

**Logic and Categories, or:
Heyting Algebras for Children**

A tutorial presented at the UniLog 2015 conference (Istanbul),
June 20-22, 2015

By Eduardo Ochs
eduardoochs@gmail.com

<http://angg.twu.net/math-b.html#istanbul>

Version: 2015jun22

1. (Non-mathematical) introduction

1.1. Why study CT now?



Public education in Brazil is being dismantled -
maybe we should be doing better things than studying
very technical & inaccessible subjects
with no research grants -

1.2. Category theory should be more accessible

Most texts about CT are for specialists in research universities...

Category theory should be more accessible.

To whom?...

- Non-specialists (in research universities)
- Grad students (in research universities)
- Undergrads (in research universities)
- Non-specialists (in conferences - where we have to be quick)
- Undergrads (e.g., in CompSci - in teaching colleges) - (“Children”)

1.3. What do we mean by "accessible"?

- 1) Done on very firm grounds:
 - mathematical objects made from numbers, sets and tuples
 - FINITE, SMALL mathematical objects whenever possible
 - avoid references to non-mathematical things like windows, cars and pizzas (like the OO people do)
 - avoid reference to Physics - avoid QM at all costs
 - time is difficult to draw - prefer static rather than changing
- 2) People have very short attention spans nowadays
- 3) Self-contained, but not isolated or isolating - our material should make the literature more accessible
- 4) We learn better by doing.
 - Our material should have lots of space for exercises.
- 5) Most people with whom I interact are not from Maths/CS/etc
- 6) Proving general cases is relatively hard.
 - Checking and calculating is much easier.
 - People can believe that something can be generalized after seeing a convincing particular case.
 - (Sometimes leave them to look for the right generalization by themselves)

1.4. Generalizations

General case	Full things
$\hat{\quad}$	$\hat{\quad}$
projection : lifting	projection : lifting
v :	v :
Particular case	Things for children

Paper:

Internal Diagrams and Archetypal Reasoning in Category Theory

Eduardo Ochs 2103 - Logica Universalis

<http://angg.twu.net/>

<- my home page

<http://angg.twu.net/math-b.html>

<http://angg.twu.net/math-b.html#idarct>

The type-theoretical view of the formalization of all that
is something that I can discuss with very few people =(

The parts "for children" are more fun

(esp. because I don't have to do them alone)

so I have to get good excuses for working on that

instead of on the very technical type-theoretical parts

1.5. Yesterday and slides

To do: apologize for yesterday

I overestimated my TeXnical skills!!! =(
Some of my diagrams need LuaLaTeX extensions
that took me `_much_` longer to write than I expected,
and the emergency tools that let me use `ascii+utf8` slides
also took me one (`_very_` frantic) day more than I had -

...so things were not ready yesterday - and as I could not
connect my computer directly to the projector, and I did not
have enough material in PDF form, I had to cancel -

These `ascii` slides will be at
<http://angg.twu.net/math-b.html>
soon.

Nice (Lua)(La)TeXed slides should be ready soon,
for some adequate, not-very-small value of "soon".

2. The very basics

2.1. Evaluation: steps

How do we evaluate an expression like $2*3+4*5$ to obtain its value, 26?

This involves a series of steps...

There are several ways to represent these steps:

$$\begin{array}{r} 2*3+4*5 \\ \backslash- / \backslash- / \\ 6 \quad 20 \\ \backslash----- / \\ 26 \end{array} \qquad \begin{array}{r} 2*3+4*5 \rightarrow 2*3+20 \\ | \qquad \qquad | \\ v \qquad \qquad v \\ 6+4*5 \rightarrow 6+20 \rightarrow 26 \end{array}$$

$$\begin{array}{r} 2*3+4*5 = 6+20 \\ = 26 \end{array} \qquad \begin{array}{r} 2*3+4*5 = 2*3+20 \\ = 6+20 \\ = 26 \end{array}$$

The technical term for the arrows is ‘reductions’.
Informally, they are ‘evaluation steps’.

2.2. Evaluation sequences

$$\begin{array}{r} 2*3+4*5 \\ \backslash- / \backslash- / \\ 6 \quad 20 \\ \backslash----- / \\ 26 \end{array} \qquad \begin{array}{r} 2*3+4*5 \rightarrow 2*3+20 \\ | \qquad \qquad | \\ v \qquad \qquad v \\ 6+4*5 \rightarrow 6+20 \rightarrow 26 \end{array}$$

$$\begin{array}{r} 2*3+4*5 = 6+20 \\ = 26 \end{array} \qquad \begin{array}{r} 2*3+4*5 = 2*3+20 \\ = 6+20 \\ = 26 \end{array}$$

To evaluate is to reduce, and reduce, and reduce until there is nothing more to do - until we reach something that is ‘fully evaluated’.

Note that two ‘evaluation sequences’ starting at the same expression may ‘diverge’ at some point, but they necessarily ‘converge’ later - this is what assures us that the result of evaluation is unique.

(Technical terms: ‘confluence’, ‘diamond lemma’, ‘normalization’...)

2.3. Algebra: tables

Some expressions, e.g. $(a+b)(a-b)$ and $a*a-b*b$, return equal values for all inputs...

We can /test/ whether $(a+b)(a-b)=a*a-b*b$ in a /finite number/ of cases using tables,

a	b	$(a+b)(a-b)$	$a*a-b*b$	$(a+b)(a-b)=a*a-b*b$
0	0	0	0	T
2	3	-5	-5	T
10	1	99	99	T

but we can not /know/ that $(a+b)(a-b)=a*a-b*b$ just by checking a finite number of table lines...

2.4. Algebra: reductions

To understand why $(a+b)(a-b)=a^2-b^2$ we need /algebra/, which includes reductions like these, which use associativity always in the "expansion" direction, i.e., $x(y+z)\rightarrow xy+xz$,

$$\begin{array}{rcl}
 (a+b)(a-b) & \rightarrow & (a+b)a-(a+b)b \rightarrow (a+b)a-ab-bb \\
 | & & | & & | \\
 | & & v & & v \\
 | & & aa+ba-(a+b)b & \rightarrow & aa+ba-ab-bb \\
 | & & & & : \\
 v & & & & : \\
 a(a-b)+b(a-b) & \rightarrow & a(a-b)+ba-bb & & : \\
 | & & | & & : \\
 v & & v & & v \\
 aa-ab+b(a-b) & \rightarrow & aa-ab+ba-bb & \text{-- --} & \rightarrow & aa-bb
 \end{array}$$

...but then we don't get confluence!

To get the two "-->" arrows we need commutativity of + and ·, plus $x-x=0$...

We need several new things to learn how to do "calculations with letters"!

We can teach evaluation to small children.
 We can only teach algebra to /older/ children, who have already learned evaluation.

The important thing:
 evaluation comes first,
 algebra comes later.

2.5. Logic: evaluation and tautologies

Logic is simpler than algebra (for children),
because we can prove lots of things just by
looking at tables with a finite number of lines.

For example, we can check that $P \& Q \rightarrow P \vee Q$ just by
calculating the result of $P \& Q \rightarrow P \vee Q$ in all four cases.

(P → Q) ∨ (Q → P)	P	Q	P → Q	Q → P	(P → Q) ∨ (Q → P)
\-/ \-/ \-/ \-/	-----				
0 1 1 0	0	0	1	1	1
\-----/ \-----/	0	1	1	0	1
1 0	1	0	0	1	1
\-----/	1	1	1	1	1
1					

P	Q	P & Q	P ∨ Q	P → Q

0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	1

When we teach logic to children it is
good karma to spend a LONG time on this:
what propositions are tautologies?

2.6. A glimpse at other logics

The truth values of classical logic are $\{0,1\}$.
The truth values of $0(2)$ -logic are $\{00,01,11\}$.
Some propositions, like $\neg\neg P \rightarrow P$, are tautologies
in classical logic but not in $0(2)$ -logic...

P	Q	P&Q	P∨Q	P→Q	P	¬P	¬¬P	¬¬P→P
-----					-----			
00	00	00	00	11	00	11	00	11
00	01	00	01	11	01	00	11	01
00	11	00	11	11	11	00	11	11
01	00	00	01	00				
01	01	01	01	11				
01	11	01	11	11				
11	00	00	11	00				
11	01	01	11	01				
11	11	11	11	11				

2.7. λ -calculus: subtopics

Lambda-calculus, even for children, is a big topic...

We will see it as this series of subtopics:

- * the usual way of defining (named) functions
- * the meaning of ':' and '->' in $f:A \rightarrow B$
- * the meaning of '|->'
- * evaluation, a.k.a., reduction
- * lambda-notation
- * beta-reduction
- * the diamond lemma
- * name clashes
- * pairing and projection
- * types
- * contexts, pre-judgments and judgments
- * unions of contexts
- * type inference
- * Set as model for `lambda1`

2.8. Defining named functions

The usual notation for defining functions is like this:

```
f: N -> R                                (*)  
  n |-> 2+sqrt(n)
```

```
name: domain -> codomain  
      variable |-> expression
```

It creates `_named_` functions.

To use this notation we have to fill up five slots,
and use a ":", an "->" and a "|->" in the right places.

After stating (*) we can "reduce" expressions with f, like this:

```
f(4+5) ---> 2+sqrt(4+5)  
:         :  
:         :  
v         v  
f(9) ----> 2+sqrt(9) ----> 2+3 ----> 5
```


2.9. λ -calculus: defining unnamed functions

Now compare

```
name: domain -> codomain
      variable |-> expression   name = \variable.expression
```

```
g: N -> R
   a |-> a*a+4                h = \a.a*a+4
```

$ \begin{array}{c} g(2+3) \text{ -----} \rightarrow g(5) \\ \\ \\ \\ \\ \vee \\ (2+3)*(2+3)+4 \\ \quad \backslash \\ \quad \vee \\ \quad (2+3)*5+4 \\ \quad \backslash \\ \vee \quad \vee \quad \vee \\ 5*(2+3)+4 \text{ -----} \rightarrow 5*5+4 \end{array} $	$ \begin{array}{c} h(2+3) \text{ -----} \rightarrow h(5) \\ \\ \vee \\ (\backslash a.a*a+4)(2+3) \text{ ----} \rightarrow (\backslash a.a*a+4)(5) \\ \\ \vee \\ (2+3)*(2+3)+4 \\ \quad \backslash \\ \quad \vee \\ \quad (2+3)*5+4 \\ \quad \backslash \\ \vee \quad \vee \quad \vee \\ 5*(2+3)+4 \text{ -----} \rightarrow 5*5+4 \end{array} $
---	---

Note that the parentheses are important...

$2+3*2+3+4 \neq (2+3)*(2+3)+4$

Note also that in $h = \backslash a.a*a+4$ we have not used
neither the domain nor the codomain...

2.10. Functions as lists of ' \mapsto 's

Functions can be regarded as sets of (input,output) pairs.
For example,

$$f:\{4,5,6\} \rightarrow \mathbb{R}$$
$$a \mapsto 10a$$

is the set $\{(4,40), (5,50), (6,60)\}$.

Just as we saw to fully evaluate expressions like $2 \cdot 3 + 4 \cdot 5$ -
the _value_ of $2 \cdot 3 + 4 \cdot 5$ is 26 because on numbers there are
no further evaluation steps to be done -
the _value_ of the f above is $\{(4,40), (5,50), (6,60)\}$.

Alternative notations:

$$\backslash\text{csm}\{(4,40), \&\& (5,50), \&\& (6,60)\} =$$
$$\backslash\text{csm}\{4\rightarrow 40, \&\& 5\rightarrow 50, \&\& 6\rightarrow 60\} =$$
$$\backslash\text{csm}\{4\rightarrow 40 \&\& 5\rightarrow 50 \&\& 6\rightarrow 60\} =$$
$$\backslash\text{sm}\{4\rightarrow 40 \&\& 5\rightarrow 50 \&\& 6\rightarrow 60\}$$

We will be sloppy with commas in the $\{\} \mapsto$ notation.

