Effective Quantum Certification via Linear Homotopy Types

Urs Schreiber (NYU Abu Dhabi)
on joint work at CQTS with
D. J. Myers, M. Riley,
and Hisham Sati

CENTER FOR
QUANTUM &
TOPOLOGICAL
SYSTEMS

presentation at:

The Topos Institute Colloquium, 13 Apr 2023

slides and further pointers at: ncatlab.org/Quantum+Certification+via+Linear+Homotopy+Types#TI2023

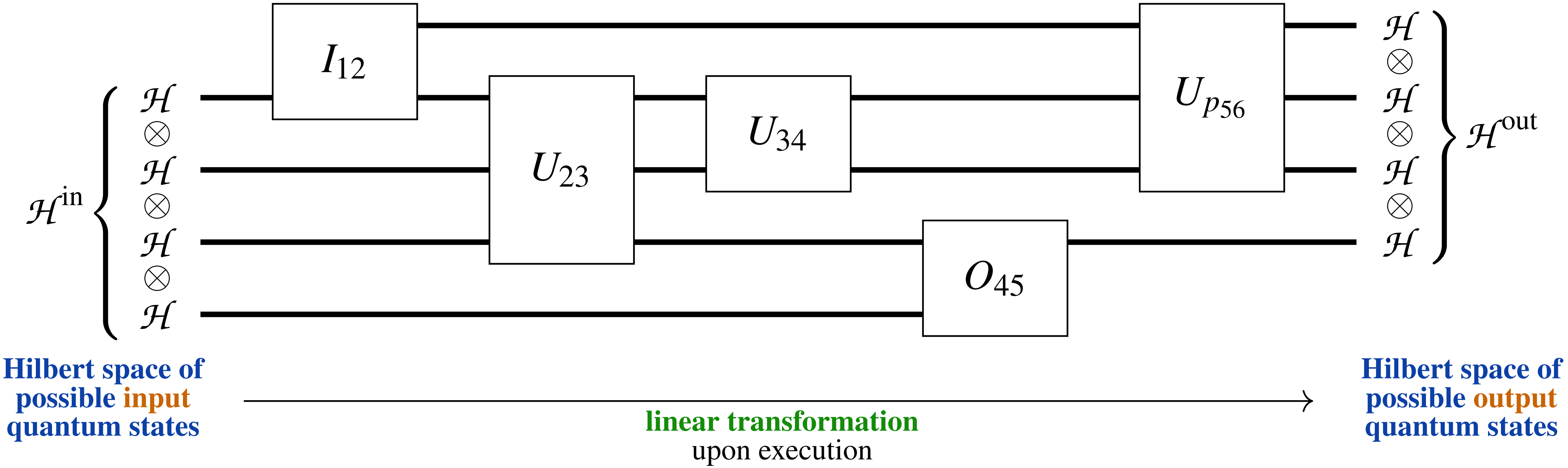
The Problem

in

Quantum Computing

Pure quantum circuits are easy...

Linear operator composed & tensored from given *quantum logic gates*



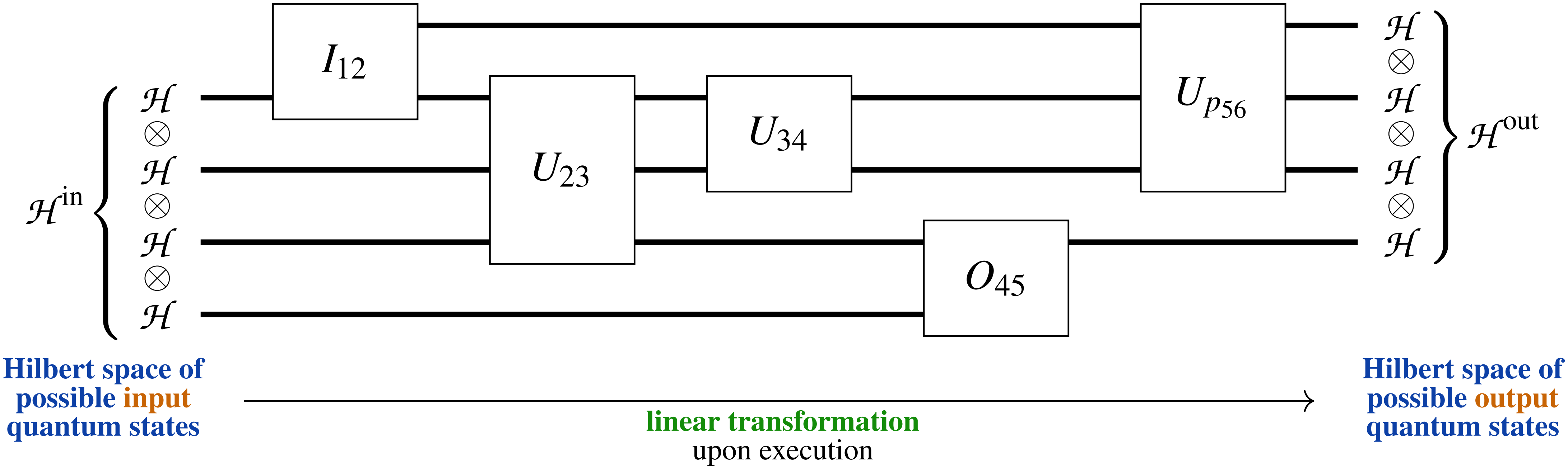
Hilbert space of possible input quantum states

linear transformation upon execution

Hilbert space of possible output quantum states

Pure quantum circuits are easy...

Linear operator composed & tensored from given *quantum logic gates*



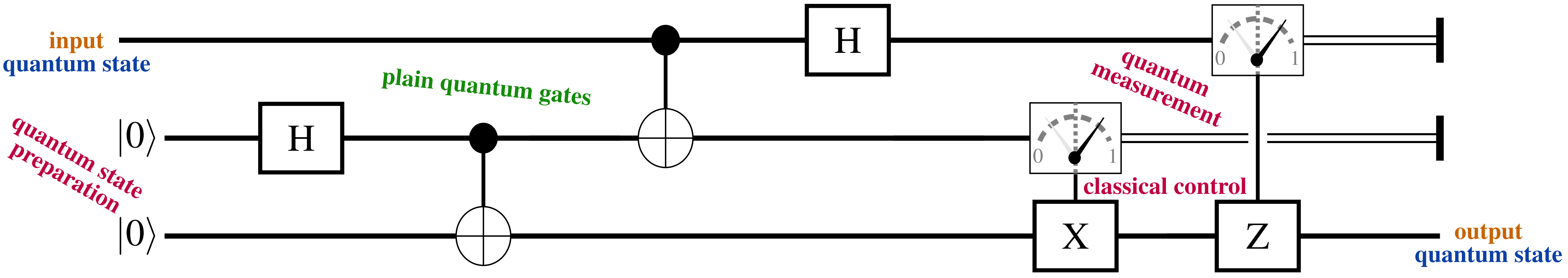
Hilbert space of possible input quantum states

linear transformation upon execution

Hilbert space of possible output quantum states

but real quantum circuits have classical control & effects

(Example: QBit Teleportation protocol)



input quantum state

quantum state preparation

plain quantum gates

quantum measurement

classical control

output quantum state

full reality is a loop: Classical $\underset{\text{prepare}}{\overset{\text{measure}}{\leftrightarrows}}$ Quantum

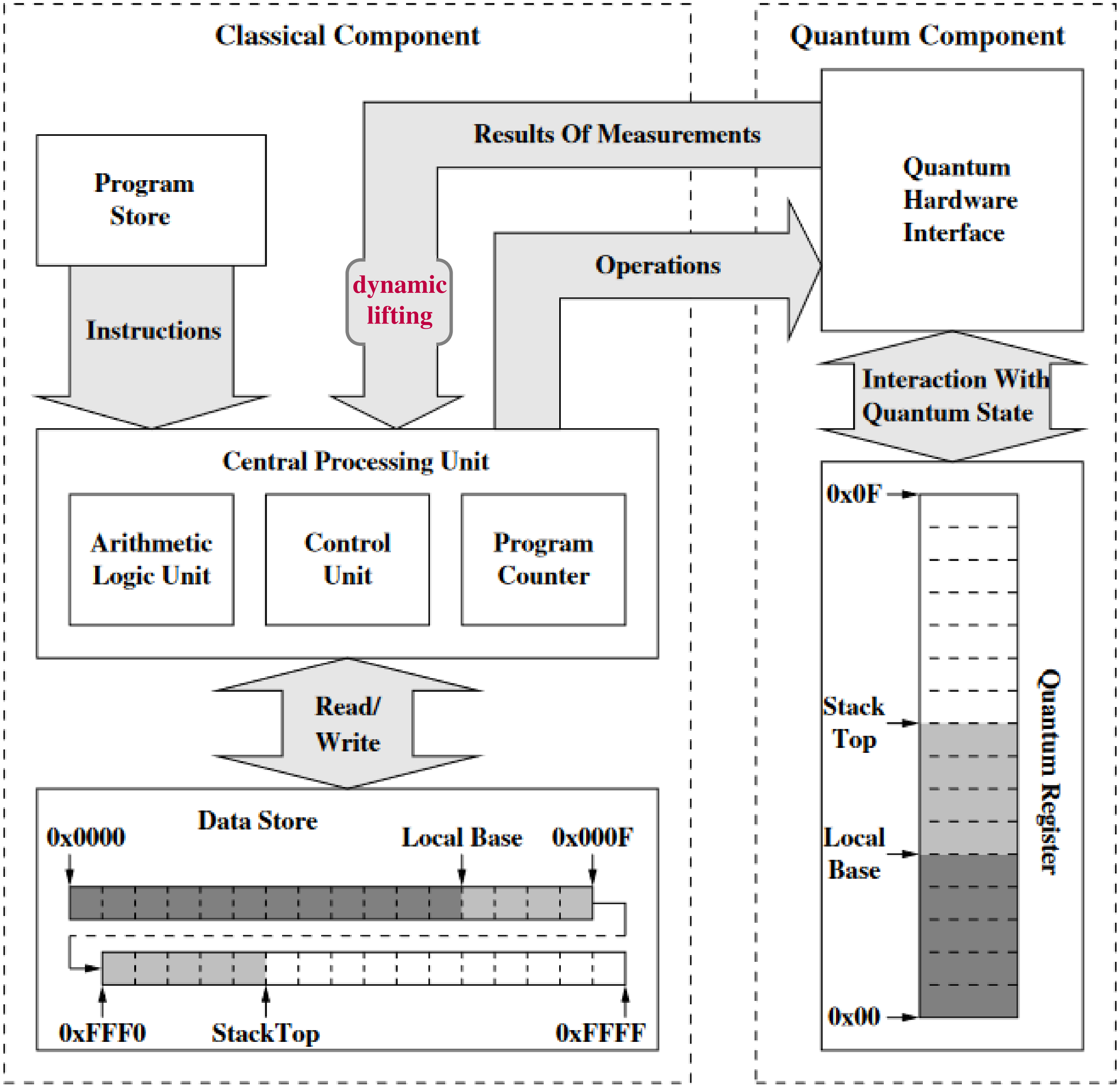
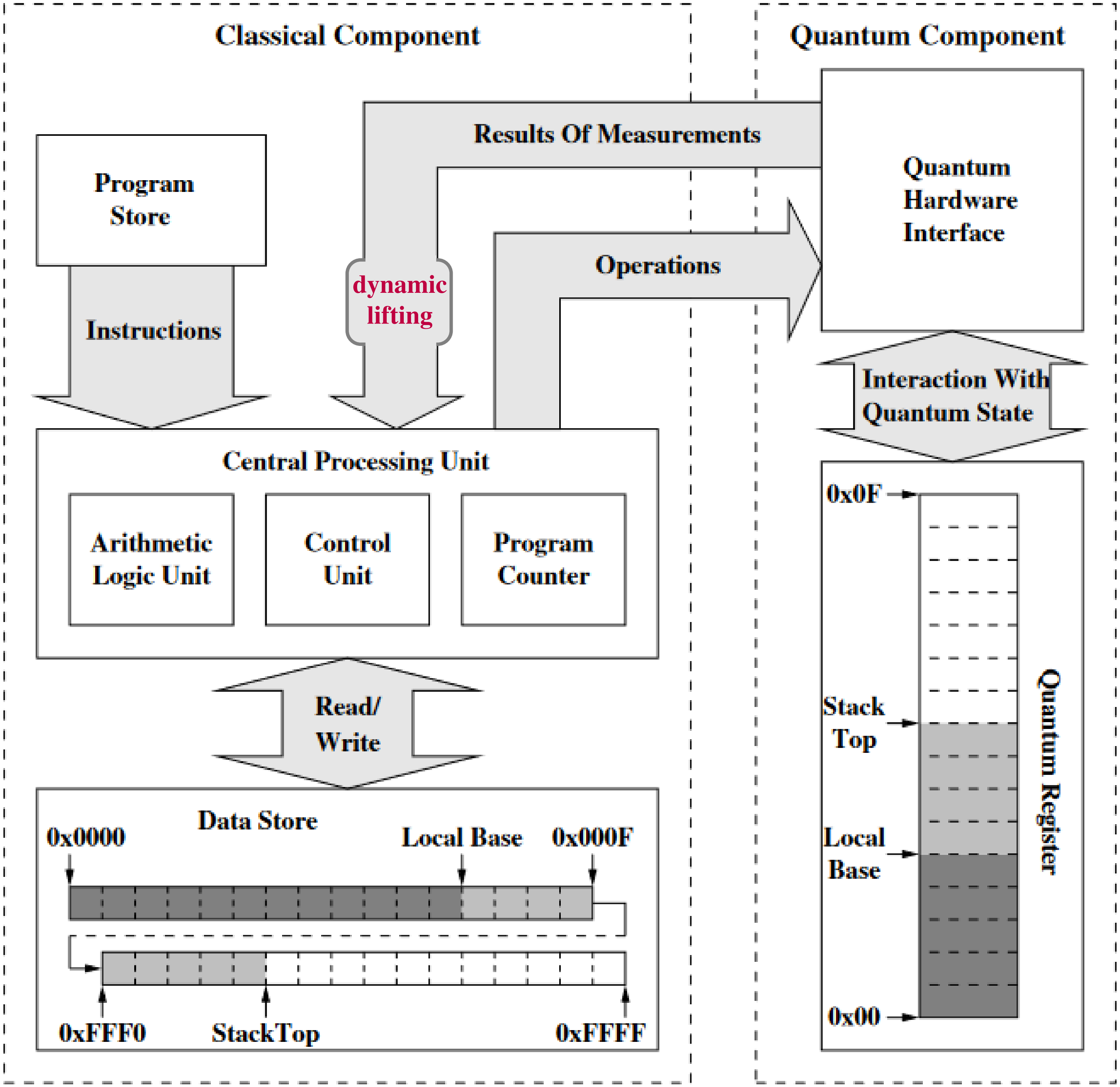


diagram adapted from Nagarajan et al. (2007)

full reality is a loop: Classical $\underset{\text{prepare}}{\overset{\text{measure}}{\leftrightarrows}}$ Quantum


