Title: Inference and Questions

Date: Mar 31, 2009 04:00 PM

URL: http://pirsa.org/09030026

Abstract: We know the mathematical laws of quantum mechanics, but as yet we are not so sure why those laws should be inevitable. In the simpler but related environment of classical inference, we also know the laws (of probability). With better understanding of quantum mechanics as the eventual goal, Kevin Knuth and I have been probing the foundations of inference. The world we wish to infer is a partially-ordered set ('poset') of states, which may as often supposed be exclusive, but need not be (e.g. A might be a requirement for B). In inference, a state of mind about the world degrades from perfect knowledge through logical OR, which allows for uncertain alternatives. We don't need AND, and we don't need NOT; we just need OR. This display of acceptable states of mind is [close to] a mathematical 'lattice'. We find that the OR structure by itself (!) forces the ordinary rules of probability calculus. No other rules are compatible with the structure of a lattice, so the ordinary rules are inevitable. The standard Shannon information/entropy is likewise inevitable. Taking this idea further, the OR of states of mind gives a lattice of 'Questions' that might be useful for automated learning. Disconcertingly, this lattice is very much larger (in class aleph-2), and the natural valuations on it exhibit large range. I will present this extension, and ask whether we can rationally foresee its use in practical application.

INFERENCE & QUESTIONS

John Skilling
with Kevin Knuth
skilling@eircom.net

I'm interested in learning about the classical world.

States $a, b, c, \ldots$ of world often have partial order.

$x \leq y$ means "x needs y"

e.g. can't have child without mother
can't have bridge without supports
can't have apple without itself.

So states form partially-ordered set ("poset").
traditionally omitted

Compute partial order with indicator function

$$I(x \mid y) = \begin{cases} 1 & \text{if } x \leq y \\ 0 & \text{otherwise} \end{cases} = \text{matrix of } 0\text{'s } \& \text{ } 1\text{'s}.$$

Definition of poset:

	MATH	COMPUTE
Reflexivity	$x \leq x$	Diagonal $I(x \mid x) = 1$
Antisymmetry	$x \leq y$ and $y \leq x$ $\Rightarrow x = y$	Transpose pairs $I(x \mid y)$, $I(y \mid x)$ not both 1.
Transitivity	$x \leq y$ and $y \leq z$ $\Rightarrow x \leq z$	$I(x \mid y) = I(y \mid z) = 1$ $\Rightarrow I(x \mid z) = 1$

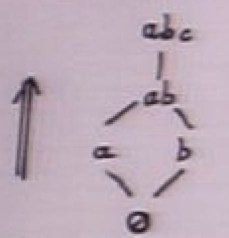
Compute partial order with indicator function

$$I(x|y) = \begin{Bmatrix} 1 & \text{if } x \le y \\ 0 & \text{otherwise} \end{Bmatrix} = \text{matrix of 0's \& 1's.}$$

Definition of poset:

	MATH	COMPUTE
Reflexivity	$x \le x$	Diagonal $I(x\|x) = 1$
Antisymmetry	$x \le y$ and $y \le x \Rightarrow x = y$	Transpose pairs $I(x\|y)$, $I(y\|x)$ not both 1.
Transitivity	$x \le y$ and $y \le z \Rightarrow x \le z$	$I(x\|y) = I(y\|z) = 1 \Rightarrow I(x\|z) = 1$

Example:



$$\begin{array}{c|ccccc|}
abc & 1 & 1 & 1 & 1 & 1 \\
ab & 1 & 1 & 1 & 1 & 0 \\
b & 1 & 0 & 1 & 0 & 0 \\
a & 1 & 1 & 0 & 0 & 0 \\
\emptyset & 1 & 0 & 0 & 0 & 0 \\
\hline
 & \emptyset & a & b & ab & abc
\end{array}$$

11111 is code for abc
11110 is code for ab
10100 is code for b
11000 is code for a
10000 is code for ∅

Order elements by #(1's in row)

This makes I upper triangular

$$I = \begin{bmatrix} ? & & & 1 \\ & & 1 & \\ & 1 & & \\ 1 & & 0 & \end{bmatrix}$$

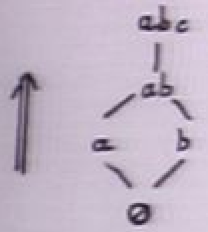
Rows all different, and can be used as a multi-bit "poset code" identifying and describing an element.

