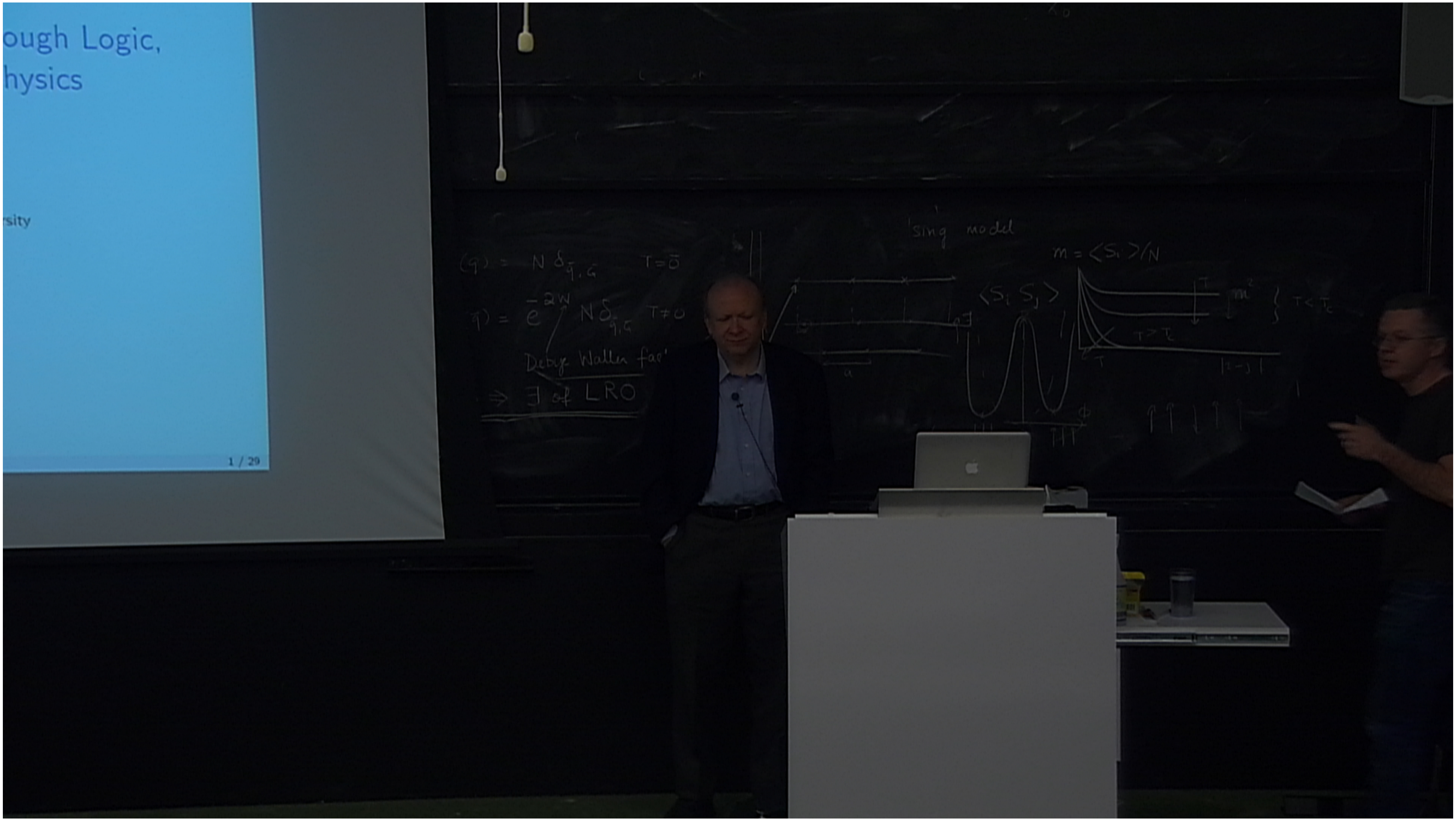
Title: Information Flow in Computation, Logic and Physics

Date: Nov 02, 2011 02:00 PM

URL: http://pirsa.org/11110076

Abstract: Ideas about information are pervasive, yet the fundamental nature and structure of information - if indeed it has one! - remains elusive. Work done from many different perspectives, including those of physics, biology, logic, computer science, statistics, and game and decision theory, has yielded insights into various aspects of information. Could there be a comprehensive, unified theory? We shall chart a particular path, focusing on the idea of *information flow*, and show how common mathematical structures arise in the description of information flow in computer science, logic and quantum information.

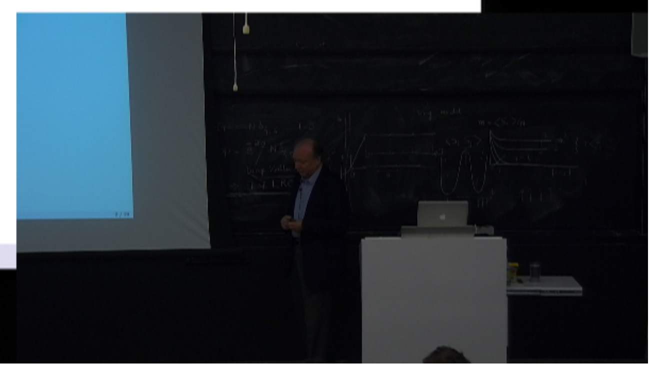

ough Logic,
hysics
rsity
1 / 29
Ising model
Debye Waller fac
LRO

Information Flow: tracing a path through Logic, Computation, Topology and Physics

Samson Abramsky

Department of Computer Science, Oxford University

Information, Information . . .



Samson Abramsky (Department of Computer Science, Information Flow: tracing a path through Logic, Comp

Information, Information . . .

Information is everywhere. We live in an Information Age.

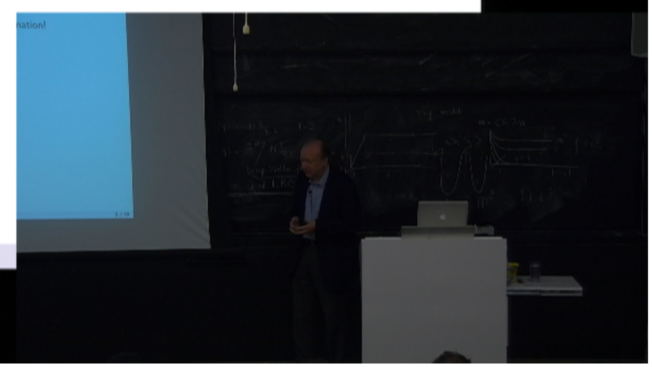
Information, Information ...

Information is everywhere. We live in an Information Age.

But what is the fundamental nature and structure of information!

(Sceptics: If it has these things!)

A lot of disciplines have things to say about information:



Information, Information . . .

Information is everywhere. We live in an Information Age.

But what is the fundamental nature and structure of information!

(Sceptics: If it has these things!)

A lot of disciplines have things to say about information:

Physics, biology, mathematics, logic, computer science, statistics, game and decision theory, linguistics, . . .

Information, Information . . .

Information is everywhere. We live in an Information Age.

But what is the fundamental nature and structure of information!

(Sceptics: If it has these things!)

A lot of disciplines have things to say about information:

Physics, biology, mathematics, logic, computer science, statistics, game and decision theory, linguistics, . . .

The blind men and the elephant?

All have yielded insights into various aspects of information. Could there be a comprehensive, unified theory?

Information, Information . . .

Information is everywhere. We live in an Information Age.

But what is the fundamental nature and structure of information!

(Sceptics: If it has these things!)

A lot of disciplines have things to say about information:

Physics, biology, mathematics, logic, computer science, statistics, game and decision theory, linguistics, . . .

The blind men and the elephant?

All have yielded insights into various aspects of information. Could there be a comprehensive, unified theory?

Sceptic: didn't Shannon give the definitive approach?

An Anecdote

An Anecdote

A cautionary tale from over 10 years back.

An Anecdote

A cautionary tale from over 10 years back.

Consider the following 'puzzle':

Why do we compute?

An Anecdote

A cautionary tale from over 10 years back.

Consider the following 'puzzle':

Why do we compute?

Obvious answer: to *gain information* (which, therefore, we didn't previously have).

An Anecdote

A cautionary tale from over 10 years back.

Consider the following ‘puzzle’:

Why do we compute?

Obvious answer: to *gain information* (which, therefore, we didn’t previously have).

Distinguished physicist (now knighted :) asks:

An Anecdote

A cautionary tale from over 10 years back.

Consider the following 'puzzle':

Why do we compute?

Obvious answer: to *gain information* (which, therefore, we didn't previously have).

Distinguished physicist (now knighted :) asks:

> *'I don't know much about computer science, but I know some information theory. How can you speak of information increase? Information theory tells us you can only lose information!*

An Anecdote

A cautionary tale from over 10 years back.

Consider the following 'puzzle':

Why do we compute?

Obvious answer: to *gain information* (which, therefore, we didn't previously have).

Distinguished physicist (now knighted :) asks:

> *'I don't know much about computer science, but I know some information theory. How can you speak of information increase? Information theory tells us you can only lose information!*

A couple of morals:

An Anecdote

A cautionary tale from over 10 years back.

Consider the following 'puzzle':

Why do we compute?

Obvious answer: to *gain information* (which, therefore, we didn't previously have).

Distinguished physicist (now knighted :) asks:

> *'I don't know much about computer science, but I know some information theory. How can you speak of information increase? Information theory tells us you can only lose information!*

A couple of morals:

- Information gain is relative to *subsystems* (or *agents*).

An Anecdote

A cautionary tale from over 10 years back.

Consider the following 'puzzle':

Why do we compute?

Obvious answer: to *gain information* (which, therefore, we didn't previously have).

Distinguished physicist (now knighted :) asks:

> *'I don't know much about computer science, but I know some information theory. How can you speak of information increase? Information theory tells us you can only lose information!*

A couple of morals:

- Information gain is relative to *subsystems* (or *agents*).
- The dynamics of computation (and language, cognition, etc.) is about *information flow* to and from these sub-systems/agents.

A simple model of information increase in computation

A simple model of information increase in computation

Recursive definitions: fundamental in computation — and in logic and foundations of mathematics.

A simple model of information increase in computation

Recursive definitions: fundamental in computation — and in logic and foundations of mathematics.

E.g. a recursive definition of factorial:

$$\mathbf{fact}(n) = \mathbf{if}\ n = 0\ \mathbf{then}\ 1\ \mathbf{else}\ n \times \mathbf{fact}(n-1).$$

A simple model of information increase in computation

Recursive definitions: fundamental in computation — and in logic and foundations of mathematics.

E.g. a recursive definition of factorial:

$$\mathbf{fact}(n) = \mathbf{if}\ n = 0\ \mathbf{then}\ 1\ \mathbf{else}\ n \times \mathbf{fact}(n-1).$$

We can understand recursive definitions as *fixpoints*: $\mathbf{fact} = F(\mathbf{fact})$, where F is a suitable *functional* (acting on partial functions of natural numbers).

A simple model of information increase in computation

Recursive definitions: fundamental in computation — and in logic and foundations of mathematics.

E.g. a recursive definition of factorial:

$$\mathbf{fact}(n) = \mathbf{if}\ n = 0\ \mathbf{then}\ 1\ \mathbf{else}\ n \times \mathbf{fact}(n-1).$$

We can understand recursive definitions as *fixpoints*: $\mathbf{fact} = F(\mathbf{fact})$, where F is a suitable *functional* (acting on partial functions of natural numbers).

Zeroth approximation: $\mathbf{fact}_0 = \varnothing$. We have *no information* about **fact**.

A simple model of information increase in computation

Recursive definitions: fundamental in computation — and in logic and foundations of mathematics.

E.g. a recursive definition of factorial:

$$\mathbf{fact}(n) = \mathbf{if}\ n = 0\ \mathbf{then}\ 1\ \mathbf{else}\ n \times \mathbf{fact}(n-1).$$

We can understand recursive definitions as *fixpoints*: $\mathbf{fact} = F(\mathbf{fact})$, where F is a suitable *functional* (acting on partial functions of natural numbers).

Zeroth approximation: $\mathbf{fact}_0 = \varnothing$. We have *no information* about **fact**.

First approximation: $\mathbf{fact}_1 = \{(0, 1)\}$. We *gain information* about one value of **fact**.

A simple model of information increase in computation

Recursive definitions: fundamental in computation — and in logic and foundations of mathematics.

E.g. a recursive definition of factorial:

$$\mathbf{fact}(n) = \mathbf{if}\ n = 0\ \mathbf{then}\ 1\ \mathbf{else}\ n \times \mathbf{fact}(n-1).$$

We can understand recursive definitions as *fixpoints*: $\mathbf{fact} = F(\mathbf{fact})$, where F is a suitable *functional* (acting on partial functions of natural numbers).

Zeroth approximation: $\mathbf{fact}_0 = \varnothing$. We have *no information* about **fact**.

First approximation: $\mathbf{fact}_1 = \{(0,1)\}$. We *gain information* about one value of **fact**.

Second approximation: $\mathbf{fact}_2 = \{(0,1),(1,1)\}$.

A simple model of information increase in computation

Recursive definitions: fundamental in computation — and in logic and foundations of mathematics.

E.g. a recursive definition of factorial:

$$\mathbf{fact}(n) = \mathbf{if}\ n = 0\ \mathbf{then}\ 1\ \mathbf{else}\ n \times \mathbf{fact}(n-1).$$

We can understand recursive definitions as *fixpoints*: $\mathbf{fact} = F(\mathbf{fact})$, where F is a suitable *functional* (acting on partial functions of natural numbers).

Zeroth approximation: $\mathbf{fact}_0 = \varnothing$. We have *no information* about **fact**.

First approximation: $\mathbf{fact}_1 = \{(0, 1)\}$. We *gain information* about one value of **fact**.

Second approximation: $\mathbf{fact}_2 = \{(0, 1), (1, 1)\}$.

k-th approximation gives us the first k values. The *limit of the increasing sequence of approximations* gives us the whole thing — which is the *least fixpoint* of the associated functional.

Reflexivity and Self-Application

Reflexivity and Self-Application

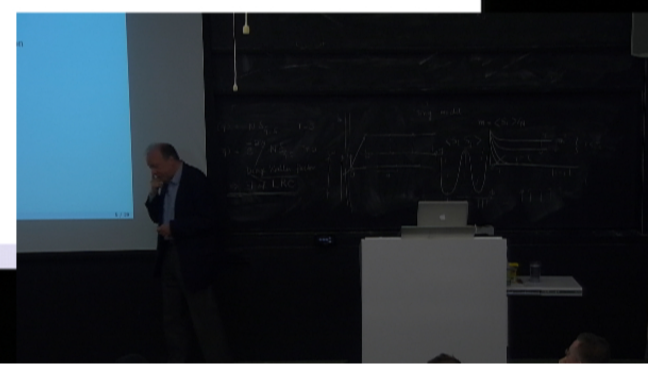
As a further illustration of the power of these ideas (elaborated in the *Domain theory* pioneered by Dana Scott), we shall apply them to something more startling.