Classical Component

Program Store

Instructions

dynamic lifting

Results Of Measurements

Operations

Central Processing Unit

Arithmetic Logic Unit

Control Unit

Program Counter

Read/ Write

Data Store

0x0000

Local Base

0x000F

0xFFF0

StackTop

0xFFFF

Quantum Component

Quantum Hardware Interface

Interaction With Quantum State

0x0F

Stack Top

Local Base

0x00

Quantum Register

diagram adapted from Nagarajan et al. (2007)

full loop needed e.g. for quantum error correction

full reality is a loop: Classical $\underset{\text{prepare}}{\overset{\text{measure}}{\leftrightarrows}}$ Quantum

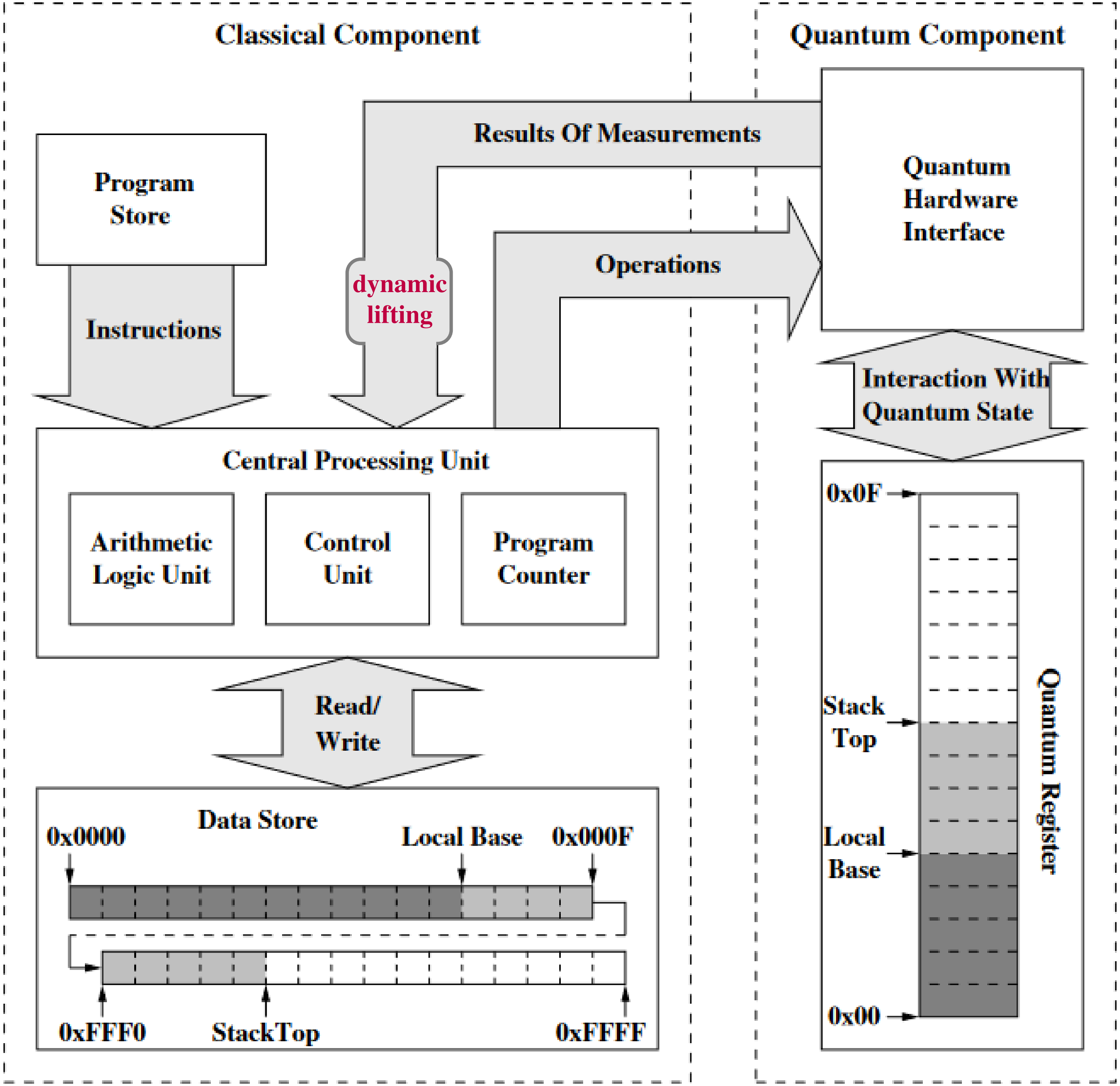


diagram adapted from Nagarajan et al. (2007)

full loop needed e.g. for quantum error correction

but its formal language theory had remained thin

full reality is a loop: Classical $\underset{\text{prepare}}{\overset{\text{measure}}{\leftrightarrows}}$ Quantum

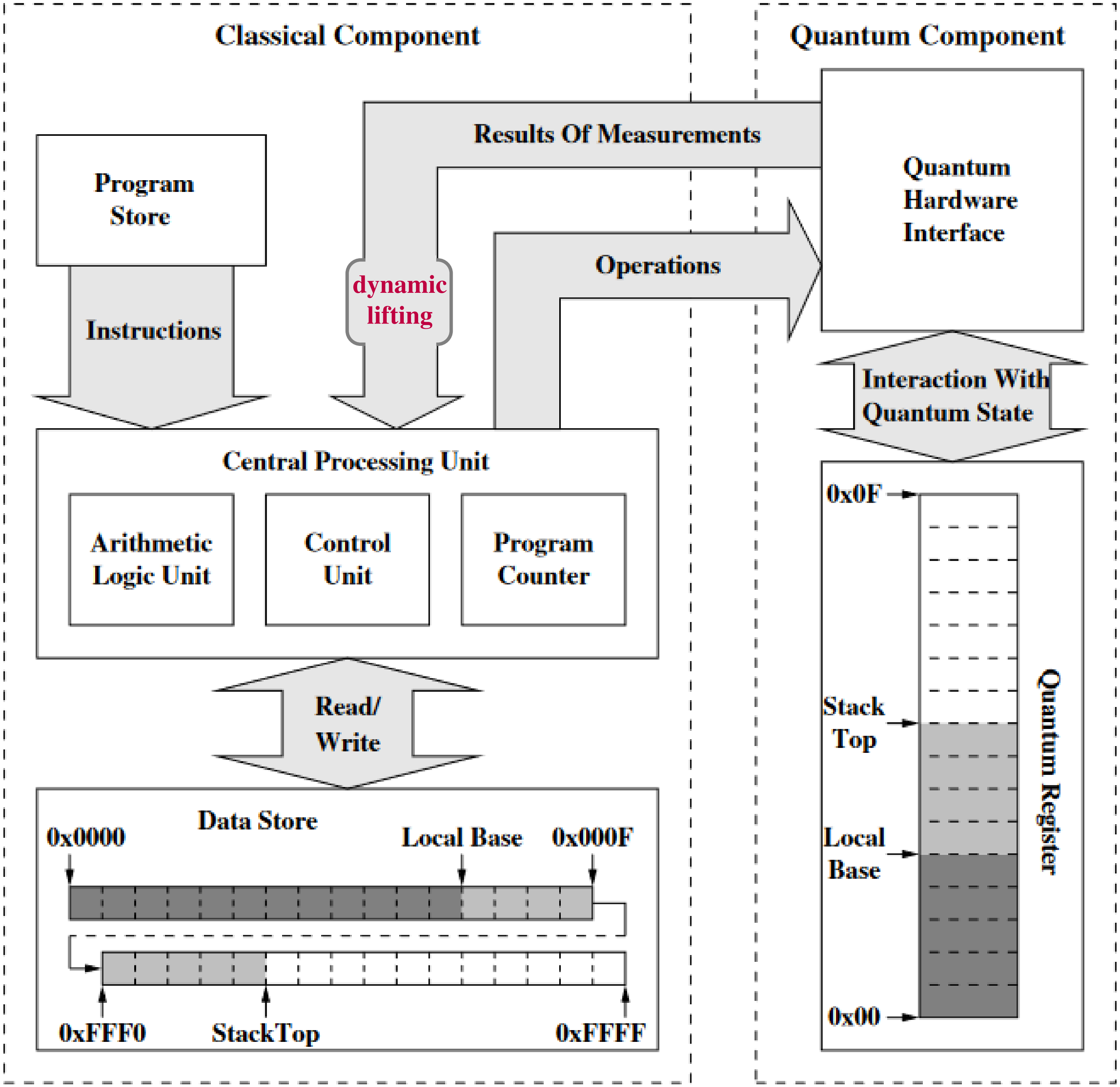


diagram adapted from Nagarajan et al. (2007)

full loop needed e.g. for quantum error correction

but its formal language theory had remained thin

existing models for dynamic lifting are ad hoc & unverified

Existing quantum typed circuit languages

are embedded inside *classical* type theories:

Existing quantum typed circuit languages

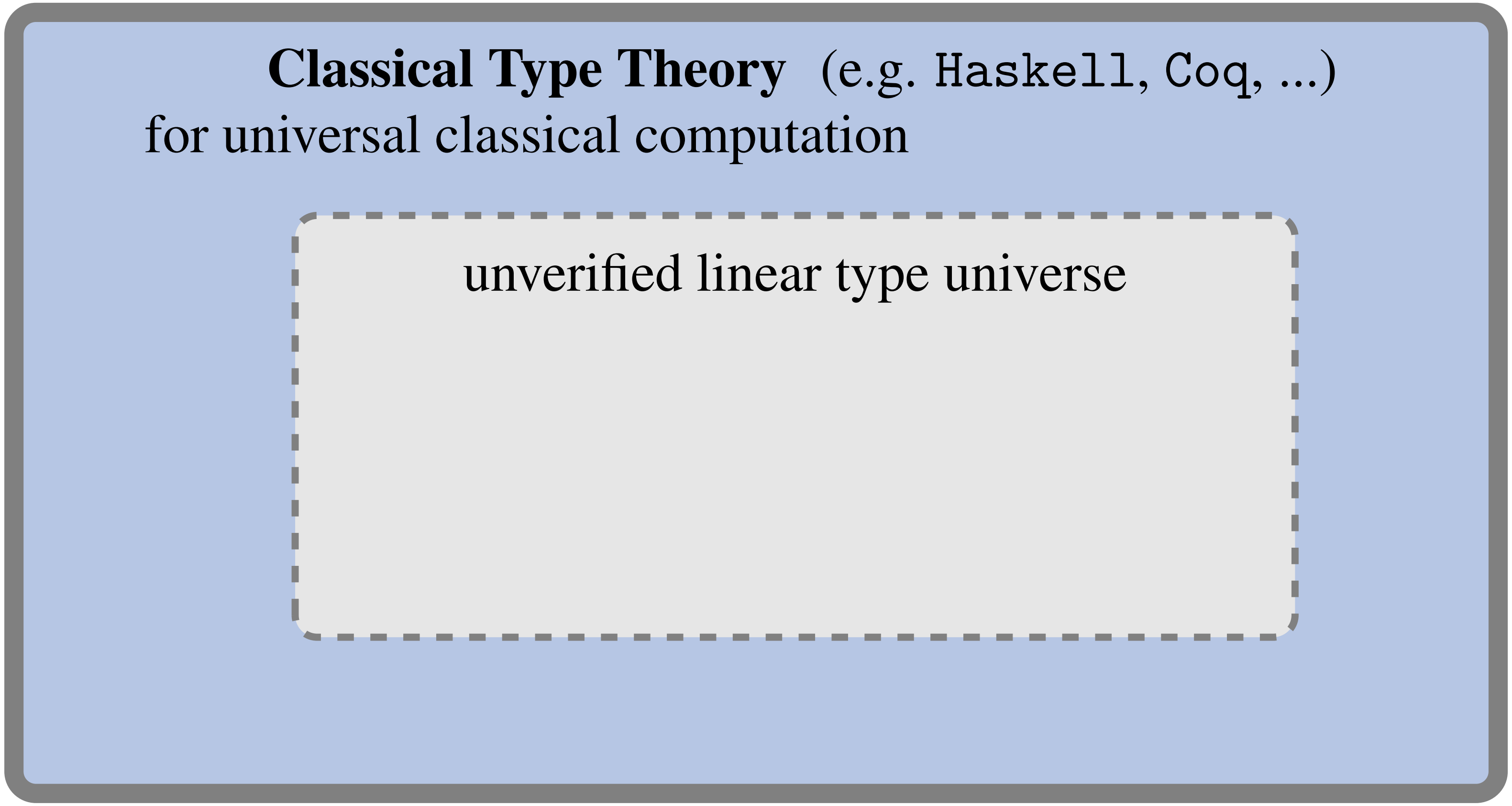
are embedded inside *classical* type theories:

Classical Type Theory (e.g. `Haskell`, `Coq`, ...)

for universal classical computation

Existing quantum typed circuit languages

are embedded inside *classical* type theories:



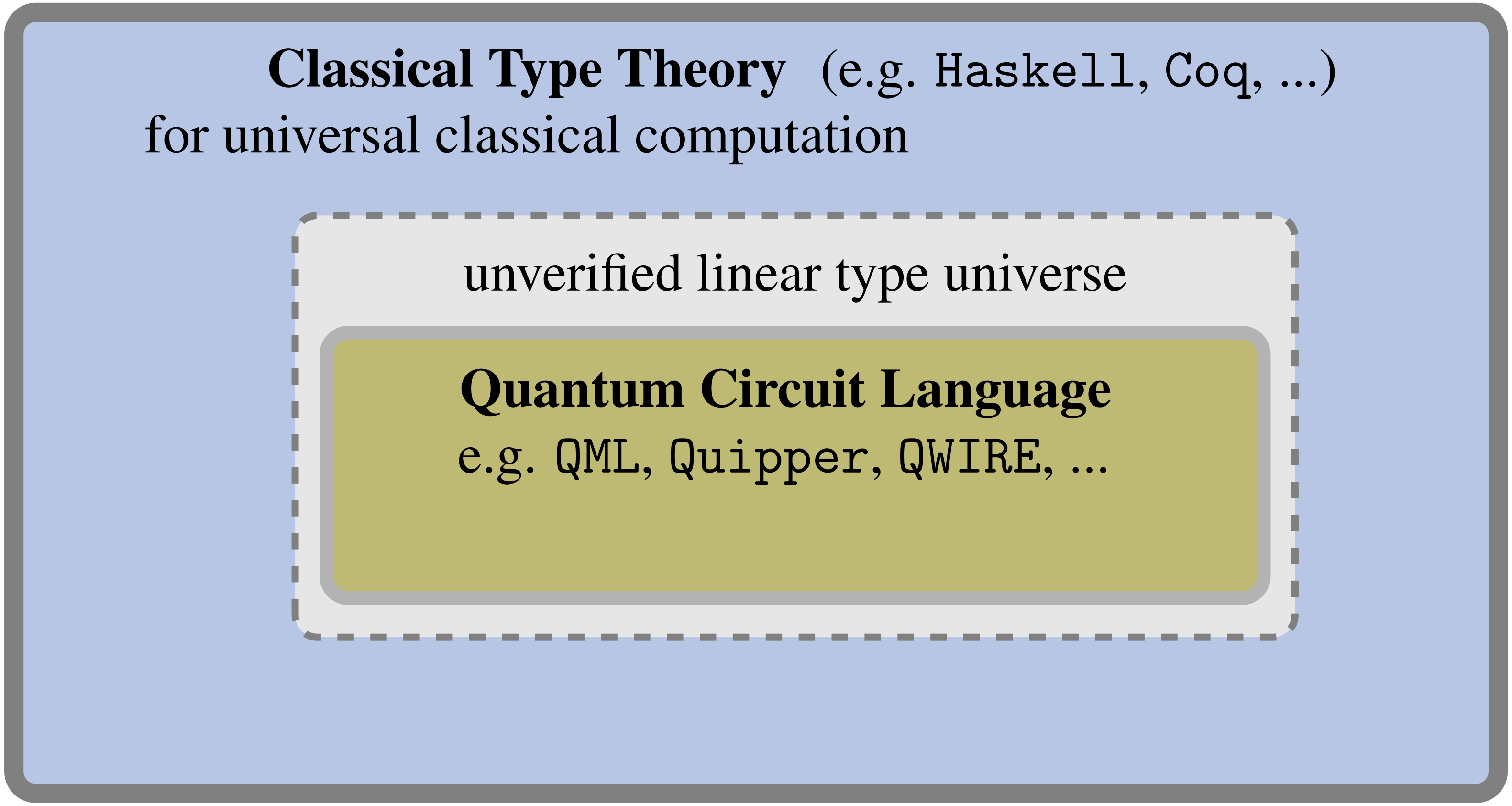
Classical Type Theory (e.g. `Haskell`, `Coq`, ...)

for universal classical computation

unverified linear type universe

Existing quantum typed circuit languages

are embedded inside *classical* type theories:



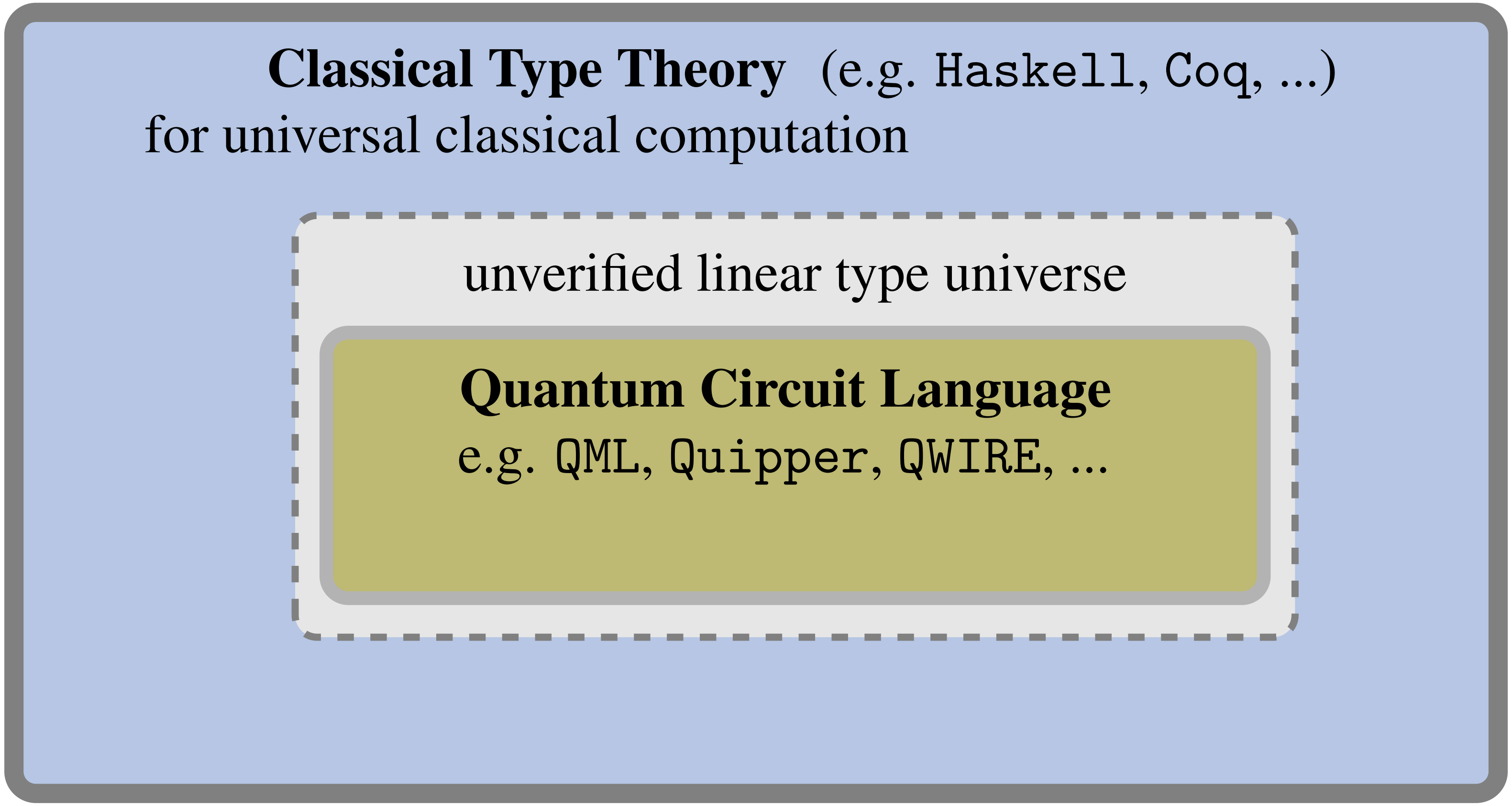
Classical Type Theory (e.g. `Haskell`, `Coq`, ...)
for universal classical computation

unverified linear type universe

Quantum Circuit Language
e.g. `QML`, `Quipper`, `QWIRE`, ...

Existing quantum typed circuit languages

are embedded inside *classical* type theories:



Classical Type Theory (e.g. `Haskell`, `Coq`, ...)
for universal classical computation

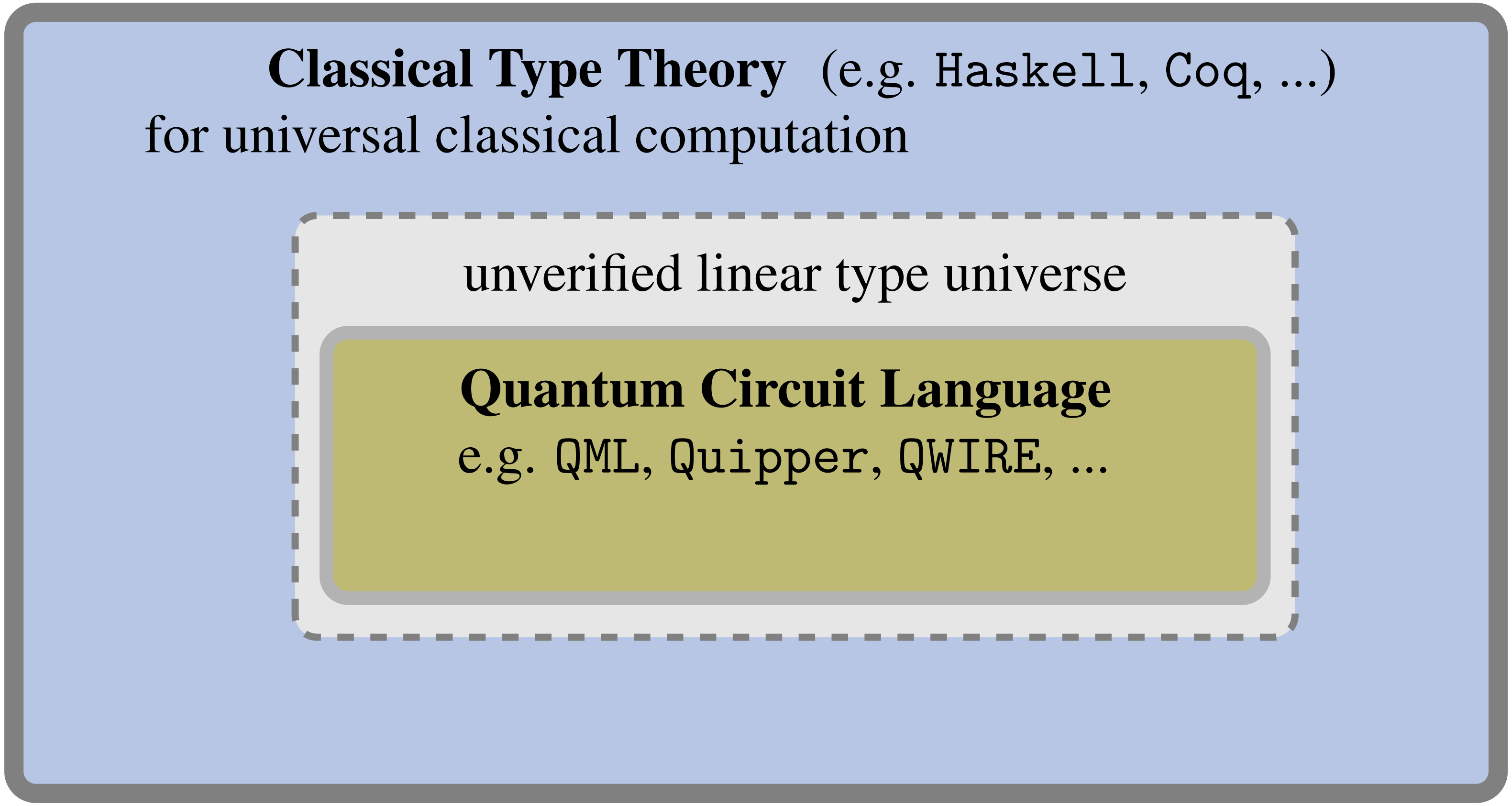
unverified linear type universe

Quantum Circuit Language
e.g. `QML`, `Quipper`, `QWIRE`, ...

for lack of a universal linear type theory.

Existing quantum typed circuit languages

are embedded inside *classical* type theories:



Classical Type Theory (e.g. `Haskell`, `Coq`, ...)
for universal classical computation

unverified linear type universe

Quantum Circuit Language
e.g. `QML`, `Quipper`, `QWIRE`, ...

for lack of a universal linear type theory.

Why did that not exist?