Relationship $x \le y$ is computable with binary OR $\veebar$ and AND $\barwedge$

$x \veebar y = y \iff x \le y \iff x \barwedge y = x$

(y has all 1's of x) (x has all 0's of y)

Example:



$$
\begin{array}{c|ccccc|}
abc & 1 & 1 & 1 & 1 & 1 \\
ab & 1 & 1 & 1 & 1 & 0 \\
b & 1 & 0 & 1 & 0 & 0 \\
a & 1 & 1 & 0 & 0 & 0 \\
\emptyset & 1 & 0 & 0 & 0 & 0 \\
\hline
 & \emptyset & a & b & ab & abc
\end{array}
$$

11111 is code for abc
11110 is code for ab
10100 is code for b
11000 is code for a
10000 is code for ∅

Order elements by #(1's in row)

This makes I upper triangular

$$
I = \begin{bmatrix} ? & & & 1 \\ & & 1 & \\ & 1 & & \\ 1 & & 0 & \end{bmatrix}
$$

Rows all different, and can be used as a multi-bit "poset code" identifying and describing an element.

Relationship $x \leq y$ is computable with binary OR $\veebar$ and AND $\barwedge$

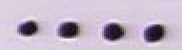
$$x \veebar y = y \iff x \leq y \iff x \barwedge y = x$$

(y has all 1's of x) (x has all 0's of y)

This is how I code my classical world.

$$
\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}
$$

Chain



$$
\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}
$$

Antichain (traditional)

But a poset has too little structure to be interesting.

(OR) My knowledge (belief, prejudice, ---) is uncertain.

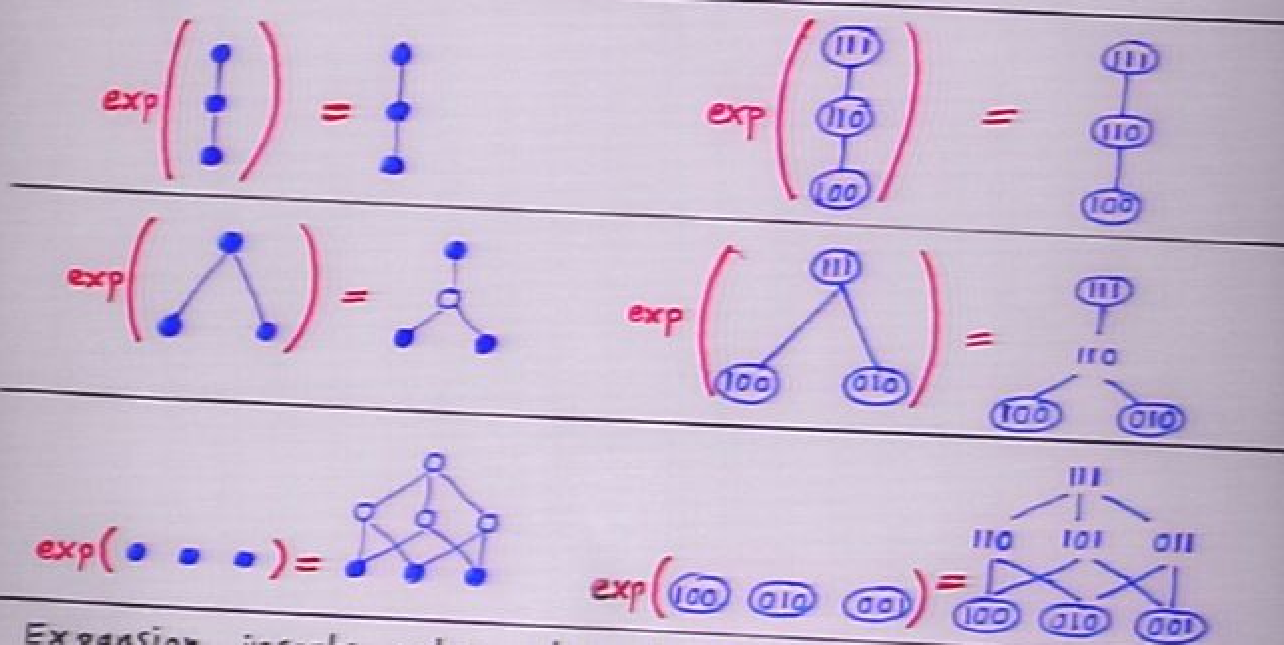
Assertion = "this state OR that state OR ----"