Reflexivity and Self-Application

As a further illustration of the power of these ideas (elaborated in the *Domain theory* pioneered by Dana Scott), we shall apply them to something more startling.

Church's λ-calculus:

$$t \; ::= \; x \;\mid\; \underbrace{tu}_{\text{application}} \;\mid\; \underbrace{\lambda x.\, t}_{\text{abstraction}}$$



Reflexivity and Self-Application

As a further illustration of the power of these ideas (elaborated in the *Domain theory* pioneered by Dana Scott), we shall apply them to something more startling.

Church's λ-calculus:

$$t \ ::= \ x \ \mid \ \underbrace{tu}_{\text{application}} \ \mid \ \underbrace{\lambda x.\, t}_{\text{abstraction}}$$

Dynamics: β-*reduction*

$$(\lambda x.\, t)u \rightarrow t[u/x]$$

Reflexivity and Self-Application

As a further illustration of the power of these ideas (elaborated in the *Domain theory* pioneered by Dana Scott), we shall apply them to something more startling.

Church's λ-calculus:

$$t \quad ::= \quad x \mid \underbrace{tu}_{\text{application}} \mid \underbrace{\lambda x.\, t}_{\text{abstraction}}$$

Dynamics: β-*reduction*

$$(\lambda x.\, t)u \rightarrow t[u/x]$$

We can form *self-application* : $\omega \equiv \lambda x.\, xx$.

Reflexivity and Self-Application

As a further illustration of the power of these ideas (elaborated in the *Domain theory* pioneered by Dana Scott), we shall apply them to something more startling.

Church's λ-calculus:

$$t \ ::= \ x \mid \underbrace{tu}_{\text{application}} \mid \underbrace{\lambda x.\, t}_{\text{abstraction}}$$

Dynamics: β-*reduction*

$$(\lambda x.\, t)u \rightarrow t[u/x]$$

We can form *self-application* : $\omega \equiv \lambda x.\, xx$.
Hence $\Omega \equiv \omega\omega$, which *diverges*:

$$\Omega \rightarrow \Omega \rightarrow \cdots$$

Reflexivity and Self-Application

As a further illustration of the power of these ideas (elaborated in the *Domain theory* pioneered by Dana Scott), we shall apply them to something more startling.

Church's λ-calculus:

$$t \;::=\; x \mid \underbrace{tu}_{\text{application}} \mid \underbrace{\lambda x.\, t}_{\text{abstraction}}$$

Dynamics: β*-reduction*

$$(\lambda x.\, t)u \to t[u/x]$$

We can form *self-application* : $\omega \equiv \lambda x.\, xx$.
Hence $\Omega \equiv \omega\omega$, which *diverges*:

$$\Omega \to \Omega \to \cdots$$

Also, $\mathbf{Y} \equiv \lambda f.\, (\lambda x.\, f(xx))(\lambda x.\, f(xx))$ — recursion.

$$\mathbf{Y}t \to (\lambda x.\, t(xx))(\lambda x.\, t(xx)) \to t((\lambda x.\, t(xx))(\lambda x.\, t(xx))) = t(\mathbf{Y}t).$$

A model for the λ-calculus

The λ-calculus is not (just :) a logician's fantasy:

A model for the λ-calculus

The λ-calculus is not (just :) a logician's fantasy:

- Foundations of LISP, many current languages

A model for the λ-calculus

The λ-calculus is not (just :) a logician's fantasy:

- Foundations of LISP, many current languages
- Universal expressive power: the *Church-Turing thesis.*

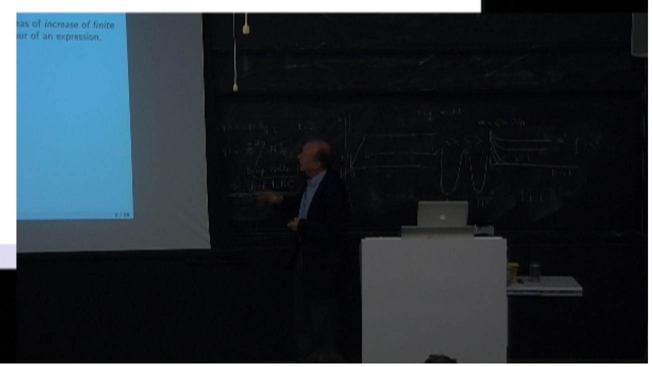
We can make a mathematical model based on the same ideas of *increase of finite pieces of information* we have about the functional behaviour of an expression.

A model for the λ-calculus

The λ-calculus is not (just :) a logician's fantasy:

- Foundations of LISP, many current languages
- Universal expressive power: the *Church-Turing thesis*.

We can make a mathematical model based on the same ideas of *increase of finite pieces of information* we have about the functional behaviour of an expression.



Samson Abramsky (Department of Computer Science, Information Flow: tracing a path through Logic, Comp

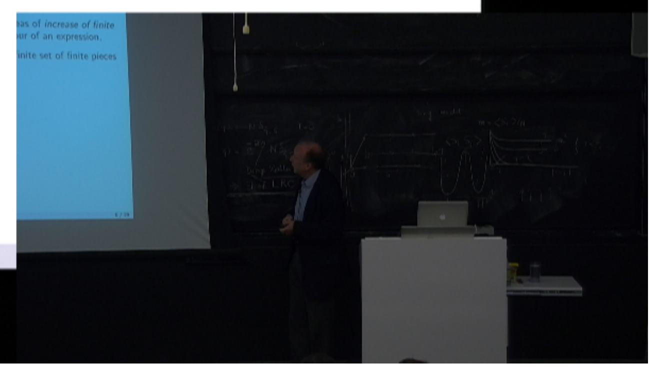
A model for the λ-calculus

The λ-calculus is not (just :) a logician's fantasy:

- Foundations of LISP, many current languages
- Universal expressive power: the *Church-Turing thesis*.

We can make a mathematical model based on the same ideas of *increase of finite pieces of information* we have about the functional behaviour of an expression.

A finite piece of information is a pair (S, a), where S is a finite set of finite pieces of information, and a is a finite piece of information.



A model for the λ-calculus

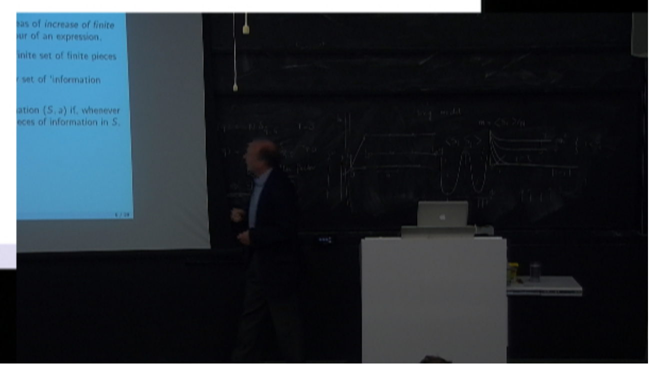
The λ-calculus is not (just :) a logician's fantasy:

- Foundations of LISP, many current languages
- Universal expressive power: the *Church-Turing thesis*.

We can make a mathematical model based on the same ideas of *increase of finite pieces of information* we have about the functional behaviour of an expression.

A finite piece of information is a pair (S, a), where S is a finite set of finite pieces of information, and a is a finite piece of information.
N.B. This is inductive! Need to start off with a non-empty set of 'information atoms' and build up from there.

The idea is that a function satisfies or contains the information (S, a) if, whenever it is applied to an argument which satisfies all the finite pieces of information in S, it produces a result containing the information a.



A model for the λ-calculus

The λ-calculus is not (just :) a logician's fantasy:

- Foundations of LISP, many current languages
- Universal expressive power: the *Church-Turing thesis*.

We can make a mathematical model based on the same ideas of *increase of finite pieces of information* we have about the functional behaviour of an expression.

A finite piece of information is a pair (S, a), where S is a finite set of finite pieces of information, and a is a finite piece of information.
N.B. This is inductive! Need to start off with a non-empty set of 'information atoms' and build up from there.

The idea is that a function satisfies or contains the information (S, a) if, whenever it is applied to an argument which satisfies all the finite pieces of information in S, it produces a result containing the information a.

The meaning of a term, $[\![t]\!]$, is the set of finite pieces of information it contains.

A model for the λ-calculus

The λ-calculus is not (just :) a logician's fantasy:

- Foundations of LISP, many current languages
- Universal expressive power: the *Church-Turing thesis*.

We can make a mathematical model based on the same ideas of *increase of finite pieces of information* we have about the functional behaviour of an expression.

A finite piece of information is a pair (S, a), where S is a finite set of finite pieces of information, and a is a finite piece of information.
N.B. This is inductive! Need to start off with a non-empty set of 'information atoms' and build up from there.

The idea is that a function satisfies or contains the information (S, a) if, whenever it is applied to an argument which satisfies all the finite pieces of information in S, it produces a result containing the information a.

The meaning of a term, $[\![t]\!]$, is the set of finite pieces of information it contains.

Definition of application:

$$[\![tu]\!] \;=\; \{a \mid \exists S.(S, a) \in [\![t]\!],\; S \subseteq [\![u]\!]\}$$

A model for the λ-calculus

The λ-calculus is not (just :) a logician's fantasy:

- Foundations of LISP, many current languages
- Universal expressive power: the *Church-Turing thesis*.

We can make a mathematical model based on the same ideas of *increase of finite pieces of information* we have about the functional behaviour of an expression.

A finite piece of information is a pair (S, a), where S is a finite set of finite pieces of information, and a is a finite piece of information.
N.B. This is inductive! Need to start off with a non-empty set of 'information atoms' and build up from there.

The idea is that a function satisfies or contains the information (S, a) if, whenever it is applied to an argument which satisfies all the finite pieces of information in S, it produces a result containing the information a.

The meaning of a term, $[\![t]\!]$, is the set of finite pieces of information it contains.

Definition of application:

$$[\![tu]\!] \;=\; \{a \mid \exists S.(S, a) \in [\![t]\!],\ S \subseteq [\![u]\!]\}$$

It works! Gives a consistent model of the calculus, self-application, fixpoints, etc.

Where is Physics in all this?

- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions ...

Where is Physics in all this?

- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions ...

Where is Physics in all this?

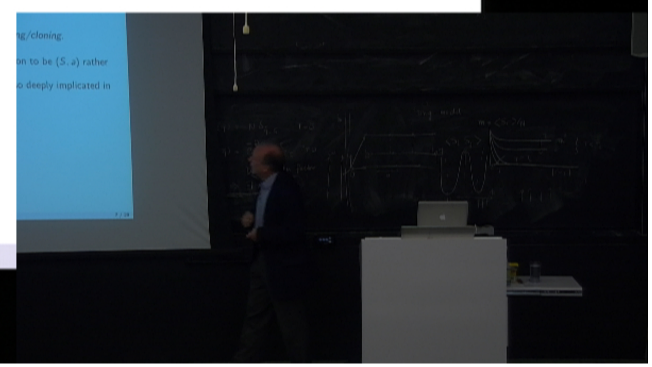
- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions . . .
- At which physical level do they arise?

Where is Physics in all this?

- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions ...
- At which physical level do they arise?
- There are subtle issues here. Recursion involves *copying/cloning*.

Where is Physics in all this?

- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions ...
- At which physical level do they arise?
- There are subtle issues here. Recursion involves *copying/cloning*.
- E.g. $\mathbf{Y} \equiv \lambda f.\,(\lambda x.\,f(xx))(\lambda x.\,f(xx))$.
  This is also why we defined a finite piece of information to be (S, a) rather than just (a, b).
  Diagonal arguments use *diagonals* $\Delta(x) = (x, x)$! Also deeply implicated in the blow-up of computational complexity.



Samson Abramsky (Department of Computer Science, Information Flow: tracing a path through Logic, Comp

Reflexivity and Self-Application

As a further illustration of the power of these ideas (elaborated in the *Domain theory* pioneered by Dana Scott), we shall apply them to something more startling.

Church's λ-calculus:

$$t \; ::= \; x \mid \underbrace{tu}_{\text{application}} \mid \underbrace{\lambda x.\, t}_{\text{abstraction}}$$

Dynamics: β-*reduction*

$$(\lambda x.\, t)u \rightarrow t[u/x]$$

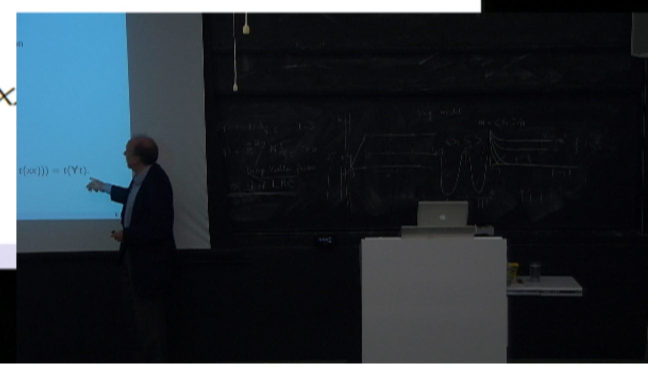
We can form *self-application* : $\omega \equiv \lambda x.\, xx$.
Hence $\Omega \equiv \omega\omega$, which *diverges*:

$$\Omega \rightarrow \Omega \rightarrow \cdots$$

Also, $\mathbf{Y} \equiv \lambda f.\, (\lambda x.\, f(xx))(\lambda x.\, f(xx))$ — recursion.

$$\mathbf{Y}t \rightarrow (\lambda x.\, t(xx))(\lambda x.\, t(xx)) \rightarrow t((\lambda x.\, t(xx))(\lambda x.\, t(x$$

It extracts a fixpoint for any term!



Samson Abramsky (Department of Computer Science, Information Flow: tracing a path through Logic, Comp

A model for the λ-calculus

The λ-calculus is not (just :) a logician's fantasy:

- Foundations of LISP, many current languages
- Universal expressive power: the *Church-Turing thesis*.

We can make a mathematical model based on the same ideas of *increase of finite pieces of information* we have about the functional behaviour of an expression.

A finite piece of information is a pair (S, a), where S is a finite set of finite pieces of information, and a is a finite piece of information.
N.B. This is inductive! Need to start off with a non-empty set of 'information atoms' and build up from there.

The idea is that a function satisfies or contains the information (S, a) if, whenever it is applied to an argument which satisfies all the finite pieces of information in S, it produces a result containing the information a.

The meaning of a term, $[\![t]\!]$, is the set of finite pieces of information it contains.

Definition of application:

$$[\![tu]\!] \;=\; \{a \mid \exists S.(S, a) \in [\![t]\!],\; S \subseteq [\![u]\!]\}$$

Where is Physics in all this?

- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions ...
- At which physical level do they arise?
- There are subtle issues here. Recursion involves *copying/cloning*.
- E.g. $\mathbf{Y} \equiv \lambda f.\,(\lambda x.\,f(xx))(\lambda x.\,f(xx))$.
  This is also why we defined a finite piece of information to be (S, a) rather than just (a, b).
  Diagonal arguments use *diagonals* $\Delta(x) = (x, x)$! Also deeply implicated in the blow-up of computational complexity.

