{0})\ x = \underline{0}\ x \\ \\ (a\times)\ (succ\ b) = (a+)\ ((a\times)\ b) \end{cases}$$

What **is** a program?

$$\begin{cases} ((a\times)\cdot \underline{0})\ x = \underline{0}\ x \\ \\ ((a\times)\cdot \mathit{succ})\ b = (a+)\ ((a\times)\ b) \end{cases}$$

What **is** a program?

$$\begin{cases} ((a\times)\cdot\underline{0})\ x = \underline{0}\ x \\ \\ ((a\times)\cdot \mathit{succ})\ b = ((a+)\cdot(a\times))\ b \end{cases}$$

What **is** a program?

$$
\begin{cases}
(a\times)\cdot\underline{0} = \underline{0} \\[2em]
((a\times)\cdot \mathit{succ})\ b = ((a+)\cdot(a\times))\ b
\end{cases}
$$

What **is** a program?

$$\begin{cases} (a\times)\cdot \underline{0} = \underline{0} \\ \\ (a\times)\cdot \mathit{succ} = (a+)\cdot(a\times) \end{cases}$$

What **is** a program?

$$[(a\times)\cdot \underline{0}, (a\times)\cdot \mathit{succ}] = [\underline{0}, (a+)\cdot(a\times)]$$

What **is** a program?

$$(a\times)\cdot[\underline{0}, succ] = [\underline{0}, (a+)\cdot(a\times)]$$

What **is** a program?

$$(a\times)\cdot[\underline{0}, succ] = [\underline{0}\cdot id, (a+)\cdot(a\times)]$$

What **is** a program?

$$(a\times)\cdot[\underline{0}, succ] = [\underline{0}, (a+)]\cdot(id + (a\times))$$

$$(a\times)\cdot[\underline{0}, succ] = [\underline{0}, (a+)]\cdot(id + (a\times))$$

$$\begin{array}{c} ? \\ {\scriptstyle (a\times)}\Big\downarrow \\ ? \end{array}$$

$$(a\times)\cdot[\underline{0}, succ] = [\underline{0}, (a+)]\cdot(id + (a\times))$$

$$\begin{tikzcd}
? \arrow[d, "(a\times)"'] & & ? \arrow[ll, "{[\underline{0}, succ]}"'] \\
?
\end{tikzcd}$$

$$(a\times)\cdot[\underline{0}, succ] = [\underline{0}, (a+)]\cdot(id + (a\times))$$

$$\begin{tikzcd}
? \arrow[d, "(a\times)"'] & & ? \arrow[ll, "{[\underline{0}, succ]}"'] \\
? & & ? \arrow[ll, "{[\underline{0}, (a+)]}"]
\end{tikzcd}$$

$$(a\times)\cdot[\underline{0}, succ] = [\underline{0}, (a+)]\cdot(id + (a\times))$$

$$\begin{tikzcd}
? \arrow[d, "(a\times)"'] & & ? \arrow[ll, "{[\underline{0}, succ]}"'] \arrow[d, "id + (a\times)"] \\
? & & ? \arrow[ll, "{[\underline{0}, (a+)]}"]
\end{tikzcd}$$

A program is a "rectangle" 

$$
\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "(a\times)"'] & ? \arrow[l, "{[\underline{0}, succ]}"'] \arrow[d, "id + (a\times)"] \\
\mathbb{N}_0 & ? \arrow[l, "{[\underline{0}, (a+)]}"]
\end{tikzcd}
$$

A program is a "rectangle"

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "(a\times)"'] & ? + ? \arrow[l, "{[\underline{0}, succ]}"'] \arrow[d, "id + (a\times)"] \\
\mathbb{N}_0 & ? + ? \arrow[l, "{[\underline{0}, (a+)]}"]
\end{tikzcd}$$

A program is a "rectangle"

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "(a\times)"'] & ? + \mathbb{N}_0 \arrow[l, "{[\underline{0}, succ]}"'] \arrow[d, "id + (a\times)"] \\
\mathbb{N}_0 & ? + \mathbb{N}_0 \arrow[l, "{[\underline{0}, (a+)]}"]
\end{tikzcd}$$

A program is a "rectangle"

$$
\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "(a\times)"'] & A + \mathbb{N}_0 \arrow[l, "{[\underline{0}, succ]}"'] \arrow[d, "id + (a\times)"] \\
\mathbb{N}_0 & A + \mathbb{N}_0 \arrow[l, "{[\underline{0}, (a+)]}"]
\end{tikzcd}
$$

Does $[\underline{0}, succ]^\circ$ exist?

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[r, "{[\underline{0}, succ]^\circ}"] \arrow[d, "(a\times)"'] & A + \mathbb{N}_0 \arrow[d, "id + (a\times)"] \\
\mathbb{N}_0 & A + \mathbb{N}_0 \arrow[l, "{[\underline{0}, (a+)]}"]
\end{tikzcd}$$

Does $[\underline{0}, succ]^\circ$ exist?

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "(a\times)"'] & A + \mathbb{N}_0 \arrow[l, "{[\underline{0}, succ]}"'] \arrow[d, "id + (a\times)"] \\
\mathbb{N}_0 & A + \mathbb{N}_0 \arrow[l, "{[\underline{0}, (a+)]}"]
\end{tikzcd}$$

Does $[\underline{0}, succ]^{\circ}$ exist?

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[rr, "{[\underline{0}, succ]^{\circ}}"] \arrow[d, "(a\times)"'] & & A + \mathbb{N}_0 \arrow[d, "id + (a\times)"] \\
\mathbb{N}_0 & & A + \mathbb{N}_0 \arrow[ll, "{[\underline{0}, (a+)]}"]
\end{tikzcd}$$

Does $[\underline{0}, succ]^{\circ}$ exist?

$$
\begin{tikzcd}
\mathbb{N}_0 \arrow[r, "{[\underline{0}, succ]^{\circ}}"] \arrow[d, "(a\times)"'] & A + \mathbb{N}_0 \arrow[d, "id + (a\times)"] \\
\mathbb{N}_0 & A + \mathbb{N}_0 \arrow[l, "{[\underline{0}, (a+)]}"]
\end{tikzcd}
$$

$$(a\times) = [\underline{0}, (a+)] \cdot (id + (a\times)) \cdot [\underline{0}, succ]^{\circ} \quad (?)$$

Does $[\underline{0}, succ]^\circ$ exist?

Conjecture: $\mathbf{out} : \mathbb{N}_0 \to A + \mathbb{N}_0$ exists and is $[\underline{0}, succ]^\circ$

$$\mathbf{out} \cdot [\underline{0}, succ] = id$$

Does $[\underline{0}, succ]^\circ$ exist?

Conjecture: $\mathbf{out} : \mathbb{N}_0 \to A + \mathbb{N}_0$ exists and is $[\underline{0}, succ]^\circ$

$$
\begin{array}{cl}
& \mathbf{out} \cdot [\underline{0}, succ] = id \\
\Leftrightarrow & \{ \ +\text{-fusion} \ \} \\
& [\mathbf{out} \cdot \underline{0}, \mathbf{out} \cdot succ] = id
\end{array}
$$

Does $[\underline{0}, succ]^{\circ}$ exist?