2.11. Trees

We are going to use tree for lots of things, e.g.:

represent constructions

compare general and particular cases (with parallel trees)

add and erase information

track free variables / undischarged hypotheses

understand translations (including Curry-Howard)

p	$(-2, 5)$	$(-2, 5)$
---	-----	-----
$p \quad \pi'p \quad f$	$(-2, 5) \quad \pi'(-2, 5) \quad \text{sqrt}$	$(-2, 5) \quad 5 \quad \text{sqrt}$
-----	-----	-----
$\pi p \quad f(\pi'p)$	$\pi(-2, 5) \quad \text{sqrt}(\pi'(-2, 5))$	$-2 \quad \text{sqrt}(5)$
-----	-----	-----
$(\pi p, f(\pi'p))$	$(\pi(-2, 5), \text{sqrt}(\pi'(-2, 5)))$	$(-2, \text{sqrt}(5))$

$p:A \times B$	$(-2, 5) \in \mathbb{Z} \times \mathbb{N}$
-----	-----
$p:A \times B \quad \pi'p:B \quad f:B \rightarrow C$	$(-2, 5) \in \mathbb{Z} \times \mathbb{N} \quad 5 \in \mathbb{N} \quad \text{sqrt}: \mathbb{N} \rightarrow \mathbb{R}$
-----	-----
$\pi p:A \quad f(\pi'p):C$	$-2 \in \mathbb{Z} \quad \text{sqrt}(5) \in \mathbb{R}$
-----	-----
$(\pi p, f(\pi'p)):A \times C$	$(-2, \text{sqrt}(5)) \in \mathbb{Z} \times \mathbb{R}$

2.12. Trees: free variables, discharges, sequents, Curry-Howard...

$$\begin{array}{c}
 \begin{array}{c}
 \text{[p]}^{\wedge 1} \\
 \hline
 \text{[p]}^{\wedge 1} \quad \pi'p \quad f \\
 \hline
 \pi p \quad f(\pi'p) \\
 \hline
 (\pi p, f(\pi'p)) \\
 \hline
 \lambda p. (\pi p, f(\pi'p))
 \end{array}
 \quad
 \begin{array}{c}
 \text{p|}^{-}p \\
 \hline
 \text{p|}^{-}p \quad \text{p|}^{-}\pi'p \quad f|^{-}f \\
 \hline
 \text{p|}^{-}\pi p \quad f, \text{p|}^{-}f(\pi'p) \\
 \hline
 f, \text{p|}^{-}(\pi p, f(\pi'p)) \\
 \hline
 f|^{-}\lambda p. (\pi p, f(\pi'p))
 \end{array}
 \quad
 \begin{array}{c}
 \text{[P\&Q]}^{\wedge 1} \\
 \hline
 \text{[P\&Q]}^{\wedge 1} \quad Q \quad Q \rightarrow R \\
 \hline
 P \quad R \\
 \hline
 P\&R \\
 \hline
 \text{P\&Q} \rightarrow P\&R
 \end{array}
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{c}
 \text{p:A}\times\text{B} \\
 \hline
 \text{p:A}\times\text{B} \quad \pi'p:\text{B} \quad f:\text{B}\rightarrow\text{C} \\
 \hline
 \pi p:\text{A} \quad f(\pi'p):\text{C} \\
 \hline
 (\pi p, f(\pi'p)):\text{A}\times\text{C}
 \end{array}
 \quad
 \begin{array}{c}
 \text{p:A}\times\text{B|}^{-}\pi'p:\text{B} \quad f:\text{B}\rightarrow\text{C} \\
 \hline
 \text{p:A}\times\text{B|}^{-}\pi p:\text{A} \quad \text{p:A}\times\text{B|}^{-}f(\pi'p):\text{C} \\
 \hline
 \text{p:A}\times\text{B|}^{-}(\pi p, f(\pi'p)):\text{A}\times\text{C}
 \end{array}
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{c}
 \pi':\text{A}\times\text{B}\rightarrow\text{B} \quad f:\text{B}\rightarrow\text{C} \\
 \hline
 \pi:\text{A}\times\text{B}\rightarrow\text{A} \quad f\circ\pi':\text{A}\times\text{B}\rightarrow\text{C} \\
 \hline
 \text{\ang}\{\pi, f\circ\pi\}:\text{A}\times\text{B}\rightarrow\text{A}\times\text{C}
 \end{array}
 \end{array}$$

2.13. Alice's logic and Bob's logic

(For students of Discrete Mathematics)

Our friends Alice and Bob have both published books on logic, with some stylistic differences...

For Alice, Classical Propositional Logic is this,

$(\Omega, \top, \perp, \&, \vee, \rightarrow, \neg, \leftrightarrow) =$

$\{ \{0,1\}, 1, 0, \{ (0,0) \rightarrow 0, (0,0) \rightarrow 0, \{ (0,0) \rightarrow 1, \{0 \rightarrow 1, \{ (0,0) \rightarrow 1, \}$
 $(0,1) \rightarrow 0, (0,1) \rightarrow 1, (0,1) \rightarrow 1, 1 \rightarrow 0 \}, (0,1) \rightarrow 0,$
 $(1,0) \rightarrow 0, (1,0) \rightarrow 1, (1,0) \rightarrow 0, (1,0) \rightarrow 0,$
 $(1,1) \rightarrow 1 \}, (1,1) \rightarrow 1 \}, (1,1) \rightarrow 1 \}, (1,1) \rightarrow 1 \}$

In Bob's book, Classical Propositional Logic is

$(\Omega, \top, \&, \neg) = \dots$

with all the other operators being defined from this one...

How do we translate from one of these notions to another?

Suppose that Alice's book has a theorem that is not in Bob's book. We may need to translate both the /theorem/ and the /proof/.

At one point, when we were studying Alice's book and feeling that we were on shaky ground, we thought of "&" as being the 4th component of the structure above...

A very low-level translation uses things like "4th component". A higher-level translation refers to components by their names.

3. (Planar) Heyting Algebras for children

3.1. A game - on N^2

Let's consider a game played on an infinite board, N^2 , with black and white pieces.

* This is a black piece.
It is solid. It is heavy.
It tends to sink.
/Black pieces move down/.

0 This is a white piece.
It is hollow. It is light.
It tends to float.
/White pieces move up/.

Black moves:	*	(x,y)	(0,y)
	/ \	/ \	\
	v v v	v v v	v v
	(x-1,y-1)	(x,y-1) (x-1,y-1)	(0,y-1) (1,y-1)

White moves:	(x-1,y+1)	(x,y+1)	(x-1,y+1)	(0,y+1)	(1,y+1)
	^ ^ ^	^ ^ ^		^ ^	
	\ /	\ /		/	
	0	(x,y)		(0,y)	

Note that moves are only possible between /valid positions/ - this rules out positions with $x=-1$.

3.2. A game - on subsets of N^2

Let's consider a game played
on an infinite board, N^2 ,
with black and white pieces...

Note that moves are only possible
between /valid positions/ -
this rules out positions with $x=-1$.

We may restrict the
valid positions even more.
Let $A \subseteq N^2$.

Let $BM(A)$ be the set of black moves
starting and ending on points of A , and
let $WM(A)$ be the set of white moves
starting and ending on points of A .

Then $BM(A)$ and $WM(A)$ are sets of arrows,
and $(A, BM(A))$ and $(A, WM(A))$ are graphs.

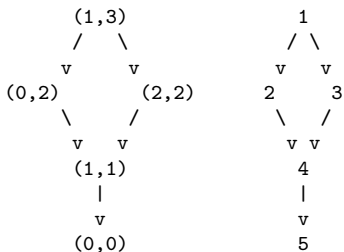
$(A, BM(A))$ and $(A, WM(A))$ are directed,
acyclical graphs - DAGs.

3.3. A game - on K

Let K ("Kite") be this subset of N^2 :

$$K = \{(1,3), (0,2), (2,2), (1,1), (1,0)\}$$

Then $(K, BM(K))$ is the DAG at the left below,
which is isomorphic to the DAG K' at the right...



$$(K, BM(K)) = (\{(1,3), (0,2), (2,2), (1,1), (1,0)\}, \\ \{((1,3), (0,2)), ((1,3), (2,2)), \\ ((0,2), (1,1)), ((2,2), (1,1)), \\ ((1,1), (0,0))\})$$

$$K' = (\{1, 2, 3, 4, 5\}, \\ \{(1,2), (1,3), (2,4), (3,4), (4,5)\})$$

3.4. A game - definition aborted

When I talked about ‘‘game’’ that was a trick...
I just wanted to define $BM(A)$ and $WM(A)$ on an $A \subseteq \mathbb{N}^2$.
We don’t need notions like ‘‘initial position’’,
‘‘turn’’, ‘‘winning’’, etc.

From now on we will forget the idea of a ‘‘game’’.

3.5. Transitive-reflexive closure

Let (A,R) be a (directed) graph; note that $R \subseteq A^2$.

We will write

(A,R^+) for the transitive closure of (A,R) , and

(A,R^*) for the transitive-reflexive closure of (A,R) .

(A,R^+) is (A,R) ‘‘plus all the identity arrows’’,

(A,R^*) is (A,R^+) ‘‘plus all the composites’’.

Remember that when $A \subseteq N^2$

then $(A, \text{BM}(A))$ and $(A, \text{WM}(A))$ are DAGs...

When we pass to $(A, \text{BM}(A)^*)$ and $(A, \text{WM}(A)^*)$

we don't get cycles - except for the identity maps!

$(A, \text{BM}(A)^*)$ and $(A, \text{WM}(A)^*)$ are directed,

‘‘almost acyclical’’ graphs...

More precisely, $(A, \text{BM}(A)^*)$ and $(A, \text{WM}(A)^*)$

are /partial orders/!

3.6. Well-positioned subsets of N^2

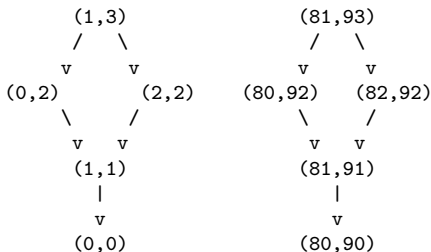
Let $A \subseteq N^2$, and let $(\Delta x, \Delta y) \in Z^2$.

Let $A' = \{(x+\Delta x, y+\Delta y) \mid (x,y) \in A\}$.

Then $(A, BM(A))$ and $(A', BM(A'))$ are isomorphic!

For example, if $A=K$ and $(\Delta x, \Delta y)=(80,90)$,

then $(A, BM(A))$ and $(A', BM(A'))$ are:



We are only interested in the /graphs/
generated by subsets of N^2 - so let's
create a notion of 'canonical position'...

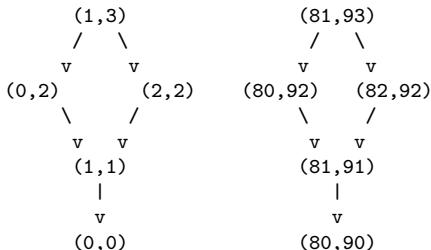
Also, we are only interested in the graphs
generated by /finite/, /nonempty/ subsets of N^2 ...

3.7. Well-positioned subsets of N^2 (2)

Definition:

a subset A of N^2 is /well-positioned/
 when it contains a point of the form $(0,y)$
 and a point of the form $(x,0)$, i.e.,
 it touches both axes, i.e.,
 it can't be translated left or down
 without going outside of N^2 ...

For any non-empty subset A of N^2 ,
 there is exactly one $(\Delta x, \Delta y) \in N^2$
 such that $A - (\Delta x, \Delta y)$ is well-positioned.



3.8. ZSets

Definition:

A /ZSet/ is a finite, nonempty,
well-positioned subset of \mathbb{N}^2 .

The nice thing about ZSets is that
we can represent them unambiguously
using a bullet notation. For example,

```
*  
* *  
*  
*
```

is $\{(1,3), (0,2), (2,2), (1,1), (0,0)\}$.

3.9. ZSets, ZDAGs, ZPosets

Definitions:

A /ZSet/ is a finite, nonempty, well-positioned subset of N^2 .

A /ZDAG/ is a DAG $(A, BM(A))$ or $(A, WM(A))$, where A is a ZSet.

A /ZPoset/ is a poset $(A, BM(A)^*)$ or $(A, WM(A)^*)$, where A is a ZSet.

Examples:

Let H ("House") be this zset here:

```
*  
* *  
* *
```

then:

$(H, BM(H))$ and $(H, WM(H))$ are ZDAGs

$(H, BM(H)^*)$ and $(H, WM(H)^*)$ are ZPosets

3.10. Zfunctions

A Zfunction is a function whose domain is a ZSet.

We can represent Zfunctions using a positional notation.

For example, with $K=\{(1,3), (0,2), (2,2), (1,1), (1,0)\}$,
the function $f:K \rightarrow \mathbb{N}$ such that

$f = \{((1,3),1), ((0,2),2), ((2,2),3), ((1,1),4), ((1,0),5)\}$
can be represented as:

```
1
2 3
4
5
```

Note that its domain, K , can be inferred from the shape

```
*
* *
*
*
```

but its codomain has to be deduced from the context.

3.11. The reading order

The /reading order/ for a ZSet A is the bijection from A to $\{1, 2, \dots, |A|\}$ that enumerates the elements of A from top to bottom, going from left to right in each line. Here are some reading orders:

			1			
1	1		2	3		
2	3	2	3	4	5	
4	4	5	6	7	8	9

The reading order can be used to formalize a way to ‘read aloud’ Zfunctions. For example, we can read aloud

			3
2	40		
1	50		

as 3, 2, 40, 1, 50.

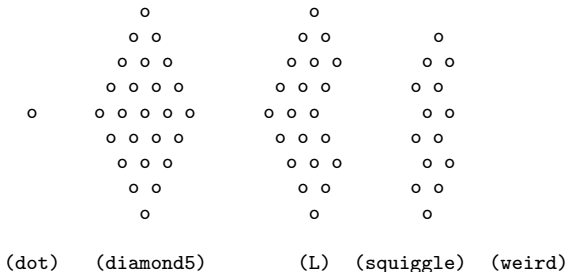
(And this gives a kind of a lexicographic order on Zfunctions on the same ZSet)

3.12. ZHAs

A Heyting Algebra, as we will see later, is a structure (a logic!) $(\Omega, \leq, \top, \perp, \&, \vee, \rightarrow, \neg)$ in which $\leq, \top, \perp, \&, \vee, \rightarrow, \neg$ obey the axioms of /intuitionistic/ propositional calculus.

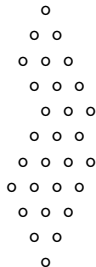
A ZHA is a ZPoset having:
 a bottom point and a top point,
 a left wall and a right wall,
 all the same-parity points between the two walls,
 and no other points.

Some examples of ZHAs:

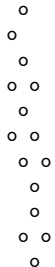


3.13. ZHAs (2)

A ZHA is a ZPoset having:
a bottom point and a top point,
a left wall and a right wall,
all the same-parity points between the two walls,
and no other points.



This is
a ZHA



This is
a ZHA



This is
NOT a ZHA

3.14. ZHAs (formally)

A ZHA is a ZPoset having:

a bottom point and a top point,

a left wall and a right wall,

all the same-parity points between the two walls,

and no other points.