Where is Physics in all this?

- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions ...
- At which physical level do they arise?
- There are subtle issues here. Recursion involves *copying/cloning*.
- E.g. $\mathbf{Y} \equiv \lambda f.\,(\lambda x.\,f(xx))(\lambda x.\,f(xx))$.
  This is also why we defined a finite piece of information to be (S, a) rather than just (a, b).
  Diagonal arguments use *diagonals* $\Delta(x) = (x, x)$! Also deeply implicated in the blow-up of computational complexity.
  So how does this emerge from the quantum level, where we have no-cloning?

Where is Physics in all this?

- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions ...
- At which physical level do they arise?
- There are subtle issues here. Recursion involves *copying/cloning*.
- E.g. $\mathbf{Y} \equiv \lambda f.\,(\lambda x.\,f(xx))(\lambda x.\,f(xx))$.
  This is also why we defined a finite piece of information to be (S, a) rather than just (a, b).
  Diagonal arguments use *diagonals* $\Delta(x) = (x, x)$! Also deeply implicated in the blow-up of computational complexity.
  So how does this emerge from the quantum level, where we have no-cloning?
- Otherwise put, how does (*logical*) non-linearity arise from linearity?

Where is Physics in all this?

- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions . . .
- At which physical level do they arise?
- There are subtle issues here. Recursion involves *copying/cloning*.
- E.g. $\mathbf{Y} \equiv \lambda f.\,(\lambda x.\, f(xx))(\lambda x.\, f(xx))$.
  This is also why we defined a finite piece of information to be (S, a) rather than just (a, b).
  Diagonal arguments use *diagonals* $\Delta(x) = (x, x)$! Also deeply implicated in the blow-up of computational complexity.
  So how does this emerge from the quantum level, where we have no-cloning?
- Otherwise put, how does (*logical*) non-linearity arise from linearity?

We shall now turn to *linear forms of information flow*.

Where is Physics in all this?

- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions ...
- At which physical level do they arise?
- There are subtle issues here. Recursion involves *copying/cloning*.
- E.g. $\mathbf{Y} \equiv \lambda f.\,(\lambda x.\,f(xx))(\lambda x.\,f(xx))$.
  This is also why we defined a finite piece of information to be (S, a) rather than just (a, b).
  Diagonal arguments use *diagonals* $\Delta(x) = (x, x)$! Also deeply implicated in the blow-up of computational complexity.
  So how does this emerge from the quantum level, where we have no-cloning?
- Otherwise put, how does (*logical*) non-linearity arise from linearity?

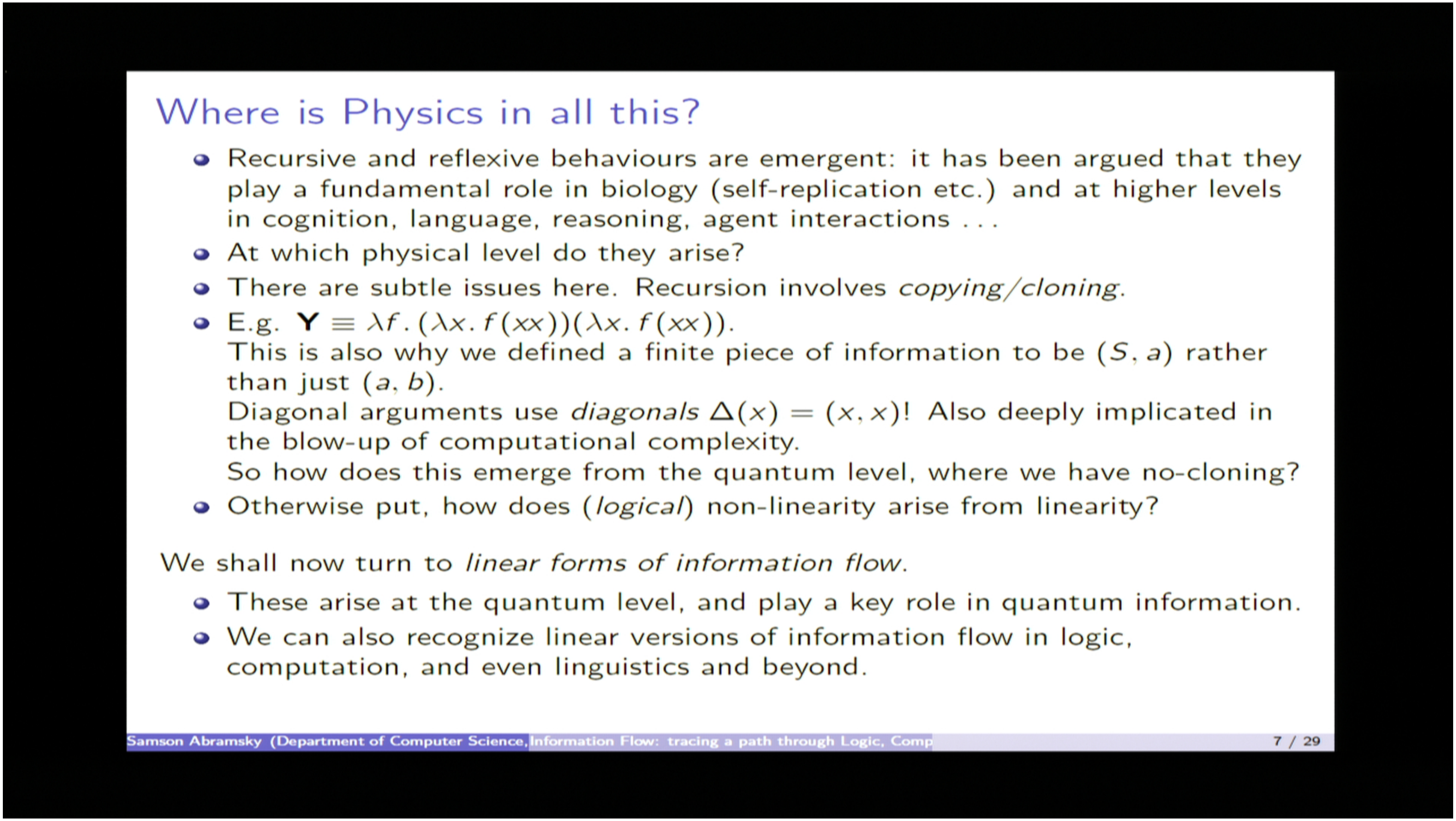
We shall now turn to *linear forms of information flow*.

Where is Physics in all this?

- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions ...
- At which physical level do they arise?
- There are subtle issues here. Recursion involves *copying/cloning*.
- E.g. $\mathbf{Y} \equiv \lambda f.\,(\lambda x.\,f(xx))(\lambda x.\,f(xx))$.
  This is also why we defined a finite piece of information to be (S, a) rather than just (a, b).
  Diagonal arguments use *diagonals* $\Delta(x) = (x, x)$! Also deeply implicated in the blow-up of computational complexity.
  So how does this emerge from the quantum level, where we have no-cloning?
- Otherwise put, how does (*logical*) non-linearity arise from linearity?

We shall now turn to *linear forms of information flow*.

- These arise at the quantum level, and play a key role in quantum information.


Where is Physics in all this?
Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions ...
At which physical level do they arise?
There are subtle issues here. Recursion involves copying/cloning.
E.g. $\mathbf{Y} \equiv \lambda f.(\lambda x. f(xx))(\lambda x. f(xx))$.
This is also why we defined a finite piece of information to be (S, a) rather than just (a, b).
Diagonal arguments use diagonals $\Delta(x) = (x, x)$! Also deeply implicated in the blow-up of computational complexity.
So how does this emerge from the quantum level, where we have no-cloning?
Otherwise put, how does (logical) non-linearity arise from linearity?
We shall now turn to linear forms of information flow.
These arise at the quantum level, and play a key role in quantum information.
We can also recognize linear versions of information flow in logic, computation, and even linguistics and beyond.
Samson Abramsky (Department of Computer Science, Information Flow: tracing a path through Logic, Comp
7 / 29

Where is Physics in all this?

- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions ...
- At which physical level do they arise?
- There are subtle issues here. Recursion involves *copying/cloning*.
- E.g. $\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$.
  This is also why we defined a finite piece of information to be (S, a) rather than just (a, b).
  Diagonal arguments use *diagonals* $\Delta(x) = (x, x)$! Also deeply implicated in the blow-up of computational complexity.
  So how does this emerge from the quantum level, where we have no-cloning?
- Otherwise put, how does (*logical*) non-linearity arise from linearity?

We shall now turn to *linear forms of information flow*.

- These arise at the quantum level, and play a key role in quantum information.
- We can also recognize linear versions of information flow in logic, computation, and even linguistics and beyond.
- Some steps towards a unified theory.

7 / 29

Where is Physics in all this?

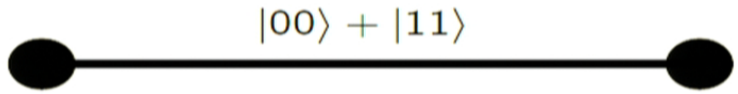
- Recursive and reflexive behaviours are emergent: it has been argued that they play a fundamental role in biology (self-replication etc.) and at higher levels in cognition, language, reasoning, agent interactions ...
- At which physical level do they arise?
- There are subtle issues here. Recursion involves *copying/cloning*.
- E.g. $\mathbf{Y} \equiv \lambda f.\,(\lambda x.\,f(xx))(\lambda x.\,f(xx))$.
  This is also why we defined a finite piece of information to be (S, a) rather than just (a, b).
  Diagonal arguments use *diagonals* $\Delta(x) = (x, x)$! Also deeply implicated in the blow-up of computational complexity.
  So how does this emerge from the quantum level, where we have no-cloning?
- Otherwise put, how does (*logical*) non-linearity arise from linearity?

We shall now turn to *linear forms of information flow*.

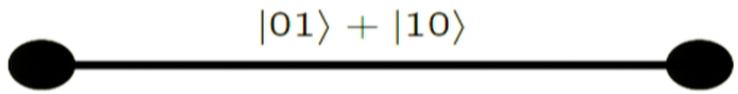
- These arise at the quantum level, and play a key role in quantum information.
- We can also recognize linear versions of information flow in logic, computation, and even linguistics and beyond.
- Some steps towards a unified theory.

Quantum Entanglement

Bell state:



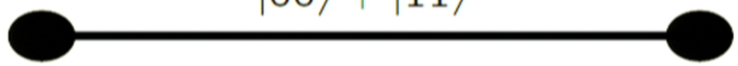
EPR state:



Quantum Entanglement

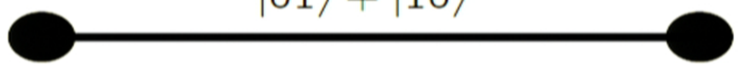
Bell state:

$$|00\rangle + |11\rangle$$



EPR state:

$$|01\rangle + |10\rangle$$



Compound systems are represented by *tensor product*: $\mathcal{H}_1 \otimes \mathcal{H}_2$. Typical element:

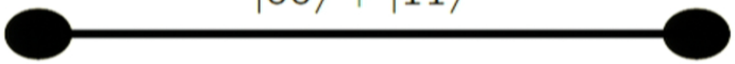
$$\sum_i \lambda_i \cdot \phi_i \otimes \psi_i$$

Superposition encodes *correlation*.

Quantum Entanglement

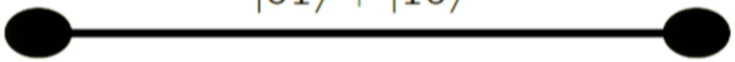
Bell state:

$|00\rangle + |11\rangle$



EPR state:

$|01\rangle + |10\rangle$



Compound systems are represented by *tensor product*: $\mathcal{H}_1 \otimes \mathcal{H}_2$. Typical element:

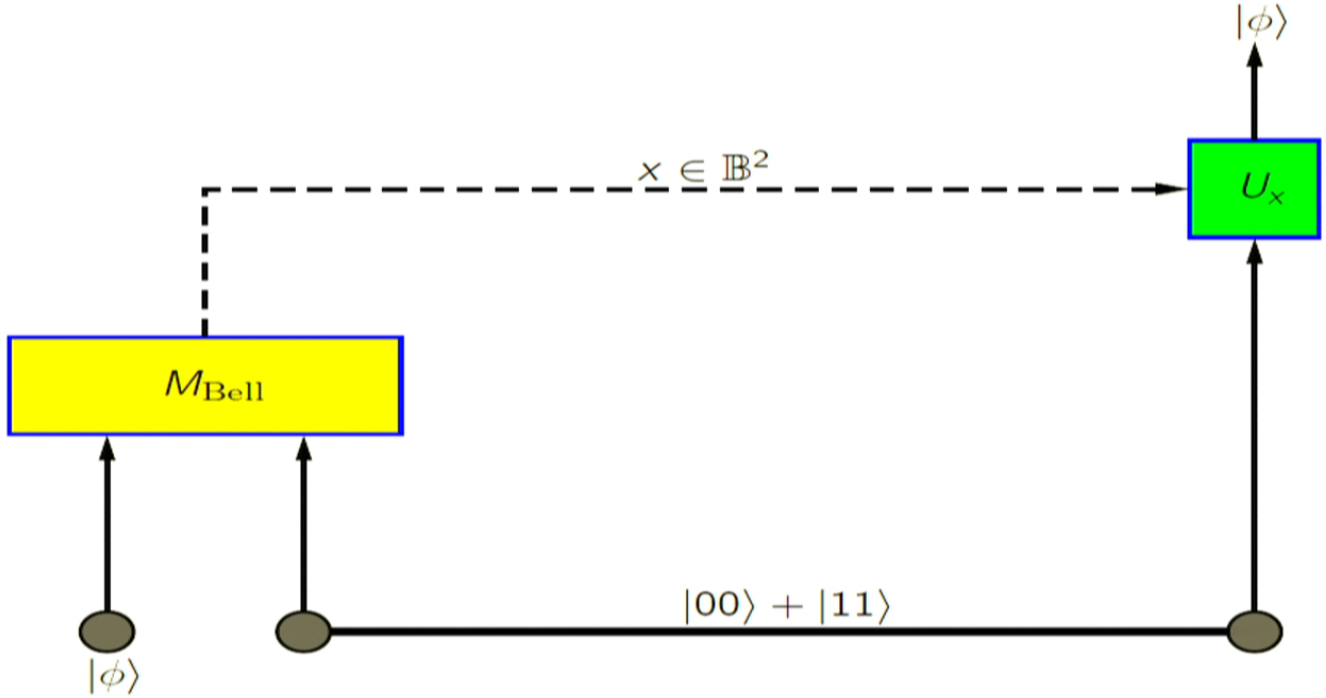
$$\sum_i \lambda_i \cdot \phi_i \otimes \psi_i$$

Superposition encodes *correlation*.

Einstein's 'spooky action at a distance'. Even if the particles are spatially separated, measuring one has an effect on the state of the other.

Bell's theorem: QM is *essentially non-local*.