The Problem

in

Type Theory

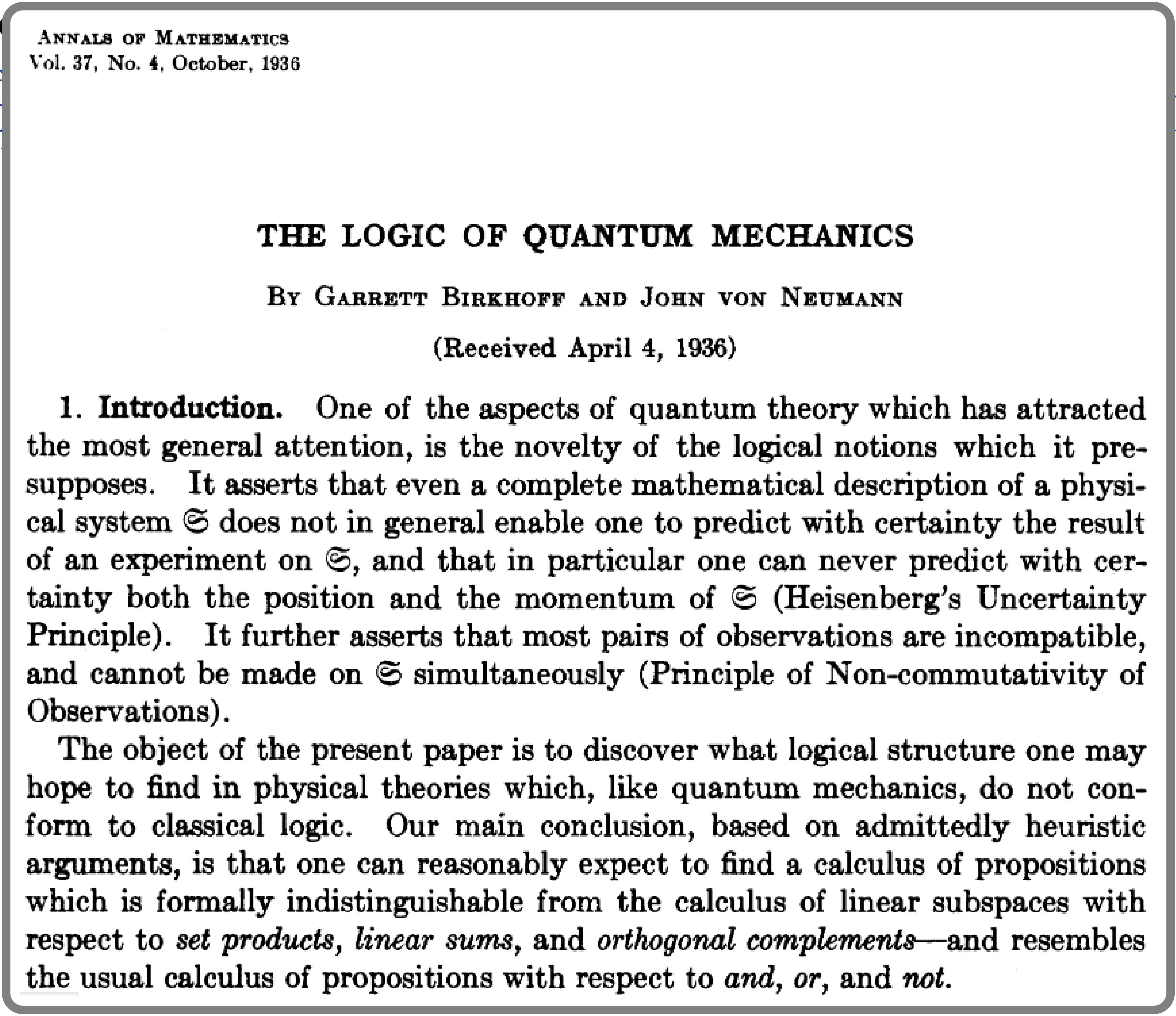
Historical search for quantum/linear logic/type-theory — a goose chase:

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Histori

Birkhof *logic*,
but seen ve, etc.)



Annals of Mathematics
Vol. 37, No. 4, October, 1936

THE LOGIC OF QUANTUM MECHANICS

By Garrett Birkhoff and John von Neumann

(Received April 4, 1936)

1. Introduction. One of the aspects of quantum theory which has attracted the most general attention, is the novelty of the logical notions which it presupposes. It asserts that even a complete mathematical description of a physical system $\mathfrak{S}$ does not in general enable one to predict with certainty the result of an experiment on $\mathfrak{S}$, and that in particular one can never predict with certainty both the position and the momentum of $\mathfrak{S}$ (Heisenberg's Uncertainty Principle). It further asserts that most pairs of observations are incompatible, and cannot be made on $\mathfrak{S}$ simultaneously (Principle of Non-commutativity of Observations).

The object of the present paper is to discover what logical structure one may hope to find in physical theories which, like quantum mechanics, do not conform to classical logic. Our main conclusion, based on admittedly heuristic arguments, is that one can reasonably expect to find a calculus of propositions which is formally indistinguishable from the calculus of linear subspaces with respect to *set products*, *linear sums*, and *orthogonal complements*—and resembles the usual calculus of propositions with respect to *and*, *or*, and *not*.

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Historical search for quantum/linear logic/type-theory — a goose chase:

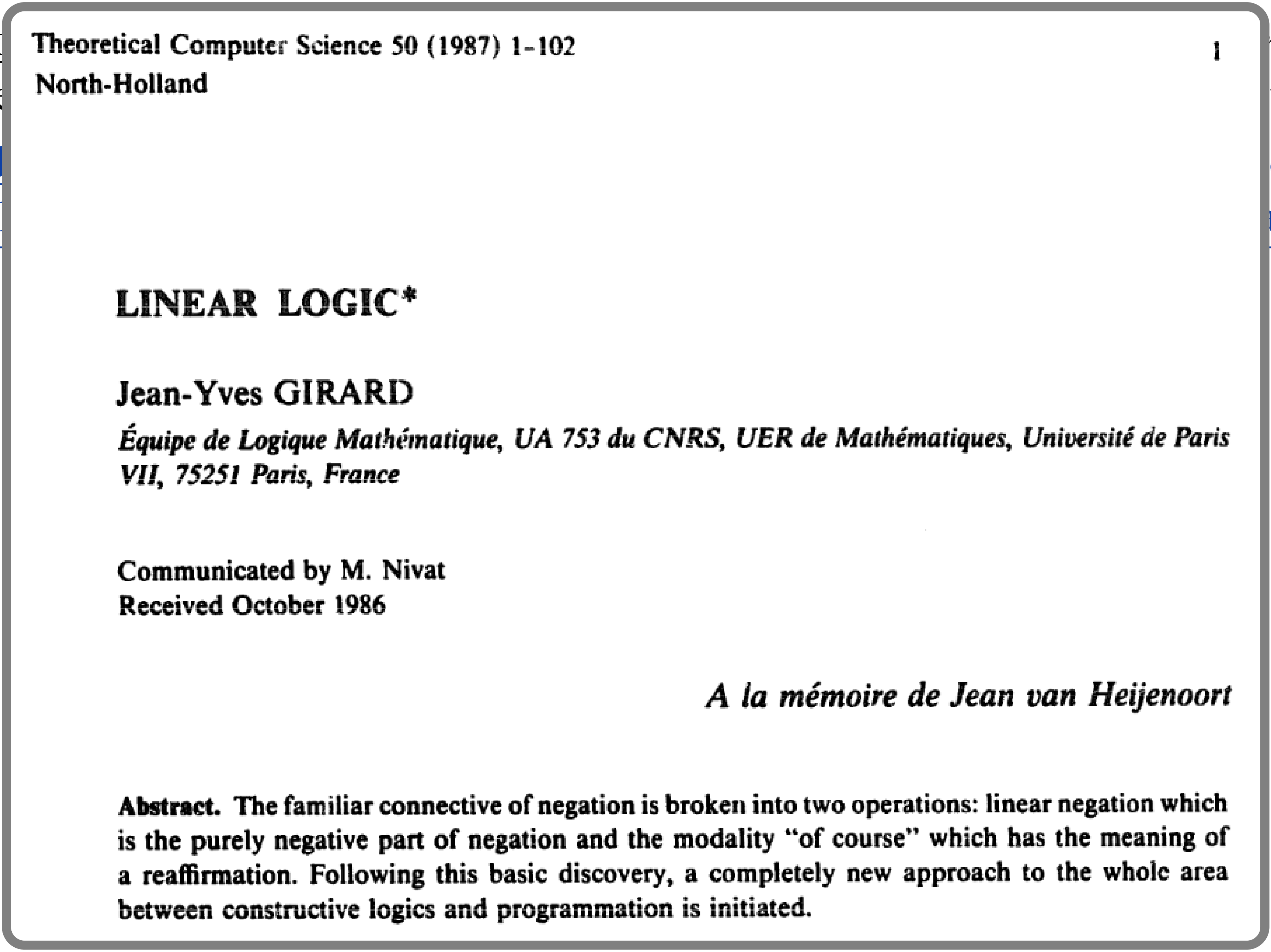
Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkho *n logic*,
but see tive, etc.)

Girard e
“wild l am!”



Theoretical Computer Science 50 (1987) 1-102 1
North-Holland

LINEAR LOGIC*

Jean-Yves GIRARD

Équipe de Logique Mathématique, UA 753 du CNRS, UER de Mathématiques, Université de Paris VII, 75251 Paris, France

Communicated by M. Nivat
Received October 1986

A la mémoire de Jean van Heijenoort

Abstract. The familiar connective of negation is broken into two operations: linear negation which is the purely negative part of negation and the modality “of course” which has the meaning of a reaffirmation. Following this basic discovery, a completely new approach to the whole area between constructive logics and programmation is initiated.

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Abramsky, & Duncan 2005 also find BvN "a non-logic...never satisfactory" consider the internal logic of †-compact closed categories, like FDHilb

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Abramsky, & Duncan 2005 also find BvN "a non-logic...never satisfactory" consider the internal logic of †-compact closed categories, like FDHilb

A categorical quantum logic

Published online by Cambridge University Press: **04 July 2006**

SAMSON ABRAMSKY and ROSS DUNCAN

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Abramsky, & Duncan 2005 also find BvN "a non-logic...never satisfactory" consider the internal logic of †-compact closed categories, like FDHilb

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

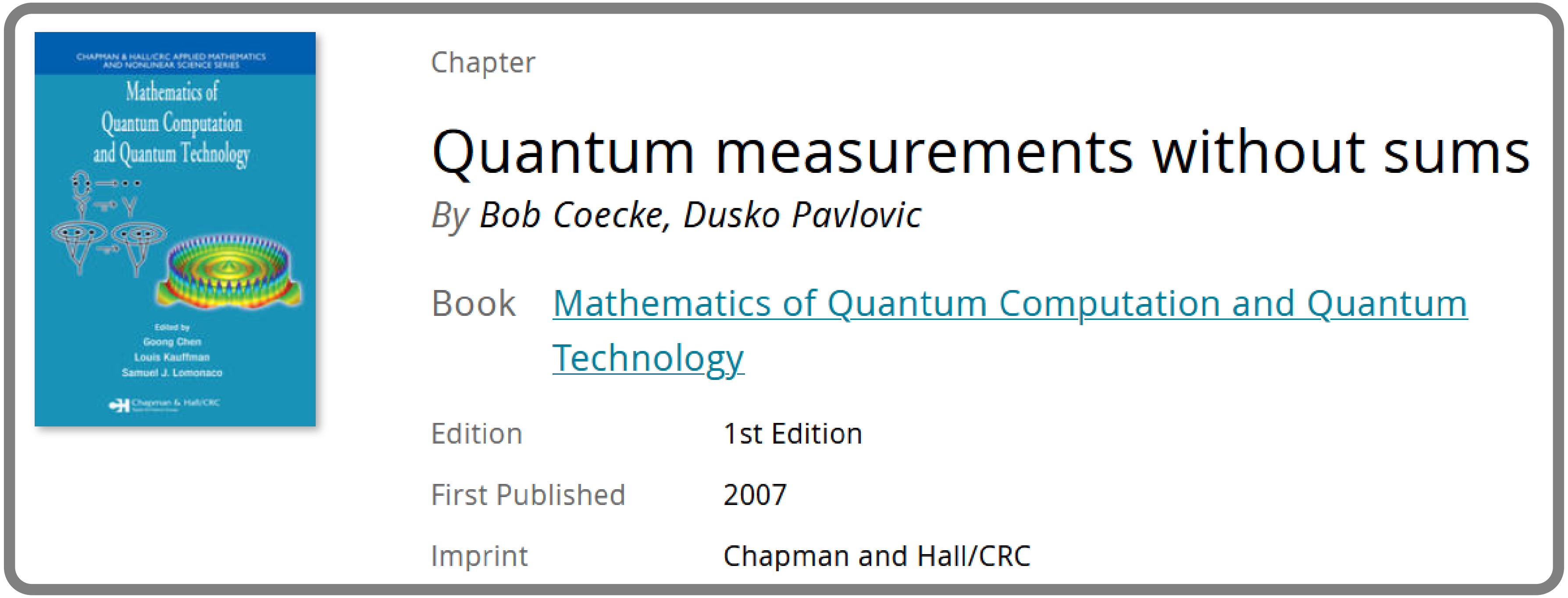
Abramsky, & Duncan 2005 also find BvN "a non-logic...never satisfactory" consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Abramsky, & Duncan 2005 also find BvN "a non-logic...never satisfactory" consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.



Chapter

Quantum measurements without sums

By Bob Coecke, Dusko Pavlovic

Book Mathematics of Quantum Computation and Quantum Technology

Edition 1st Edition

First Published 2007

Imprint Chapman and Hall/CRC

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a “magisterial mistake”, introduces *linear logic* in the “wild hope of direct connection with quantum mechanics – but let’s not dream!”