A ZHA D is generated by:

a height $\text{maxy} \geq 0$

a bottom point $(x_0, 0)$

left wall and right wall functions $L, R: \{0, 1, \dots, \text{maxy}\} \rightarrow \mathbb{N}$

obeying:

$L(0) = R(0) = x_0$ (bottom point)

$L(\text{maxy}) = R(\text{maxy})$ (top point)

$0 \leq L(i) \leq R(i)$ (left wall \leq right wall)

$L(y+1) = L(y) \pm 1$ (left wall changes in unit steps)

$R(y+1) = R(y) \pm 1$ (right wall changes in unit steps)

$L(y) = 0$ for some y (to ensure well-positionedness)

then

$D \cap \{(x, y) \mid x \in \mathbb{N}\}$ is empty when $y > \text{maxy}$,

$D \cap \{(x, y) \mid x \in \mathbb{N}\} = \{L(y), L(y)+2, L(y)+4, \dots, R(y)\}$
when $y \in \{0, 1, \dots, \text{maxy}\}$.

3.15. ZHAs (formally) (2)

A ZHA is a ZPoset having:

a bottom point and a top point,

a left wall and a right wall,

all the same-parity points between the two walls,

and no other points.

o L(9)=1 R(9)=1 maxy=9 L(maxy)=R(maxy)
o o
o
o
o
o o
o o o
o o
o o L(1)=3 R(1)=5
o L(0)=4 R(0)=4 L(0)=R(0)=x_0=4

3.16. ZHAs: the (l,r) coordinates

We can define the (l,r) coordinates on a ZHA /informally/ in this way:

the bottom point has (l,r)=(0,0);

when we go one unit northwest,
i.e., from (x,y) to (x-1,y+1),
the l coordinate increases by 1;

when we go one unit northeast,
i.e., from (x,y) to (x+1,y+1),
the r coordinate increases by 1;

We usually write the (l,r) coordinates of a point as just lr - two digits together, without the '(', ',', ')'.
(Note: the original image contains a typo in the text above: 'without the '(, ', ')'. It should be 'without the '(', ',', ')'.')

Here are two examples:

46		
45 36		63
35 26		62 53
34 25		52 43
24		51 42 33
23		50 41 32 23
22		40 31 22
21		30 21 12
20 11		20 11 02
10 01		10 01
00		00

3.17. ZHAs are Heyting Algebras (introduction)

Magic fact / big theorem:
ZHAs are Heyting Algebras.

Version for children: we will learn

1. how to interpret $\leq, \top, \perp, \&, \vee, \rightarrow, \neg, \leftrightarrow$ in any given ZHA
2. what properties that $\leq, \top, \perp, \&, \vee, \rightarrow, \neg, \leftrightarrow$ have to obey in a HA
3. that our ZHA operations from (1) obey the properties in (2)

The properties are these:

$$\text{(id)} \quad \forall P. (P \leq P)$$

$$\text{(comp)} \quad \forall P, Q, R. (P \leq Q) \& (Q \leq R) \rightarrow (P \leq R)$$

$$(\top) \quad \forall P. (P \leq \top)$$

$$(\perp) \quad \forall P. (\perp \leq P)$$

$$(\&) \quad \forall P, Q, R. (P \leq Q \& R) \rightarrow (P \leq Q) \& (P \leq R)$$

$$(\vee) \quad \forall P, Q, R. (P \vee Q \leq R) \rightarrow (P \leq R) \& (Q \leq R)$$

$$(\rightarrow) \quad \forall P, Q, R. (P \leq Q \rightarrow R) \rightarrow (P \& Q \leq R)$$

$$(\neg) \quad \neg P := P \rightarrow \perp$$

$$(\leftrightarrow) \quad P \leftrightarrow Q := (P \rightarrow Q) \& (Q \rightarrow P)$$

3.18. Operations in ZHAs: \leq , \top , \perp , $\&$ and \vee

In a ZHA H ,

\top is the top element of H ,

\perp is the bottom element (00), and

$P \leq Q$ is true iff $P_L \leq Q_L$ and $P_R \leq Q_R$, or, equivalently,

$P \leq Q$ is true iff the ZPoset $(H, \text{WM}(H)^*)$ has an arrow $P \rightarrow Q$.

Also:

$$(P_L, P_R) \& (Q_L, Q_R) := (\min(P_L, Q_L), \min(P_R, Q_R))$$

$$(P_L, P_R) \vee (Q_L, Q_R) := (\max(P_L, Q_L), \max(P_R, Q_R))$$

Visually, this is:

(diagram)

3.19. Operations in ZHAs: \rightarrow

In a ZHA H ,

$P \leq Q$ returns 0 or 1, $(\leq: H \times H \rightarrow \{0,1\})$

$P \rightarrow Q$ returns an element of H $(\rightarrow: H \times H \rightarrow H)$.

The definition of $\rightarrow: H \times H \rightarrow H$ is by cases.

(Note: before I explain this to people I usually ask them to remember how weird they felt the definition of $\rightarrow: \{0,1\} \times \{0,1\} \rightarrow \{0,1\}$ was when they first saw it...)

```

Q → R := if      Q is below R then ⊤                <-- case 1
          elseif Q at left  of R then (↗)(Q&R)      <-- case 2
          elseif Q at right of R then (↖)(Q&R)      <-- case 3
          elseif Q is above R then R                 <-- case 4
          end
    
```

Note that the order matters!

If $R=22$, this is $\lambda Q.$ (which case we use to calculate $Q \rightarrow R$):

```

      4
     4 4
    2 4 3
   2 2 3 3
  2 2 1 3 3
 2  1 1 3
 1  1 1
 1  1
 1
    
```

3.20. Operations in ZHAs: \neg and \leftrightarrow

The operations \neg and \leftrightarrow in a ZHA are defined in terms of others.

$$\neg P := P \rightarrow \perp$$

$$P \leftrightarrow Q := (P \rightarrow Q) \& (Q \rightarrow P)$$

3.21. Some non-tautologies: DeMorgan

In classical logic both DeMorgan laws hold:

$$\neg(P \& Q) \leftrightarrow (\neg P \vee \neg Q)$$

$$\neg(P \vee Q) \leftrightarrow (\neg P \& \neg Q)$$

In intuitionistic logic these three implications are provable,

$$(\neg P \vee \neg Q) \rightarrow \neg(P \& Q)$$

$$\neg(P \vee Q) \rightarrow (\neg P \& \neg Q) \quad (\neg P \& \neg Q) \rightarrow \neg(P \vee Q)$$

While this one

$$\neg(P \& Q) \rightarrow (\neg P \vee \neg Q)$$

is false in some ZHAs, for example here:

32			
22			
21		12	
20	11	02	
10		01	
00			
$\neg(P \quad \& \quad Q) \rightarrow (\neg P \vee \neg Q)$			
--		--	
10 01		10 01	
-----		-----	
00		02 20	
-----		-----	
32		22	
-----		-----	
22			

3.22. Some non-tautologies: $\neg\neg P \leq P$

Exercise (homework!):

check that in the obvious ZHA -
the same as in the previous slide -
we have:

$$\begin{array}{r}
 00 \\
 00 \\
 00 \ 00 \\
 (\lambda P. \neg P) = 02 \ 00 \ 20 \\
 02 \ 20 \\
 32
 \end{array}$$

$$\begin{array}{r}
 32 \\
 32 \\
 32 \ 32 \\
 (\lambda P. \neg\neg P) = 20 \ 32 \ 02 \\
 20 \ 02 \\
 00
 \end{array}$$

$$\begin{array}{r}
 1 \\
 0 \\
 (\lambda P. \neg\neg P = P) = 0 \ 0 \\
 (\lambda P. \neg\neg P \leq P) = 1 \ 0 \ 1 \\
 0 \ 0 \\
 1
 \end{array}$$

3.23. Lindenbaum Algebras

We've been working with algebras of truth values.

What happens when we try to pass to an algebra of propositions?
"Wff" = "well-formed formula".

Let $A = \text{Wffs}(\top, \perp, \&, \vee, \rightarrow, \neg, \leftrightarrow, P, Q, R)$ be the set of wffs built from the constants \top, \perp , the operators $\&, \vee, \rightarrow, \neg, \leftrightarrow$ and the atomic propositions P, Q, R - plus parentheses.

Let D be the set of triples (P, Q, δ) where

δ is a /intuitionistic/ proof of $P \vdash Q$.

<- what is that????

Let D^- be the set of pairs (P, Q) obtained by deleting the third component of each triple $(P, Q, \delta) \in D$.

Then (A, D^-) is a directed graph.

(A, D^-) is reflexive and transitive - it is a /preorder/ on A .

It induces a quotient \sim on A .

$(A/\sim, D^-/\sim)$ is a partial order - the Lindenbaum Algebra generated by etc etc etc etc.

3.24. Lindenbaum Algebras (2)

Let $A = \text{Wffs}(\top, \perp, \&, \vee, \rightarrow, P, Q, R)$.

Let D be the set of triples (P, Q, δ) where
 δ is an intuitionistic proof of $P \vdash Q$ ($P, Q \in A$).

Let D^{\sim} be the set of pairs (P, Q) obtained by deleting
the third component of each triple $(P, Q, \delta) \in D$.

Then (A, D^{\sim}) is a directed graph,
and the partial order $(A/\sim, D^{\sim}/\sim)$ is the Lindenbaum Algebra (...)

Lindenbaum Algebras are (free) Heyting Algebras.

We can visualize Lindenbaum Algebras by writing some
wffs in A and

- 1) treating them as vertices in a directed graph
- 2) drawing the arrows in D between them
- 3) making each wff stand for its equivalence class.

3.25. Lindenbaum Algebra for classical logic

Let $A = \text{Wffs}(\top, \perp, \&, \vee, \rightarrow, \neg, \leftrightarrow, P)$ be the set of wffs...

Let D be the set of triples (P, Q, δ) where

δ is a /classical/ proof of $P \vdash Q$.

Then the Lindenbaum Algebra $(A/\sim, D^{\sim}/\sim)$ looks like this -
more precisely, it is transitive-reflexive closure of this:

\top

$P \quad \neg P$

\perp

This is a Heyting Algebra - but it is not free.

It is a Boolean Algebra - the free one on one generator.

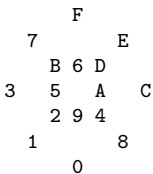
3.26. Lindenbaum Algebras for classical logic (2)

Let $A = \text{Wffs}(\top, \perp, \&, \vee, \rightarrow, \neg, \leftrightarrow, P, Q)$ be the set of wffs...

Let D be the set of triples (P, Q, δ) where

δ is a /classical/ proof of $P \vdash Q$.

Then the Lindenbaum Algebra $(A/\sim, D^{\sim}/\sim)$ looks like this - more precisely, it is the transitive-reflexive closure of this hypercube:



This is the free Boolean Algebra on two generators.

(Hint: take $P=3$ and $Q=5$)

3.27. Lindenbaum Algebras for IPL are infinite

Let $I_1 = \text{Wffs}(\top, \perp, \&, \vee, \rightarrow, \neg, \leftrightarrow, A)$ be the set of wffs...

Let D be the set of triples (P, Q, δ) where

δ is an /intuitionistic/ proof of $P \vdash Q$.

Then the Lindenbaum Algebra $(I_1/\sim, D^{\sim}/\sim)$ has at least these distinct points:

43		G		$G := F \vee F'$
42	33	F'	F	$F := E \vee E' \quad F' := F \rightarrow E$
	32	23	E	$E := D \vee D' \quad E' := E \rightarrow D$
31	22	D'	D	$D := C \vee C' \quad D' := D \rightarrow C$
	21	12	C	$C := B \vee B' \quad C' := C \rightarrow B$
20	11	B'	B	$B := A \vee A' \quad B' := B \rightarrow A$
	10	01	A	$A := 10 \quad A' := \neg A$
	00		00	

A, A', B, B', \dots are all distinct in the ZHA at the left above, so there cannot be an intuitionistic proof that any two of them - say, E and C - are equal.

~ This is an argument for adults!!!

We will make it children-friendly soon.

3.28. A computer library

The "spec" for a ZHA uses one char per line...

```
> ZHA.fromspec1r("123LLR432L1"):drawf()
  64
63 54
 53 44
52 43 34   <- 3
51 42 33 24 <- 4
 41 32 23   <- R (means 3 again, but moved right)
40 31 22    <- L (means 3 again, but moved left)
 30 21 12   <- L (means 3 again, but moved left)
  20 11 02  <- 3
    10 01   <- 2
      00    <- 1   <- first char of the spec
>
```

3.29. A computer library (2)

We can define functions using a very compact lambda-notation, and use them to draw the values of Zfunctions.

```
LO "P Eq(P, Not(Not(P)))"  
--> "function (P) return Eq(P, Not(Not(P))) end"
```

The ":drawL" method draws Zfunctions.

The global functions Eq, Not, And, Imp, etc use the global variable "z" - that's why the "z =".

```
> z = ZHA.fromspec1r("123LLRR21"):drawf()  
(...)  
> z:drawL "P           Not(P) "  
(...)  
> z:drawL "P           Not(Not(P)) "  
(...)  
> z:drawL "P Eq(P, Not(Not(P)))"  
      11  
      00 00  
      00 00 00  
      00 00 00  
11 00 00  
      00 00 00  
      00 00 11  
      00 00  
      11  
>
```

3.30. When in doubt use brute force

When in doubt use brute force.

Dennis Ritchie

In the beginning, and for several years, I had very little intuition about how the \rightarrow behaved in a Heyting Algebra...

And to make things worse, there are lots of things that are important in Topos Theory - namely: the $\neg\neg$ operator, modalities, sheaves, forcing - about which I had no intuition whatsoever!

It *had* to be possible to visualize them -

That was my main motivation for working with ZSets, ZHAs, etc.

Status of the project:

now I know how to visualise *most* of the operations and theorems in Fourman/Scott79 -

and *some* things about sheaves and geometric morphisms, mainly from Peter Johnstone's "Sketches of an Elephant"

and now I am in a position to get help from adult toposophers!