From 'paradox' to 'feature': Teleportation



Entangled states as linear maps

$\mathcal{H}_1 \otimes \mathcal{H}_2$ is spanned by

$$\begin{matrix} |11\rangle & \cdots & |1m\rangle \\ \vdots & \ddots & \vdots \\ |n1\rangle & \cdots & |nm\rangle \end{matrix}$$

hence

$$\sum_{i,j} \alpha_{ij} |ij\rangle \quad \longleftrightarrow \quad \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1m} \\ \vdots & \ddots & \vdots \\ \alpha_{n1} & \cdots & \alpha_{nm} \end{pmatrix} \quad \longleftrightarrow \quad |i\rangle \mapsto \sum_{j} \alpha_{ij} |j\rangle$$

Pairs $|\psi_1, \psi_2\rangle$ are a special case — $|ij\rangle$ in a well-chosen basis.

This is **Map-State Duality**.

Entangled states as linear maps

$\mathcal{H}_1 \otimes \mathcal{H}_2$ is spanned by

$$\begin{matrix} |11\rangle & \cdots & |1m\rangle \\ \vdots & \ddots & \vdots \\ |n1\rangle & \cdots & |nm\rangle \end{matrix}$$

hence

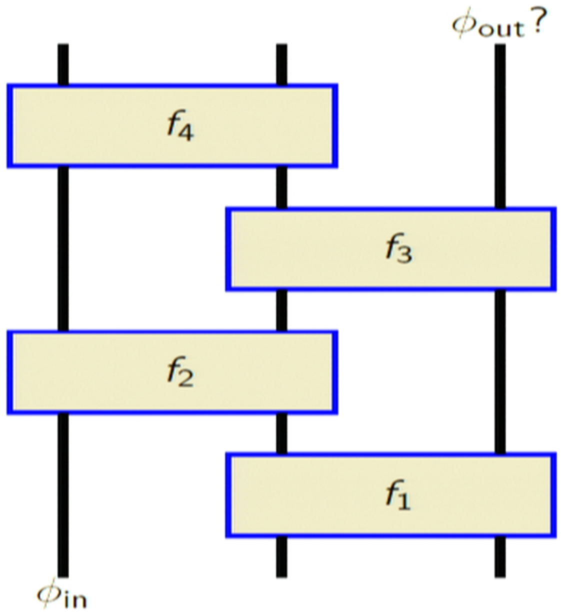
$$\sum_{i,j} \alpha_{ij} |ij\rangle \quad \longleftrightarrow \quad \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1m} \\ \vdots & \ddots & \vdots \\ \alpha_{n1} & \cdots & \alpha_{nm} \end{pmatrix} \quad \longleftrightarrow \quad |i\rangle \mapsto \sum_{j} \alpha_{ij} |j\rangle$$

Pairs $|\psi_1, \psi_2\rangle$ are a special case — $|ij\rangle$ in a well-chosen basis.

This is **Map-State Duality**.

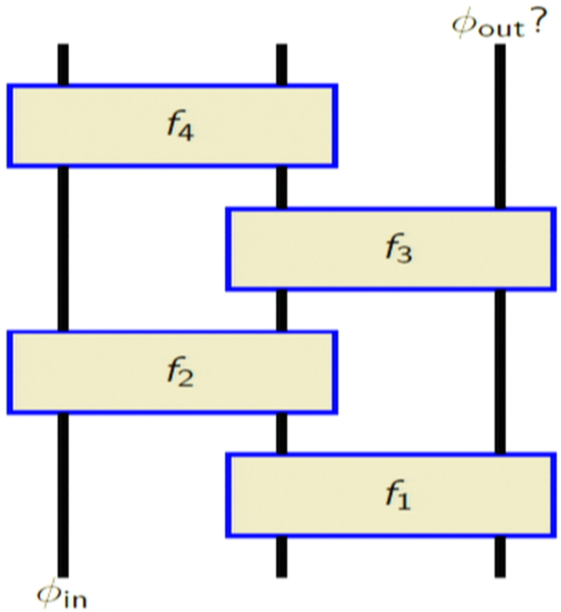
Notation. Given a linear map $f : \mathcal{H} \to \mathcal{H}$, we write $\mathbf{P}_f$ for the projector on $\mathcal{H} \otimes \mathcal{H}$ determined by the vector corresponding to f under Map-State duality.

What is the output?



$$(\mathbf{P}_{f_4} \otimes 1) \circ (1 \otimes \mathbf{P}_{f_3}) \circ (\mathbf{P}_{f_2} \otimes 1) \circ (1 \otimes \mathbf{P}_{f_1}) : \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \mathcal{H}_3 \longrightarrow \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \mathcal{H}_3$$

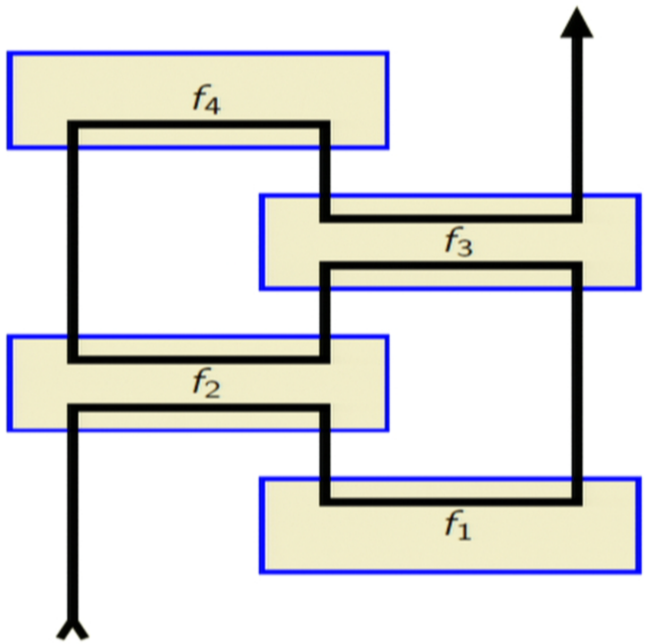
What is the output?



$$(\mathbf{P}_{f_4} \otimes 1) \circ (1 \otimes \mathbf{P}_{f_3}) \circ (\mathbf{P}_{f_2} \otimes 1) \circ (1 \otimes \mathbf{P}_{f_1}) : \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \mathcal{H}_3 \longrightarrow \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \mathcal{H}_3$$

$$\phi_{out} = f_3 \circ f_4 \circ f_2^\dagger \circ f_3^\dagger \circ f_1 \circ f_2(\phi_{in})$$

Follow the line!



$$f_3 \circ f_4 \circ f_2^{\dagger} \circ f_3^{\dagger} \circ f_1 \circ f_2$$

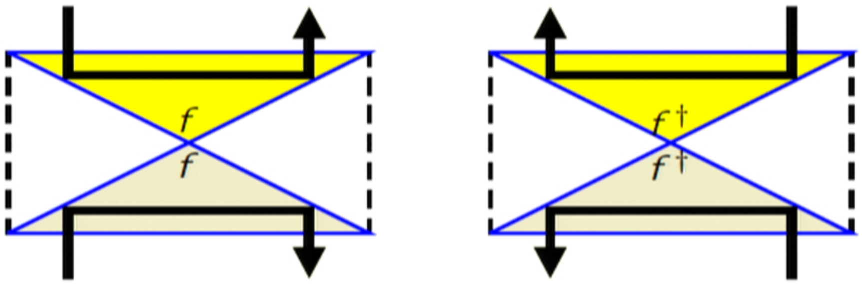
Bipartite Projectors

Information flow in entangled states can be captured mathematically by the isomorphism

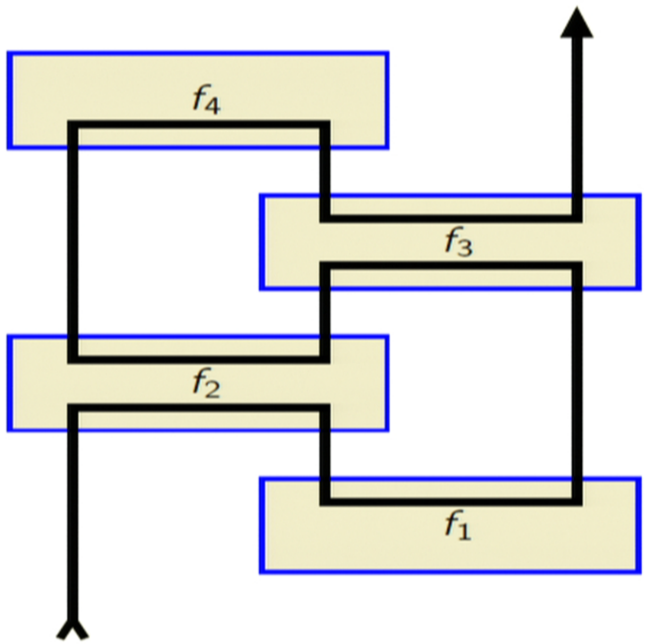
$$\mathbf{Hom}(A, B) \cong A^* \otimes B.$$

This leads to a *decomposition* of bipartite projectors into "names" (preparations) and "conames" (measurements).

In graphical notation:



Follow the line!



$$f_3 \circ f_4 \circ f_2^\dagger \circ f_3^\dagger \circ f_1 \circ f_2$$

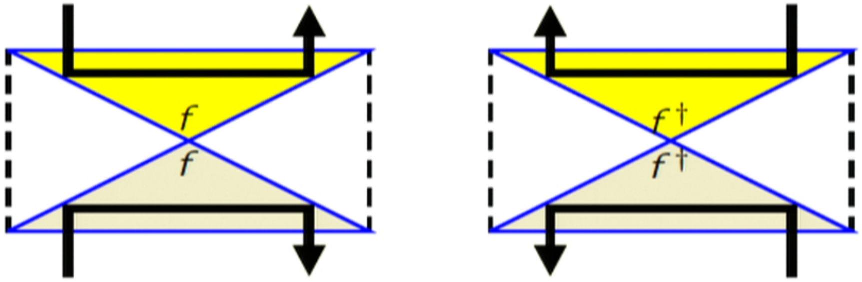
Bipartite Projectors

Information flow in entangled states can be captured mathematically by the isomorphism

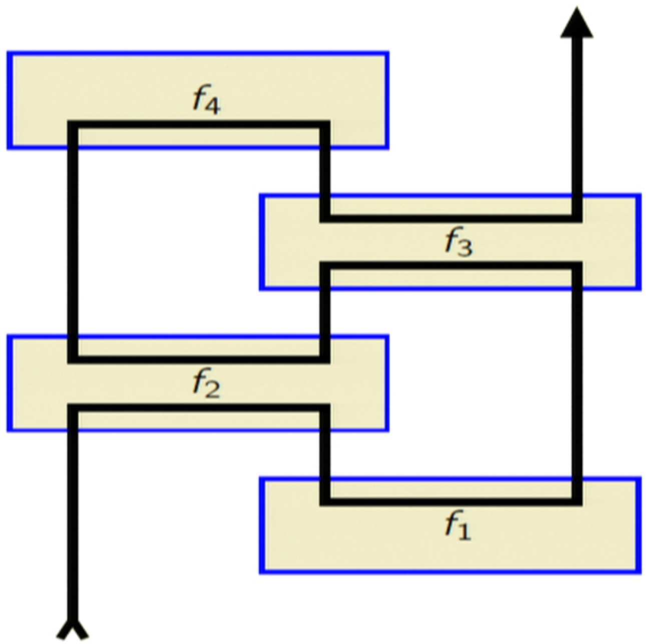
$$\mathbf{Hom}(A, B) \cong A^* \otimes B.$$

This leads to a *decomposition* of bipartite projectors into "names" (preparations) and "conames" (measurements).

In graphical notation:



Follow the line!



$$f_3 \circ f_4 \circ f_2^\dagger \circ f_3^\dagger \circ f_1 \circ f_2$$

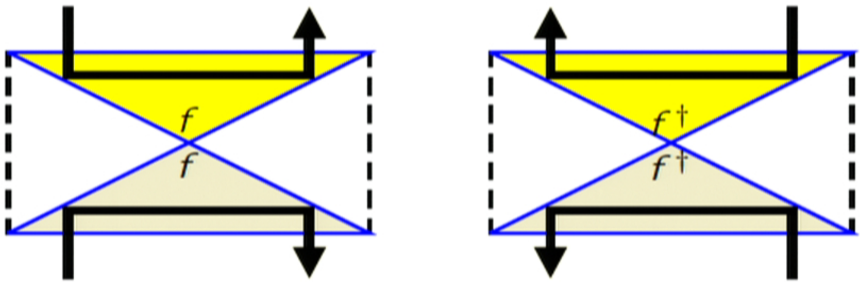
Bipartite Projectors

Information flow in entangled states can be captured mathematically by the isomorphism

$$\mathbf{Hom}(A, B) \cong A^* \otimes B.$$

This leads to a *decomposition* of bipartite projectors into "names" (preparations) and "conames" (measurements).

In graphical notation:



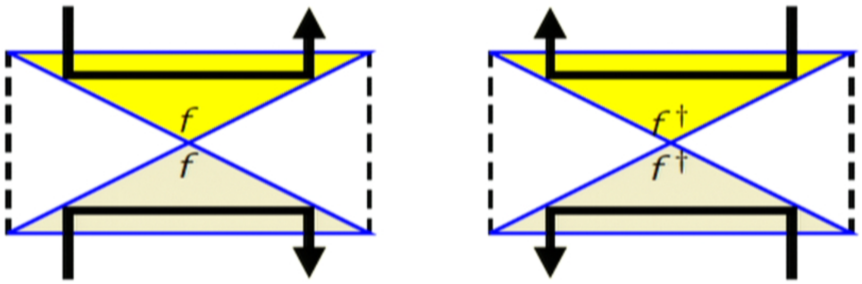
Bipartite Projectors

Information flow in entangled states can be captured mathematically by the isomorphism

$$\mathbf{Hom}(A, B) \cong A^* \otimes B.$$

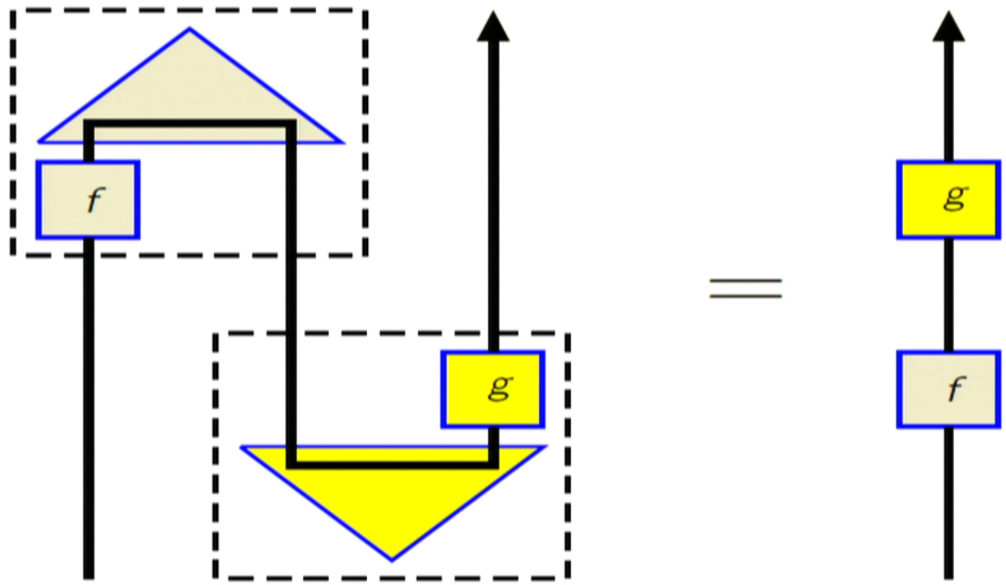
This leads to a *decomposition* of bipartite projectors into "names" (preparations) and "conames" (measurements).