Conjecture: $\mathbf{out} : \mathbb{N}_0 \to A + \mathbb{N}_0$ exists and is $[\underline{0}, succ]^{\circ}$

$$
\begin{array}{ll}
 & \mathbf{out} \cdot [\underline{0}, succ] = id \\
\Leftrightarrow & \quad \{ \ +\text{-fusion} \ \} \\
 & [\mathbf{out} \cdot \underline{0}, \mathbf{out} \cdot succ] = id \\
\Leftrightarrow & \quad \{ \ \text{universal-}+ \ \} \\
 & \left\{ \begin{array}{l} \mathbf{out} \cdot \underline{0} = i_1 \\ \mathbf{out} \cdot succ = i_2 \end{array} \right.
\end{array}
$$

Does $[\underline{0}, \mathit{succ}]^{\circ}$ **exist?**

$$\begin{cases} \mathbf{out} \cdot \underline{0} = i_1 \\ \mathbf{out} \cdot \mathit{succ} = i_2 \end{cases}$$

Does $[\underline{0}, succ]^\circ$ exist?

$$\left\{\begin{array}{l} \mathbf{out} \cdot \underline{0} = i_1 \\ \mathbf{out} \cdot succ = i_2 \end{array}\right.$$

$$\Leftrightarrow \qquad \{ \text{ introduce variables } \}$$

$$\left\{\begin{array}{l} \mathbf{out}\ (\underline{0}\ x) = i_1\ x \\ \mathbf{out}\ (succ\ x) = i_2\ x \end{array}\right.$$

Does $[\underline{0}, succ]^{\circ}$ exist?

$$\begin{cases} \mathbf{out} \cdot \underline{0} = i_1 \\ \mathbf{out} \cdot succ = i_2 \end{cases}$$

$$\Leftrightarrow \qquad \{ \text{ introduce variables } \}$$

$$\begin{cases} \mathbf{out}\ (\underline{0}\ x) = i_1\ x \\ \mathbf{out}\ (succ\ x) = i_2\ x \end{cases}$$

$$\Leftrightarrow \qquad \{ \text{ simplificar } \}$$

$$\begin{cases} \mathbf{out}\ 0 = i_1\ x \\ \mathbf{out}\ (x + 1) = i_2\ x \end{cases}$$

Does $[\underline{0}, succ]^{\circ}$ exist?

$$\begin{cases} \mathbf{out} \cdot \underline{0} = i_1 \\ \mathbf{out} \cdot succ = i_2 \end{cases}$$

$\Leftrightarrow \quad \{ \text{ introduce variables } \}$

$$\begin{cases} \mathbf{out}\ (\underline{0}\ x) = i_1\ x \\ \mathbf{out}\ (succ\ x) = i_2\ x \end{cases}$$

$\Leftrightarrow \quad \{ \text{ simplificar } \}$

$$\begin{cases} \mathbf{out}\ 0 = i_1\ x \\ \mathbf{out}\ (x+1) = i_2\ x \end{cases}$$

$$\begin{cases} \mathbf{out}\ 0 = i_1\ x \\ \mathbf{out}\ (x+1) = i_2\ x \end{cases}$$

?

Problem

$$\left\{\begin{array}{l} \mathbf{out}\ 0 = i_1\ x \\ \mathbf{out}\ (x+1) = i_2\ x \end{array}\right.$$

$$\begin{tikzcd} \mathbb{N}_0 \arrow[rr, bend left] \arrow[rr, phantom, "?"] & & A + \mathbb{N}_0 \arrow[ll, bend left, "\mathbf{in}=[\underline{0},succ]"] \end{tikzcd}$$

?

Solution

$$\begin{tikzcd} \mathbb{N}_0 \arrow[rr, bend left, "{\begin{cases} \mathbf{out}\ 0 = i_1\ () \\ \mathbf{out}\ (x+1) = i_2\ x \end{cases}}"] \arrow[rr, phantom, "\cong"] && 1 + \mathbb{N}_0 \arrow[ll, bend left, "{\mathbf{in}=[\underline{0}, succ]}"] \end{tikzcd}$$

Solution

$$\begin{cases} \mathbf{out}\ 0 = i_1\ () \\ \mathbf{out}\ (x+1) = i_2\ x \end{cases}$$

$$\begin{tikzcd}[column sep=huge, row sep=large]
\mathbb{N}_0 \arrow[r, bend left] \arrow[d, "(a\times)"'] \arrow[r, phantom, "\cong"] & 1 + \mathbb{N}_0 \arrow[l, bend left, "\mathbf{in}=[\underline{0}, succ]"] \arrow[d, "id+(a\times)"] \\
\mathbb{N}_0 & 1 + \mathbb{N}_0 \arrow[l, bend left, "{[\underline{0}, (a+)]}"]
\end{tikzcd}$$

Diagram

$$
\begin{tikzcd}[column sep=huge, row sep=large]
\mathbb{N}_0 \arrow[r, bend left, "\mathbf{out}"] \arrow[r, phantom, "\cong"] \arrow[d, "(a\times)"'] & 1 + \mathbb{N}_0 \arrow[l, bend left, "\mathbf{in}"] \arrow[d, "id + (a\times)"] \\
\mathbb{N}_0 & 1 + \mathbb{N}_0 \arrow[l, bend left, "{[\underline{0}, a+]}"]
\end{tikzcd}
$$

$$(a\times) = [\underline{0}, (a+)] \cdot (id + (a\times)) \cdot \mathbf{out}$$

Renaming

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
\mathbb{N}_0 \arrow[r, bend left, "\mathsf{out}"] \arrow[r, phantom, "\cong"] \arrow[d, "k"'] & 1 + \mathbb{N}_0 \arrow[l, bend left, "\mathsf{in}"] \arrow[d, "id + k"] \\
\mathbb{N}_0 & 1 + \mathbb{N}_0 \arrow[l, bend left, "{[\underline{0}, a+]}"]
\end{tikzcd}
$$

$$(a\times) = k \ \textbf{where} \ k = [\underline{0}, (a+)] \cdot (id + k) \cdot \mathsf{out}$$

Variations in diagram

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[rr, bend left, "\mathsf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "k"'] & & 1 + \mathbb{N}_0 \arrow[ll, bend left, "\mathsf{in}"] \arrow[d, "id + k"] \\
\mathbb{N}_0 & & 1 + \mathbb{N}_0 \arrow[ll, bend left, "{[\underline{1}, (a\times)]}"]
\end{tikzcd}$$

Variations in diagram

$$\begin{tikzcd}[column sep=huge, row sep=large]
\mathbb{N}_0 \arrow[r, bend left, "\mathsf{out}"] \arrow[r, phantom, "\cong"] \arrow[d, "k"'] & 1 + \mathbb{N}_0 \arrow[l, bend left, "\mathsf{in}"] \arrow[d, "id+k"] \\
\mathbb{N}_0 & 1 + \mathbb{N}_0 \arrow[l, bend left, "{[\underline{1}, (a\times)]}"]
\end{tikzcd}$$

$$a^b = k \textbf{ where } k = [\underline{1}, (a\times)] \cdot (id + k) \cdot \mathsf{out}$$

Variations in diagram

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "k"'] & & 1 + \mathbb{N}_0 \arrow[ll, bend left, "\mathbf{in}"] \arrow[d, "id+k"] \\
\mathbb{N}_0 & & 1 + \mathbb{N}_0 \arrow[ll, bend left, "{[\underline{1}, (a\times)]}"]
\end{tikzcd}$$

$$\begin{cases} a^0 = 1 \\ a^{b+1} = a \times a^b \end{cases}$$

$$a^b = k \textbf{ where } k = [\underline{1}, (a\times)] \cdot (id + k) \cdot \mathbf{out}$$