Abramsky, & Duncan 2005 also find BvN “a non-logic...never satisfactory” consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Abramsky, & Duncan 2005 also find BvN "a non-logic...never satisfactory" consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Coecke & Duncan 2008 make this the first *practically* successful quantum logic.

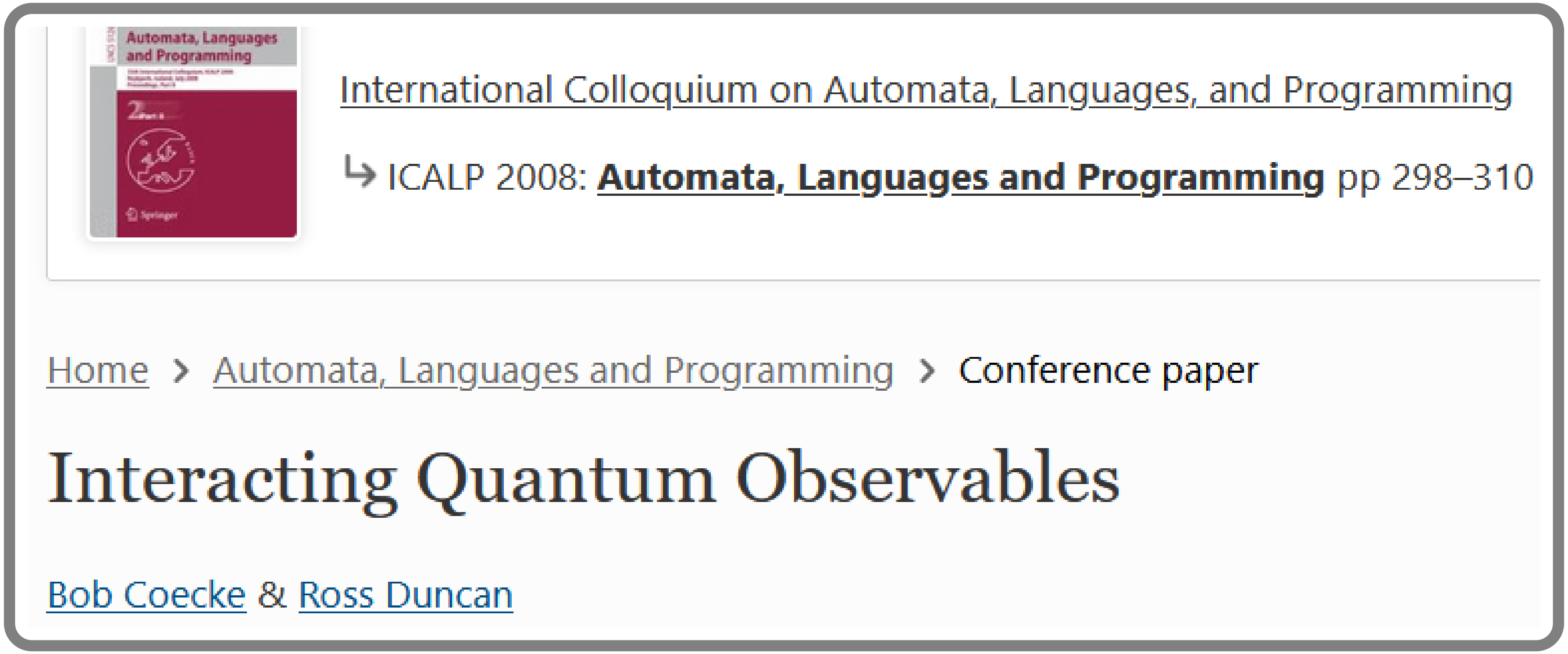
Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a “magisterial mistake”, introduces *linear logic* in the “wild hope of direct connection with quantum mechanics – but let’s not dream!”

Abramsky, & Duncan 2005 also find BvN “a non-logic...never satisfactory” consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Coecke & Duncan 2008 make this the first *practically* successful quantum logic.



International Colloquium on Automata, Languages, and Programming

↳ ICALP 2008: **Automata, Languages and Programming** pp 298–310

Home › Automata, Languages and Programming › Conference paper

Interacting Quantum Observables

Bob Coecke & Ross Duncan

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Abramsky, & Duncan 2005 also find BvN "a non-logic...never satisfactory" consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Coecke & Duncan 2008 make this the first *practically* successful quantum logic.

Historical search for quantum/linear logic/type-theory — a goose chase:

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Abramsky, & Duncan 2005 also find BvN "a non-logic...never satisfactory" consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Coecke & Duncan 2008 make this the first *practically* successful quantum logic.

Murfet 2014 gives rare amplification that (FD)Vect, of course, interprets Girard's linear logic: So this was secretly a successful quantum logic all along!

Birkhoff-von Neumann 1936 (BvN) give physically well-motivated *quantum logic*, but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Abramsky, & Duncan 2005 also find BvN "a non-logic...never satisfactory" consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Coecke & Duncan 2008 make this the first *practically* successful quantum logic.

Murfet 2014 gives rare amplification that (FD)Vect, of course, interprets Girard's linear logic: So this was secretly a successful quantum logic all along!

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Abramsky, & Duncan 2005 also find BvN "a non-logic...never satisfactory" consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Coecke & Duncan 2008 make this the first *practically* successful quantum logic.

Murfet 2014 gives rare amplification that (FD)Vect, of course, interprets Girard's linear logic: So this was secretly a successful quantum logic all along!

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices FDVect$_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

but seemingly formally unsatisfactory (infamous lack of *implication*-connective, etc.)

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Abramsky, & Duncan 2005 also find BvN "a non-logic...never satisfactory" consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Coecke & Duncan 2008 make this the first *practically* successful quantum logic.

Murfet 2014 gives rare amplification that (FD)Vect, of course, interprets Girard's linear logic: So this was secretly a successful quantum logic all along!

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices $\mathrm{FDVect}_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

<table>
<tr><th colspan="3">Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$</th></tr>
<tr><th>proposition</th><th>conjunction</th><th>disjunction</th></tr>
<tr><td>$\mathcal{P}$</td><td>$\begin{tikzcd} \mathcal{P}_1 & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] & \mathcal{P}_2 \end{tikzcd}$</td><td>$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] & \mathrm{Span}(\mathcal{P}_1, \mathcal{P}_2) & \mathcal{P}_2 \arrow[ul] \end{tikzcd}$</td></tr>
</table>

Girard 1987 finds BvN a "magisterial mistake", introduces *linear logic* in the "wild hope of direct connection with quantum mechanics – but let's not dream!"

Abramsky, & Duncan 2005 also find BvN "a non-logic...never satisfactory" consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Coecke & Duncan 2008 make this the first *practically* successful quantum logic.

Murfet 2014 gives rare amplification that (FD)Vect, of course, interprets Girard's linear logic: So this was secretly a successful quantum logic all along!

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices $\mathrm{FDVect}_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$		
proposition	**conjunction**	**disjunction**
$\begin{tikzcd} \mathcal{P} \\ {} \arrow[u, hook, "p"] \end{tikzcd}$	$\begin{tikzcd} \mathcal{P}_1 & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] & \mathcal{P}_2 \\ & {} \arrow[ul, hook, "p_1"] \arrow[u, hook, "p_1 \wedge p_2"] \arrow[ur, hook', "p_2"'] & \end{tikzcd}$	$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] & \mathrm{Span}(\mathcal{P}_1, \mathcal{P}_2) & \mathcal{P}_2 \arrow[ul] \\ & {} \arrow[ul, hook, "p_1"] \arrow[u, hook, "p_1 \vee p_2"] \arrow[ur, hook', "p_2"'] & \end{tikzcd}$

Abramsky, & Duncan 2005 also find BvN “a non-logic...never satisfactory” consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Coecke & Duncan 2008 make this the first *practically* successful quantum logic.

Murfet 2014 gives rare amplification that (FD)Vect, of course, interprets Girard’s linear logic: So this was secretly a successful quantum logic all along!

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices $\mathrm{FDVect}_{/\mathcal{H}}$) would immediately recover BvN’s quantum logic:

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$

proposition	**conjunction**	**disjunction**
$$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p"] \\ \mathcal{H} \end{tikzcd}$$	$$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1"'] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook', "p_2"] \\ & \mathcal{H} & \end{tikzcd}$$	$$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"'] & \mathrm{Span}\big(\mathcal{P}_1, \mathcal{P}_2\big) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook', "p_2"] \\ & \mathcal{H} & \end{tikzcd}$$

consider the internal logic of †-compact closed categories, like FDHilb, which Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Coecke & Duncan 2008 make this the first *practically* successful quantum logic.

Murfet 2014 gives rare amplification that (FD)Vect, of course, interprets Girard's linear logic: So this was secretly a successful quantum logic all along!

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices FDVect$_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

Internal logic in FDVect$_{/\mathcal{H}}$

proposition	**conjunction**	**disjunction**
$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p"] \\ \mathcal{H} \end{tikzcd}$	$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1"'] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook', "p_2"] \\ & \mathcal{H} & \end{tikzcd}$	$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"'] & \mathrm{Span}(\mathcal{P}_1, \mathcal{P}_2) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook', "p_2"] \\ & \mathcal{H} & \end{tikzcd}$

Coecke & Pavlović 2008 equip with Frobenius monads for quantum measurement.

Coecke & Duncan 2008 make this the first *practically* successful quantum logic.

Murfet 2014 gives rare amplification that (FD)Vect, of course, interprets Girard's linear logic: So this was secretly a successful quantum logic all along!

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices $\mathrm{FDVect}_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$

proposition	**conjunction**	**disjunction**
$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p"] \\ \mathcal{H} \end{tikzcd}$	$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1"] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook', "p_2"] \\ & \mathcal{H} & \end{tikzcd}$	$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"] & \mathrm{Span}\left(\mathcal{P}_1, \mathcal{P}_2\right) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook', "p_2"] \\ & \mathcal{H} & \end{tikzcd}$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$ – these ought to be interpreted as vector (Hilbert) *bundles*.

Coecke & Duncan 2008 make this the first *practically* successful quantum logic.

Murfet 2014 gives rare amplification that (FD)Vect, of course, interprets Girard's linear logic: So this was secretly a successful quantum logic all along!

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices $\mathrm{FDVect}_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$		
proposition	**conjunction**	**disjunction**
$$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p"] \\ \mathcal{H} \end{tikzcd}$$	$$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1"'] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook', "p_2"] \\ & \mathcal{H} & \end{tikzcd}$$	$$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"'] & \mathrm{Span}(\mathcal{P}_1, \mathcal{P}_2) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook', "p_2"] \\ & \mathcal{H} & \end{tikzcd}$$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$ – these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory

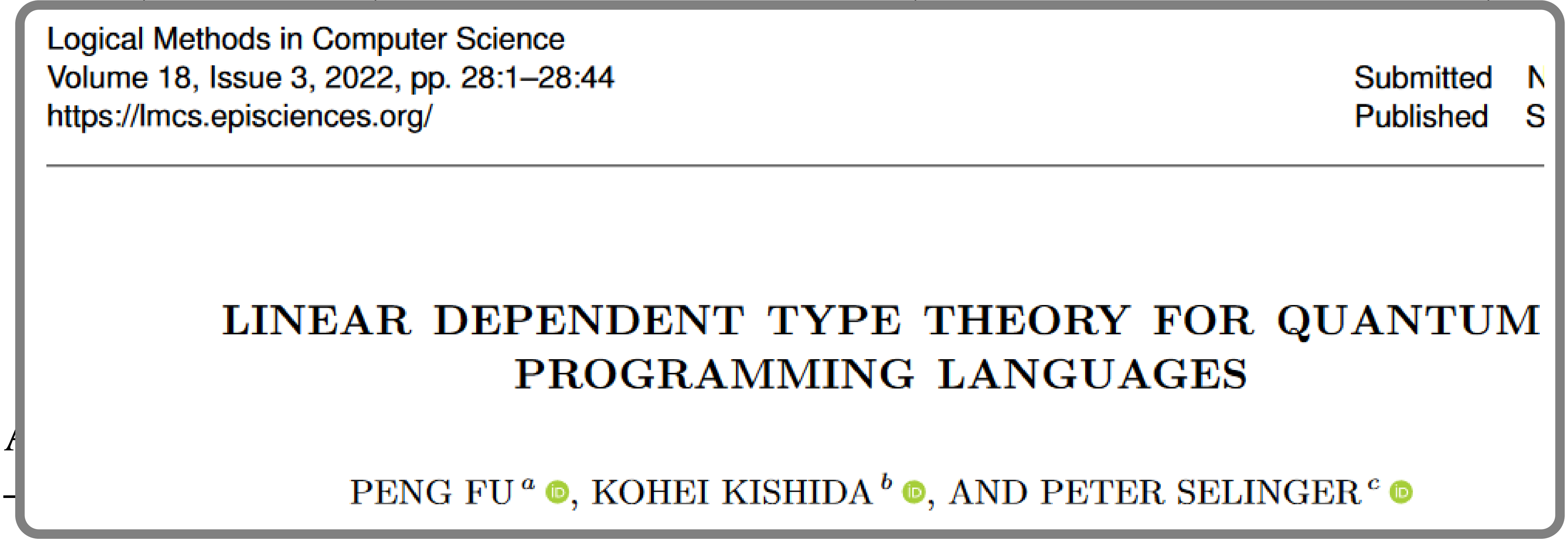
Coecke & Duncan 2008 make this the first *practically* successful quantum logic.

Murfet 2014 gives rare amplification that (FD)Vect, of course, interprets Girard's linear logic: So this was secretly a successful quantum logic all along!