=) =) =)

4. Categories

4.1. Categories

A category

$$C = (C_0, \text{Hom}_C, \text{id}_C, \text{o}_C; \text{idL}_C, \text{idR}_C, \text{assoc}_C)$$

is a directed graph on steroids.

C_0 is the set (<- version for children) of ‘‘objects’’ of C .

$\text{Hom}_C(A,B)$ returns the set (<- same)

of arrows (‘‘morphisms’’, ‘‘maps’’) from A to B .

$\text{id}_C(A)$ returns and ‘‘identity map’’ $\text{id}_A:A \rightarrow A$.

o_C tells how ‘‘composable’’ maps (whose tails and heads meet)

compose; we use ‘‘;’’ to write the composition in ‘‘diagrammatic order’’.

$$\begin{array}{ccccc} & f & & g & \\ A & \text{---}> & B & \text{---}> & C \\ & \text{-----}> & & & \\ & (\text{gof})=(f;g) & & & \end{array}$$

idL_C and idR_C tell us that identity maps are cancellable

at the left and at the right: $(\text{id}_A;f)=f$, $(f;\text{id}_B)=f$.

assoc_C tells us that composition is associative: $(f;g);h=f;(g;h)$.

A proto-category is just this:

$$C^- = (C_0, \text{Hom}_C, \text{id}_C, \text{o}_C)$$

Note (for adults!!!):

the ‘‘properties’’ idL_C , idR_C , assoc_C only make sense

when we know how to compare morphisms for equality.

4.2. Set is a category

This is our archetypal category:

$\text{Set} = (\text{Set}_0, \text{Hom_Set}, \text{id_Set}, \text{o_Set}; \text{idL_Set}, \text{idR_Set}, \text{assoc_Set})$

The objects of Set are sets - so, for example,
 $\{0,1\}, \{2,3,4,5,6\}, \{7,8,9\}, \emptyset, \mathbb{N}, \mathbb{R} \in \text{Set}_0$.

The morphisms of Set are the maps between sets - so, for example,
 $\text{Hom_Set}(\{0,1\}, \{2,3,4,5,6\})$ has 2^5 elements (functions!),
 $\text{Hom_Set}(\{2,3,4,5,6\}, \{7,8,9\})$ has 5^3 elements.

Composition in Set is composition of functions. For example:

		2 ->7		
		3 ->8		
		4 ->9		
	0 ->4	5 ->9		
	1 ->2	6 ->9		
{0,1}	----->	{2,3,4,5,6}	----->	{7,8,9}
	----->			
		0 ->9		
		1 ->7		

And $\text{id}_{\{7,8,9\}} = \{7|->7, 8|->8, 9|->9\}$.

4.3. ZPosets are categories

A ZPoset, like $K_{\downarrow*}$

(the "kite" with the (transitive-reflexive closure of) $BM(K)$) can be seen as a category, in which

$$\text{Hom}_{K_{\downarrow*}}(A,B) = \begin{cases} \{(A,B)\} & \text{when there is an arrow from } A \text{ to } B, \\ \emptyset & \text{when there isn't an arrow from } A \text{ to } B. \end{cases}$$

For example:

$$\begin{array}{ccccc} & ((1,3), (0,2)) & & ((0,2), (1,0)) & \\ (1,3) & \xrightarrow{\hspace{2cm}} & (0,2) & \xrightarrow{\hspace{2cm}} & (1,0) \\ & \xrightarrow{\hspace{4cm}} & & & \\ & & ((1,3), (1,0)) & & \end{array}$$

Posets can be seen as categories in which each Hom-set has at most one element.

Warning (for children!):

The "can be seen" is important!!!

Many slides ago we saw $K_{\downarrow*}$ as a pair made of a ZSet K and subset of $K \times K$...

But $K_{\downarrow*}$ "as a category" is a 7-uple!

Abuse of language!

Also, morphisms here are funny things - they are not functions, they are /pairs/!

4.4. Categories: associativity

Associativity assures us that diagrams whose cells commute commute fully.

Example:

$$\begin{array}{ccccc} A & \longrightarrow & B & \longrightarrow & C \\ | & & | & & | \\ v & & v & & v \\ D & \longrightarrow & E & \longrightarrow & F \end{array}$$

$$\begin{aligned} ab;bc;cf &= ab;(bc;cf) \\ &= ab;(be;ef) \\ &= (ab;be);ef \\ &= (ad;de);ef \end{aligned}$$

4.5. Categories: iso(morphisms) and identities

An iso is a map $f:A \rightarrow B$
 for which there is a $f^{-1}:B \rightarrow A$
 such that $(f;f^{-1})=id_A$ and $(f^{-1};f)=id_B$.

$$\begin{array}{ccc}
 & f & \\
 A & \xrightarrow{\quad} & B \\
 \backslash & | & \backslash \\
 id_A \backslash & f^{-1} & \backslash id_B \\
 & v & v & v \\
 & A & \xrightarrow{\quad} & B \\
 & & f &
 \end{array}$$

Inverses are unique. When we have both
 $(f;g)=id_A$ and $(g;f)=id_B$ and
 $(f;h)=id_A$ and $(h;f)=id_B$
 then we can prove that $g=h$:

$$\begin{array}{ccc}
 g,h & & \\
 B \xrightarrow{\quad} A & & g = g;id_A \\
 \backslash & | & \backslash & = g;(f;h) \\
 id_B \backslash & f & \backslash id_A & = (g;f);h \\
 & v & v & v \\
 & B & \xrightarrow{\quad} & A & = id_B;h \\
 & & & & = h \\
 & & & & g,h
 \end{array}$$

Note that we used identities (id_C) to define isos
 and the properties (idL_C , idR_C)
 that say that they identities are neutral elements
 to prove that inverses are unique.

4.6. Functors

A functor $F: C \rightarrow D$ is:

$$F = (F_0, F_1; \text{respids}_F, \text{respcomp}_F)$$

it is made of an `_action` on `objects_`, `F_0`,
that takes objects of `C` to objects of `D`,
and an `_action` of `morphisms_`, `F_1`,
that takes morphisms in `C` to morphisms in `D`...
and ‘‘properties’’ (that appear after the “;”):

$$\text{respids}_F = \forall A \in C_0. \text{id}_D(F_0(A)) = F_1(\text{id}_C(A))$$

$$\text{respcomp}_F = \forall A, B, C \in C_0. \forall f: A \rightarrow B, g: B \rightarrow C. F_1(f;g) = F_1(f);F_1(g)$$

4.7. Functors - on diagrams

A functor $F: \mathcal{C} \rightarrow \mathcal{D}$ takes

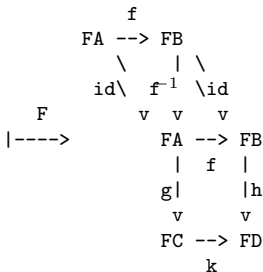
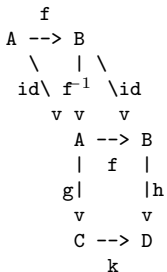
1. composable maps in \mathcal{C} to composable maps in \mathcal{D}
2. diagrams in \mathcal{C} to diagrams in \mathcal{D}
3. commutative diagrams in \mathcal{C} to commutative diagrams in \mathcal{D}
4. isos in \mathcal{C} to isos in \mathcal{D}

Exercise / homework:

show that if the diagram at the left below commutes then its image by F commutes too, and $F_1(f^{-1}) = F_1(f)^{-1}$.

Note: it has lots of (common!) shorthands...

You will have to figure them all out.



4.8. Functors: $(A \times)$ and $(0, 1x)$

$$\begin{array}{ccc}
 B & \dashrightarrow & FB \\
 | & & | \\
 f| & \mapsto & |Ff \\
 \downarrow & & \downarrow \\
 C & \dashrightarrow & FC
 \end{array}
 \qquad
 \begin{array}{ccc}
 B & \dashrightarrow & F_0(B) \\
 | & & | \\
 f| & \mapsto & |F_1(f) \\
 \downarrow & & \downarrow \\
 C & \dashrightarrow & F_0(C)
 \end{array}$$

$$\begin{array}{ccc}
 F & & F \\
 \backslash C & \dashrightarrow & \backslash D \\
 & & \backslash C & \dashrightarrow & \backslash D
 \end{array}$$

$$\begin{array}{ccc}
 B & \dashrightarrow & (A \times)B \\
 | & & | \\
 f| & \mapsto & |(A \times)f \\
 \downarrow & & \downarrow \\
 C & \dashrightarrow & (A \times)C
 \end{array}
 \qquad
 \begin{array}{ccc}
 B & \dashrightarrow & (A \times)_0(B) \\
 | & & | \\
 f| & \mapsto & |(A \times)_1(f) \\
 \downarrow & & \downarrow \\
 C & \dashrightarrow & (A \times)_0(C)
 \end{array}
 \qquad
 \begin{array}{ccc}
 B & \dashrightarrow & A \times B \\
 | & & | \\
 f| & \mapsto & |\lambda p. (\pi p, f(\pi' p)) \\
 \downarrow & & \downarrow \\
 C & \dashrightarrow & A \times C
 \end{array}$$

$$\begin{array}{ccc}
 (A \times) & & (A \times) & & (A \times) \\
 \backslash \text{Set} & \dashrightarrow & \backslash \text{Set} & & \backslash \text{Set} & \dashrightarrow & \backslash \text{Set}
 \end{array}$$

$$\begin{array}{ccc}
 \{2, 3\} & \dashrightarrow & \{0, 1\} \times \{2, 3\} \\
 | & & | \\
 2| \rightarrow 4 & | & \mapsto & | & (0, 2) \mapsto (0, 4), & (1, 2) \mapsto (1, 4), \\
 3| \rightarrow 6 & | & & | & (0, 3) \mapsto (0, 6), & (1, 3) \mapsto (1, 6) \\
 \downarrow & & & \downarrow & & \\
 \{4, 5, 6\} & \dashrightarrow & \{0, 1\} \times \{4, 5, 6\}
 \end{array}$$

4.9. Functors: why $(A \times)_1(f) = \lambda p.(\pi p, f(\pi' p))$?

$$\frac{\frac{\frac{p : ?}{\pi p : ?}}{\frac{p : ?}{\pi p : ?}} \quad \frac{\frac{p : ?}{\pi' p : ?} \quad f : B \rightarrow C}{f(\pi' p) : ?}}{(\pi p, f(\pi' p)) : ?}}{\lambda p.(\pi p, f(\pi' p)) : A \times B \rightarrow A \times C} \quad \frac{\frac{\frac{p : A \times B}{\pi p : A} \quad \frac{p : A \times B}{\pi' p : B} \quad f : B \rightarrow C}{f(\pi' p) : C}}{(\pi p, f(\pi' p)) : A \times C}}{\lambda p.(\pi p, f(\pi' p)) : A \times B \rightarrow A \times C}$$

How would someone *find* the expression $\lambda p.(\pi p, f(\pi' p))$? Idea: let's try to construct a function $? : A \times B \rightarrow A \times C$ from $f : B \rightarrow C$...

$$\frac{\frac{\frac{? : A \times B}{? : A} \quad \pi \quad \frac{\frac{? : A \times B}{? : B} \quad \pi' \quad f : B \rightarrow C}{? : C}}{\langle, \rangle}}{\frac{? : A \times C}{? : A \times B \rightarrow A \times C} \lambda}$$

4.10. Categories with

Almost all the time in Category Theory we work with categories with extra structure (and extra properties).

For example, Set , and each ZHA, is

- (1) a category with a terminal object
- (2) a category with binary products
- (3) a category with finite products (= 1+2)
- (4) a category with exponentials (needs 2)
- (5) a cartesian-closed category (= all the above)
- (6) a category with an initial object
- (7) a category with binary coproducts (a.k.a. binary sums)
- (8) a category with finite coproducts (a.k.a. finite sums; = 6+7)
- (9) a cartesian-biclosed category (= all the above)

4.11. BiCCCs

A Cartesian-biclosed category has these operations
(we use some nested uples for clarity):

$(C_0, \text{Hom}, \text{id}, \circ,$
 $(1, !), (\times, \pi, \pi', <, >), (\dashv, b, \#),$
 $(0, j), (+, \iota, \iota', [,]))$

We are omitting the ‘‘properties’’ that these operations obey.

Trick: a (bi-)Heyting Algebra has exactly the same structure as a (bi-)CCC.

A (bi-)Heyting Algebra is a (bi-)CCC in which each Hom-set has at most one element.

(See the handouts!)

5. Logic on subsets of ZDAGs

5.1. Order topologies

Let (A,R) be a graph.

Example: $(A,R) = (\{1,2,3\}, \{(1,3), (2,3)\})$.



A topology on a set X is a set of subsets of X (the "open sets").

A graph $R \subseteq A \times A$ induces a topology on A .

Here's how:

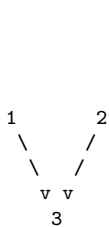
$$\begin{array}{l} \mathcal{O}(A) = \{ \underline{U} \subseteq A \mid (1 \in U \rightarrow 3 \in U) \ \& \ (2 \in U \rightarrow 3 \in U) \} \\ \qquad \qquad \qquad \backslash \text{-----} / \qquad \qquad \backslash \text{-----} / \\ \qquad \qquad \qquad \text{condition} \qquad \qquad \text{condition} \\ \qquad \qquad \qquad \text{induced by} \qquad \qquad \text{induced by} \\ \qquad \qquad \qquad 1 \rightarrow 3 \qquad \qquad 2 \rightarrow 3 \end{array}$$

If we order the open sets by inclusion

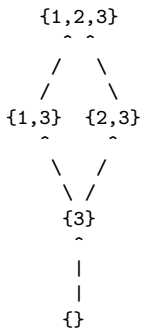
we get a partial order, $(\mathcal{O}(A), \subseteq)$.

Let's draw it as a graph.