{Assertions} = {states **closed under OR**}

Canvas of assertions = **exp(**poset of states**)**

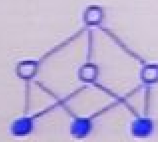
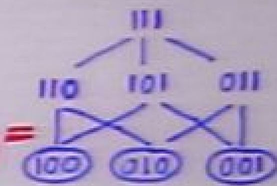
	MATH	COMPUTE
Closure under OR	Order-theoretic exponentiation	Expansion (exp)
Result of closure	Distributive join semi-lattice	Canvas

If $x \leq y$, knowledge "x OR y" is equivalent to just "y", so logical OR $\vee$ is coded as binary OR $\dot{\vee}$.



Expansion inserts extra elements ○ which are "OR" of others, leaving original generators ● identifiable.

∴ Expansion is reversible $\boxed{\text{Canvas} = \exp(\text{poset})}$

exp(• • •) =    exp(100 010 001) = 

Expansion inserts extra elements ○ which are "OR" of others, leaving original generators • identifiable.

∴ Expansion is reversible

$$\boxed{\begin{array}{l}\text{Canvas} = \exp(\text{poset}) \\ \log(\text{canvas}) = \text{poset}\end{array}}$$

(Birkhoff representation theorem)

AND   By construction, $x \tilde{\vee} y$ is in canvas.   3.

It follows that $x \tilde{\wedge} y$ is also in canvas

(unless $x \tilde{\wedge} y = 0000$, in which case $\nexists\, x \wedge y$,

though we may write $x \wedge y = \perp =$ "absurdity" $\notin$ canvas).

Basically, OR gives us AND as well.

NOT   Negation would invert codes $0 \Leftrightarrow 1$, but is <u>not</u> present.

A canvas always has 1111, but this never has negation 0000.

Other elements too may lack negation.

(The near-Boolean expansion of the traditional antichain <u>almost</u> allows negation, but that's a special case).

We may not use NOT.

Now for calculus - - -

AND By construction, $x \overset{\vee}{\vee} y$ is in canvas.

It follows that $x \overset{\wedge}{\wedge} y$ is also in canvas

(unless $x \overset{\wedge}{\wedge} y = 0000$, in which case $\nexists\ x \wedge y$,

though we may write $x \wedge y = \perp =$ "absurdity" $\notin$ canvas).

Basically, OR gives us AND as well.

NOT Negation would invert codes $0 \Leftrightarrow 1$, but is not present.

A canvas always has 1111, but this never has negation 0000.

Other elements too may lack negation.

(The near-Boolean expansion of the traditional antichain almost allows negation, but that's a special case).

We may not use NOT.

Now for calculus ----

VALUATION   Seek values $V(x)$ on a canvas, conforming to structure through relation between

$$V(x),\ V(y),\ V(x \vee y),\ V(x \wedge y).$$

Justification: $\vee$ and $\wedge$ are the canvas structure.

In special case $x \wedge y = \perp$, relation reduces to

$$V(x \vee y) = V(x) \oplus V(y)$$

for some operator $\oplus : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$

OR is associative,

$$(x \vee y) \vee z = x \vee (y \vee z).$$

In conformity, require

$$\underbrace{\big(V(x) \oplus V(y)\big) \oplus V(z) = V(x) \oplus \big(V(y) \oplus V(z)\big)}_{\text{associativity equation}}$$

Solution: $\exists\ F(V)$ such that

$$F(V(x \vee y)) = F(V(x)) + F(V(y))$$

Re-grade arbitrary initial V to $m(x) = F(V(x))$ with

$$m(x \vee y) = m(x) + m(y) \quad \text{when } x \wedge y = \perp.$$

General sum rule quickly follows.

$$\boxed{m(x \vee y) = m(x) + m(y) - m(x \wedge y)} \quad \text{with } m(\perp) = 0.$$

Structure $\Longrightarrow$ Calculus !