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices $\mathrm{FDVect}_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$		
proposition	**conjunction**	**disjunction**



Logical Methods in Computer Science
Volume 18, Issue 3, 2022, pp. 28:1–28:44
https://lmcs.episciences.org/

Submitted N
Published S

LINEAR DEPENDENT TYPE THEORY FOR QUANTUM PROGRAMMING LANGUAGES

PENG FU [a], KOHEI KISHIDA [b], AND PETER SELINGER [c]

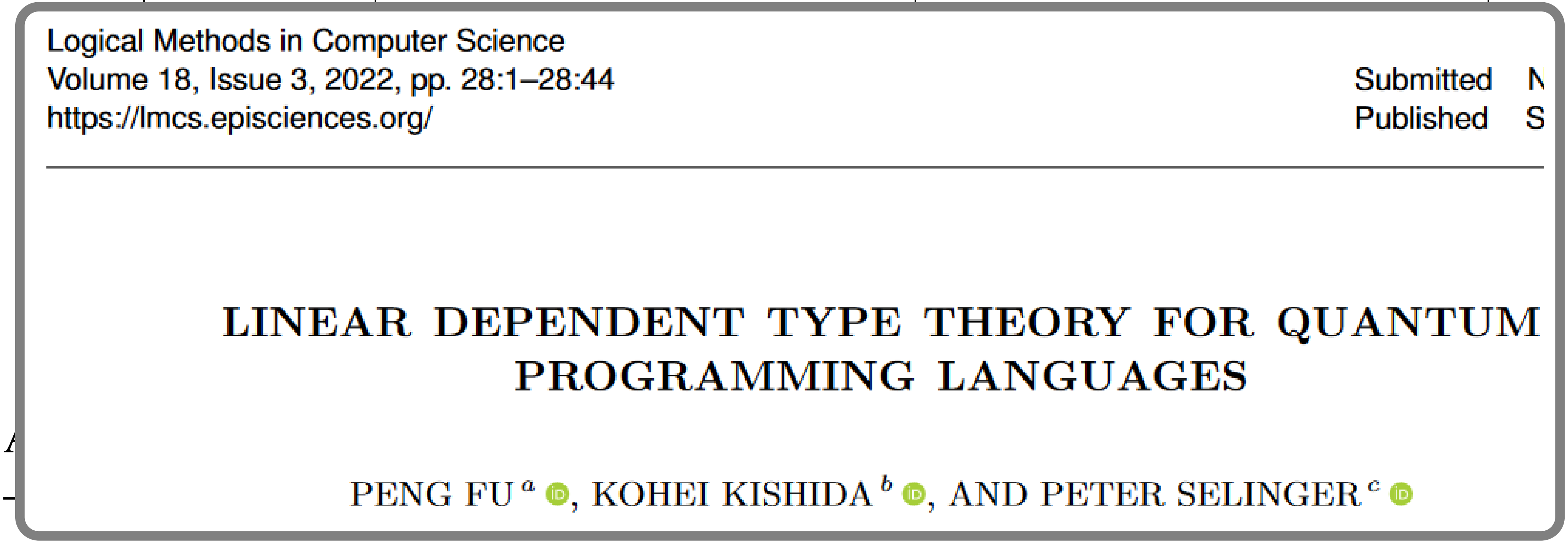
Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory

Murfet 2014 gives rare amplification that (FD)Vect, of course, interprets Girard's linear logic: So this was secretly a successful quantum logic all along!

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices $\mathrm{FDVect}_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$		
proposition	**conjunction**	**disjunction**



Logical Methods in Computer Science
Volume 18, Issue 3, 2022, pp. 28:1–28:44
https://lmcs.episciences.org/

Submitted
Published

LINEAR DEPENDENT TYPE THEORY FOR QUANTUM PROGRAMMING LANGUAGES

PENG FU [a], KOHEI KISHIDA [b], AND PETER SELINGER [c]

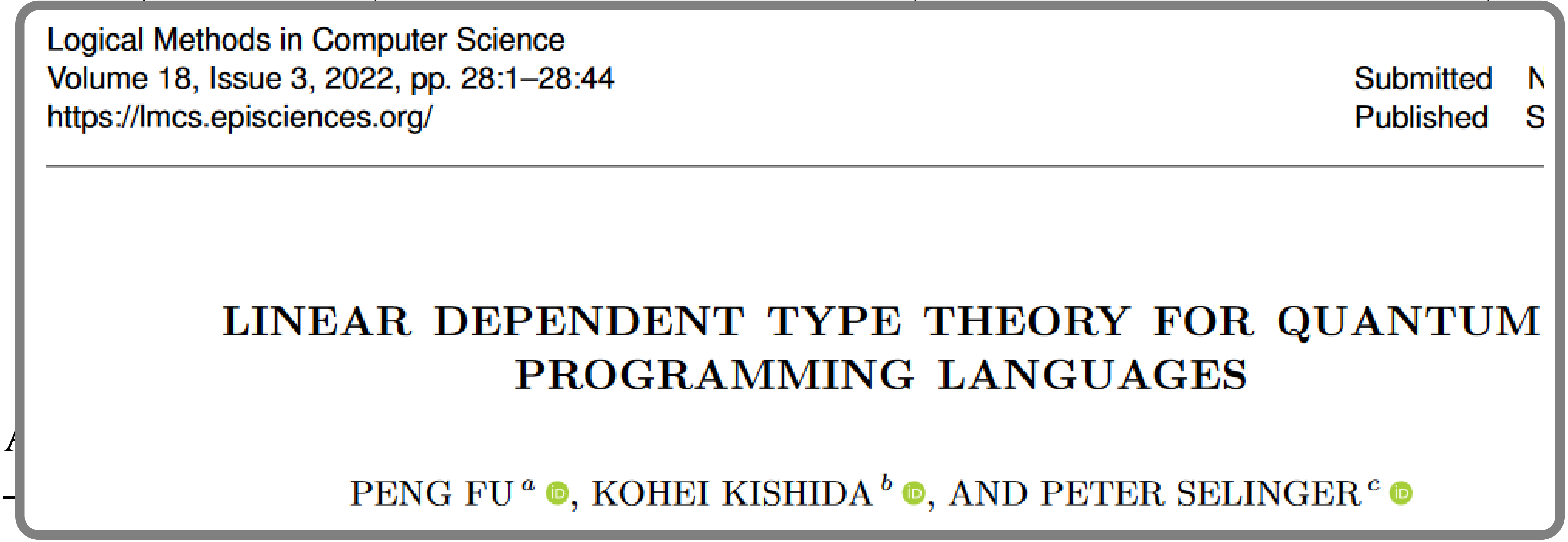
Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) linear logic: So this was secretly a successful quantum logic all along!

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices $\mathrm{FDVect}_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$

proposition	**conjunction**	**disjunction**



Logical Methods in Computer Science
Volume 18, Issue 3, 2022, pp. 28:1–28:44
https://lmcs.episciences.org/

Submitted N
Published S

LINEAR DEPENDENT TYPE THEORY FOR QUANTUM PROGRAMMING LANGUAGES

PENG FU [a], KOHEI KISHIDA [b], AND PETER SELINGER [c]

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

linear logic: So this was secretly a successful quantum logic all along!

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices $\mathrm{FDVect}_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$

proposition	**conjunction**	**disjunction**
$$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p"] \\ \mathcal{H} \end{tikzcd}$$	$$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1"'] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook, "p_2"] \\ & \mathcal{H} & \end{tikzcd}$$	$$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"'] & \mathrm{Span}\big(\mathcal{P}_1, \mathcal{P}_2\big) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook, "p_2"] \\ & \mathcal{H} & \end{tikzcd}$$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$ – these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

But it is still unsatisfactory as a type theory, notably lacking type-dependency.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices $\mathrm{FDVect}_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$		
proposition	**conjunction**	**disjunction**
$$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p"] \\ \mathcal{H} \end{tikzcd}$$	$$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1"'] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook', "p_2"] \\ & \mathcal{H} & \end{tikzcd}$$	$$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"'] & \mathrm{Span}\left(\mathcal{P}_1, \mathcal{P}_2\right) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook', "p_2"] \\ & \mathcal{H} & \end{tikzcd}$$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$ – these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

NB: A *linear*-dependent internal logic of $\mathrm{Vect}_{\mathrm{Set}}$ would *still* see BvN quantum logic.

Ironically, *enhancing* the internal linear logic of FDVect with type dependency, (to see the slices $\mathrm{FDVect}_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$

proposition	**conjunction**	**disjunction**
$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p"] \\ \mathcal{H} \end{tikzcd}$	$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1"'] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook, "p_2"] \\ & \mathcal{H} & \end{tikzcd}$	$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"'] & \mathrm{Span}\left(\mathcal{P}_1, \mathcal{P}_2\right) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook, "p_2"] \\ & \mathcal{H} & \end{tikzcd}$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \;\vdash\; \mathbb{C}^n : \mathrm{LinType}$ – these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

NB: A *linear*-dependent internal logic of $\mathrm{Vect}_{\mathrm{Set}}$ would *still* see BvN quantum logic.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

(to see the slices $\mathrm{FDVect}_{/\mathcal{H}}$) would immediately recover BvN's quantum logic:

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$		
proposition	**conjunction**	**disjunction**
$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p"] \\ \mathcal{H} \end{tikzcd}$	$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1"'] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook, "p_2"] \\ & \mathcal{H} & \end{tikzcd}$	$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"'] & \mathrm{Span}\big(\mathcal{P}_1, \mathcal{P}_2\big) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook, "p_2"] \\ & \mathcal{H} & \end{tikzcd}$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$
– these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

NB: A *linear*-dependent internal logic of $\mathrm{Vect}_{\mathrm{Set}}$ would *still* see BvN quantum logic.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$

proposition	**conjunction**	**disjunction**
$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p"] \\ \mathcal{H} \end{tikzcd}$	$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1"'] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook', "p_2"] \\ & \mathcal{H} & \end{tikzcd}$	$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"'] & \mathrm{Span}\big(\mathcal{P}_1, \mathcal{P}_2\big) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook', "p_2"] \\ & \mathcal{H} & \end{tikzcd}$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$
– these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

NB: A *linear*-dependent internal logic of $\mathrm{Vect}_{\mathrm{Set}}$ would *still* see BvN quantum logic.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian $(\times)$ and "external" $(\otimes)$ tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

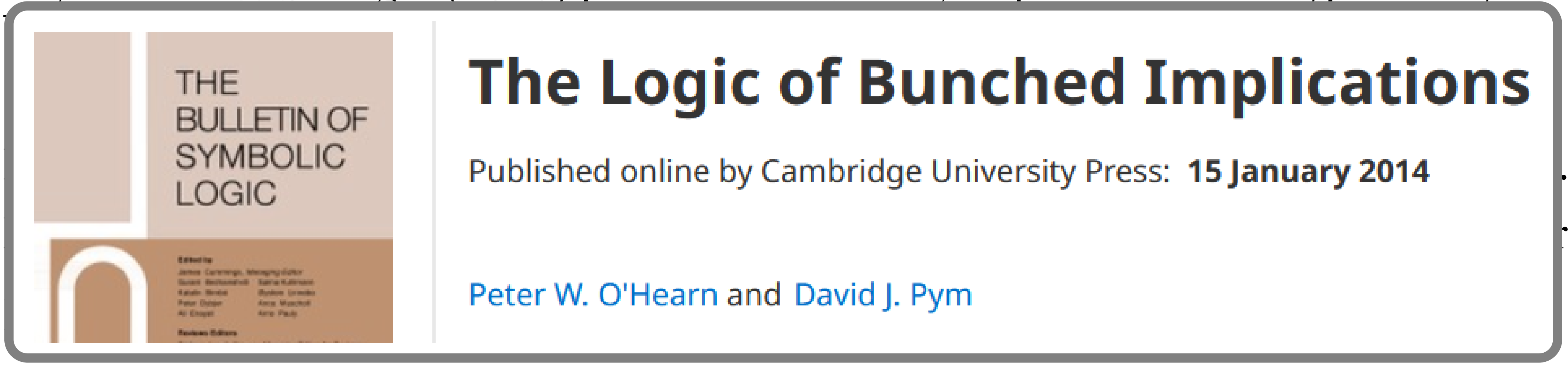
O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but

Internal logic in $\mathrm{FDVect}_{/\mathcal{H}}$

proposition	**conjunction**	**disjunction**
$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p"] \\ \mathcal{H} \end{tikzcd}$	$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1"] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook', "p_2"'] \\ & \mathcal{H} & \end{tikzcd}$	$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"] & \mathrm{Span}(\mathcal{P}_1, \mathcal{P}_2) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook', "p_2"'] \\ & \mathcal{H} & \end{tikzcd}$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$ – these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory



THE BULLETIN OF SYMBOLIC LOGIC

The Logic of Bunched Implications

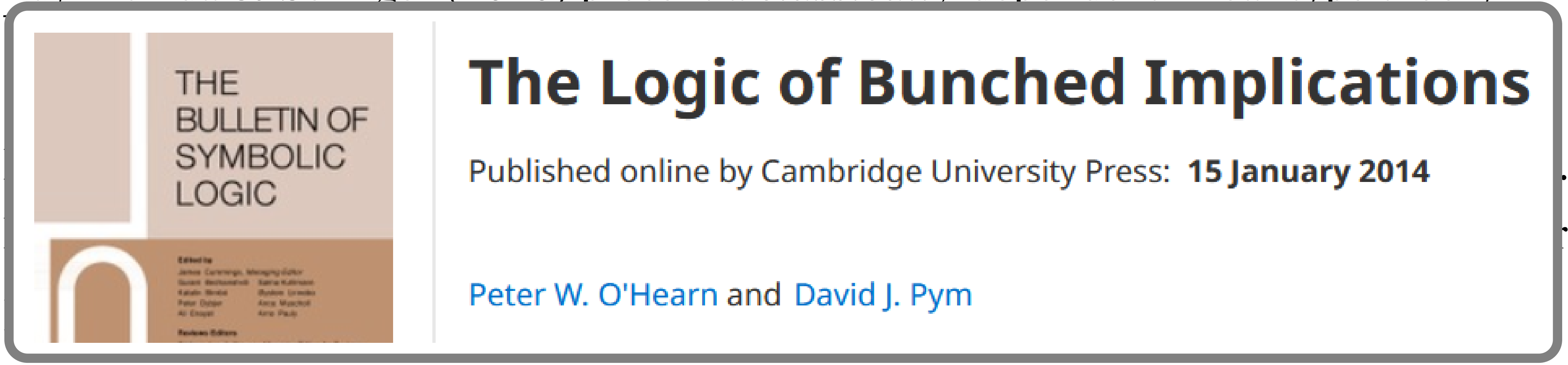
Peter W. O'Hearn and David J. Pym

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but

proposition	**conjunction**	**disjunction**
$$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p"] \\ \mathcal{H} \end{tikzcd}$$	$$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1"'] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook, "p_2"] \\ & \mathcal{H} & \end{tikzcd}$$	$$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"'] & \mathrm{Span}(\mathcal{P}_1, \mathcal{P}_2) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook, "p_2"] \\ & \mathcal{H} & \end{tikzcd}$$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$ – these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory



THE
BULLETIN OF
SYMBOLIC
LOGIC

The Logic of Bunched Implications

Published online by Cambridge University Press: **15 January 2014**

Peter W. O'Hearn and David J. Pym

O'Hearn & Pym (1999) already experimented with such “bunched” contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

proposition	**conjunction**	**disjunction**
$$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p" description] \\ \mathcal{H} \end{tikzcd}$$	$$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1" description] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2" description] & \mathcal{P}_2 \arrow[dl, hook, "p_2" description] \\ & \mathcal{H} & \end{tikzcd}$$	$$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1" description] & \mathrm{Span}\big(\mathcal{P}_1, \mathcal{P}_2\big) \arrow[d, hook, "p_1 \vee p_2" description] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook, "p_2" description] \\ & \mathcal{H} & \end{tikzcd}$$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \;\vdash\; \mathbb{C}^n : \mathrm{LinType}$
– these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

NB: A *linear*-dependent internal logic of $\mathrm{Vect}_{\mathrm{Set}}$ would *still* see BvN quantum logic.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

$$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p"] \\ \mathcal{H} \end{tikzcd}$$	$$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1"'] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook, "p_2"] \\ & \mathcal{H} & \end{tikzcd}$$	$$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"'] & \mathrm{Span}\left(\mathcal{P}_1, \mathcal{P}_2\right) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook, "p_2"] \\ & \mathcal{H} & \end{tikzcd}$$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$ – these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

NB: A *linear*-dependent internal logic of $\mathrm{Vect}_{\mathrm{Set}}$ would *still* see BvN quantum logic.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and “external” ($\otimes$) tensor.

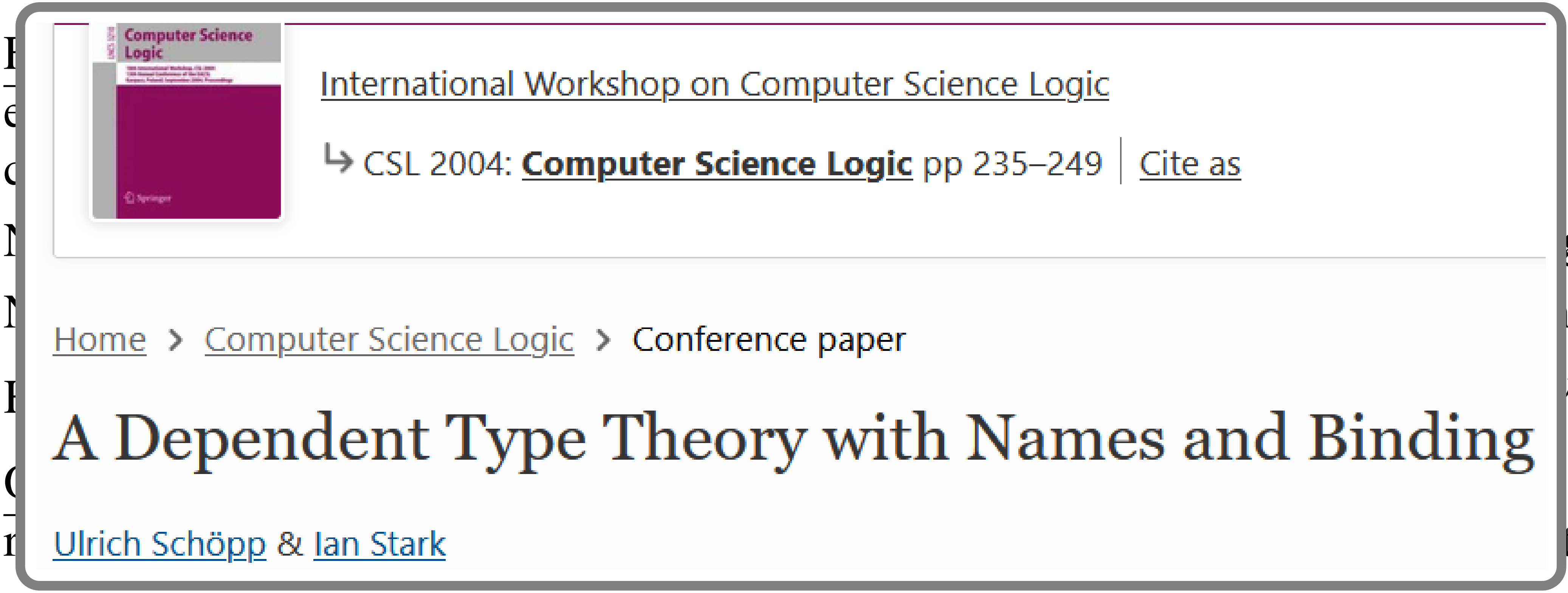
Hence full dependency requires handling nested contexts like $X \times \left(\mathcal{H} \otimes \mathcal{V}\right) \times \mathcal{W}$.

O’Hearn & Pym (1999) already experimented with such “bunched” contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory

$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p" description] \\ \mathcal{H} \end{tikzcd}$	$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1" description] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2" description] & \mathcal{P}_2 \arrow[dl, hook, "p_2" description] \\ & \mathcal{H} & \end{tikzcd}$	$\begin{tikzcd} & \mathcal{P}_1 \oplus \mathcal{P}_2 \arrow[d, two heads] & \\ \mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1" description] & \mathrm{Span}\left(\mathcal{P}_1, \mathcal{P}_2\right) \arrow[d, hook, "p_1 \vee p_2" description] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook, "p_2" description] \\ & \mathcal{H} & \end{tikzcd}$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$ – these ought to be interpreted as vector (Hilbert) *bundles*.



International Workshop on Computer Science Logic

↳ CSL 2004: **Computer Science Logic** pp 235–249 | Cite as

Home › Computer Science Logic › Conference paper

A Dependent Type Theory with Names and Binding

Ulrich Schöpp & Ian Stark

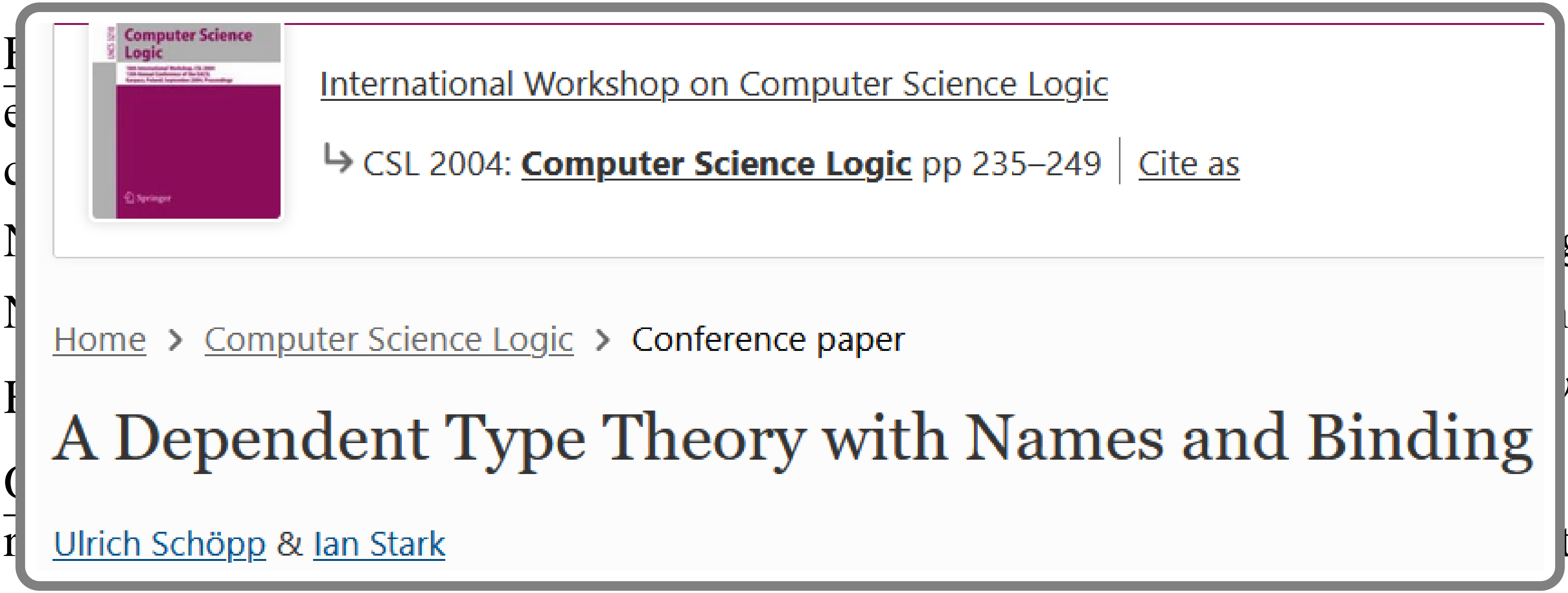
Schöpp & Stark (2004) make progress towards bunched dependent type theory

$$\begin{tikzcd}
\mathcal{P} \arrow[d, hook, "p"] \\
\mathcal{H}
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{P}_1 \arrow[dr, hook, "p_1"'] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2"] & \mathcal{P}_2 \arrow[dl, hook, "p_2"] \\
& \mathcal{H} &
\end{tikzcd}$$

$$\begin{tikzcd}
& {} \arrow[d, two heads] & \\
\mathcal{P}_1 \arrow[ur] \arrow[dr, hook, "p_1"'] & \mathrm{Span}(\mathcal{P}_1, \mathcal{P}_2) \arrow[d, hook, "p_1 \vee p_2"] & \mathcal{P}_2 \arrow[ul] \arrow[dl, hook, "p_2"] \\
& \mathcal{H} &
\end{tikzcd}$$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$ – these ought to be interpreted as vector (Hilbert) *bundles*.



International Workshop on Computer Science Logic

↳ CSL 2004: **Computer Science Logic** pp 235–249 | Cite as

Home › Computer Science Logic › Conference paper

A Dependent Type Theory with Names and Binding

Ulrich Schöpp & Ian Stark

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

$$\begin{tikzcd}
\mathcal{P} \arrow[d, hook, "p" description] \\
\mathcal{H}
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{P}_1 \arrow[dr, hook, "p_1" description] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2" description] & \mathcal{P}_2 \arrow[dl, hook', "p_2" description] \\
& \mathcal{H} &
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{P}_1 \arrow[dr, hook, "p_1" description] & \mathrm{Span}\left(\mathcal{P}_1, \mathcal{P}_2\right) \arrow[d, hook, "p_1 \vee p_2" description] & \mathcal{P}_2 \arrow[dl, hook', "p_2" description] \\
& \mathcal{H} &
\end{tikzcd}$$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$ – these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

NB: A *linear*-dependent internal logic of $\mathrm{Vect}_{\mathrm{Set}}$ would *still* see BvN quantum logic.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \left(\mathcal{H} \otimes \mathcal{V}\right) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

$$\begin{tikzcd} \mathcal{P} \arrow[d, hook, "p" description] \\ \mathcal{H} \end{tikzcd}$$	$$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1" description] & \mathcal{P}_1 \cap \mathcal{P}_2 \arrow[l] \arrow[r] \arrow[d, hook, "p_1 \wedge p_2" description] & \mathcal{P}_2 \arrow[dl, hook, "p_2" description] \\ & \mathcal{H} & \end{tikzcd}$$	$$\begin{tikzcd} \mathcal{P}_1 \arrow[dr, hook, "p_1" description] & \mathrm{Span}\left(\mathcal{P}_1, \mathcal{P}_2\right) \arrow[d, hook, "p_1 \vee p_2" description] & \mathcal{P}_2 \arrow[dl, hook, "p_2" description] \\ & \mathcal{H} & \end{tikzcd}$$

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$
– these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

NB: A *linear*-dependent internal logic of $\mathrm{Vect}_{\mathrm{Set}}$ would *still* see BvN quantum logic.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and “external” ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \left(\mathcal{H} \otimes \mathcal{V}\right) \times \mathcal{W}$.