In graphical notation:



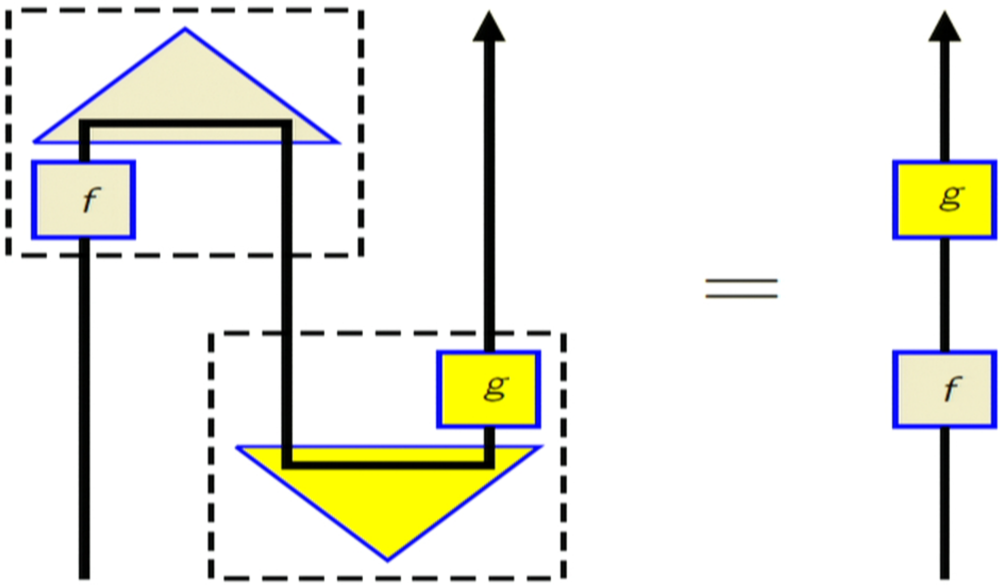
Compositionality

The key algebraic fact from which teleportation (and many other protocols) can be derived.

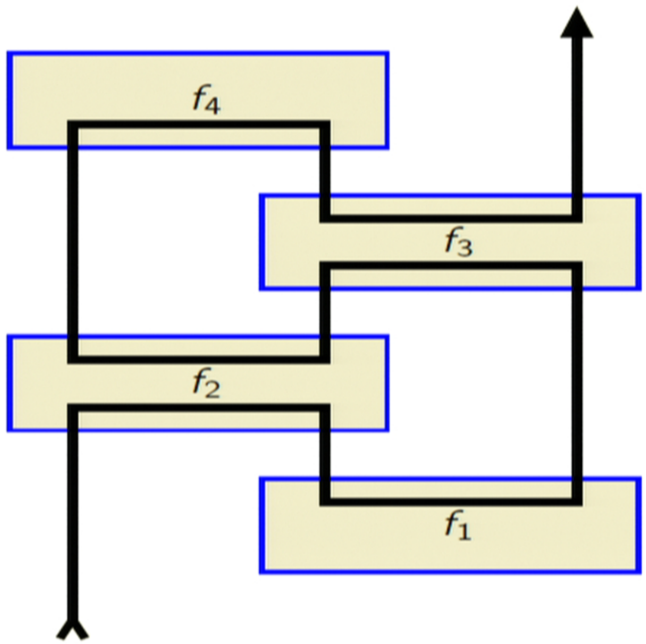


Compositionality

The key algebraic fact from which teleportation (and many other protocols) can be derived.



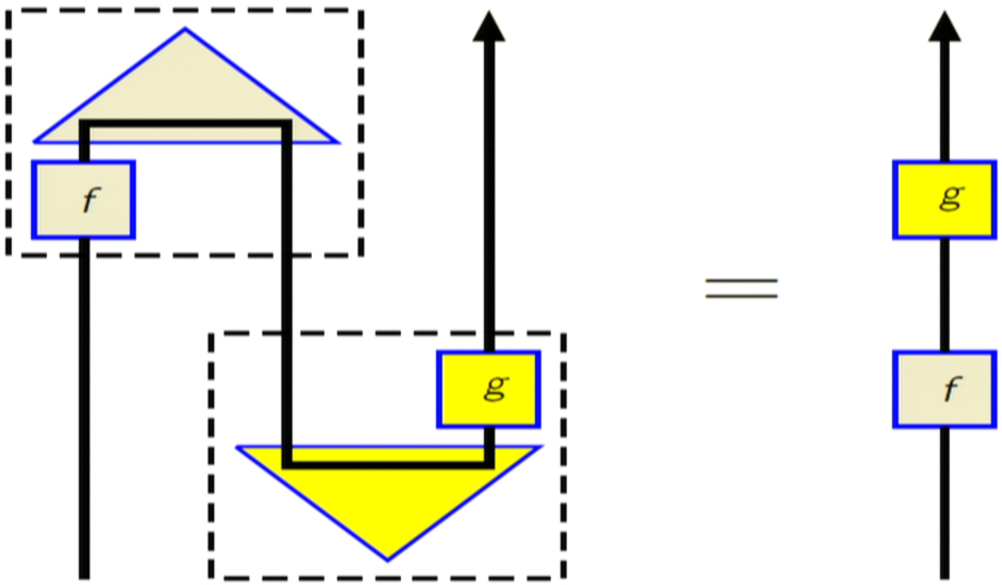
Follow the line!



$$f_3 \circ f_4 \circ f_2^\dagger \circ f_3^\dagger \circ f_1 \circ f_2$$

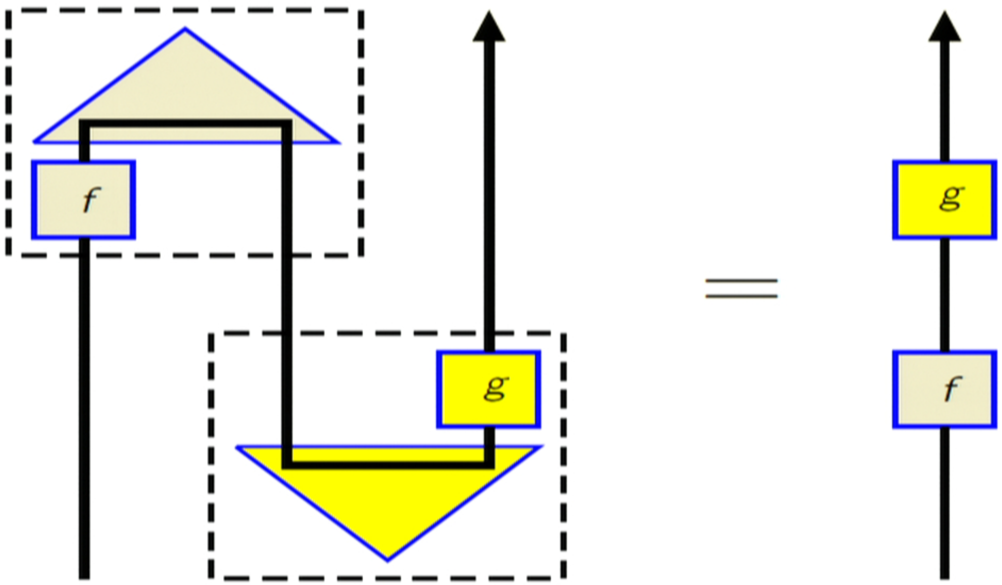
Compositionality

The key algebraic fact from which teleportation (and many other protocols) can be derived.

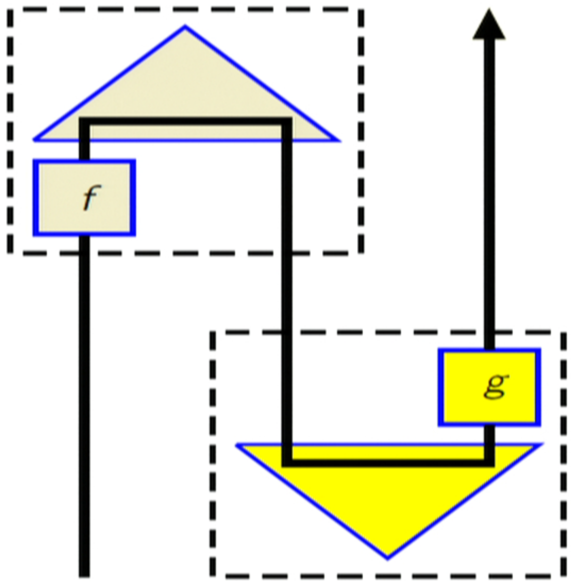


Compositionality

The key algebraic fact from which teleportation (and many other protocols) can be derived.



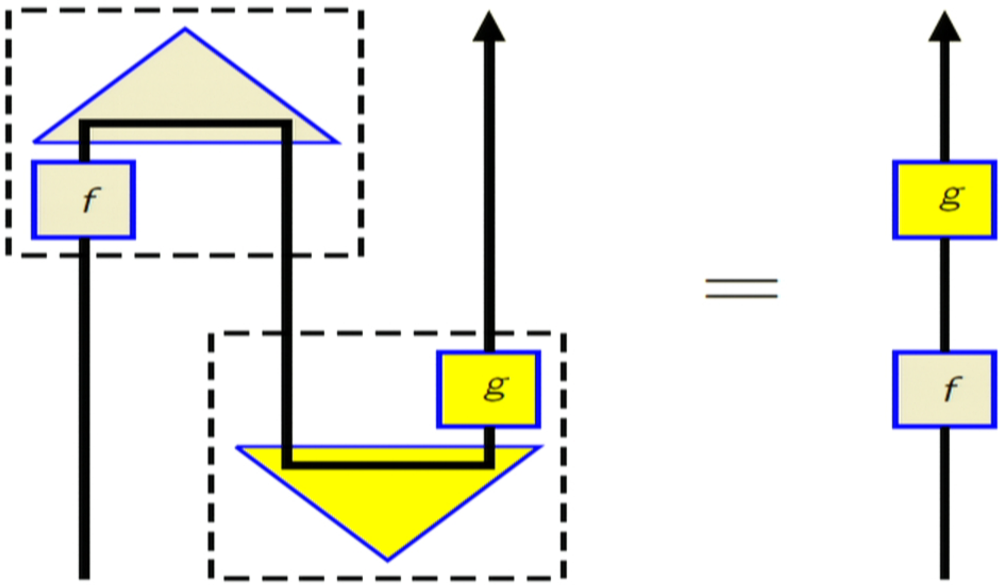
Compositionality ctd



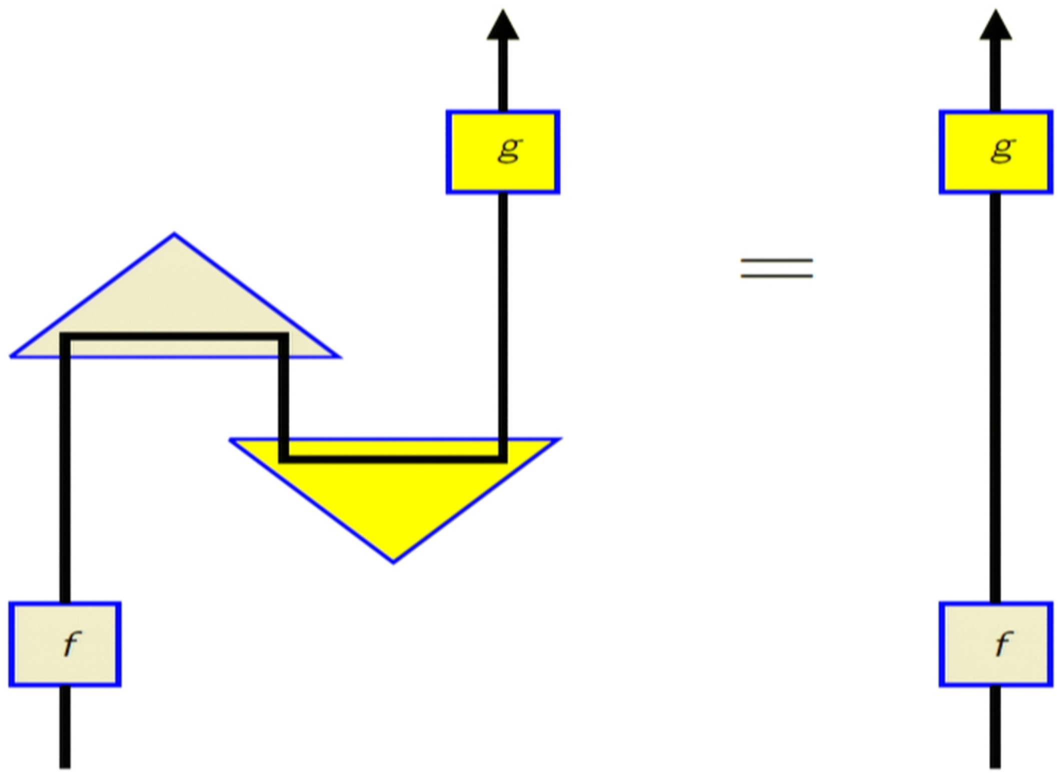
=

Compositionality

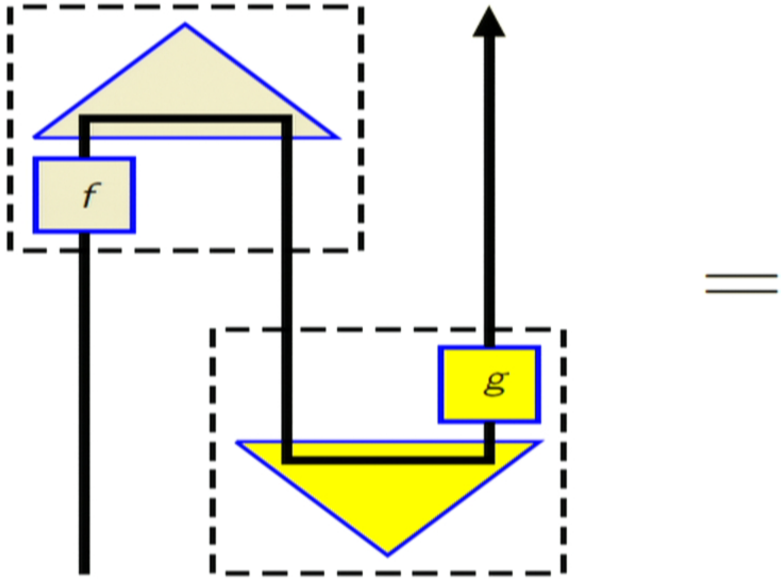
The key algebraic fact from which teleportation (and many other protocols) can be derived.