Equivalently, valuation is

$$m(x) = \sum_{\text{bits } k} \underbrace{I(k \mid x)}_{\text{Select "1"s}} \ \underbrace{\alpha_k}_{\text{weight of bit}}$$

e.g. $m(10110) = \alpha_1 + \alpha_3 + \alpha_4$

MEASURE.

If values conform to $\leq$, as well as $\vee$ and $\wedge$,

$$x \leq y \implies m(x) \leq m(y),$$

then the α's must be ≥ 0.

Valuation (initially + or −) becomes <u>measure</u> $m \geq 0$

e.g. $m(10110) = \alpha_1 + \alpha_3 + \alpha_4$

This is <u>why</u> measure theory is constructed as Σ weights.
It's forced by the structure of the canvas
on which applications are painted.
True for <u>any</u> canvas (not just traditional Boolean).

MASS and SHAPE

What re-grade $f(m)$ remains consistent with sum rule?
Only $m \longrightarrow Am$ (A = constant)
This suggests decomposing a canvas beneath its top element T as

$$m(x) = \underbrace{M}_{\text{mass} = m(T)} . \underbrace{p(x)}_{\text{shape } p(x|T) = \frac{m(x)}{m(T)}}$$

PROBABILITY

Is there any bi-valuation $w(x|y)$ for $x \leq y$ *other* than $m(x)/m(y)$?

Take ordering in two steps $x \leq y \leq z$.

Require $w(x|z) = w(x|y) \odot w(y|z)$ for some $\odot$

(else relation between x & z would depend on intermediate y).

Ordering is associative

$$(x \leq y \leq z) \leq t \quad = \quad x \leq (y \leq z \leq t)$$

$$\underbrace{\big(w(x|y) \odot w(y|z)\big) \odot w(z|t) = w(x|y) \odot \big(w(y|z) \odot w(z|t)\big)}_{\text{associativity equation}}$$

Solution : $\Phi\big(w(x|z)\big) = \Phi\big(w(x|y)\big) + \Phi\big(w(y|z)\big)$

But we are no longer free to re-grade w arbitrarily.
Need $w(\cdot|t)$ to stay consistent with sum rule.
This fixes Φ as log.

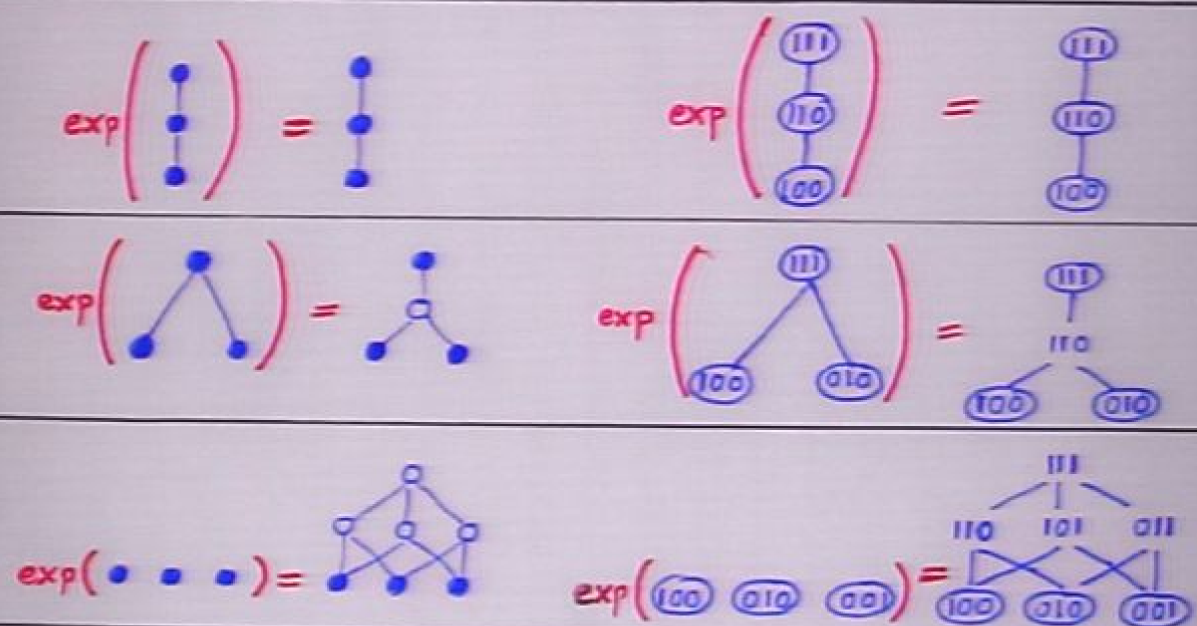
Assertion = "this state OR that state OR ----"