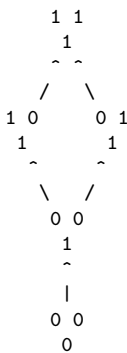
5.2. Order topologies (2)



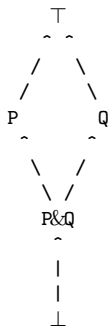
(A, R)



$(O(A), \subseteq)$



$(O(A), \subseteq)$



\perp

5.3. Topologies are Heyting Algebras

Look into almost any CT book
and you find this:

for any topological space $(X, \mathcal{O}(X))$
the poset $(\mathcal{O}(X), \subseteq)$ is a Heyting Algebra.

A HA is a structure $(\Omega, \leq, \top, \perp, \&, \vee, \rightarrow, \neg)$
obeying certain properties.

What are $\Omega, \leq, \top, \perp, \&, \vee, \rightarrow, \neg$?

$\Omega := \mathcal{O}(X)$	the open sets are the truth values.
$\top := X$	because \top is true everywhere
$\perp := \{\}$	because \perp is true everywhere
$\leq := \subseteq$	

$P \& Q := \{x \in X \mid x \in P \ \& \ x \in Q\}$	$P \& Q$ is true where both P and Q are	
$P \vee Q := \{x \in X \mid x \in P \ \vee \ x \in Q\}$	$P \vee Q$ is true where either P or Q are	
$P \rightarrow Q := \{x \in X \mid x \in P \rightarrow x \in Q\}$	$P \rightarrow Q$ is true where P implies Q	\leftarrow oops.

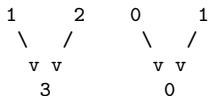
Problem: the definition of $P \rightarrow Q$ does not work always -
it sometimes produces a non-open output even when
both its inputs are open... example:

0	0	1	
0 0	\rightarrow	1 0	= 1 1
1 1		1 0	1 0

(open) (open) (non-open)

5.4. Order topologies: non-open subsets

A subset $B \subseteq A$ is non-open when it has a 1 above a 0 with an arrow $1 \rightarrow 0$ between them... for example, $\{2\}$ is not open:



5.5. Order topologies: stable subsets

When I explain these things to children I find it better to start using the term ‘stable subset’ (or stable function) instead of ‘open subset’.

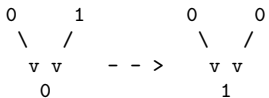
The idea:

‘1’s are ‘things’ (which are heavy),

‘0’s are ‘empty spaces’.

The ‘1’s want to fall down.

For example, 010 is unstable:



Then

$$P(* *) = \{0\ 0, 0\ 0, 0\ 1, 0\ 1, 1\ 0, 1\ 0, 1\ 1, 1\ 1\}$$
$$* \quad \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1$$

$$O(* *) = \{0\ 0, 0\ 0, \quad 0\ 1, \quad 1\ 0, \quad 1\ 1\}$$
$$* \quad \quad 0 \quad 1 \quad \quad 1 \quad \quad 1 \quad \quad 1$$

The ‘P’ means ‘parts’, but

The ‘O’ means ‘open sets’,

for technical reasons that we will understand much later...

(Children don’t know topology - I mean, usually)

5.6. Fixing the implication

The fix:

```
P→Q := {x∈X | x∈P → x∈Q}    <-- bad  
P→Q := {x∈X | x∈P → x∈Q}◦  <-- good ("◦" means "interior")
```

5.7. Modal logic for children (...but not now)

We can do MUCH better than just adding an " \circ ".

Let (A,R) be a graph, and $(A,O(A))$ its order topology.

These are two logics:

$$I = (\underset{I}{\Omega}, \underset{I}{\leq}, \underset{I}{\top}, \underset{I}{\perp}, \underset{I}{\&}, \underset{I}{\vee}, \underset{I}{\rightarrow}, \underset{I}{\neg})$$

$$M = (\underset{M}{\Omega}, \underset{M}{\leq}, \underset{M}{\top}, \underset{M}{\perp}, \underset{M}{\&}, \underset{M}{\vee}, \underset{M}{\rightarrow}, \underset{M}{\neg}, \underset{M}{\square}, \underset{M}{\diamond})$$

"I" is the intuitionistic logic we've been working on.

"M" is a modal logic (S4).

$$\Omega_I = O(A)$$

$$\Omega_M = P(A)$$

$$P \underset{I}{\rightarrow} Q := \underset{M}{\square} (P \underset{M}{\rightarrow} Q)$$

This gives us

- 1) Modal Logic for children
- 2) Godels embedding of intuitionistic logic into S4, for children

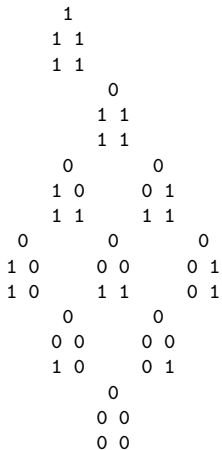
But we won't go that way now!!!

5.8. Some order topologies are ZHAs

For example,



---->



5.9. The generators of a ZHA

Let $(A, BM(A))$ be a ZDAG, and

let $(H, WM(H))$ be a ZHA such that $(H, WM(H)^*) \simeq (O(A), \sqsubseteq)$.

Let $G = \text{Gen}(H)$ - the ‘‘generators’’ of H - be the points of H with exactly one arrow going into them (in the ZDAG).

Then $\text{Gen}(H)$ is /exactly/ the image of $\$A\$$ in $\$H\$$ by the inclusion ‘ \downarrow ’!
 Magic! =)

$$\begin{array}{ccc}
 & & \begin{array}{c} 1 \\ 1\ 1 \\ 1\ 1 \\ 0 \\ 1\ 1 \\ 1\ 1 \\ 0\ 0 \\ 1\ 0\ 0\ 1 \\ 1\ 1\ 1\ 1 \\ 0\ 0\ 0 \\ 1\ 0\ 0\ 0\ 0\ 1 \\ 1\ 0\ 1\ 1\ 0\ 1 \\ 0\ 0\ 0\ 0 \\ 1\ 0\ 0\ 1 \\ 0 \\ 0\ 0 \\ 0\ 0 \end{array} \\
 \begin{array}{cc} 1 \\ / \ \backslash \\ v\ v \\ 2\ 3 \\ | \ | \\ v\ v \\ 4\ 5 \end{array} & \text{---->} & \begin{array}{ccc} 0 & 0 & 0 \\ 1\ 0 & 0\ 0 & 0\ 1 \\ 1\ 0 & 1\ 1 & 0\ 1 \\ 0 & 0 & \\ 0\ 0 & 0\ 0 & \\ 1\ 0 & 0\ 1 & \\ 0 & & \\ 0\ 0 & & \\ 0\ 0 & & \end{array}
 \end{array}$$

Can we recover $(A, BM(A))$ from (G, \sqsubseteq) ?

Well, sort of.

5.10. The generators of a ZHA (2)

Let $(A, \text{BM}(A))$ be a ZDAG, and

let $(H, \text{WM}(H))$ be a ZHA such that $(H, \text{WM}(H)^*) \simeq (\text{Opens}(A), \sqsubseteq)$.

Let $G = \text{Gen}(H)$.

Can we recover $(A, \text{BM}(A))$ from (G, \sqsubseteq) ?

Almost - we can make G inherit an order from H ...

Important: we need to look at H as a ZPoset, not as a ZDAGs,
and we need to invert the direction of the arrows.

5.11. The generators of a ZHA (3)

Look: $(A, BM(A)^*) \simeq (G, R^*)$.

		33	..	32	13
		32	23	32	..
32	13	22	13	..	13
20	12	21	12	..	12
10	01	20	11	20	..
		10	01	10	01
		00		..	10
					01

$$(A, BM(A)) \quad (H, WM(H)) \quad (G, R)$$

$$\simeq (\text{Opens}(A), \subseteq)$$

Now let's divide the generators in $G \subseteq H$ into two classes according to the direction of the (only) arrow that points to them. In the example above we have $\text{Gennw}(H) = \{10, 20, 32\}$ and $\text{Genne}(H) = \{01, 12, 13\}$.

The order on G is the columns $32 \rightarrow 20 \rightarrow 10$ and $13 \rightarrow 12 \rightarrow 01$, plus some intercolumn arrows.

5.12. The generators of a ZHA (4)

What happens when we "omit the other digit",
 and write $1_ , 2_ , 3_ \dots$ for the left-wall generators
 and $_1 , _2 , _3 \dots$ for the right-wall generators?

	33	..	3_	_3
	3_ 23	3_ ..	\	
3_ _3	22 _3	.. _3	v v v	
2_ _2	21 _2	.. _2	2_ _2	
1_ _1	2_ 11	2_ ..	/	
	1_ _1	1_ _1	v v v	
	00	..	1_	_1

(A, BM(A)) (H, WM(H)) (G, R)
 $\simeq (\backslash \text{Opens}(A), \square)$

It becomes easier to create homework problems!!!

We can specify the two-column graphs as

(height of the left column,
 height of the right column,
 inter-wall arrows from left to right,
 inter-wall arrows from right to right)

The ZHA above is generated by the two-column graph

(3, 3, {3->2}, {2->1}).

5.13. The generators of a ZHA: homework ideas

- 1) Draw the ZHA corresponding to each of the 2-column graphs specified below:

(3, 4, {}, {})

(3, 4, {}, {_4->_4})

(3, 4, {}, {_4->_3})

(3, 4, {}, {_4->_2})

(3, 4, {}, {_3->_2})

(3, 4, {}, {_4->_3, _3->_2})

(3, 4, {_2,_1}, {})

(3, 4, {_2,_1}, {_4->_3})

(3, 4, {_2,_1}, {_4->_3, _3->_2})

- 2) Draw the ZHA etc etc for these string specs:

"123RRLL2L321"

(...)

- 3) Find the 2-column graph and the string spec that generate each of the ZHAs below:

(...)

- 4) For each of the ZPosets below draw its topology as a graph.



(They are not going to be ZDAG-ish or ZPoset-ish, because when we have three independent points the topology contains a cube)

6. Modalities for children

6.1. The axioms

A modality on Ω is an operation $J: \Omega \rightarrow \Omega$ obeying these axioms:

$$M1: P \leq J(P)$$

$$M2: J(J(P)) \leq J(P)$$

$$M3: J(P \& Q) = J(P) \& J(Q)$$

What do these axioms mean (e.g., geometrically, on ZDAGs)?

- Equivalent elements form diamond-shaped regions
- Stable elements form a sub-ZDAG

To understand and prove this we start with consequences of M1, M2, M3

What familiar operations obey these axioms?

- $I(P) := P$ (the "identity" modality)
- $T(P) := \top$ (the "true" modality)
- $O_A(P) := A \text{ or } P$ (the "or" modality)
- $E_A(P) := A \rightarrow P$ (the "implication" (exponential) modality)
- $B_A(P) := (P \rightarrow A) \rightarrow A$ ("boolean quotient" - general case)
- $B(P) := B_{\perp}(P)$ ("boolean quotient" - particular case)
= $(P \rightarrow \perp) \rightarrow \perp$
= $\neg\neg P$

What other operations obey these axioms?

- Modalities on ZDAGs correspond to dividing by diagonal lines

6.2. Fourman/Scott1979: “Sheaves and Logic” This is an excerpt – pages 329-331 – from M.P. Fourman and D.S. Scott’s “Sheaves and Logic”, that was published in SLNM0753 (“Applications of Sheaves: Proceedings of the Research Symposium on Applications of Sheaf Theory to Logic, Algebra and Analysis - Durham, July 9-21, 1977”).

2.18. ELEMENTARY J-OPERATORS. In these examples Ω is a given cHa.

(i) *The closed quotient.* The operator is defined by

$$J_a p = a \vee p.$$

This is obviously a J-operator, and the congruence relation is:

$$a \vee p = a \vee q.$$

The set of fixed points (quotient lattice) is:

$$\{p \in \Omega \mid a \leq p\}.$$

Classically speaking in the spatial case where $\Omega = \mathcal{O}(X)$, the quotient corresponds to the topology on the *closed* subspace complementary to the open set a .

This quotient makes the element a “false” and is the least such.

6.3. Fourman/Scott1979 (2)

(ii) *The open quotient.* The operator is defined by:

$$J^a p = a \rightarrow p.$$

The congruence relation is:

$$a \& p = a \& q \quad (\text{equivalently, } a \leq p \leftrightarrow q).$$

The set of fixed points is thus isomorphic to

$$\Omega_a = \{ p \in \Omega \mid p \leq a \}.$$

Intuitionistically speaking in the spatial case, this quotient corresponds to the topology on the open subspace a . This quotient makes a “true” and is the least such.

6.4. Fourman/Scott1979 (3)

(iii) *The Boolean quotient.* The operator is defined by:

$$B_a p = (p \rightarrow a) \rightarrow a.$$

The congruence relation is:

$$p \rightarrow a = q \rightarrow a.$$

The set of fixed points is

$$\{p \in \Omega \mid (p \rightarrow a) \rightarrow a \leq p\}.$$

In case $a = \perp$, this is the well-known $\neg\neg$ -quotient giving the (complete) Boolean algebra of “stable” elements. For arbitrary a , we could first form Ω/J_a and follow this by the $\neg\neg$ -quotient to obtain Ω/B_a . (In general, if $J \leq K$, then Ω/K is a quotient of Ω/J .)

We remark that in general Ω/J is a cBa iff $J = B_{J_\perp}$. Further, if Ω is already Boolean, then every J -operator on Ω is of the form $B_a = J_a$.