Generalizing the diagram

$$\begin{tikzcd}[column sep=huge, row sep=large]
\mathbb{N}_0 \arrow[r, bend left, "\mathsf{out}"] \arrow[r, phantom, "\cong"] \arrow[d, "k\ g"'] & 1 + \mathbb{N}_0 \arrow[l, bend left, "\mathsf{in}"] \arrow[d, "id+k\ g"] \\
A & 1 + A \arrow[l, bend left, "g"]
\end{tikzcd}$$

$$k\ g = g \cdot (id + k\ g) \cdot \mathsf{out}$$

Cálculo de Programas

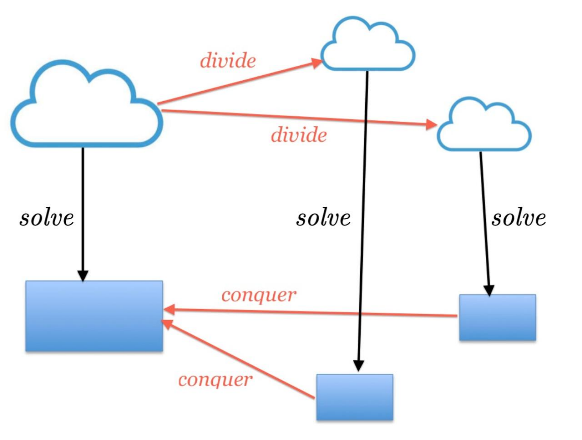
Class T06 — planned for 31-Oct

Generalizing the diagram

$$
\begin{tikzcd}
\mathbb{N}_0 \arrow[r, bend left, "\mathbf{out}"] \arrow[r, phantom, "\cong"] \arrow[d, "(\!| g |\!)"'] & 1+\mathbb{N}_0 \arrow[l, bend left, "\mathbf{in}"] \arrow[d, "id+(\!| g |\!)"] \\
A & 1+A \arrow[l, bend left, "g"]
\end{tikzcd}
$$

$$(\!| g |\!) = g \cdot (id + (\!| g |\!)) \cdot \mathbf{out}$$

Anticipation of what is to come...



Catamorphism

$(\!| g |\!)$

$$\underbrace{\mathit{cata}}_{\mathit{downwards}} \; (\kappa\alpha\tau\alpha)$$

Catamorphism

$(\!| g |\!)$

$$\underbrace{\textit{cata}}_{\textit{downwards}} \; (\kappa\alpha\tau\alpha) + \underbrace{\textit{morphism}}_{\text{transformation}} \; (\mu o \rho \phi \iota \sigma \mu o \zeta)$$

"Downwards transformation"

Definition

$$\begin{tikzcd}[column sep=huge, row sep=huge]
\mathbb{N}_0 \arrow[r, "\mathsf{out}"] \arrow[d, swap, "(\!| g |\!)"] & 1 + \mathbb{N}_0 \arrow[d, "id + (\!| g |\!)"] \\
A & 1 + A \arrow[l, "g"]
\end{tikzcd}$$

$$(\!| g |\!) = g \cdot (id + (\!| g |\!)) \cdot \mathsf{out}$$

Property

$$\begin{tikzcd} \mathbb{N}_0 \arrow[d, "(\!| g |\!)"'] & 1 + \mathbb{N}_0 \arrow[l, "\mathsf{in}"'] \arrow[d, "id + (\!| g |\!)"] \\ A & 1 + A \arrow[l, "g"] \end{tikzcd}$$

$$(\!| g |\!) \cdot \mathsf{in} = g \cdot (id + (\!| g |\!))$$

Property

$$
\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "(\!| g |\!)"'] & & 1 + \mathbb{N}_0 \arrow[ll, "\mathbf{in}"'] \arrow[d, "id + (\!| g |\!)"] \\
A & & 1 + A \arrow[ll, "g"]
\end{tikzcd}
$$

$$k = (\!| g |\!) \Rightarrow k \cdot \mathbf{in} = g \cdot (id + k)$$

Uniqueness

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "(\!|\, g \,|\!)"'] & 1 + \mathbb{N}_0 \arrow[l, "\mathsf{in}"'] \arrow[d, "id + (\!|\, g \,|\!)"] \\
A & 1 + A \arrow[l, "g"]
\end{tikzcd}$$

$$k = (\!|\, g \,|\!) \Leftarrow k \cdot \mathsf{in} = g \cdot (id + k)$$

Universal property

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "k"'] & 1 + \mathbb{N}_0 \arrow[l, "\mathbf{in}"'] \arrow[d, "id+k"] \\
A & 1 + A \arrow[l, "g"]
\end{tikzcd}$$

$$k = (\!|\, g \,|\!) \quad \Leftrightarrow \quad k \cdot \mathbf{in} = g \cdot (id + k)$$

Universal-$\times$:

$$k = \langle f, g \rangle \;\Leftrightarrow\; \left\{ \begin{array}{l} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{array} \right.$$

Universal-$\times$:

$$k = \langle f, g \rangle \;\Leftrightarrow\; \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases}$$

Universal-$+$:

$$k = [f, g] \;\Leftrightarrow\; \begin{cases} k \cdot i_1 = f \\ k \cdot i_2 = g \end{cases}$$

Universal-$\times$:

$$k = \langle f, g \rangle \;\Leftrightarrow\; \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases}$$

Universal-$+$:

$$k = [f, g] \;\Leftrightarrow\; \begin{cases} k \cdot i_1 = f \\ k \cdot i_2 = g \end{cases}$$

Universal-expo:

$$k = \overline{f} \;\Leftrightarrow\; ap \cdot (k \times id) = f$$

Universal-$\times$:

$$k = \langle f , g \rangle \;\Leftrightarrow\; \left\{ \begin{array}{l} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{array} \right.$$

Universal-$+$:

$$k = [f , g] \;\Leftrightarrow\; \left\{ \begin{array}{l} k \cdot i_1 = f \\ k \cdot i_2 = g \end{array} \right.$$

Universal-expo:

$$k = \overline{f} \;\Leftrightarrow\; ap \cdot (k \times id) = f$$

Cata-Universal:

$$k = (\!| g |\!) \;\Leftrightarrow\; k \cdot \mathbf{in} = g \cdot (id + k)$$

Cata-universal ($\mathbb{N}_0$)

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "k"'] & 1 + \mathbb{N}_0 \arrow[l, "\mathbf{in}"'] \arrow[d, "id+k"] \\
A & 1 + A \arrow[l, "g"]
\end{tikzcd}$$

$$k = (\!| g |\!) \Leftrightarrow k \cdot \mathbf{in} = g \cdot (id + k)$$

Cata-reflexion

To be derived from

$$k = (\!| g |\!) \Leftrightarrow k \cdot \mathbf{in} = g \cdot (id + k)$$

Cata-reflexion ($k := id$)

$$id = (\!| g |\!) \Leftrightarrow id \cdot \mathbf{in} = g \cdot (id + id)$$

Cata-reflexion ($k := id$)

$$id = (\!| g |\!) \Leftrightarrow \mathbf{in} = g$$

Cata-reflexion

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "id = (\!|\,\mathbf{in}\,|\!)"'] & 1 + \mathbb{N}_0 \arrow[l, "\mathbf{in} = [\underline{0}, succ]"'] \arrow[d, "id + (\!|\,\mathbf{in}\,|\!)"] \\
\mathbb{N}_0 & 1 + \mathbb{N}_0 \arrow[l, "\mathbf{in} = [\underline{0}, succ]"]
\end{tikzcd}$$

$$(\!|\,\mathbf{in}\,|\!) = id$$

Cata-cancellation

Another corollary of

$$k = (\!|\, g \,|\!) \Leftrightarrow k \cdot \mathbf{in} = g \cdot (id + k)$$

for $k := (\!|\, g \,|\!)$.