{Assertions} = {states **closed under OR**}

Canvas of assertions = **exp**(poset of states)

	MATH	COMPUTE
Closure under OR	Order-theoretic exponentiation	Expansion (exp)
Result of closure	Distributive join semi-lattice	Canvas

If $x \le y$, knowledge "x OR y" is equivalent to just "y",
so logical OR $\vee$ is coded as binary OR $\dot{\vee}$.



Expansion inserts extra elements ○ which are "OR" of others, leaving original generators ● identifiable.

∴ Expansion is reversible

$\boxed{\begin{array}{l} \text{Canvas} = \exp(\text{poset}) \\ \log(\text{canvas}) = \text{poset} \end{array}}$

PROBABILITY

Is there any bi-valuation $w(x|y)$ for $x \leq y$ <u>other</u> than $m(x)/m(y)$?

Take ordering in two steps $x \leq y \leq z$.

Require $w(x|z) = w(x|y) \odot w(y|z)$ for some $\odot$ (else relation between x & z would depend on intermediate y).

Ordering is associative

$$(x \leq y \leq z) \leq t \quad = \quad x \leq (y \leq z \leq t)$$

$$\underbrace{\big(w(x|y) \odot w(y|z)\big) \odot w(z|t) = w(x|y) \odot \big(w(y|z) \odot w(z|t)\big)}_{\text{associativity equation}}$$

Solution : $\Phi\big(w(x|z)\big) = \Phi\big(w(x|y)\big) + \Phi\big(w(y|z)\big)$

But we are no longer free to re-grade w arbitrarily.
Need $w(\cdot|t)$ to stay consistent with sum rule.
This fixes Φ as log.

Hence we get product rule

$$\boxed{p(x|z) = p(x|y)\, p(y|z)} \quad \text{for } x \leq y \leq z$$

This means $p(x|z) = \dfrac{m(x)}{m(z)}$ = shape as before

For general elements it soon follows that

$$\boxed{p(x|t) = \frac{m(x \wedge t)}{m(t)}}$$

called "probability"

This neo-frequentist definition subsumes the traditional rules

$p(x \vee y) = p(x) + p(y) - p(x \wedge y)$ sum

$p(x|t) = p(x|y)\, p(y|t)$ product

$0 \leq p(x) \leq 1$ range

NO ALTERNATIVE !

VARIATIONAL POTENTIAL

Is there a potential $H(\underset{\sim}{m})$ for measures ?

Write $\underset{\sim}{m} = M \underset{\sim}{p}$

Data on <u>mass</u> should not give us shape.

Data on <u>shape</u> should not give us mass.

$$\therefore \quad \nabla H(\underset{\sim}{m}) = \underbrace{f(M)}_{} + \underbrace{g(\underset{\sim}{p})}_{} \qquad \text{(shorthand)}$$

Lagrange multipliers

Solution :

$$H(x) = \sum_{\text{bit } k} \underbrace{I(k|x)}_{\text{Select "1"s}} \underbrace{\left(A + B\alpha_k + C\alpha_k \log \alpha_k\right)}_{\text{Potential for bit } k.}$$

e.g.

$$m(10110) = \alpha_1 + \alpha_3 + \alpha_4$$

$$H(10110) = \sum_{k=1,3,4} \left(A + B\alpha_k + C\alpha_k \log \alpha_k\right)$$

SUMMARY SO FAR

Measure, Probability, Information / Entropy all fixed by "closure under OR" structure.

Measure obeys sum rule and is ≥ 0.

Probability is ratio of measures.

Information and Entropy take "$p \log p$" form.

But inference is only one application.

These are pure math results.