6.5. Fourman/Scott1979 (4)

(iv) *The forcing quotient.* The operator is a combination of previous ones:

$$(J_a \& J^b)p = (a \vee b) \& (b \rightarrow p).$$

The congruence relation is a conjunction:

$$a \vee b = a \vee q \quad \text{and} \quad b \& p = b \& q.$$

The set of fixed points is:

$$\{p \in \Omega \mid b \rightarrow p \leq a \rightarrow p\}.$$

The point of the quotient is that it provides the *least* J-operator such that $Ja \leq Jb$; that is, we take the least quotient that “forces” $a \rightarrow b$ to be true. If we want to force a sequence of statements $a_i \rightarrow b_i$, for $i < n$, the operator needed is $\bigvee_{i < n} (J_{a_i} \& J_{b_i})$. It is important to note that in general sup’s of J-operators cannot be calculated pointwise. We shall see below, however, that it is possible to find a finite expression for this particular sup. (We owe this remark to John Cartmell).

6.6. Fourman/Scott1979 (5)

(vi) *A mixed quotient.* The interest of this example lies in the fact that it has a neat finite definition:

$$(B_a \& J^a)p = (p \rightarrow a) \rightarrow p.$$

The congruence relation is:

$$(p \rightarrow a) \rightarrow p = (q \rightarrow a) \rightarrow q,$$

which is equivalent to the conjunction:

$$a \& p = a \& q \quad \text{and} \quad p \rightarrow a = q \rightarrow a.$$

The set of fixed points is:

$$\{p \in \Omega \mid (p \rightarrow a) \rightarrow p \leq p\}.$$

It is difficult to make this set vivid except to say that it is the set of elements p satisfying Pierce's law (for a fixed a).

6.7. Fourman/Scott1979 (6)

If we take a polynomial in $\rightarrow, \&, \vee, \perp$, say $f(p, a, b, \dots)$, it is a decidable question whether for all a, b, \dots it defines a J-operator. This does not, however, help us very much in cataloguing such operators (nor in seeing what good they are!) Some techniques can be developed from the following formulae, which were pointed out to us by Cartmell, see also [33] and [47].

2.19. PROPOSITION. In the following, K is an arbitrary J-operator, \top is the constant function (the greatest J-operator with the most trivial quotient), and \perp is the least J-operator (namely, the identity function on Ω):

$$\begin{array}{ll}
 \text{(i)} & J_a \vee J_b = J_{a \vee b} & \text{(ii)} & J^a \vee J^b = J^{a \& b} \\
 \text{(iii)} & J_a \& J_b = J_{a \& b} & \text{(iv)} & J^a \& J^b = J^{a \vee b} \\
 \text{(v)} & J_a \& J_a = \perp & \text{(vi)} & J_a \vee J^a = \top \\
 \text{(vii)} & J_a \vee K = K \circ J_a & \text{(viii)} & J^a \vee K = J^a \circ K \\
 \text{(ix)} & J_a \vee B_a = B_a & \text{(x)} & J^a \vee B_b = B_{a \rightarrow b}
 \end{array}$$

Proof. Equations (i)-(iv) are easy calculations; while (v) comes down to showing $p = (a \vee p) \& (a \rightarrow p)$. Formula (vi) is a direct consequence of (vii) (equally, of (viii)).

6.8. How different modalities interact

(There is an algebra of modalities!)

How does a modality interact with $\&$, \vee , \rightarrow , etc?

What interesting things can we do with modalities?

- Quotients
- Sheafness, sheaves, sheafification (from topos theory)
- Geometric morphisms (from topos theory)

6.9. Diamond-shaped regions

A region (i.e., a subset) of a ZDAG is `_diamond-shaped_` if it has

- a top element
- a bottom element
- all elements between the top and the bottom, and nothing else

Formally, the `_diamond` between A and B_ is

$$\text{Db}(A,B) := \{P \text{ in } \Omega \mid A \leq P \leq B\}$$

Example: in

33			..
32	23	
22	13		22 ..
12	03	we have Db(01,22) =	12 ..
11	02	{01,02,11,12,22} =	11 02
10	01		.. 01
00			..

(Note: the standard terminology is "intervals")

6.10. Stable and equivalent truth-values

Remember: a modality on Ω is an operation $J: \Omega \rightarrow \Omega$ obeying these axioms:

$$M1: P \leq J(P)$$

$$M2: J(J(P)) \leq J(P)$$

$$M3: J(P \& Q) = J(P) \& J(Q)$$

M1 means that applying J once moves up,

M2+M1 mean that applying J a second time does not move further up.

(What is M3?)

Let's say that P is stable when $P = J(P)$,

and that P and Q are equivalent when $J(P) = J(Q)$.

Formally:

$$\text{St}(P) := (P = J(P)) \quad (\text{returns true or false})$$

$$\text{Sts} := \{P \text{ in } \Omega_m \mid \text{St}(P)\} \quad (\text{returns a set})$$

$$P == Q := (J(P) = J(Q)) \quad (\text{returns true or false})$$

$$E(P) := \{Q \text{ in } \Omega_m \mid P == Q\} \quad (\text{returns a set})$$

$$E_s := \{E(P) \mid P \text{ in } \Omega_m\} \quad (\text{returns a partition})$$

6.11. Stable and equivalent truth-values (2)

Let's look at an example.

When $J(P) := B(P) = \text{not not } P$, we have

	33		33		..
	32 23		33 33	
	22 13		33 33	
0m ==	12 03	(P ->J(P)) ==	33 03	Sts ==	.. 03
	11 02		33 03	
	10 01		10 03		10 ..
	00		00		10

and the equivalence classes are:

..	33
..	32 23
..	22 13
.. 03	12 ..
.. 02	11 ..
.. ..	10 01
00

i.e., $Es = \{\{00\}, \{10\}, \{01,02,03\}, \{11,12,13,22,23,32,33\}\}$

6.12. What is axiom M3? Introduction

Remember: a modality on Ω is an operation $J: \Omega \rightarrow \Omega$ obeying these axioms:

$$M1: P \leq J(P)$$

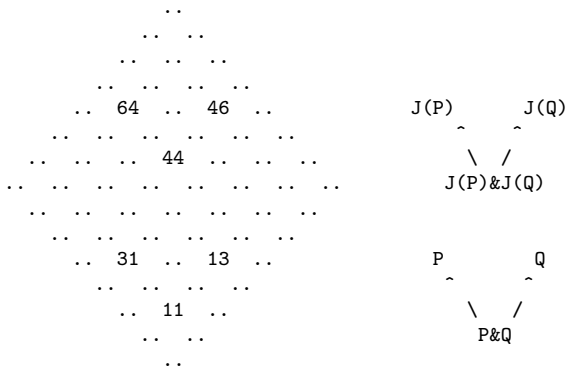
$$M2: J(J(P)) \leq J(P)$$

$$M3: J(P \& Q) = J(P) \& J(Q)$$

M3 implies `_monotonicity_`, which is:

$$M0: P \leq Q \text{ implies } J(P) \leq J(Q)$$

but this is not obvious. Let's prove that in several steps.



Let $P=31$, $Q=13$, $J(P)=64$, $J(Q)=46$.

Then $P \& Q=11$ and $J(P) \& J(Q)=44$, but without M3 we wouldn't know where is $J(P \& Q)=J(11)$... it could be anything above 11.

M3 tells us that $J(11)=J(P \& Q)=J(P) \& J(Q)=64 \& 46=44$,

which is below $J(P)=64$ and $J(Q)=46$...

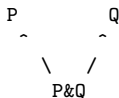
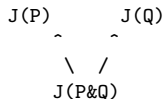
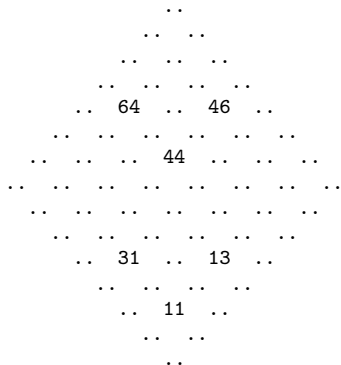
(Actually $J(11)$ is the topmost value below $J(P)$ and $J(Q)$.)

6.13. What is axiom M3? Monotonicity for products

M3 implies that $J(P \& Q) = J(P) \& J(Q)$,
 so let's rewrite the " $J(P) \& J(Q)$ " as " $J(P \& Q)$ " in our diagram.
 M3 implies that when we apply J to all elements of a
 product diagram $P \leftarrow P \& Q \rightarrow Q$, we do have arrows
 in its image $J(P) \leftarrow J(P \& Q) \rightarrow J(Q)$.

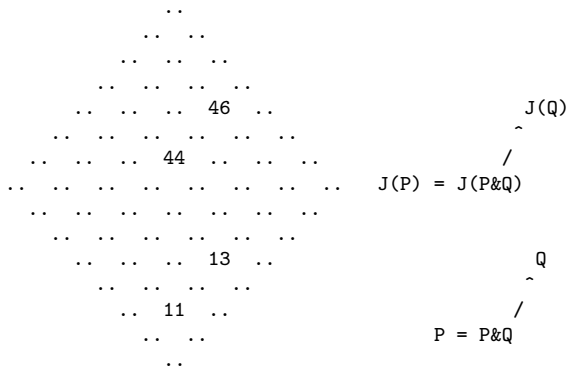
Formally, this is

Mp: $J(P \& Q) \leq J(P)$, and $J(P \& Q) \leq J(Q)$
 ("monotonicity for products")



6.14. What is axiom M3? Monotonicity

When $P \leq Q$ we have $P = P \& Q$,
 and a part of the previous diagram collapses..
 If $P=P \& Q=11$, $Q=13$, $J(P)=J(P \& Q)=44$, $J(Q)=46$,
 and we have:



6.15. What is axiom M3? Monotonicity (2)

Formally:

$$\text{-----Mp} \quad := \quad \frac{\text{-----M3} \quad \text{-----}}{\text{J(P\&Q) <= J(Q)}} \quad \frac{\text{J(P\&Q)=J(P)\&J(Q)} \quad \text{J(P)\&P(Q) <= J(Q)}}{\text{J(P\&Q) <= J(Q)}}$$

$$\text{-----Mo} \quad := \quad \frac{\text{P <= Q} \quad \text{P = P\&Q} \quad \text{-----Mp}}{\text{J(P) <= J(Q)}} \quad \frac{\text{J(P)=J(P\&Q)} \quad \text{J(P\&Q) <= J(Q)}}{\text{J(P) <= J(Q)}}$$

6.16. The sandwich lemma

From

$$M1: P \leq J(P)$$

$$M2: J(J(P)) \leq J(P)$$

$$M3: J(P \& Q) = J(P) \& J(Q)$$

We can prove that if $P \leq Q \leq J(P)$ then P and Q (and $J(P)$) are equivalent

More formally ("sandwich lemma"):

$$SL: \text{ If } P \leq Q \leq J(P) \text{ then } P = Q.$$

Proof:

$$\begin{array}{rcl}
 & & Q \leq J(P) \\
 & & \text{-----} M_0 \\
 & P \leq Q & J(Q) \leq J(J(P)) \quad J(J(P)) \leq J(P) \\
 & & \text{-----} M_2 \\
 P \leq Q \quad Q \leq J(P) & & \\
 \text{-----} SL & := & \text{-----} \\
 J(P) = J(Q) & & J(P) = J(Q)
 \end{array}$$

6.17. The sandwich lemma: equivalence classes don't have holes

Sandwich lemma:

SL: If $P \leq Q \leq J(P)$ then $P = Q$.

What does that mean? Let's look again at our example.
This was an equivalence class,

```
33
32 23
22 13
12 ..
11 ..
.. ..
..
```

and $J(12) = 33$.

Making $P=12$ and $Q=22$ in the sandwich lemma,
we have $P \leq Q \leq J(P)$, i.e., $12 \leq 22 \leq 33$,
and, by C1, $P=Q$, i.e., $12=22$, i.e., $J(12)=J(22)=33$.

The sandwich lemma is very powerful.

If we know that $J(11)=33$, then,
using it for all elements between 11 and 33,
we get that $J(12)=J(13)=J(22)=J(23)=J(32)=33$.

The sandwich lemma says that equivalence classes do not have "holes".

6.18. Equivalence classes: the bottom element lemma

The axiom

$$M3: J(P\&Q) = J(P)\&J(Q)$$

can be extended to $P\&Q\&R$, $P\&Q\&R\&S$, etc. Look:

$$\begin{aligned} J(P\&Q\&R) &= J(P\&(Q\&R)) \\ &= J(P)\&J(Q\&R) \\ &= J(P)\&J(Q)\&J(R) \end{aligned}$$

What happens when we take the "&" of all the truth-values in an equivalence class, and then we apply J to it?

Let's look at an example.

Let $E := \{11,12,13,22,23,32,33\}$

and $\&E := 11\&12\&13\&22\&23\&32\&33 \quad (= 11).$

```
  33
32 23
 22 13
   12 ..
  11 ..
.. ..
.. ..
```

$$\begin{aligned} J(\&E) &= J(11 \& 12 \& 13 \& 22 \& 23 \& 32 \& 33) \\ &= J(11) \& J(12) \& J(13) \& J(22) \& J(23) \& J(32) \& J(33) \\ &= 33 \& 33 \& 33 \& 33 \& 33 \& 33 \& 33 \\ &= 33 \end{aligned}$$

6.19. Equivalence classes: the bottom element lemma (2)

Let $E := \{11, 12, 13, 22, 23, 32, 33\}$
and $\&E := 11 \& 12 \& 13 \& 22 \& 23 \& 32 \& 33$ ($= 11$).

```
  33
32 23
  22 13
    12 ..
  11 ..
.. ..
..
```

$$\begin{aligned} J(\&E) &= J(11 \& 12 \& 13 \& 22 \& 23 \& 32 \& 33) \\ &= J(11) \& J(12) \& J(13) \& J(22) \& J(23) \& J(32) \& J(33) \\ &= 33 \& 33 \& 33 \& 33 \& 33 \& 33 \& 33 \\ &= 33 \end{aligned}$$

So, $\&E$ is $==$ -equivalent to all other elements of E ,
and $\&E$ is below all other elements of E ...