Cata-cancellation $(k := (\!| g |\!))$

$$(\!| g |\!) = (\!| g |\!) \Leftrightarrow (\!| g |\!) \cdot \mathbf{in} = g \cdot (id + (\!| g |\!))$$

Cata-cancellation

$$(\!| g |\!) \cdot \mathsf{in} = g \cdot (id + (\!| g |\!))$$

Cata-definition

$$(\!|\, g \,|\!) = g \cdot (id + (\!|\, g \,|\!)) \cdot \mathbf{out}$$

Cata-definition

$$\begin{tikzcd} \mathbb{N}_0 \arrow[r, "\mathsf{out}"] \arrow[d, "(\!| g |\!)"'] & 1 + \mathbb{N}_0 \arrow[d, "id + (\!| g |\!)"] \\ A & 1 + A \arrow[l, "g"] \end{tikzcd}$$

$$(\!| g |\!) = g \cdot (id + (\!| g |\!)) \cdot \mathsf{out}$$

Cata-fusion

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "(\!| g |\!)"'] & 1 + \mathbb{N}_0 \arrow[l, "\mathbf{in}=[\underline{0}, succ]"'] \arrow[d, "id+(\!| g |\!)"] \\
A \arrow[d, "f"'] & 1 + A \arrow[l, "g"] \arrow[d, "id+f"] \\
B & 1 + B \arrow[l, "h"]
\end{tikzcd}$$

Cata-fusion

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "(\!| g |\!)"'] \arrow[dd, bend right, "(\!| h |\!)"'] & & 1 + \mathbb{N}_0 \arrow[ll, "\mathbf{in} = [\underline{0}, succ]"'] \arrow[d, "id + (\!| g |\!)"] \\
A \arrow[d, "f"'] & & 1 + A \arrow[ll, "g"] \arrow[d, "id + f"] \\
B & & 1 + B \arrow[ll, "h"]
\end{tikzcd}$$

Cata-fusion

$$f \cdot (\!| g |\!) = (\!| h |\!)$$

Cata-fusion

$$
\begin{aligned}
& f \cdot (\!| g |\!) = (\!| h |\!) \\
\Leftrightarrow \quad & \{ \text{ Cata-universal for } k = f \cdot (\!| g |\!) \ \} \\
& f \cdot (\!| g |\!) \cdot \mathbf{in} = h \cdot (id + f \cdot (\!| g |\!))
\end{aligned}
$$

Cata-fusion

$$
\begin{array}{ll}
 & f \cdot (\!| g |\!) = (\!| h |\!) \\
\Leftrightarrow & \{ \text{ Cata-universal for } k = f \cdot (\!| g |\!) \ \} \\
 & f \cdot (\!| g |\!) \cdot \mathbf{in} = h \cdot (id + f \cdot (\!| g |\!)) \\
\Leftrightarrow & \{ \text{ Cata-cancellation } \} \\
 & f \cdot g \cdot (id + (\!| g |\!)) = h \cdot (id + f \cdot (\!| g |\!))
\end{array}
$$

Cata-fusion

$$
\begin{array}{ll}
& f \cdot (\!|\, g \,|\!) = (\!|\, h \,|\!) \\
\Leftrightarrow & \quad \{ \text{ Cata-universal for } k = f \cdot (\!|\, g \,|\!) \ \} \\
& f \cdot (\!|\, g \,|\!) \cdot \mathbf{in} = h \cdot (id + f \cdot (\!|\, g \,|\!)) \\
\Leftrightarrow & \quad \{ \text{ Cata-cancellation } \} \\
& f \cdot g \cdot (id + (\!|\, g \,|\!)) = h \cdot (id + f \cdot (\!|\, g \,|\!)) \\
\Leftrightarrow & \quad \{ \ id\text{-natural } \} \\
& f \cdot g \cdot (id + (\!|\, g \,|\!)) = h \cdot (id \cdot id + f \cdot (\!|\, g \,|\!))
\end{array}
$$

Cata-fusion

$$f \cdot g \cdot (id + (\!|\, g \,|\!)) = h \cdot (id \cdot id + f \cdot (\!|\, g \,|\!))$$

Cata-fusion

$$
\begin{array}{ll}
 & f \cdot g \cdot (id + (\!| g |\!)) = h \cdot (id \cdot id + f \cdot (\!| g |\!)) \\
\Leftrightarrow & \qquad \{ \text{ functor-}+ \text{ } \} \\
 & f \cdot g \cdot (id + (\!| g |\!)) = h \cdot (id + f) \cdot (id + (\!| g |\!))
\end{array}
$$

Cata-fusion

$$
\begin{array}{ll}
 & f \cdot g \cdot (id + (\!|\, g \,|\!)) = h \cdot (id \cdot id + f \cdot (\!|\, g \,|\!)) \\
\Leftrightarrow & \quad \{ \ \text{functor-}+ \ \} \\
 & f \cdot g \cdot (id + (\!|\, g \,|\!)) = h \cdot (id + f) \cdot (id + (\!|\, g \,|\!)) \\
\Leftarrow & \quad \{ \ \text{Leibniz} \ \} \\
 & f \cdot g = h \cdot (id + f)
\end{array}
$$

Cata-fusion

$$
\begin{array}{cl}
& f \cdot g \cdot (id + (\!| g |\!)) = h \cdot (id \cdot id + f \cdot (\!| g |\!)) \\
\Leftrightarrow & \qquad \{ \text{ functor-}+ \ \} \\
& f \cdot g \cdot (id + (\!| g |\!)) = h \cdot (id + f) \cdot (id + (\!| g |\!)) \\
\Leftarrow & \qquad \{ \text{ Leibniz } \} \\
& f \cdot g = h \cdot (id + f)
\end{array}
$$

Summing up:

$$
f \cdot (\!| g |\!) = (\!| h |\!) \quad \Leftarrow \quad f \cdot g = h \cdot (id + f)
$$

Cata-fusion

$$
\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "\llparenthesis g \rrparenthesis"'] \arrow[dd, bend right=60, "\llparenthesis h \rrparenthesis"'] & 1 + \mathbb{N}_0 \arrow[l, "\mathbf{in} = {[\underline{0}, succ]}"'] \arrow[d, "id + \llparenthesis g \rrparenthesis"] \\
A \arrow[d, "f"'] & 1 + A \arrow[l, "g"] \arrow[d, "id + f"] \\
B & 1 + B \arrow[l, "h"]
\end{tikzcd}
$$

What is a program?

What **is** a program?

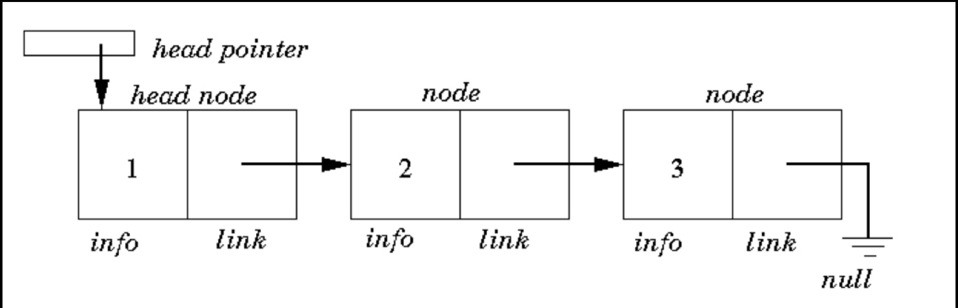
Programs are "rectangles" 

What **is** a program?

Programs are "rectangles" 

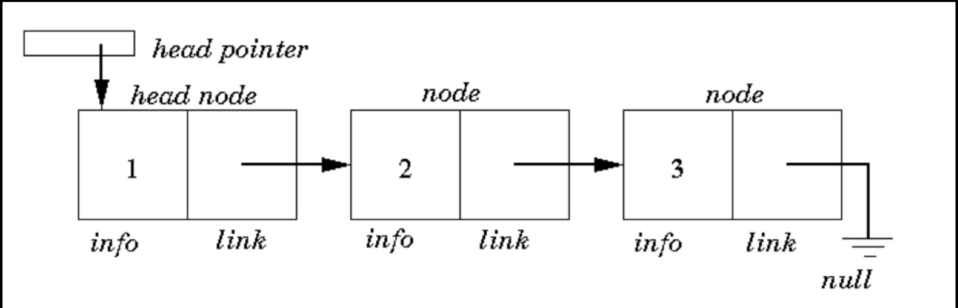
But there is a long way to the general case...