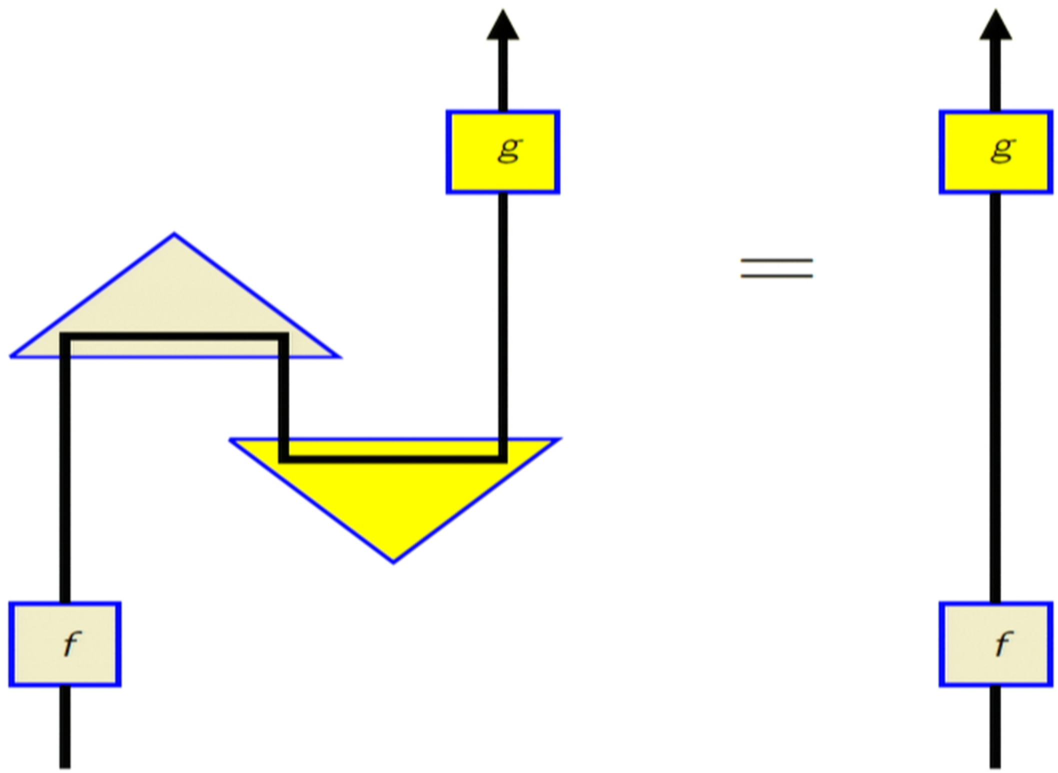
Compositionality ctd



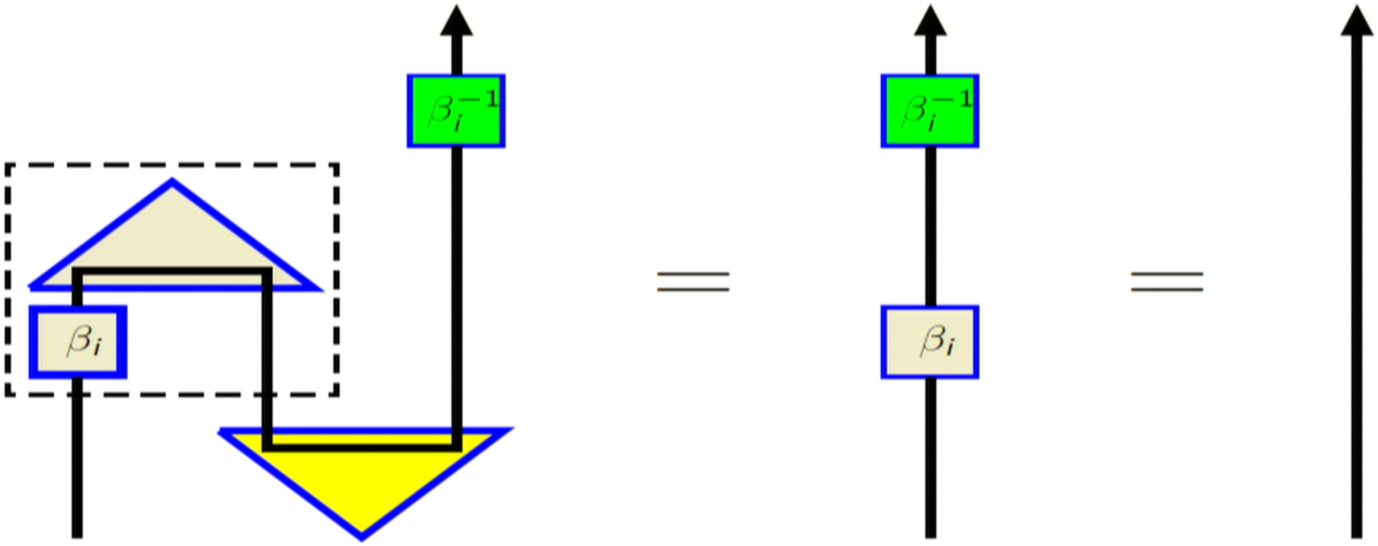
Compositionality ctd



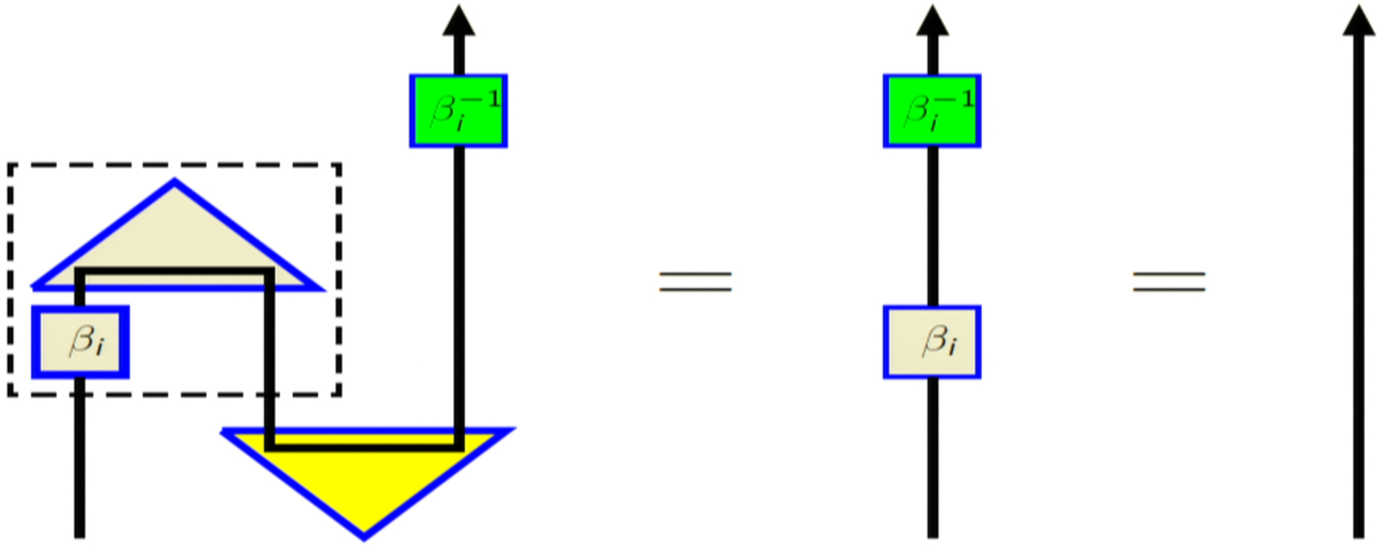
Compositionality ctd



Teleportation diagrammatically



Teleportation diagrammatically



Categorical Quantum Mechanics

Work of many people, both in the Quantum Group at Oxford CS Dept and elsewhere.

Categorical Quantum Mechanics

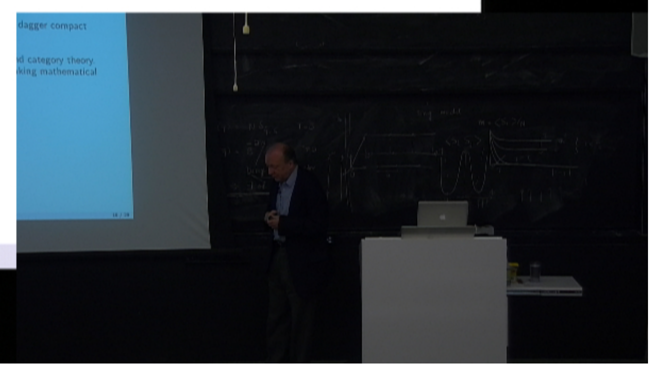
Work of many people, both in the Quantum Group at Oxford CS Dept and elsewhere.

- Underlying mathematics: monoidal dagger categories, dagger compact structure, Frobenius algebras, bialgebras . . .

Categorical Quantum Mechanics

Work of many people, both in the Quantum Group at Oxford CS Dept and elsewhere.

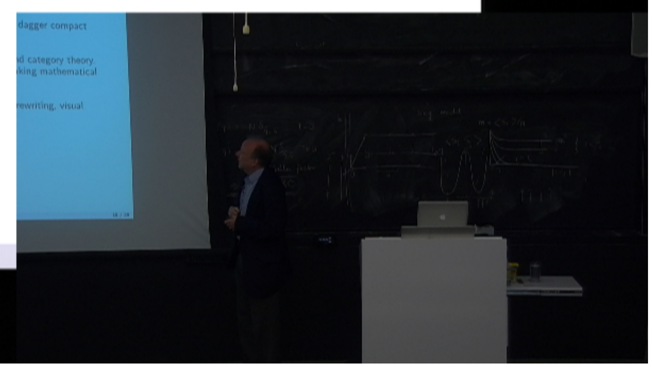
- Underlying mathematics: monoidal dagger categories, dagger compact structure, Frobenius algebras, bialgebras . . .
- Diagrammatic representation. Connections to logic and category theory. Underpinning mathematics, effective visualization, making mathematical structures accessible.



Categorical Quantum Mechanics

Work of many people, both in the Quantum Group at Oxford CS Dept and elsewhere.

- Underlying mathematics: monoidal dagger categories, dagger compact structure, Frobenius algebras, bialgebras . . .
- Diagrammatic representation. Connections to logic and category theory. Underpinning mathematics, effective visualization, making mathematical structures accessible.
- Software tool support: Quantomatic. Tactics, graph rewriting, visual interface.



Samson Abramsky (Department of Computer Science, Information Flow: tracing a path through Logic, Comp

Categorical Quantum Mechanics

Work of many people, both in the Quantum Group at Oxford CS Dept and elsewhere.

- Underlying mathematics: monoidal dagger categories, dagger compact structure, Frobenius algebras, bialgebras . . .
- Diagrammatic representation. Connections to logic and category theory. Underpinning mathematics, effective visualization, making mathematical structures accessible.
- Software tool support: Quantomatic. Tactics, graph rewriting, visual interface.
- Applications. Formalization of quantum protocols, QKD, measurement-based quantum computation, etc. Analysis of determinism in MBQC, compositional structure of multipartite entanglement. Foundational topics: e.g. analysis of non-locality.

String Diagrams Are Everywhere

Categorical Quantum Mechanics

Work of many people, both in the Quantum Group at Oxford CS Dept and elsewhere.

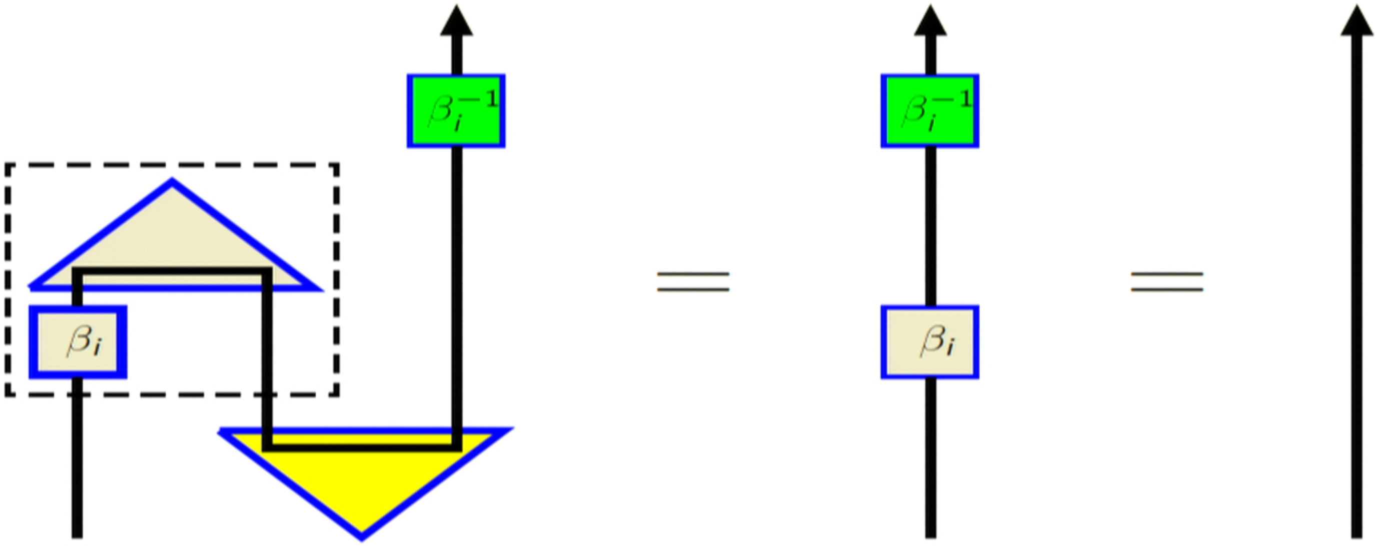
- Underlying mathematics: monoidal dagger categories, dagger compact structure, Frobenius algebras, bialgebras . . .
- Diagrammatic representation. Connections to logic and category theory. Underpinning mathematics, effective visualization, making mathematical structures accessible.
- Software tool support: Quantomatic. Tactics, graph rewriting, visual interface.
- Applications. Formalization of quantum protocols, QKD, measurement-based quantum computation, etc. Analysis of determinism in MBQC, compositional structure of multipartite entanglement. Foundational topics: e.g. analysis of non-locality.

Categorical Quantum Mechanics

Work of many people, both in the Quantum Group at Oxford CS Dept and elsewhere.