That means that $\&E$ is the bottom element of that equivalence class,
and, by the sandwich lemma, the equivalence class is made of
all elements between $\&E$ and $J(\&E)$.

TYPESET THIS IN RED:

(That only works for `_finite_` conjunctions, but we are working only
with finite cases...)

6.20. Equivalence classes: the bottom element lemma (3)

A small modification of that previous argument leads to this:
let E be any non-empty set of equivalent elements of $0m$.
Then the diamond-shaped region

$$E' := D(\&E, J(\&E))$$

having $\&E$ as its bottom element and $J(\&E)$ as its top element
contains E , and all elements in E' are equivalent.

For example, in

```
  33
32 23
 22 13
   12 ..
  11 ..
.. ..
.. ..
```

let $E := \{13, 22\}$; note that $J(13)=J(22)=33$, and that $\&E = 13\&22 = 12$.
Then

$$\begin{aligned} J(\&E) &= J(13 \& 22) \\ &= J(13) \& J(22) \\ &= 33 \& 33 \\ &= 33\dots \end{aligned}$$

6.21. Equivalence classes: the bottom element lemma (4)

let $E := \{13, 22\}$; note that $J(13)=J(22)=33$, and that $\&E = 13\&22 = 12$.
Then

$$\begin{aligned} J(\&E) &= J(13 \& 22) \\ &= J(13) \& J(22) \\ &= 33 \& 33 \\ &= 33, \end{aligned}$$

and all elements in the diamond-shaped region

$$E' := D(\&E, J(\&E)) = D(12, 33) = \{12, 13, 22, 23, 32, 33\}$$

are equivalent; this new lemma takes any arbitrary non-empty set of equivalent elements, and extends it to a diamond-shaped set of equivalent elements. In our example it extends $E=\{13,22\}$ to $E'=\{12,13,22,23,32,33\}$, ie.,

$$\begin{array}{cccc} & & & 33 \\ & & & 32 \ 23 \\ & & & 22 \ 13 \\ 22 \ 13 & & & \\ & & & 12 \ .. \\ & & & .. \ .. \\ & & & .. \ .. \\ & & & .. \ .. \\ & & & .. \end{array} \quad \text{to:} \quad \begin{array}{cccc} & & & 33 \\ & & & 32 \ 23 \\ & & & 22 \ 13 \\ & & & 12 \ .. \\ & & & .. \ .. \\ & & & .. \ .. \\ & & & .. \end{array}$$

6.22. An operation dual to J

Let $\text{Eq}(P)$ be the equivalence class of P .

Formally,

$$\text{Eq}(P) := \{Q \mid J(P)=J(Q)\}.$$

We know that $J(P)$ returns the top element of $\text{Eq}(P)$.

We saw, by the bottom element lemma, that each $\text{Eq}(P)$ has a bottom element

Let's denote that bottom element by $B(P)$.

Formally, $B(P)$ is the "&" of all elements of $\text{Eq}(P)$.

(A finite conjunction because we are in the finite case!)

We have that $\text{Eq}(P)$ is made of all elements between $B(P)$ and $J(P)$,
and nothing else...

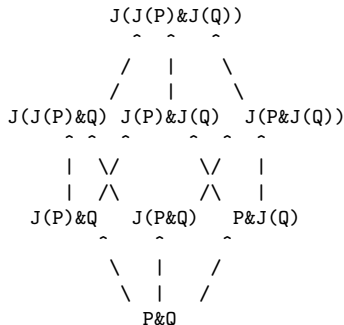
We will return to this later.

6.23. How J interacts with &

We know (proofs later!) that "&" is monotonous on both its inputs, and that $P \leq J(P)$ and that J is monotonous, i.e.,

$$\begin{array}{cccc}
 P \leq P' & Q \leq Q' & & P \leq Q \\
 \hline
 P \& Q \leq P' \& Q & P \& Q \leq P \& Q' & P \leq J(P) & J(P) \leq J(Q)
 \end{array}$$

These things mean that for any truth-values P and Q we know some relations between $P \& Q$, $J(P) \& Q$, ..., $J(J(P) \& J(Q))$:



But M3 tells us that $J(P \& Q) = J(P) \& J(Q)$, so at least the two middle vertices in the cube-shaped diagram above yield the same result... Actually we can prove more: $J(P \& Q)$ is equal to all the four upper truth-values in the diagram above. Proofs:

$$\begin{aligned}
 J(J(P) \& Q) &= J(J(P)) \& J(Q) &= J(P) \& J(Q) \\
 J(P \& J(Q)) &= J(P) \& J(J(Q)) &= J(P) \& J(Q) \\
 J(J(P) \& J(Q)) &= J(J(P)) \& J(J(Q)) &= J(P) \& J(Q)
 \end{aligned}$$

6.24. How J interacts with \vee

We know (proofs later!) that " \vee " is monotonous on both its inputs, and (again!) that $P \leq J(P)$ and that J is monotonous, i.e.,

$$\begin{array}{cccc}
 P \leq P' & Q \leq Q' & & P \leq Q \\
 \hline
 P \vee Q \leq P' \vee Q & P \vee Q \leq P \vee Q' & P \leq J(P) & J(P) \leq J(Q)
 \end{array}$$

These things mean that for any truth-values P and Q we know these relations between $P \vee Q$, $J(P) \vee Q$, ..., $J(J(P) \& J(Q))$:

$$\begin{array}{ccccc}
 & & J(J(P) \vee J(Q)) & & \\
 & / & | & \backslash & \\
 J(J(P) \vee Q) & J(P) \vee J(Q) & & J(P \vee J(Q)) & \\
 \wedge & \vee & & \vee & \wedge \\
 J(P) \vee Q & J(P \vee Q) & & P \vee J(Q) & \\
 \wedge & | & & | & \wedge \\
 & P \vee Q & & &
 \end{array}$$

But we know more. We can prove that we have $J(J(P) \vee J(Q)) \leq J(P \vee Q)$,

6.25. How J interacts with \vee (2)

But we know more. We can prove that we have $J(J(P)\vee J(Q))\leq J(P\vee Q)$,

$$\begin{array}{c}
 \begin{array}{cc}
 \text{-----} & \text{-----} \\
 P\leq P\vee Q & Q\leq P\vee Q \\
 \text{-----} \text{---} \text{Mo} & \text{-----} \text{---} \text{Mo} \\
 J(P)\leq J(P\vee Q) & J(Q)\leq J(P\vee Q) \\
 \text{-----} & \\
 J(P)\vee J(Q)\leq J(P\vee Q) \\
 \text{-----} \text{---} \text{Mo} & \text{-----} \text{---} \text{M2} \\
 J(J(P)\vee J(Q))\leq J(J(P\vee Q)) & J(J(P\vee Q))=J(P\vee Q) \\
 \text{-----} & \\
 J(J(P)\vee J(Q))\leq J(P\vee Q)
 \end{array}
 \end{array}$$

and this implies that all the 4 truth-values in the upper lozenge are equal...

6.26. How ZHAs are cut into equivalence classes

We saw that the equivalence classes of J are diamond-shaped regions, but we don't know yet how which ways of splitting a ZHA into diamond-shaped regions correspond to "J"s that obey M1, M2 and M3... For example, does this correspond a valid J?

```

    55
  55 55
 55 55 35
55 55 35 35
51 55 33 35 35
50 51 33 33 35 35
 50 51 33 13 35
  50 21 13 13
   21 13 13
    13 13
     13
  
```



(*)

It turns out that no.

Look at the frontiers between equivalence classes.

We will see two separate arguments.

6.27. How ZHAs are cut into equivalence classes (2)

The first, based on the fact that $J(Q)=J(R)$ implies $J(P\&Q)=J(P\&R)$, shows that borders that make lambda and mirrored lambdas are not possible,



(lambda)



(mirrored lambda)



(y)



(mirrored y)

and a second one, based on the fact that $St(P)$ and $St(Q)$ implies $St(P\&Q)$, that shows that borders that make "y"s and mirrored "y"s are not possible.

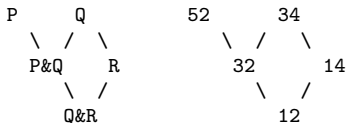
6.28. Borders like lambdas are not possible

Lemma: if $Q=R$ then $P&Q=P&R$ ("&' preserves equivalence").

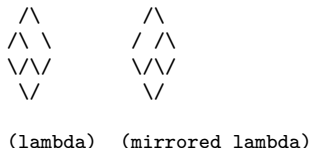
Proof:

$$\begin{array}{c}
 J(Q)=J(R) \\
 \hline
 J(Q)=J(R) \\
 \hline
 J(P&Q)=J(P&R)
 \end{array}
 \quad \&pe \quad := \quad
 \begin{array}{c}
 J(Q)=J(R) \\
 \hline
 J(P)\&J(Q)=J(P)\&J(R) \\
 \hline
 J(P&Q)=J(P&R)
 \end{array}$$

Diagrammatically, we can use "&" to "move southwest", as here:



In the diagram (*) we have $J(34)=J(14)=35$, but $J(32)=33$ and $J(12)=13$; if we take $P=52$, $Q=34$, $R=14$, the lemma &pe would imply that $J(32)=J(12)$, which does not hold. Diagrammatically, the lemma "&pe" implies that the lines marking the frontiers between different equivalence classes cannot make either a "lambda" or its mirror image...



6.29. Borders like "y"s are not possible

Lemma: $\text{St}(P)$ and $\text{St}(Q)$ implies $\text{St}(P \& Q)$ ("& preserves stability")

Proof:

$$\begin{array}{rcl}
 & P=J(P) & Q=J(Q) \\
 & \text{=====} & \text{-----} \\
 P=J(P) & Q=J(Q) & P \& Q = J(P) \& J(Q) \quad J(P) \& J(Q) = J(P \& Q) \\
 \text{-----} \& \text{ps} & := \text{-----} \\
 P \& Q = J(P \& Q) & & P \& Q = J(P \& Q) \\
 & & & \text{-----} \\
 & & & \text{-----} \text{M3}
 \end{array}$$

We start with the case in which our ZHA is a lozenge.

We know that equivalence classes are diamond-shaped regions.

When the ZHA is a lozenge, the equivalence classes are lozenges too.

Let's look at this example. Here are a ZHA H and a function $J: H \rightarrow H$ on it; J obeys $M1$ and $M2$ - remember the geometrical meaning of $M1$ and $M2$! - and we want to show that that J cannot obey $M3$ too.

```

  33
 32 23
 31 22 13
30 21 12 03
 20 11 02
 10 01
  00

```

```

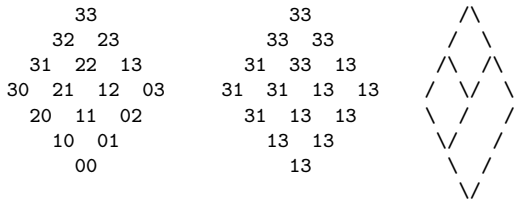
  33
 33 33
 31 33 13
31 31 13 13
 31 13 13
 13 13
  13

```



6.30. Borders like "y"s are not possible (2)

Let's look at this example. Here are a ZHA H and a function $J: H \rightarrow H$ on it; J obeys $M1$ and $M2$ - remember the geometrical meaning of $M1$ and $M2$! - and we want to show that that J cannot obey $M3$ too.



Let P' be the truth-value at the left of the vertex of the "y", and Q' be the truth-value at the right of the vertex of the "y". In this example, $P'=12$ and $Q'=21$.

Let P be $J(P')$,
and Q be $J(Q')$.

The line from P' to P points exactly northwest, and the line from Q' to Q points exactly northeast; because of this, $P' \& Q'$ is equal to $P \& Q$, and $P' \& Q'$ is the truth-value just below the vertex of the "y".

Let R be $P \& Q = P' \& Q'$.

P and Q are stable, and so, by "&ps", $R = P \& Q = P' \& Q'$ must be stable too. But, by hypothesis, the top element of the equivalence class of R is Q , so R is not stable, and thus the J in our example cannot obey $M3$.

6.31. Stable elements and diamond-shaped regions

Two slogans:

Equivalent elements form diamond-shaped regions

Stable elements form a sub-ZHA

Definitions:

$P==Q := P* \leftrightarrow Q*$ (P==Q is true or false)
 $St(P) := P == P*$ (St(P) is true or false)
 $Sts := \{P \text{ in } \Omega \mid P == P*\}$ (Sts is a set)
 $Di(P) := \{Q \text{ in } \Omega \mid P == Q\}$ (Di(P) is a set)
 $Dis := \{Di(P) \mid P \text{ in } \Omega\}$ (Dis is a partition of Ω)

33	33	33
32 23	33 33	33 33
22 13	33 13	33 33
12 03	13 13	33 03
11 02	11 13	33 03
10 01	11 11	10 03
00	11	00
Pid	PJ	Pnotnot

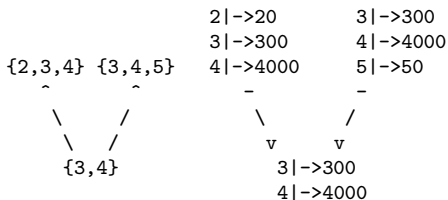
6.32. Modalities induce Z-quotients

6.33. Z-quotients are modalities

6.34. Modalities correspond to morphisms of HAs

7. Sheaves

7.1. Restriction, compatibility, glueing



The domain of 20,300,4000 is $\{2,3,4\}$.

The domain of 300,4000,50 is $\{3,4,5\}$.

The common domain of 20,300,4000 and 300,4000,50

is $\{2,3,4\} \cap \{3,4,5\} = \{3,4\}$.

The restriction of 20,300,4000 to $\{3,4\}$ is 300,4000.

The restriction of 300,4000,50 to $\{3,4\}$ is 300,4000. ← same!