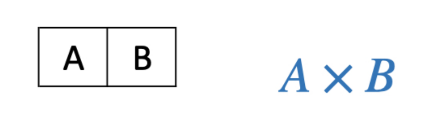
Linked list



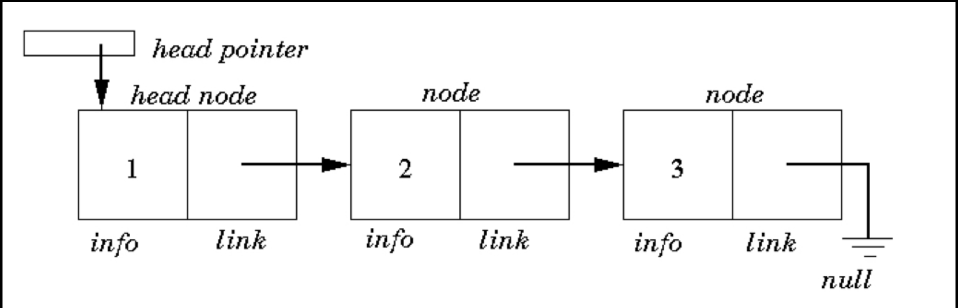
head pointer
head node
node
node
1
2
3
info
link
info
link
info
link
null

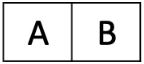
Linked list


head pointer
head node
node
node
1
2
3
info
link
info
link
info
link
null


A
B
A × B

Linked list

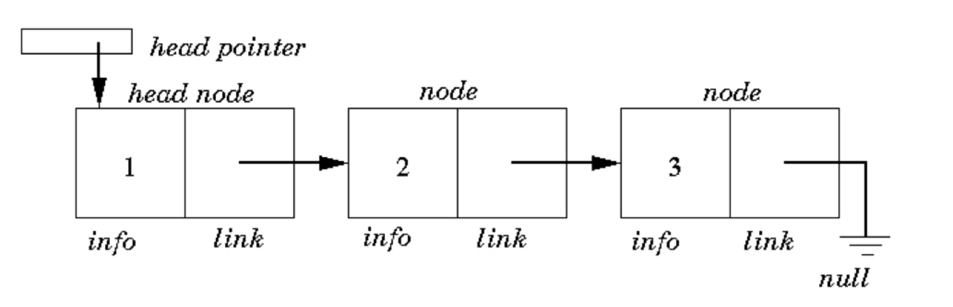




$$A \times B$$

$$1 + A$$

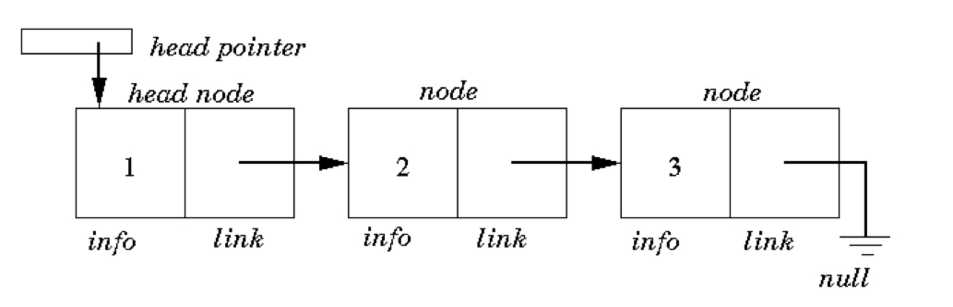
Linked list



"pointer"

$L = 1 + N$

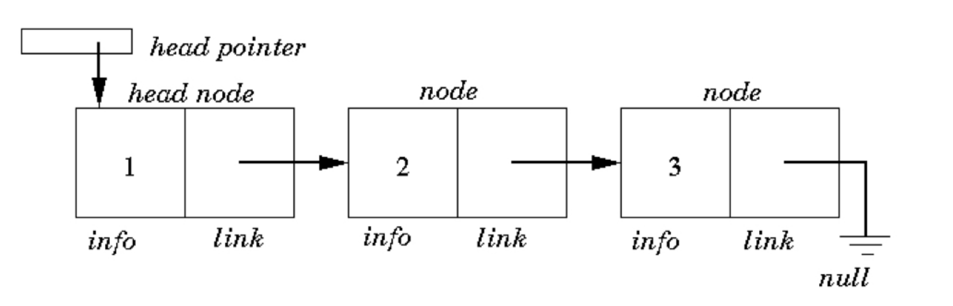
Linked list



"pointer" $L = 1 + N$

"node" $N = \mathbb{N}_0 \times L$

Linked list

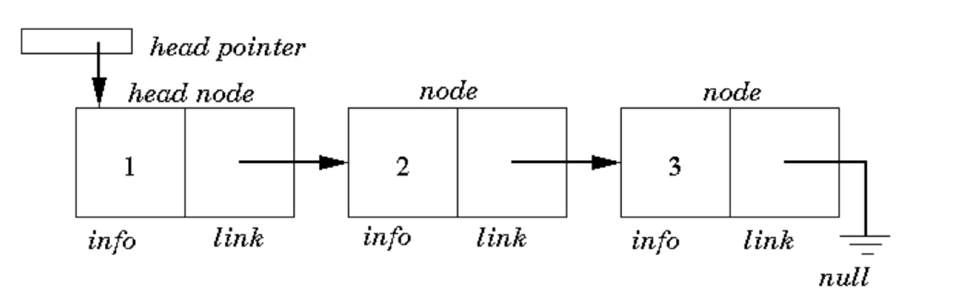


Substitution

$$\begin{cases} L = 1 + N \\ N = \mathbb{N}_0 \times L \end{cases}$$

$$L = 1 + \mathbb{N}_0 \times L$$

Linked list



In fact $$L \cong 1 + \mathbb{N}_0 \times L$$

Linked list

$$L \cong 1 + \mathbb{N}_0 \times L$$

Linked list

$$L \cong 1 + \mathbb{N}_0 \times L$$

$$\begin{tikzcd} L \arrow[r, bend left, "?"] \arrow[r, phantom, "\cong"] & 1 + \mathbb{N}_0 \times L \arrow[l, bend left, "?"] \end{tikzcd}$$

Length of a list

Recall list **concatenation** $x +\!\!+\, y$

Length of a list

Recall list **concatenation** $x \mathbin{+\!\!+} y$ such that

$$[\,] \mathbin{+\!\!+} x = x$$

Length of a list

Recall list **concatenation** $x +\!\!+\, y$ such that

$$[\,] +\!\!+\, x = x$$
$$[a] +\!\!+\, x = a : x$$

Length of a list

Recall list **concatenation** $x +\!\!+ y$ such that

$$[\,] +\!\!+ x = x$$
$$[a] +\!\!+ x = a : x$$

Properties ('specification'):

Length of a list

Recall list **concatenation** $x +\!\!+\, y$ such that

$$[\,] +\!\!+\, x = x$$
$$[a] +\!\!+\, x = a : x$$

Properties ('specification'):

$$\mathit{length}\ [\,] = 0$$

Length of a list

Recall list **concatenation** $x +\!\!+\, y$ such that

$$[\,] +\!\!+\, x = x$$
$$[a] +\!\!+\, x = a : x$$

Properties ('specification'):

$$\mathit{length}\ [\,] = 0$$
$$\mathit{length}\ [a] = 1$$

Length of a list

Recall list **concatenation** $x \mathbin{+\!\!+} y$ such that

$$\begin{aligned} [\,] \mathbin{+\!\!+} x &= x \\ [a] \mathbin{+\!\!+} x &= a : x \end{aligned}$$