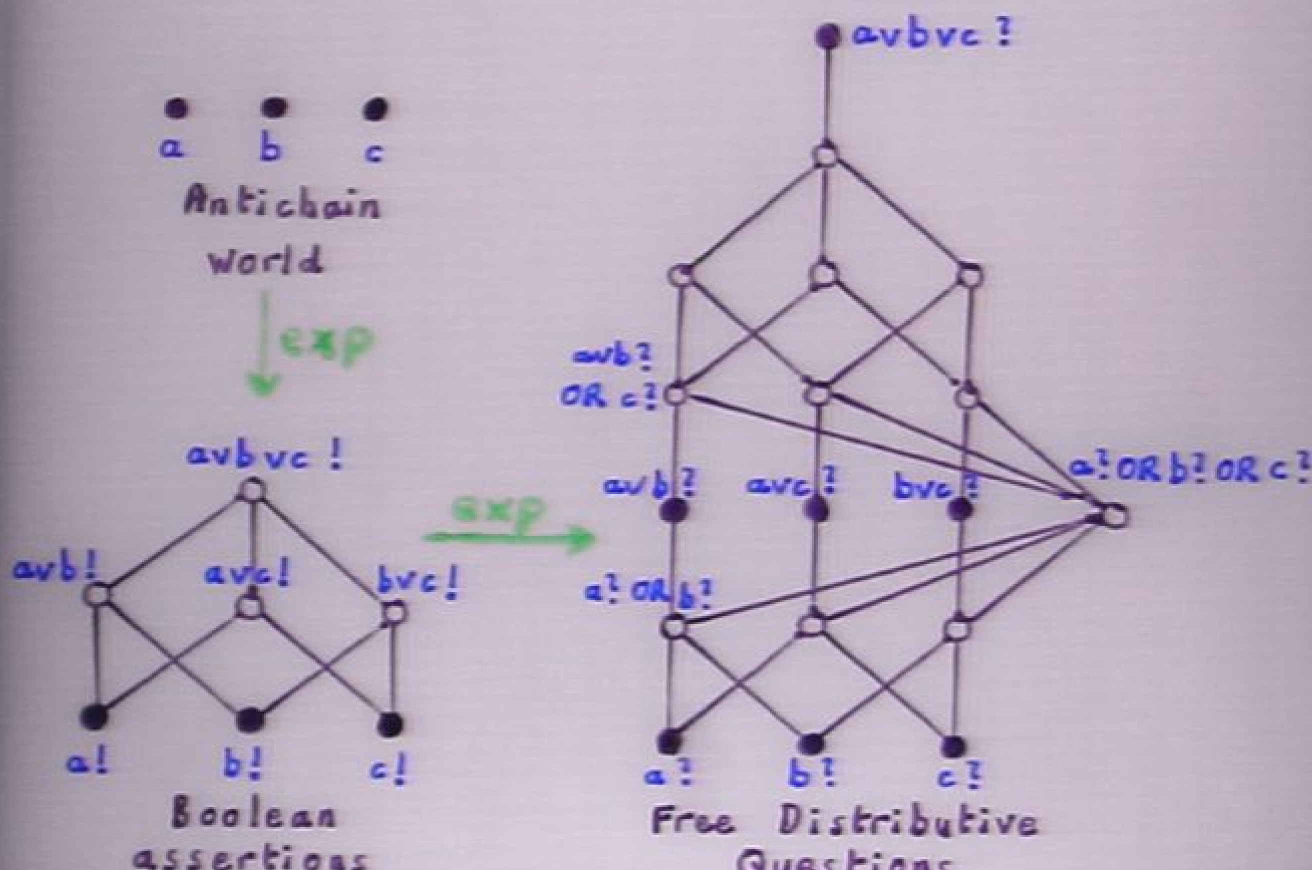
The standard rules are inevitable !

QUESTIONS

Canvas of assertions is itself a poset.
Let's expand it, OR-ing states of mind.