20,300,4000 and 300,4000,50 are compatible.

We are moving down, toward smaller sets, using intersections.

Is there a way to move up - to unions?

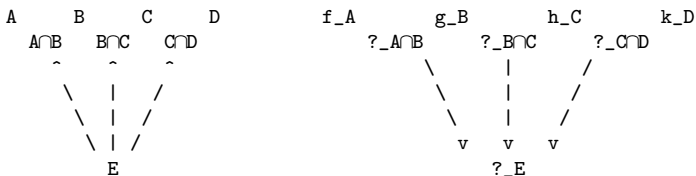
Yes, by glueing.

But we have to see some other things first.

7.2. Compatible families

Let's write the domain of a function in its subscript.
 Suppose that $f_A \text{ compat } g_B \text{ compat } h_C \text{ compat } k_D$,
 and that $E \subset A \cap B \cap C \cap D$.

Then the restrictions of f_A , g_B , h_C , k_D to E all coincide!!!
 The diagram below should make



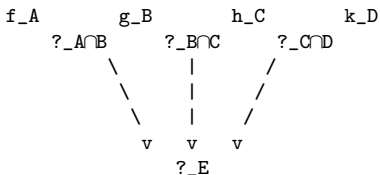
$$\begin{aligned}
 ?_{A \cap B} &= f_A|_{A \cap B} = g_B|_{A \cap B} & f_A|_E &= f_A|_{A \cap B}|_E = g_B|_{A \cap B}|_E = g_B|_E \\
 ?_{B \cap C} &= g_B|_{B \cap C} = h_C|_{B \cap C} & g_B|_E &= g_B|_{B \cap C}|_E = h_C|_{B \cap C}|_E = h_C|_E \\
 ?_{C \cap D} &= h_C|_{C \cap D} = k_D|_{C \cap D} & h_C|_E &= h_C|_{C \cap D}|_E = k_D|_{C \cap D}|_E = k_D|_E
 \end{aligned}$$

We say that the family $\{f_A, g_B, h_C, k_D\}$ is
 /pairwise-compatible/ when any two elements in it
 are compatible - $f_A \text{ compat } g_B \text{ compat } h_C \text{ compat } k_D$
 plus $f_A \text{ compat } h_C$, $g_B \text{ compat } k_D$, $f_A \text{ compat } k_D$.

7.3. Compatible families: naming

A notation trick...

Compare:

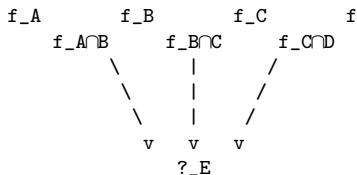


$$?_{A \cap B} := f_A|_{A \cap B} = g_B|_{A \cap B}$$

$$?_{B \cap C} := g_B|_{B \cap C} = h_C|_{B \cap C}$$

$$?_{C \cap D} := h_C|_{C \cap D} = k_D|_{C \cap D}$$

$$?_E := f_A|_E = g_B|_E = h_C|_E = k_D|_E$$



$$f_{A \cap B} := f_A|_{A \cap B} = f_B|_{A \cap B}$$

$$f_{B \cap C} := f_B|_{B \cap C} = f_C|_{B \cap C}$$

$$f_{C \cap D} := f_C|_{C \cap D} = f_D|_{C \cap D}$$

$$f_E := f_A|_E = f_B|_E = f_C|_E = f_D|_E$$

A compatible family $\{f_A, f_B, f_C, f_D\}$ induces a bigger family having one f_U for each empty set U that is smaller than $_some_$ of the sets A, B, C, D, \dots

And these ' f_U 's are well-defined, all ways of defining them from the original $\{f_A, f_B, f_C, f_D\}$ by restriction yield the same result.

7.4. Saturating downwards

Remember that we had an operation ‘ \downarrow ’ on DAGs that regarded each ‘1’ as black paint that was too liquid and that would flow down, painting the ‘0’s below it in black...

$$\begin{array}{r} 0 \qquad 0 \\ 0\ 0 \qquad 0\ 0 \\ 0\ 0\ 0 \qquad 0\ 0\ 0 \\ \downarrow 0\ 0\ 1\ 0 = 0\ 0\ 1\ 0 \\ 1\ 0\ 0 \qquad 1\ 1\ 1 \\ 0\ 0 \qquad 1\ 1 \\ 0 \qquad 1 \end{array}$$

Let’s use the same symbol, \downarrow , for an operation that starts with a compatible family, say $\{f_A, f_B, f_C, f_D\}$, and produces a (possible much bigger!) family made of all the restrictions of f_A, f_B, f_C, f_D .

Also:

$$\text{dom } f_A = A$$

$$\text{dom } \{f_A, f_B, f_C, f_D\} = \{A, B, C, D\}$$

7.5. An operation for all kinds of restrictions

This operation simplifies things magically.
(I learned it from Simmons's paper on Omega Sets).

$$\begin{aligned}U \cdot V &= U \& V \\U \cdot g_V &= (\text{error}) \\f_U \cdot V &= f_U | U \& V \\f_U \cdot g_V &= f_U | U \& V\end{aligned}$$

So: $U \cdot V = V \cdot U$ is true always
 $f_U \cdot g_V = g_V \cdot f_U$ is true iff f_U and g_V
are compatible

Now: $U \cdot \{V, W\} = \{U \cdot V, U \cdot W\} = \{U \cap V, U \cap W\}$
 $U \cdot 0(X) = \text{all (open) subsets of } U = \downarrow\{U\}$
 $f_U \cdot 0(X) = \text{all restrictions of } f_U$
 $\{f_U, g_V, h_W\} \cdot 0(X) = \text{all restrictions of } f_U, g_V, h_W,$
which yields something nice when
 f_U, g_V, h_W are compatible,
and something messy when they are not.

We can define $\downarrow A := A \cdot 0(X)$, and this works for all interesting cases!

7.6. Sheaves of functions

For each open set $U \subseteq \mathbb{R}$,

$\mathcal{C}^\infty(U, \mathbb{R})$ is the set of smooth functions from U to \mathbb{R} .

Warning: here (a, b) sometimes mean an open interval, not a pair!

$$\begin{array}{ccc}
 (4, 6) & \xrightarrow{\quad} & \mathcal{C}^\infty((4, 6), \mathbb{R}) & & f_{\{(4, 6)\}}: (4, 6) \rightarrow \mathbb{R} \\
 \uparrow & & | & & \downarrow \\
 \text{incl} & & | \text{restr} & & | \text{restr} \\
 | & & \downarrow & & \downarrow \\
 (5, 6) & \xrightarrow{\quad} & \mathcal{C}^\infty((5, 6), \mathbb{R}) & & f_{\{(4, 5)\}}: (4, 5) \rightarrow \mathbb{R}
 \end{array}$$

$$0(\mathbb{R}) \text{ --- } \rightarrow \text{Set}$$

We can regard $\mathcal{C}^\infty(-, \mathbb{R})$ as a (contravariant) functor.

Hey:

$$\mathcal{C}^\infty(-, \mathbb{R}): 0(\mathbb{R})^{\text{op}} \rightarrow \text{Set}$$

is a sheaf because compatible families can be glued in a unique way!

7.7. Limited functions are not sheaves

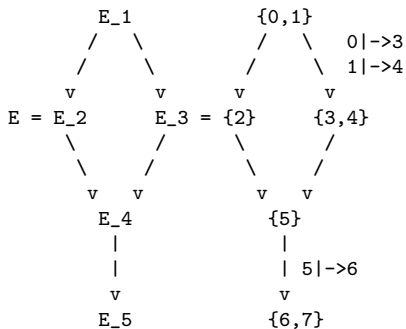
Let $L(-,R): \mathcal{O}(R)^{\text{op}} \rightarrow \text{Set}$
be the contravariant functor that returns
the `_limited_` functions on open sets.

Certain families of limited functions
are compatible, but their glueing is not limited,
so it "does not exist".

$L(-,R)$ violates one of the conditions for being a sheaf:
that glueings must exist.

The other condition is that glueings must be unique.
To violate that we need to go away from the realm of
the presheaves of functions, and use arbitrary functors.

7.8. The evil presheaf



- The compatible family $\{2,3\}$ has two different glueings: 0 and 1.
 The compatible family $\{2,4\}$ can't be glued.
 The compatible family $\{5,6,7\}$ has two different glueings: 6 and 7.

7.9. Sheaves in $Set^{K^{op}}$

...with the "obvious" notion of sheafness, are the ones that obey:

$$\begin{array}{c}
 A_1 = A_2 \times_{A_4} A_3 \\
 \swarrow \quad \searrow \\
 \begin{array}{c} \vee \\ A = A_2 \end{array} \quad \begin{array}{c} \vee \\ A_3 \end{array} \\
 \swarrow \quad \searrow \\
 \begin{array}{c} \vee \\ A_4 \end{array} \\
 | \\
 | \\
 \vee \\
 A_5 = 1
 \end{array}$$

So: A_1 and A_5 carry only redundant information, and sheaves on $Set^{K^{op}}$ are equivalent to objects of Set^V !!!

8. Toposes

8.1. A formal definition for children

A `_pullback_` for the diagram $A \rightarrow C \leftarrow B$ is an object D , together with morphisms $A \leftarrow D \rightarrow B$, that behaves like this:

$$\begin{array}{ccc}
 \{(a,b) \in A \times B \mid f(a)=g(b)\} & \dashrightarrow & B \\
 | & & | \\
 | & & | \quad g \\
 v & \quad f & v \\
 A & \dashrightarrow & C
 \end{array}$$

A `_subobject_` of A is a diagram $A' \rightarrow A$ that behaves like an inclusion.

Notation: $A' \dashrightarrow A$ means that the arrow is a monic.

A `_subobject classifier_` is an object Ω that behaves like $\{0,1\}$, plus an arrow $1 \dashrightarrow \Omega$ (called "true") that makes the curved arrow at the left have an inverse.

$$\begin{array}{ccc}
 \{a \in A \mid P(a)\} & \dashrightarrow & 1 & & A' & \dashrightarrow & 1 \\
 v & & v & & v & & v \\
 | \dashleftarrow \backslash & & | & & | \dashrightarrow \backslash & & | \\
 | & | & | & & i | & | & | \\
 v & | & P & v & v & v & v \\
 A & \dashrightarrow & \Omega & & A & \dashrightarrow & \Omega \\
 & & & & \lambda a. a \in i(A) & &
 \end{array}$$

A `_topos_` is a CCC that has pullbacks and a subobject classifier (\rightarrow extra structure and extra properties).

8.2. The main theorems

- 1) CCCs are exactly the categories in which we can interpret (the system $\lambda 1$ or simply-typed) λ -calculus
- 2) HAs are exactly the categories in which we can interpret (intuitionistic) propositional logic
- 3) Every HA is a CCC
- 4) Set is a CCC
- 5) Every ZHA is a HA
- 6) Every $(O(X), \underline{\subseteq})$ is a HA
- 7) Every ZTopos is a CCC <-- ZToposes are in the next slide
- 8) Every ZTopos is a topos
- 9) Topos are exactly the categories in which we can interpret (the system IST of intuitionistic, typed) set theory

There are many other ways of constructing toposes,
but I believe that ZToposes are the perfect tool for
doing Topos Theory for Children -

8.3. ZToposes

When we did logic on subsets of a ZHA - K, say -
each truth-value P was in fact a K-shaped diagram, e.g.,

$$P = \begin{array}{ccc} & 0 & P_1 \\ 1 & 0 & = P_2 & P_3 \\ & 1 & P_4 \\ & 1 & P_5 \end{array}$$

We are going to do the same with sets.

An object A of $\text{Set}^{\text{K}^{\text{op}}}$ is a K-shaped diagram of sets
and maps.

$$A = \begin{array}{ccc} & A_1 & \\ & / & \backslash \\ v & & v \\ A_2 & & A_3 \\ \backslash & & / \\ v & & v \\ & A_4 & \\ & | & \\ & v & \\ & A_5 & \end{array}$$

A notion of sheafness on $\text{Set}^{\text{K}^{\text{op}}}$ is something that tells us
which objects of $\text{Set}^{\text{K}^{\text{op}}}$ "are" sheaves.

The "sheaves" of $\text{Set}^{\text{K}^{\text{op}}}$ form a subcategory of $\text{Set}^{\text{K}^{\text{op}}}$
which is also a topos. <- Magic!!!

In the case of ZToposes these subcategories of sheaves are
not something weird and new - we have this:

for any ZDAG D,

8.4. Quotes from the Elephant

8.5. A translation diagram for Heyting Algebras

8.6. A translation diagram for BiCCCs

8.7. Do our operations $\&$, \vee , \rightarrow on ZHAs really deserve their names?

Many slides ago we talked about Alice's logic and Bob's logic... Alice's logic was this,

(copy)

and we were sort of leaving implicit that any other similar tuple, like

(random logic)

would be "a logic" in Alice's book-world...

But things are not like that!!!

I just explained a version for children first!!!!!!!!!!

What's missing:

```
          structure      properties (new!)
          /-----\ /-----\
a logic = (0m,and,or,...;eq_T,eq...)
```

where eq_T, ... are the properties (equations) that T,...

have to obey to "deserver their names", to "behave as expected".

They are:

(...)

8.8. The properties of \perp , \top , $\&$, \vee , \rightarrow : a diagram

[diagrammatic mnemonic]

[deduction rules (sequents with le)]

[forall-ish formulation]

We will return to this diagram /many/ times...

8.9. Definition 4.1.1

Definition 4.1.1 (a) Let \mathscr{E} and \mathscr{F} be toposes. A /geometric morphism/ $f:\mathscr{F} \rightarrow \mathscr{E}$ consists of a pair of functors $f_*:\mathscr{F}\rightarrow\mathscr{E}$