Properties ('specification'):

$$\begin{aligned} \mathit{length}\ [\,] &= 0 \\ \mathit{length}\ [a] &= 1 \\ \mathit{length}\ (x \mathbin{+\!\!+} y) &= \mathit{length}\ x + \mathit{length}\ y \end{aligned}$$

Length of a list

$$\mathit{length}\,[\,] = 0$$
$$\mathit{length}\,[a] = 1$$
$$\mathit{length}\,([a] +\!\!+\; y) = \mathit{length}\,[a] + \mathit{length}\;y$$

Length of a list

$$length\ [\,] = 0$$

$$length\ [a] = 1$$

$$length\ ([a] +\!\!+\ y) = length\ [a] + length\ y$$

$$length\ (a : y) = 1 + length\ y$$

Length of a list

$$\mathit{length}\ [\,] = 0$$
$$\mathit{length}\ (a : y) = 1 + \mathit{length}\ y$$

Length of a list

$$\mathit{length}\ [\,] = 0$$
$$\mathit{length}\ (a : y) = 1 + \mathit{length}\ y$$

$$(\mathit{length} \cdot \mathit{nil})\ \_ = \underline{0}\ \_$$
$$(\mathit{length} \cdot \mathit{cons})\ (a, y) = 1 + \mathit{length}\ (\pi_2\ (a, y))$$

Length of a list

$$\mathit{length}\ [\,] = 0$$
$$\mathit{length}\ (a : y) = 1 + \mathit{length}\ y$$

$$(\mathit{length} \cdot \mathit{nil})\ \_ = \underline{0}\ \_$$
$$(\mathit{length} \cdot \mathit{cons})\ (a, y) = 1 + \mathit{length}\ (\pi_2\ (a, y))$$

$$\mathit{length} \cdot \mathit{nil} = \underline{0}$$
$$\mathit{length} \cdot \mathit{cons} = \mathit{succ} \cdot \mathit{length} \cdot \pi_2$$

Length of a list

$$length \cdot nil = \underline{0}$$
$$length \cdot cons = succ \cdot length \cdot \pi_2$$

Length of a list

$$length \cdot nil = \underline{0}$$

$$length \cdot cons = succ \cdot length \cdot \pi_2$$

$$length \cdot [nil, cons] = [\underline{0}, succ \cdot \pi_2] \cdot (id + id \times length)$$

Length of a list

$$length \cdot nil = \underline{0}$$
$$length \cdot cons = succ \cdot length \cdot \pi_2$$

$$length \cdot [nil\,, cons] = [\underline{0}, succ \cdot \pi_2] \cdot (id + id \times length)$$

$$L \xleftarrow[\mathbf{in}=[nil\,,cons]]{\cong} 1 + \mathbb{N}_0 \times L$$

Length of a list

$$length \cdot nil = \underline{0}$$
$$length \cdot cons = succ \cdot length \cdot \pi_2$$

$$length \cdot [nil, cons] = [\underline{0}, succ \cdot \pi_2] \cdot (id + id \times length)$$

$$\begin{tikzcd} L \arrow[d, "length"'] \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] & & 1 + \mathbb{N}_0 \times L \arrow[ll, bend left, "\mathbf{in}"] \arrow[d, "id + id \times length"] \\ \mathbb{N}_0 & & 1 + \mathbb{N}_0 \times \mathbb{N}_0 \arrow[ll, bend left, "{[\underline{0}, succ \cdot \pi_2]}"] \end{tikzcd}$$

$$\begin{cases} \mathbf{in} = [nil, cons] \\ \mathbf{out} = \mathbf{in}^\circ \end{cases}$$

List sum

$$\begin{tikzcd}
L \arrow[d, "sum"'] \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] & & 1 + \mathbb{N}_0 \times L \arrow[ll, bend left, "\mathbf{in}"] \arrow[d, "id + id \times sum"] \\
\mathbb{N}_0 & & 1 + \mathbb{N}_0 \times \mathbb{N}_0 \arrow[ll, bend left, "{[\underline{0}, add]}"]
\end{tikzcd}$$

$$\begin{cases} \mathbf{in} = [nil, cons] \\ \mathbf{out} = \mathbf{in}^{\circ} \end{cases}$$

$$sum \cdot \mathbf{in} = [\underline{0}, add] \cdot (id + id \times sum)$$

Linked list

As earlier on:

$$
\begin{tikzcd}
L \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] & & 1 + \mathbb{N}_0 \times L \arrow[ll, bend left, "\mathbf{in}"]
\end{tikzcd}
$$

$$
\begin{cases}
\mathbf{out} \cdot \mathbf{in} = id \\
\mathbf{in} = [nil, cons]
\end{cases}
$$

Linked list

As earlier on:

$$
\begin{tikzcd}
L \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] & & 1 + \mathbb{N}_0 \times L \arrow[ll, bend left, "\mathbf{in}"]
\end{tikzcd}
$$

$$
\left\{ \begin{array}{l} \mathbf{out} \cdot \mathbf{in} = id \\ \mathbf{in} = [nil, cons] \end{array} \right. \quad \text{leads to} \quad \left\{ \begin{array}{l} \mathbf{out}\ [\,] = i_1\ () \\ \mathbf{out}\ (a : x) = i_2\ (a, x) \end{array} \right.
$$

Generalizing

$$
\begin{tikzcd}
L \arrow[r, bend left, "\mathbf{out}"] \arrow[r, phantom, "\cong"] \arrow[d, "k"'] & 1 + \mathbb{N}_0 \times L \arrow[l, bend left, "\mathbf{in}"] \arrow[d, "id + id \times k"] \\
B & 1 + \mathbb{N}_0 \times B \arrow[l, bend left, "g"]
\end{tikzcd}
$$

Generalizing

$$
\begin{tikzcd}
L \arrow[r, bend left, "\mathsf{out}"] \arrow[r, phantom, "\cong"] \arrow[d, "k"'] & 1 + \mathbb{N}_0 \times L \arrow[l, bend left, "\mathsf{in}"] \arrow[d, "id + id \times k"] \\
B & 1 + \mathbb{N}_0 \times B \arrow[l, bend left, "g"]
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{N}_0 \arrow[r, bend left, "\mathsf{out}"] \arrow[r, phantom, "\cong"] \arrow[d, "k"'] & 1 + \mathbb{N}_0 \arrow[l, bend left, "\mathsf{in}"] \arrow[d, "id + k"] \\
B & 1 + B \arrow[l, bend left, "g"]
\end{tikzcd}
$$