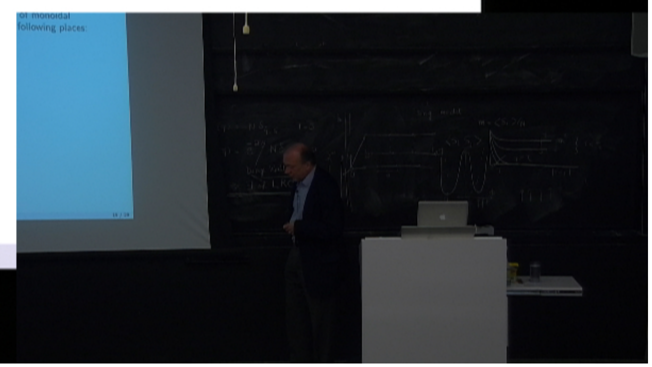
- Underlying mathematics: monoidal dagger categories, dagger compact structure, Frobenius algebras, bialgebras . . .
- Diagrammatic representation. Connections to logic and category theory. Underpinning mathematics, effective visualization, making mathematical structures accessible.
- Software tool support: Quantomatic. Tactics, graph rewriting, visual interface.
- Applications. Formalization of quantum protocols, QKD, measurement-based quantum computation, etc. Analysis of determinism in MBQC, compositional structure of multipartite entanglement. Foundational topics: e.g. analysis of non-locality.

Teleportation diagrammatically



String Diagrams Are Everywhere

This graphical formalism, with the underlying mathematics of monoidal categories, compact closure etc., turns up in (at least) the following places:



String Diagrams Are Everywhere

This graphical formalism, with the underlying mathematics of monoidal categories, compact closure etc., turns up in (at least) the following places:

- Quantum mechanics, quantum information.

String Diagrams Are Everywhere

This graphical formalism, with the underlying mathematics of monoidal categories, compact closure etc., turns up in (at least) the following places:

- Quantum mechanics, quantum information.
- Logic: (linear version of) cut-elimination

String Diagrams Are Everywhere

This graphical formalism, with the underlying mathematics of monoidal categories, compact closure etc., turns up in (at least) the following places:

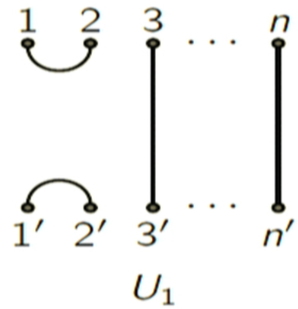
- Quantum mechanics, quantum information.
- Logic: (linear version of) cut-elimination
- Computation: (linear version of) λ-calculus, feedback, processes.
- Linguistics: Lambek pregroup grammars, lifting vector space models of word meaning
- Topology, knot theory: Temperley-Lieb algebra, braided, pivotal and ribbon categories.

We will trace a path through some of these ...

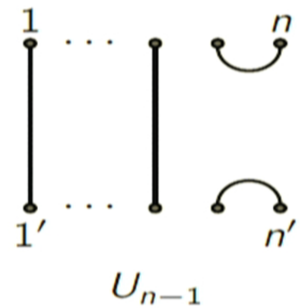
The Temperley-Lieb Algebra

The Temperley-Lieb Algebra

Generators:



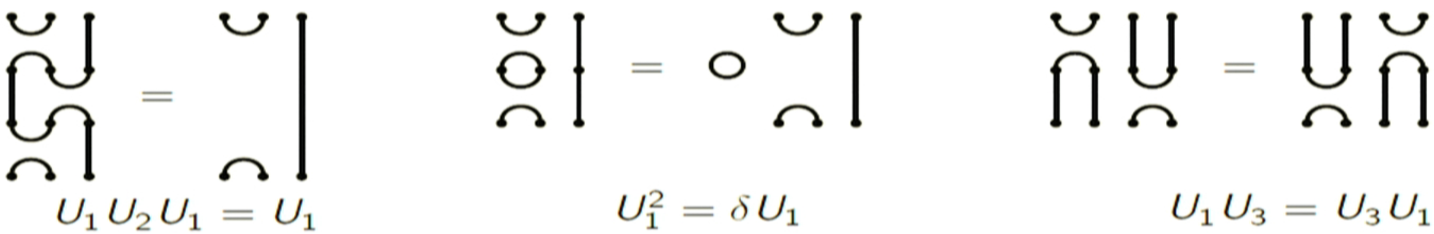
…



The Temperley-Lieb Algebra

Generators:

Relations:



Structure of Temperley-Lieb category

General form of composition:

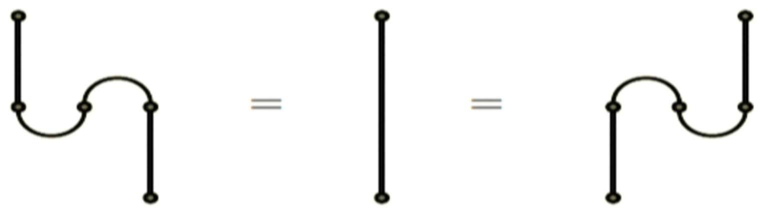
Structure of Temperley-Lieb category

General form of composition:

Structure of Temperley-Lieb category

General form of composition:

Compact closure/rigidity:

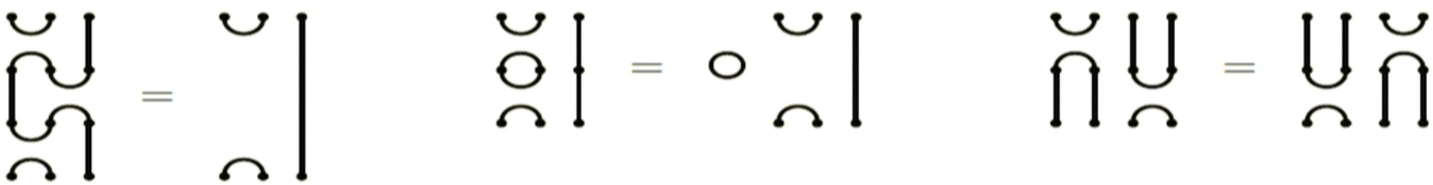


The Temperley-Lieb Algebra

Generators:

U_1 U_{n-1}

Relations:



$U_1 U_2 U_1 = U_1$ $U_1^2 = \delta U_1$ $U_1 U_3 = U_3 U_1$

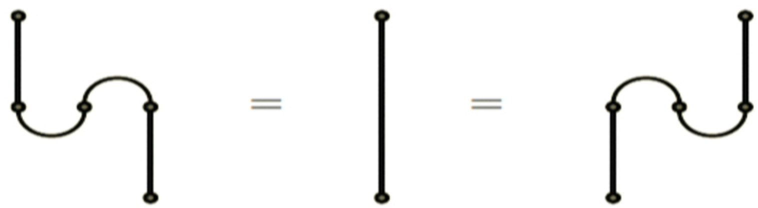
Structure of Temperley-Lieb category

General form of composition:

Structure of Temperley-Lieb category

General form of composition:

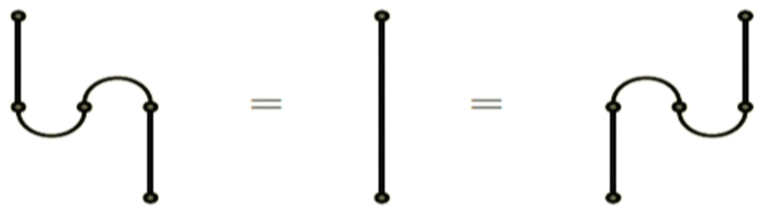
Compact closure/rigidity:



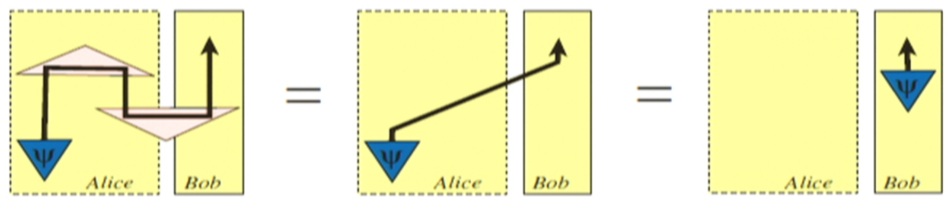
Structure of Temperley-Lieb category

General form of composition:

Compact closure/rigidity:



The same structure which accounts for teleportation:



Temperley-Lieb: expressiveness of the generators

All planar diagrams can be expressed as products of generators.

E.g. the 'left wave' can be expressed as the product $U_2 U_1$:

Temperley-Lieb: expressiveness of the generators

All planar diagrams can be expressed as products of generators.

E.g. the 'left wave' can be expressed as the product $U_2 U_1$:

Diagrammatic trace:

The Ear is a Circle

Trace of Identity is the Dimension

The Connection to Knots

How does this connect to knots? A key conceptual insight is due to Kauffman, who saw how to recast the Jones polynomial in elementary combinatorial form in terms of his *bracket polynomial*.

The Connection to Knots

How does this connect to knots? A key conceptual insight is due to Kauffman, who saw how to recast the Jones polynomial in elementary combinatorial form in terms of his *bracket polynomial*.

The basic idea of the bracket polynomial is expressed by the following equation:

The Connection to Knots

How does this connect to knots? A key conceptual insight is due to Kauffman, who saw how to recast the Jones polynomial in elementary combinatorial form in terms of his *bracket polynomial*.

The basic idea of the bracket polynomial is expressed by the following equation:

Each over-crossing in a knot or link is evaluated to a weighted sum of the two possible planar smoothings in the Temperley-Lieb algebra. With suitable choices for the coefficients A and B (as Laurent polynomials), this is invariant under the second and third Reidemeister moves. With an ingenious choice of normalizing factor, it becomes invariant under the first Reidemeister move — and yields the Jones polynomial!

Computation: back to the λ-calculus

We shall consider the *bracketing combinator*

$$\mathbf{B} \equiv \lambda x.\lambda y.\lambda z.\, x(yz) : (B \to C) \to (A \to B) \to (A \to C).$$

This is characterized by the equation $\mathbf{B}abc = a(bc)$.

Computation: back to the λ-calculus

We shall consider the *bracketing combinator*

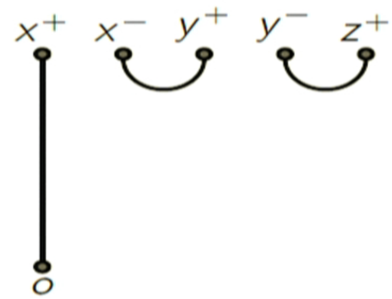
$$\mathbf{B} \equiv \lambda x.\lambda y.\lambda z.\, x(yz) : (B \to C) \to (A \to B) \to (A \to C).$$

This is characterized by the equation $\mathbf{B}abc = a(bc)$.

We take $A = B = C = \mathbf{1}$ in **TL**. The interpretation of the open term

$$x : B \to C, y : A \to B, z : A \vdash x(yz) : C$$

is as follows:



Computation: back to the λ-calculus

We shall consider the *bracketing combinator*

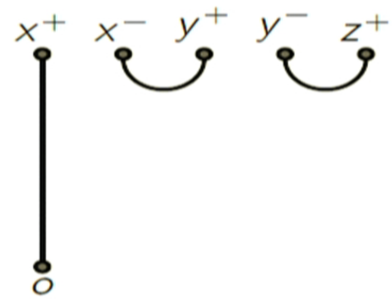
$$\mathbf{B} \equiv \lambda x.\lambda y.\lambda z.\, x(yz) : (B \to C) \to (A \to B) \to (A \to C).$$

This is characterized by the equation $\mathbf{B}abc = a(bc)$.

We take $A = B = C = \mathbf{1}$ in **TL**. The interpretation of the open term

$$x : B \to C, y : A \to B, z : A \vdash x(yz) : C$$

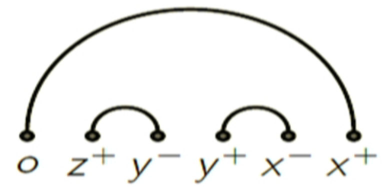
is as follows:



Diagrammatic Simplification as β-Reduction

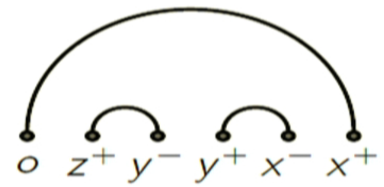
Diagrammatic Simplification as β-Reduction

When we abstract the variables, we obtain the following caps-only diagram:



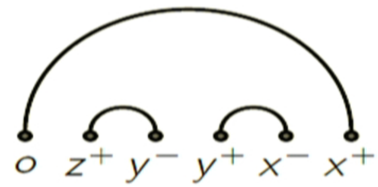
Diagrammatic Simplification as β-Reduction

When we abstract the variables, we obtain the following caps-only diagram:

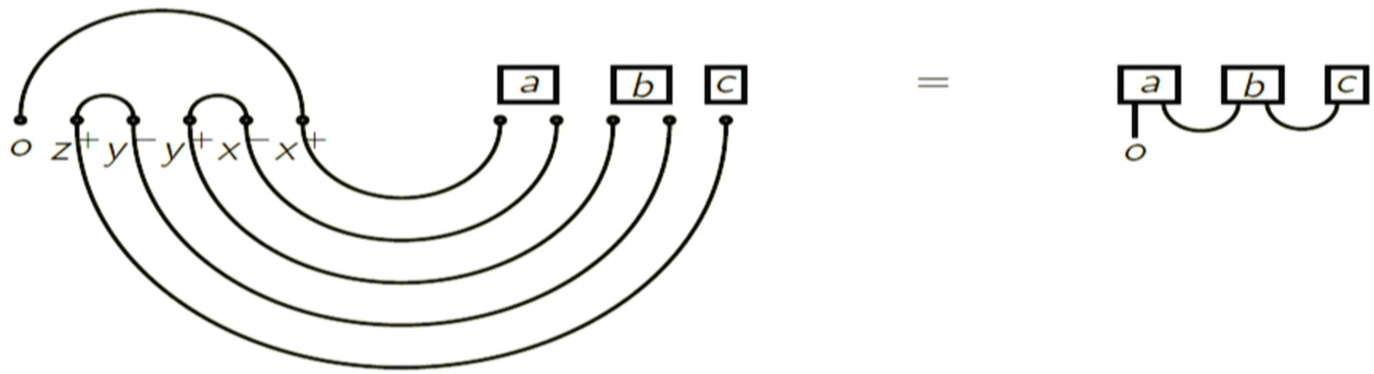


Diagrammatic Simplification as β-Reduction

When we abstract the variables, we obtain the following caps-only diagram:



Now we consider an application $\mathbf{B}abc$ (where application is represented by cups):



A Non-Planar Example

A Non-Planar Example

We shall consider the *commuting combinator*

$$\mathbf{C} \equiv \lambda x.\lambda y.\lambda z.\, xzy : (A \to B \to C) \to B \to A \to C.$$

A Non-Planar Example

We shall consider the *commuting combinator*

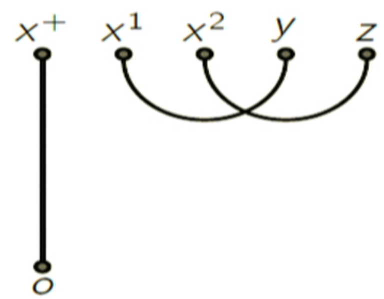
$$\mathbf{C} \equiv \lambda x.\lambda y.\lambda z.\, xzy : (A \to B \to C) \to B \to A \to C.$$

This is characterized by the equation $\mathbf{C}abc = acb$.