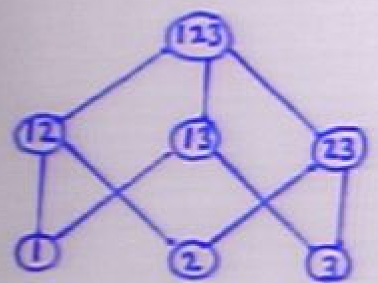
$$\exp(\text{states}) = \text{assertions}$$

$$\exp(\text{assertions}) = \text{Questions}$$

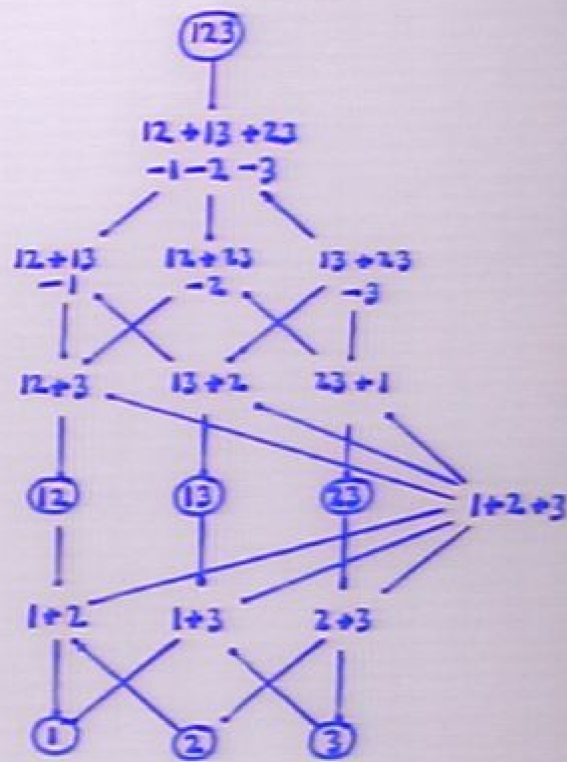


VALUATIONS

Assign values to assertions



Get values on Questions $\xrightarrow[\text{rule}]{\text{sum}}$



a) Assign probability values, $12 = p_1 + p_2$ etc.
Question values overlap. Nothing new.

b) Assign entropy, $12 = -p_1 \log p_1 - p_2 \log p_2$ etc.
Question values = "higher order information"
Ratios = "relevance of one Question to another".

Size

n	# states	# assertions	# Questions
1	1	1	
2	2	3	1
3	3	7	4
4	4	15	18
5	5	31	166
6	6	63	7579
7	7	127	7828352
8	8	255	2414682040996
9	9	511	56130437228687557907786
			may never be known
	n	2^n-1	$\sim 2^{\left(2^{n+1}/\sqrt{2\pi n}\right)}$

Size

n	# states	# assertions	# Questions
1	1	1	
2	2	3	1
3	3	7	4
4	4	15	18
5	5	31	166
6	6	63	7579
7	7	127	7828352
8	8	255	2414682040996
9	9	511	56130437228687557907786
			may never be known
	n	$2^n - 1$	$\sim 2^{(2^{n+1}/\sqrt{2\pi n})}$

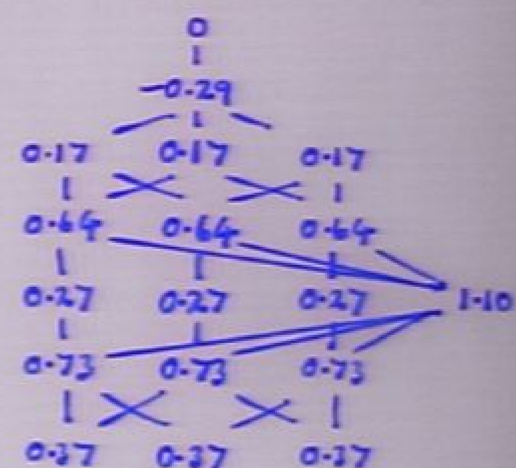
Values

With $n=3$ and $p_1 = p_2 = p_3 = \frac{1}{3}$

Get wrong ordering & signs

— inevitably because

$$S(\tau) = -1 \log 1 = 0$$



With $n = 50$, $p_1 = p_2 = \cdots \simeq 1/50$

values range $\pm$ several $\times 10^4$,

can be tiny or 0.

"relevance of one Question to another".

CONCLUSION

We have polished the foundations of measure, probability, information/entropy.

It's all firmly based on "OR".

"Questions" may be a step too far. Help?