Generalizing

$$\begin{tikzcd}[column sep=huge, row sep=large]
L \arrow[r, bend left, "\mathbf{out}"] \arrow[r, phantom, "\cong"] \arrow[d, "k"'] & 1 + \mathbb{N}_0 \times L \arrow[l, bend left, "\mathbf{in}"] \arrow[d, "id + id \times k"] \\
B & 1 + \mathbb{N}_0 \times B \arrow[l, bend left, "g"]
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=huge, row sep=large]
\mathbb{N}_0 \arrow[r, bend left, "\mathbf{out}"] \arrow[r, phantom, "\cong"] \arrow[d, "k"'] & 1 + \mathbb{N}_0 \arrow[l, bend left, "\mathbf{in}"] \arrow[d, "id + k"] \\
B & 1 + B \arrow[l, bend left, "g"]
\end{tikzcd}$$

$$k \cdot \mathbf{in} = g \cdot \underbrace{id + id \times k}_{\mathbf{F}\ k}$$

Generalizing

$$\begin{tikzcd} L \arrow[rr, bend left, "\mathsf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "k"'] & & 1 + \mathbb{N}_0 \times L \arrow[ll, bend left, "\mathsf{in}"] \arrow[d, "id + id \times k"] \\ B & & 1 + \mathbb{N}_0 \times B \arrow[ll, bend left, "g"] \end{tikzcd}$$

$$\begin{tikzcd} \mathbb{N}_0 \arrow[rr, bend left, "\mathsf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "k"'] & & 1 + \mathbb{N}_0 \arrow[ll, bend left, "\mathsf{in}"] \arrow[d, "id + k"] \\ B & & 1 + B \arrow[ll, bend left, "g"] \end{tikzcd}$$

$$k \cdot \mathsf{in} = g \cdot \underbrace{id + id \times k}_{\mathsf{F}\ k}$$

$$k \cdot \mathsf{in} = g \cdot \underbrace{id + k}_{\mathsf{F}\ k}$$

Abstraction ($\mathbb{N}_0$)

$$
\begin{tikzcd}
\mathbb{N}_0 \arrow[d, "k"'] \arrow[r, bend left, "\mathsf{out}"] \arrow[r, phantom, "\cong"] & 1 + \mathbb{N}_0 \arrow[l, bend left, "\mathsf{in}"] \arrow[d, "id+k"] \\
B & 1 + B \arrow[l, bend left, "g"]
\end{tikzcd}
$$

Abstraction ($\mathbb{N}_0$)

$$\begin{tikzcd}
\mathbb{N}_0 \arrow[rr, bend left, "\mathsf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "k"'] & & 1 + \mathbb{N}_0 \arrow[ll, bend left, "\mathsf{in}"] \arrow[d, "id+k"] \\
B & & 1 + B \arrow[ll, bend left, "g"]
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{N}_0 \arrow[rr, bend left, "\mathsf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "k"'] & & \mathbf{F}\ \mathbb{N}_0 \arrow[ll, bend left, "\mathsf{in}"] \arrow[d, "\mathbf{F}\ k"] \\
B & & \mathbf{F}\ B \arrow[ll, bend left, "g"]
\end{tikzcd}$$

$$\begin{cases} \mathbf{F}\ k = id + k \\ \mathbf{F}\ X = 1 + X \end{cases}$$

Abstraction (lists)

$$\begin{tikzcd}
L \arrow[d, "k"'] \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] & & 1 + \mathbb{N}_0 \times L \arrow[ll, bend left, "\mathbf{in}"] \arrow[d, "id + id \times k"] \\
B & & 1 + \mathbb{N}_0 \times B \arrow[ll, bend left, "g"]
\end{tikzcd}$$

Abstraction (lists)

$$\begin{tikzcd}
L \arrow[rr, bend left, "\mathsf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "k"'] & & 1 + \mathbb{N}_0 \times L \arrow[ll, bend left, "\mathsf{in}"] \arrow[d, "id + id \times k"] \\
B & & 1 + \mathbb{N}_0 \times B \arrow[ll, bend left, "g"]
\end{tikzcd}$$

$$\begin{cases} \mathbf{F}\ k = id + id \times k \\ \mathbf{F}\ X = 1 + \mathbb{N}_0 \times X \end{cases}$$

Abstraction (lists)

$$\begin{tikzcd} L \arrow[rr, bend left, "\mathsf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "k"'] & & 1 + \mathbb{N}_0 \times L \arrow[ll, bend left, "\mathsf{in}"] \arrow[d, "id + id \times k"] \\ B & & 1 + \mathbb{N}_0 \times B \arrow[ll, bend left, "g"] \end{tikzcd}$$

$$\begin{tikzcd} L \arrow[rr, bend left, "\mathsf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "k"'] & & \mathbf{F}\ L \arrow[ll, bend left, "\mathsf{in}"] \arrow[d, "\mathbf{F}\ k"] \\ B & & \mathbf{F}\ B \arrow[ll, bend left, "g"] \end{tikzcd}$$

$$\begin{cases} \mathbf{F}\ k = id + id \times k \\ \mathbf{F}\ X = 1 + \mathbb{N}_0 \times X \end{cases}$$

Abstraction

$$
\begin{tikzcd}
T \arrow[r, bend left, "\mathbf{out}"] \arrow[r, phantom, "\cong"] \arrow[d, "k"'] & \mathbf{F}\ T \arrow[l, bend left, "\mathbf{in}"] \arrow[d, "\mathbf{F}\ k"] \\
B & \mathbf{F}\ B \arrow[l, bend left, "g"]
\end{tikzcd}
$$

Abstraction

$$\begin{tikzcd}
T \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "k"'] & & \mathbf{F}\ T \arrow[ll, bend left, "\mathbf{in}"] \arrow[d, "\mathbf{F}\ k"] \\
B & & \mathbf{F}\ B \arrow[ll, bend left, "g"]
\end{tikzcd}$$

$$k \cdot \mathbf{in} = g \cdot \mathbf{F}\ k$$

Abstraction

$$
\begin{tikzcd}
T \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "k"'] & & \mathbf{F}\ T \arrow[ll, bend left, "\mathbf{in}"] \arrow[d, "\mathbf{F}\ k"] \\
B & & \mathbf{F}\ B \arrow[ll, bend left, "g"]
\end{tikzcd}
$$

$$k = (\!|\, g \,|\!)$$

$$k \cdot \mathbf{in} = g \cdot \mathbf{F}\ k$$

Abstraction

$$\begin{tikzcd}
T \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "k"'] & & \mathbf{F}\,T \arrow[ll, bend left, "\mathbf{in}"] \arrow[d, "\mathbf{F}\,k"] \\
B & & \mathbf{F}\,B \arrow[ll, bend left, "g"]
\end{tikzcd}$$

$$k = (\!|\, g \,|\!)$$

$$k \cdot \mathbf{in} = g \cdot \mathbf{F}\ k$$

Read $k = (\!|\, g \,|\!)$ as before: k is the *catamorphism* of g.

Catamorphisms

Catamorphism

$$
\begin{tikzcd}
T \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "(\!| g |\!)"'] & & \mathbf{F}\ T \arrow[ll, bend left, "\mathbf{in}"] \arrow[d, "\mathbf{F}\ (\!| g |\!)"] \\
B & & \mathbf{F}\ B \arrow[ll, bend left, "g"]
\end{tikzcd}
$$

Catamorphism

$$
\begin{tikzcd}
T \arrow[rr, bend left, "\mathbf{out}"] \arrow[rr, phantom, "\cong"] \arrow[d, "(\!|\, g \,|\!)"'] & & \mathbf{F}\ T \arrow[ll, bend left, "\mathbf{in}"] \arrow[d, "\mathbf{F}\ (\!|\, g \,|\!)"] \\
B & & \mathbf{F}\ B \arrow[ll, bend left, "g"]
\end{tikzcd}
$$

Universal property:

$$k = (\!|\, g \,|\!) \quad \Leftrightarrow \quad k \cdot \mathbf{in} = g \cdot \mathbf{F}\ k$$

Summary

$$\text{Lists: } \begin{cases} T = L \\ \mathbf{in} = [nil, cons] \\ \begin{cases} \mathbf{F}\ X = 1 + \mathbb{N}_0 \times X \\ \mathbf{F}\ f = id + id \times f \end{cases} \end{cases}$$