The interpretation of the open term

$$x : A \to B \to C, y : B, z : A \vdash xzy : C$$

is as follows:



A Non-Planar Example

We shall consider the *commuting combinator*

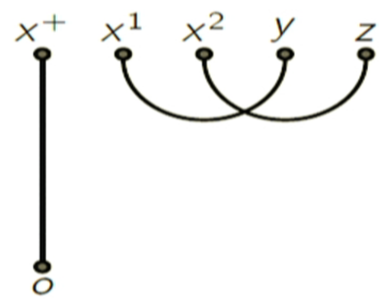
$$\mathbf{C} \equiv \lambda x.\lambda y.\lambda z.\, xzy : (A \to B \to C) \to B \to A \to C.$$

This is characterized by the equation $\mathbf{C}abc = acb$.

The interpretation of the open term

$$x : A \to B \to C, y : B, z : A \vdash xzy : C$$

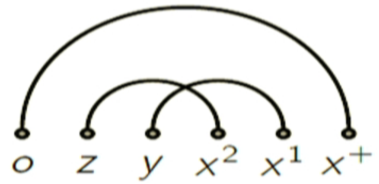
is as follows:



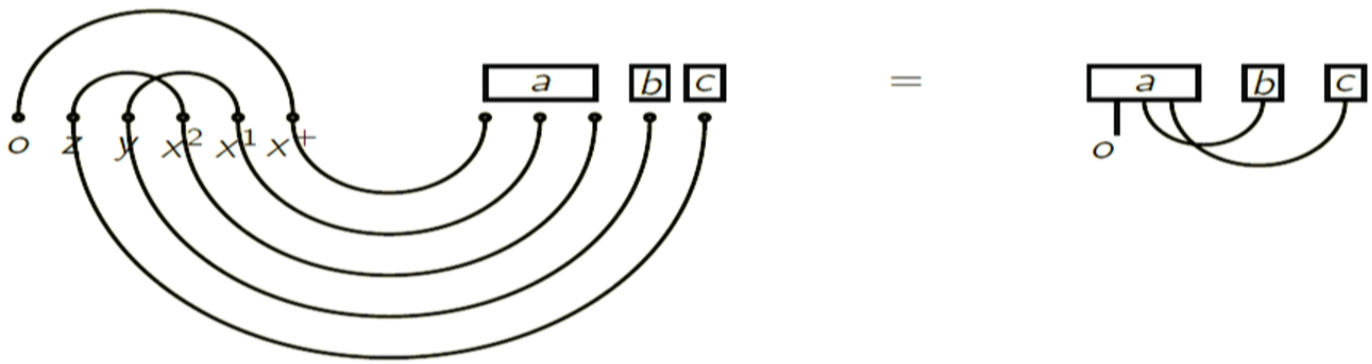
Here x^+ is the output of x, x^1 the first input, and x^2 the second input. The output of the whole expression is o.

Diagrammatic Simplification as β-Reduction

When we abstract the variables, we obtain the following caps-only diagram:



Now we consider an application $\mathbf{C}abc$:



Linguistics

Linguistics

Clark, Coecke and Sadrzadeh: Compositional Distributional Models of Meaning.

Linguistics

Clark, Coecke and Sadrzadeh: Compositional Distributional Models of Meaning.

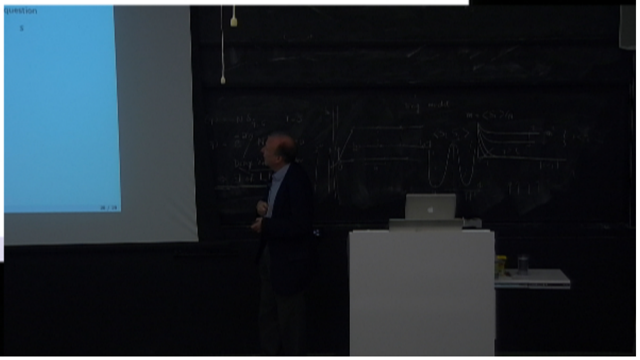
Lambek grammars: π pronoun, i infinitive, o direct object, ...

Linguistics

Clark, Coecke and Sadrzadeh: Compositional Distributional Models of Meaning.

Lambek grammars: π pronoun, i infinitive, o direct object, ...

Does	he	like	her?		question
$s\, i^l \pi^l$	π	$i\, o^l$	o	$\to$	s



Linguistics

Clark, Coecke and Sadrzadeh: Compositional Distributional Models of Meaning.

Lambek grammars: π pronoun, i infinitive, o direct object, ...

Does	he	like	her?		question
$s\ i^l \pi^l$	π	$i\ o^l$	o	$\rightarrow$	s

Linguistics

Clark, Coecke and Sadrzadeh: Compositional Distributional Models of Meaning.

Lambek grammars: π pronoun, i infinitive, o direct object, ...

Does he like her? $\quad\quad$ question

$$s\, i^l \pi^l \quad \pi \quad i\, o^l \quad o \quad \longrightarrow \quad s$$

Distributional models: words interpreted as vectors of frequency counts of co-occurrences of a set of reference words (the basis) within a fixed (small) word radius in a large text corpus. Widely used in information retrieval.

Linguistics

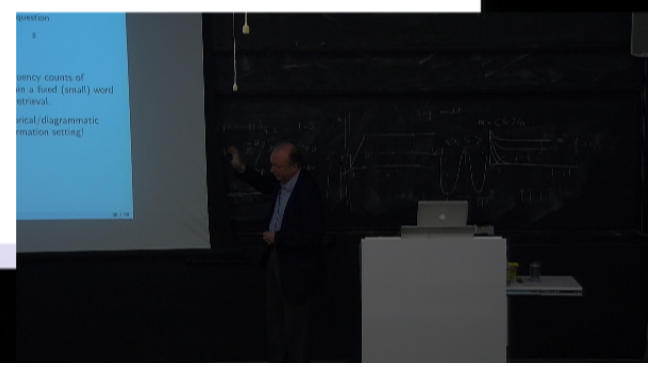
Clark, Coecke and Sadrzadeh: Compositional Distributional Models of Meaning.

Lambek grammars: π pronoun, i infinitive, o direct object, ...

Does	he	like	her?		question
$s\, i^l \pi^l$	π	$i\, o^l$	o	$\rightarrow$	s

Distributional models: words interpreted as vectors of frequency counts of co-occurrences of a set of reference words (the basis) within a fixed (small) word radius in a large text corpus. Widely used in information retrieval.

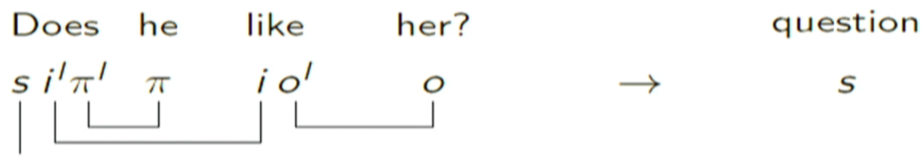
These seem very different: but they have the same categorical/diagrammatic structure — vector spaces treated as in the quantum information setting!



Linguistics

Clark, Coecke and Sadrzadeh: Compositional Distributional Models of Meaning.

Lambek grammars: π pronoun, i infinitive, o direct object, ...



Distributional models: words interpreted as vectors of frequency counts of co-occurrences of a set of reference words (the basis) within a fixed (small) word radius in a large text corpus. Widely used in information retrieval.

These seem very different: but they have the same categorical/diagrammatic structure — vector spaces treated as in the quantum information setting!

So we can functorially map Lambek pregroup parses into vector spaces to lift the distributional word meanings compositionally to meanings for phrases and sentences.

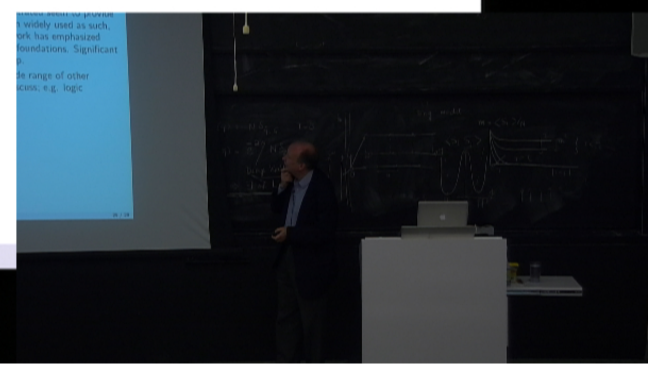
Implementations and benchmarks look promising: see recent work by Sadrzadeh and Graefenstette.

Final Remarks

- Structures in monoidal categories, involving compact structure, trace etc., which support the diagrammatic calculus we have illustrated seem to provide a canonical setting for discussing *processes*. Have been widely used as such, implicitly or explicitly, in Computer Science. Recent work has emphasized their relevance in quantum information and quantum foundations. Significant links to work at PI in the Quantum Foundations group.

Final Remarks

- Structures in monoidal categories, involving compact structure, trace etc., which support the diagrammatic calculus we have illustrated seem to provide a canonical setting for discussing *processes*. Have been widely used as such, implicitly or explicitly, in Computer Science. Recent work has emphasized their relevance in quantum information and quantum foundations. Significant links to work at PI in the Quantum Foundations group.
- As we have seen, the same structures reach into a wide range of other disciplines. There are more we didn't have time to discuss; e.g. logic (cut-elimination as information flow through proofs).

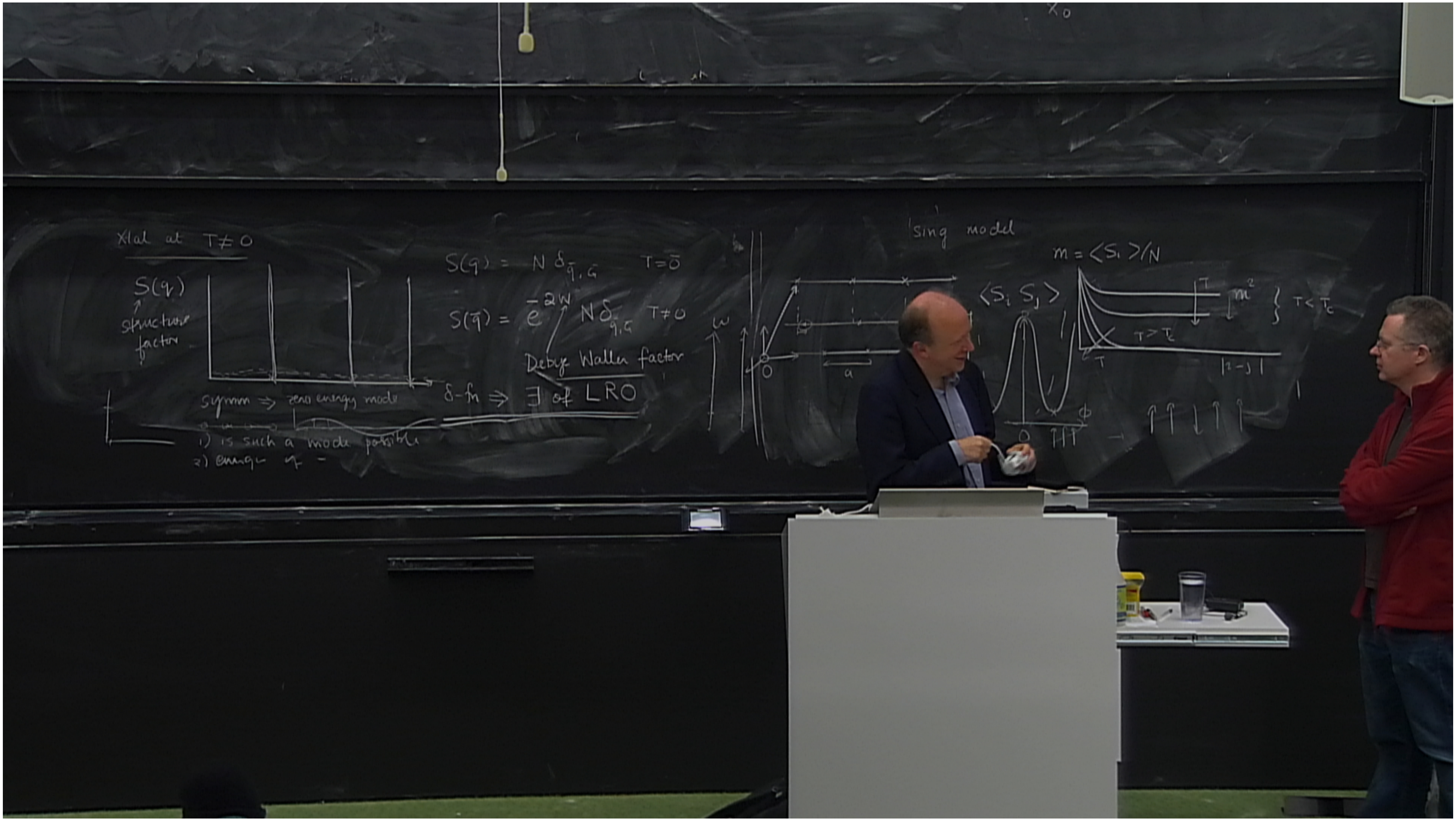


Final Remarks

- Structures in monoidal categories, involving compact structure, trace etc., which support the diagrammatic calculus we have illustrated seem to provide a canonical setting for discussing *processes*. Have been widely used as such, implicitly or explicitly, in Computer Science. Recent work has emphasized their relevance in quantum information and quantum foundations. Significant links to work at PI in the Quantum Foundations group.
- As we have seen, the same structures reach into a wide range of other disciplines. There are more we didn't have time to discuss; e.g. logic (cut-elimination as information flow through proofs).
- There are other promising ingredients for a general theory of information flow. In particular, sheaves as a general 'logic of contextuality'. Again, very direct connections to quantum foundations, quantum information. The subject of my talk in the Quantum Foundations seminar yesterday.

Final Remarks

- Structures in monoidal categories, involving compact structure, trace etc., which support the diagrammatic calculus we have illustrated seem to provide a canonical setting for discussing *processes*. Have been widely used as such, implicitly or explicitly, in Computer Science. Recent work has emphasized their relevance in quantum information and quantum foundations. Significant links to work at PI in the Quantum Foundations group.
- As we have seen, the same structures reach into a wide range of other disciplines. There are more we didn't have time to discuss; e.g. logic (cut-elimination as information flow through proofs).
- There are other promising ingredients for a general theory of information flow. In particular, sheaves as a general 'logic of contextuality'. Again, very direct connections to quantum foundations, quantum information. The subject of my talk in the Quantum Foundations seminar yesterday.
- The interface between physics and computer science is vibrant and fruitful. Long may this continue!


Xtal at T≠0
S(q)
structure factor
S(q) = N δ q,G T=0
S(q) = e^-2W N δ q,G T≠0
Debye Waller factor
δ-fn ⇒ ∃ of LRO
symm ⇒ zero energy mode
1) is such a mode possible
Ising model
m = <Si>/N
<Si Sj>
T < Tc
T > Tc