Summary

Lists: $\begin{cases} T = L \\ \mathbf{in} = [nil, cons] \\ \begin{cases} \mathbf{F}\, X = 1 + \mathbb{N}_0 \times X \\ \mathbf{F}\, f = id + id \times f \end{cases} \end{cases}$

$$\mathbf{foldr}\ f\ i = (\!|\, [\underline{i}, \widehat{f}] \,|\!)$$

Summary

Lists: $\begin{cases} T = L \\ \mathbf{in} = [\mathit{nil}, \mathit{cons}] \\ \begin{cases} \mathbf{F}\ X = 1 + \mathbb{N}_0 \times X \\ \mathbf{F}\ f = \mathit{id} + \mathit{id} \times f \end{cases} \end{cases}$

$\mathbf{foldr}\ f\ i = (\!|\, [\underline{i}, \widehat{f}]\, |\!)$

Natural numbers: $\begin{cases} T = \mathbb{N}_0 \\ \mathbf{in} = [\underline{0}, \mathit{succ}] \\ \begin{cases} \mathbf{F}\ X = 1 + X \\ \mathbf{F}\ f = \mathit{id} + f \end{cases} \end{cases}$

Summary

Lists: $\begin{cases} T = L \\ \mathbf{in} = [\mathit{nil}, \mathit{cons}] \\ \begin{cases} \mathbf{F}\ X = 1 + \mathbb{N}_0 \times X \\ \mathbf{F}\ f = \mathit{id} + \mathit{id} \times f \end{cases} \end{cases}$

$$\mathbf{foldr}\ f\ i = (\!|\, [\underline{i}, \widehat{f}]\, |\!)$$

Natural numbers: $\begin{cases} T = \mathbb{N}_0 \\ \mathbf{in} = [\underline{0}, \mathit{succ}] \\ \begin{cases} \mathbf{F}\ X = 1 + X \\ \mathbf{F}\ f = \mathit{id} + f \end{cases} \end{cases}$

$$\mathbf{for}\ b\ i = (\!|\, [\underline{i}, b]\, |\!)$$

Examples already seen

Natural numbers:

$$(a\times) = (\!|\, [\underline{0}, (a+)] \,|\!)$$
$$a^b = (\!|\, [\underline{1}, (a\times)] \,|\!)\ b$$

Lists:

$$sum = (\!|\, [\underline{0}, add] \,|\!)$$
$$length = (\!|\, [\underline{0}, succ \cdot \pi_2] \,|\!)$$

Examples already seen

Natural numbers:

$$(a\times) = (\!|\, [\underline{0}, (a+)] \,|\!) = \mathbf{for}\ (a+)\ 0$$

$$a^b = (\!|\, [\underline{1}, (a\times)] \,|\!)\ b = \mathbf{for}\ (a\times)\ 1\ b$$

Lists:

$$sum = (\!|\, [\underline{0}, add] \,|\!) = \mathbf{foldr}\ 0\ (+)$$

$$length = (\!|\, [\underline{0}, succ \cdot \pi_2] \,|\!) = \mathbf{foldr}\ 0\ \overline{succ \cdot \pi_2}$$

Cata-cancellation

In

$$k = (\!|\, g \,|\!) \;\Leftrightarrow\; k \cdot \mathbf{in} = g \cdot \mathsf{F}\, k$$

make $k := (\!|\, g \,|\!)$:

Cata-cancellation

In

$$k = (\!| g |\!) \Leftrightarrow k \cdot \mathbf{in} = g \cdot \mathbf{F}\ k$$

make $k := (\!| g |\!)$:

$$(\!| g |\!) = (\!| g |\!) \Leftrightarrow (\!| g |\!) \cdot \mathbf{in} = g \cdot \mathbf{F}\ (\!| g |\!)$$

Cata-cancellation

In

$$k = (\!| g |\!) \;\Leftrightarrow\; k \cdot \mathbf{in} = g \cdot \mathsf{F}\, k$$

make $k := (\!| g |\!)$:

$$
\begin{array}{rl}
 & (\!| g |\!) = (\!| g |\!) \;\Leftrightarrow\; (\!| g |\!) \cdot \mathbf{in} = g \cdot \mathsf{F}\, (\!| g |\!) \\
\Leftrightarrow & \quad \{\; x = x \Leftrightarrow \mathit{true} \;\} \\
 & \mathit{true} \;\Leftrightarrow\; (\!| g |\!) \cdot \mathbf{in} = g \cdot \mathsf{F}\, (\!| g |\!)
\end{array}
$$

Cata-cancellation

In

$$k = (\!|\, g \,|\!) \;\Leftrightarrow\; k \cdot \mathbf{in} = g \cdot \mathsf{F}\; k$$

make $k := (\!|\, g \,|\!)$:

$$
\begin{array}{ll}
 & (\!|\, g \,|\!) = (\!|\, g \,|\!) \;\Leftrightarrow\; (\!|\, g \,|\!) \cdot \mathbf{in} = g \cdot \mathsf{F}\; (\!|\, g \,|\!) \\
\Leftrightarrow & \qquad \{\; x = x \Leftrightarrow \mathit{true} \;\} \\
 & \mathit{true} \;\Leftrightarrow\; (\!|\, g \,|\!) \cdot \mathbf{in} = g \cdot \mathsf{F}\; (\!|\, g \,|\!) \\
\Leftrightarrow & \qquad \{\; \text{trivial} \;\} \\
 & (\!|\, g \,|\!) \cdot \mathbf{in} = g \cdot \mathsf{F}\; (\!|\, g \,|\!)
\end{array}
$$

Cata-cancellation

$$(\!| g |\!) \cdot \mathsf{in} = g \cdot \mathsf{F}\,(\!| g |\!)$$

$$\Leftrightarrow \quad \{\ \text{isomorphism } \mathsf{in} \,/\, \mathsf{out}\ \}$$

$$(\!| g |\!) = g \cdot \mathsf{F}\,(\!| g |\!) \cdot \mathsf{out}$$

Cata-cancellation

$$(\!| g |\!) \cdot \mathsf{in} = g \cdot \mathsf{F}\,(\!| g |\!)$$

$$\Leftrightarrow \qquad \{ \text{ isomorphism } \mathsf{in} \,/\, \mathsf{out} \ \}$$

$$(\!| g |\!) = g \cdot \mathsf{F}\,(\!| g |\!) \cdot \mathsf{out}$$

$$\begin{tikzcd}
T \arrow[r, bend left, "\mathsf{out}"] \arrow[r, phantom, "\cong"] \arrow[d, "(\!| g |\!)"'] & \mathsf{F}\,T \arrow[l, bend left, "\mathsf{in}"] \arrow[d, "\mathsf{F}\,(\!| g |\!)"] \\
B & \mathsf{F}\,B \arrow[l, bend left, "g"]
\end{tikzcd}$$

Cata-cancellation

$$\llparenthesis\, g \,\rrparenthesis \cdot \mathsf{in} = g \cdot \mathsf{F}\,\llparenthesis\, g \,\rrparenthesis$$

$$\Leftrightarrow \qquad \{\ \text{isomorphism } \mathsf{in} \,/\, \mathsf{out}\ \}$$

$$\llparenthesis\, g \,\rrparenthesis = g \cdot \mathsf{F}\,\llparenthesis\, g \,\rrparenthesis \cdot \mathsf{out}$$

$$\begin{tikzcd}
T \arrow[r, bend left, "\mathsf{out}"] \arrow[d, "\llparenthesis\, g \,\rrparenthesis"'] \arrow[r, phantom, "\cong"] & \mathsf{F}\,T \arrow[l, bend left, "\mathsf{in}"] \arrow[d, "\mathsf{F}\,\llparenthesis\, g \,\rrparenthesis"] \\
B & \mathsf{F}\,B \arrow[l, bend left, "g"]
\end{tikzcd}$$

$$\llparenthesis\, g \,\rrparenthesis = g \cdot \mathsf{F}\,\llparenthesis\, g \,\rrparenthesis \cdot \mathsf{out}$$

executable definition

TBC…