O’Hearn & Pym (1999) already experimented with such “bunched” contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$
– these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

NB: A *linear*-dependent internal logic of $\mathrm{Vect}_{\mathrm{Set}}$ would *still* see BvN quantum logic.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Also, we need classically-dependent linear types, eg. $n : \mathbb{N} \vdash \mathbb{C}^n : \mathrm{LinType}$ – these ought to be interpreted as vector (Hilbert) *bundles*.

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

NB: A *linear*-dependent internal logic of $\mathrm{Vect}_{\mathrm{Set}}$ would *still* see BvN quantum logic.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Fu, Kishida & Selinger (2020) present a *classically*-dependent linear type theory explicitly as a practical quantum programming language (proto-`Quipper`) with categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathsf{Set}}$.

NB: A *linear*-dependent internal logic of $\mathrm{Vect}_{\mathsf{Set}}$ would *still* see BvN quantum logic.

NB: $\mathrm{Vect}_{\mathsf{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

categorical semantics in (notably) vector bundles over sets: $\mathrm{Vect}_{\mathrm{Set}}$.

NB: A *linear*-dependent internal logic of $\mathrm{Vect}_{\mathrm{Set}}$ would *still* see BvN quantum logic.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and “external” ($\otimes$) tensor.

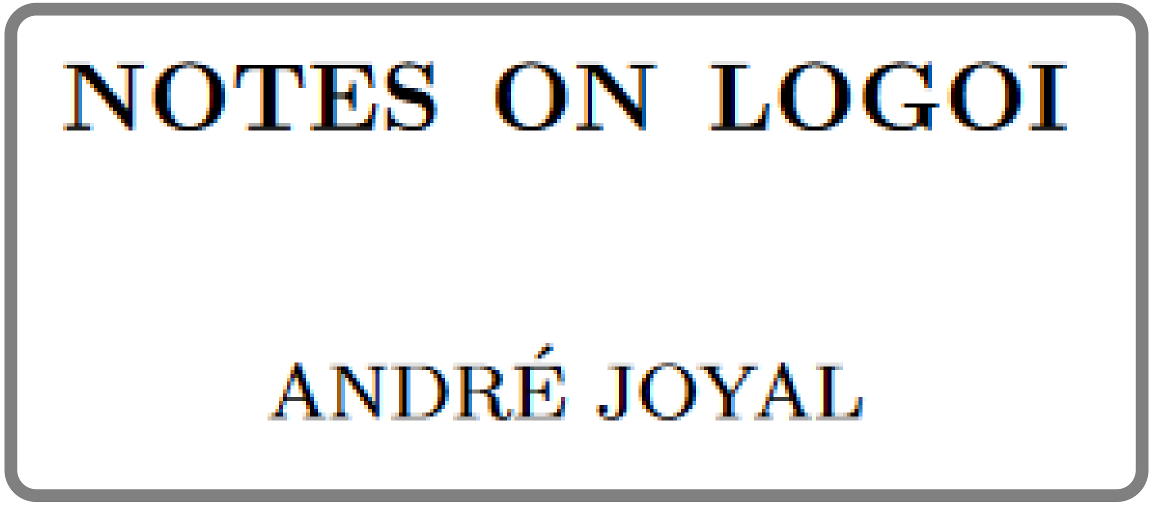
Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O’Hearn & Pym (1999) already experimented with such “bunched” contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!



NOTES ON LOGOI

ANDRÉ JOYAL

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret a modal, namely *infinitesimally cohesive* Homotopy Type Theory (`HoTT`),

NB: Vect$_{\text{Set}}$ ca
nd “external” ($\otimes$) tensor.

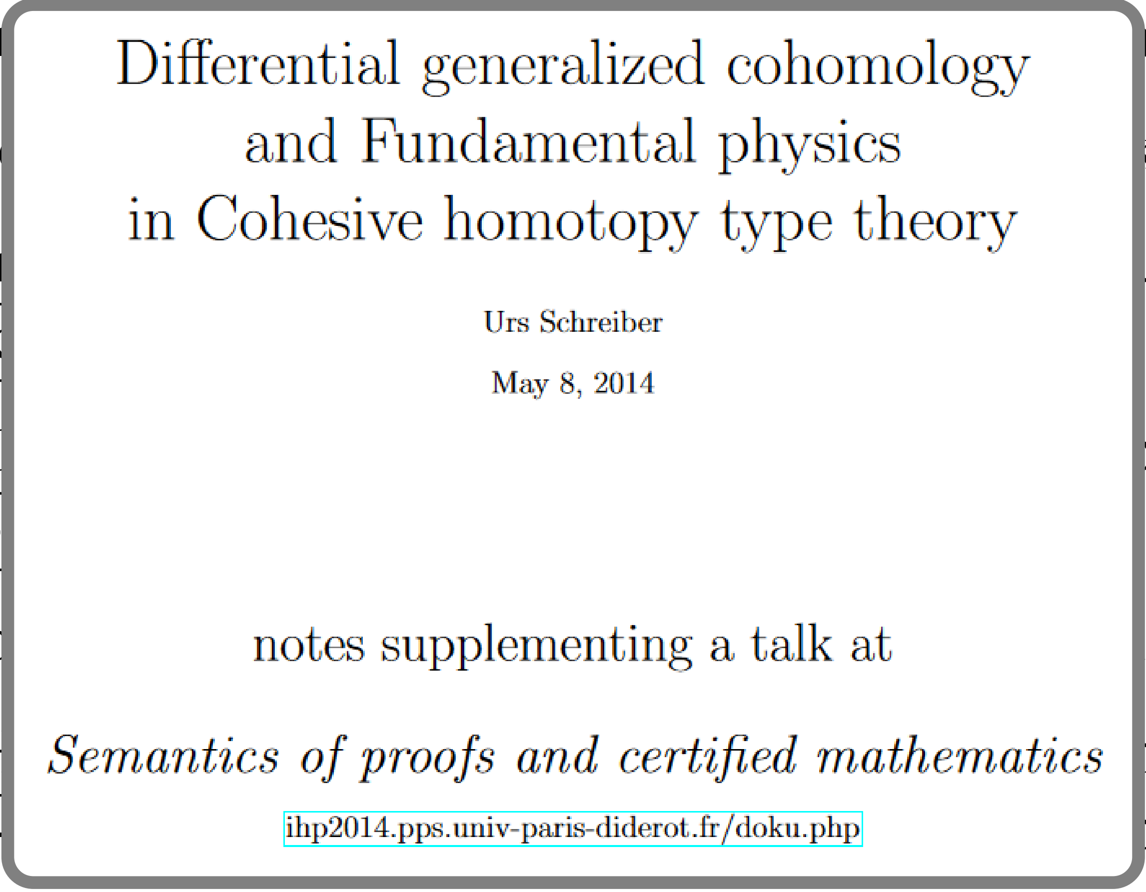
Hence full dep
ke $X \times (\mathcal{H} \otimes \mathcal{V}) \times \mathcal{W}$.

O’Hearn & Py
unched” contexts, but
ran into techni
assical and linear unit.

Schöpp & Star
pendent type theory
but do not reso

The key hint fo
n higher topos theory:

Joyal (2008), H
nd linear *homotopy*-types:
remarkably: ∞-
pes are again ∞-toposes!



Differential generalized cohomology
and Fundamental physics
in Cohesive homotopy type theory

Urs Schreiber

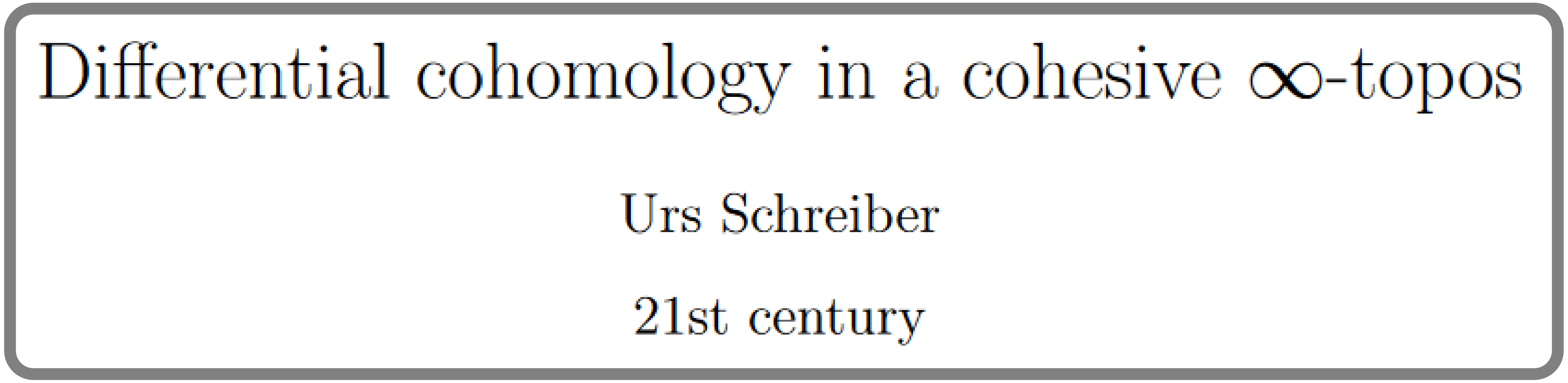
May 8, 2014

notes supplementing a talk at

Semantics of proofs and certified mathematics

ihp2014.pps.univ-paris-diderot.fr/doku.php

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret a modal, namely *infinitesimally cohesive* Homotopy Type Theory (`HoTT`),



Differential cohomology in a cohesive ∞-topos

Urs Schreiber

21st century

NB: $\mathrm{Vect}_{\mathrm{Set}}$ ca nd “external” ($\otimes$) tensor.

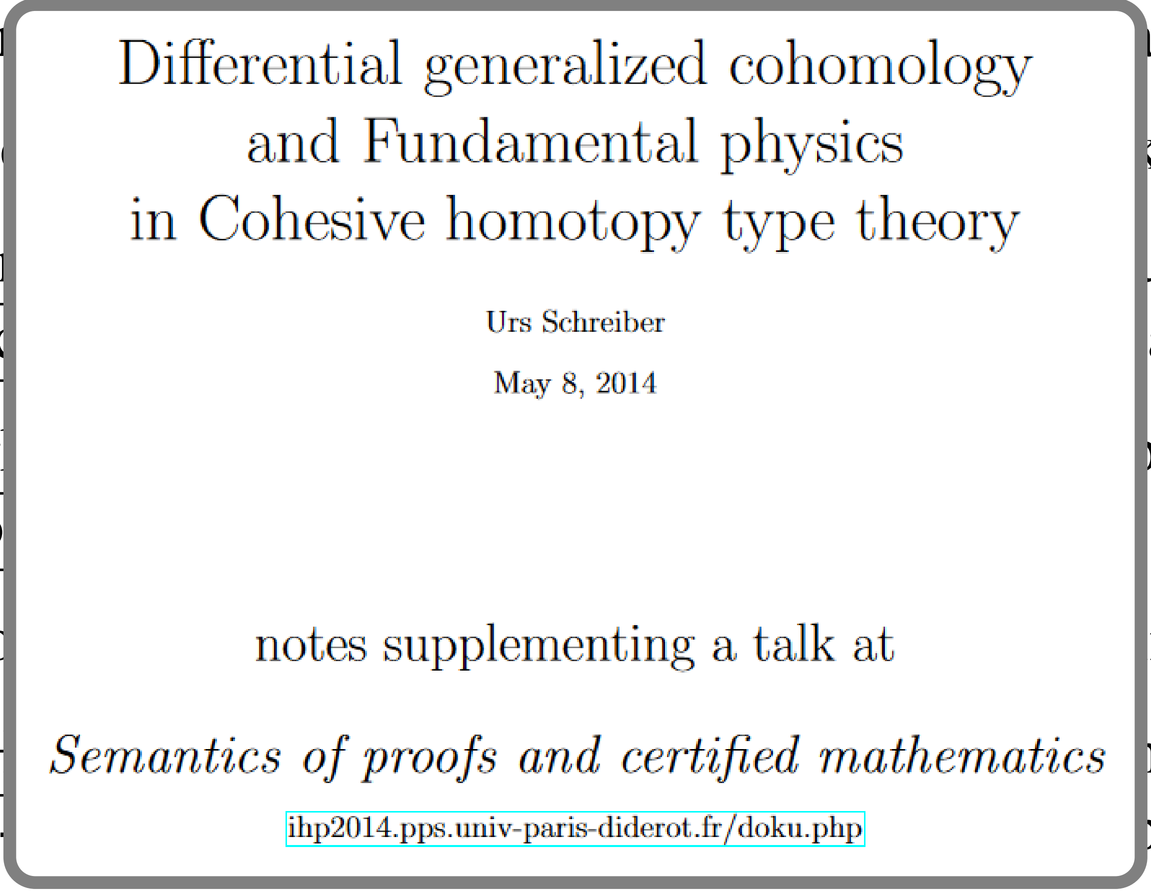
Hence full dep ke $X \times (\mathcal{H} \otimes \mathcal{V}) \times \mathcal{W}$.

O’Hearn & Py unched” contexts, but
ran into technic assical and linear unit.

Schöpp & Star pendent type theory
but do not reso

The key hint fo n higher topos theory:

Joyal (2008), H nd linear *homotopy*-types:
remarkably: ∞- pes are again ∞-toposes!



Differential generalized cohomology
and Fundamental physics
in Cohesive homotopy type theory

Urs Schreiber

May 8, 2014

notes supplementing a talk at

Semantics of proofs and certified mathematics

ihp2014.pps.univ-paris-diderot.fr/doku.php

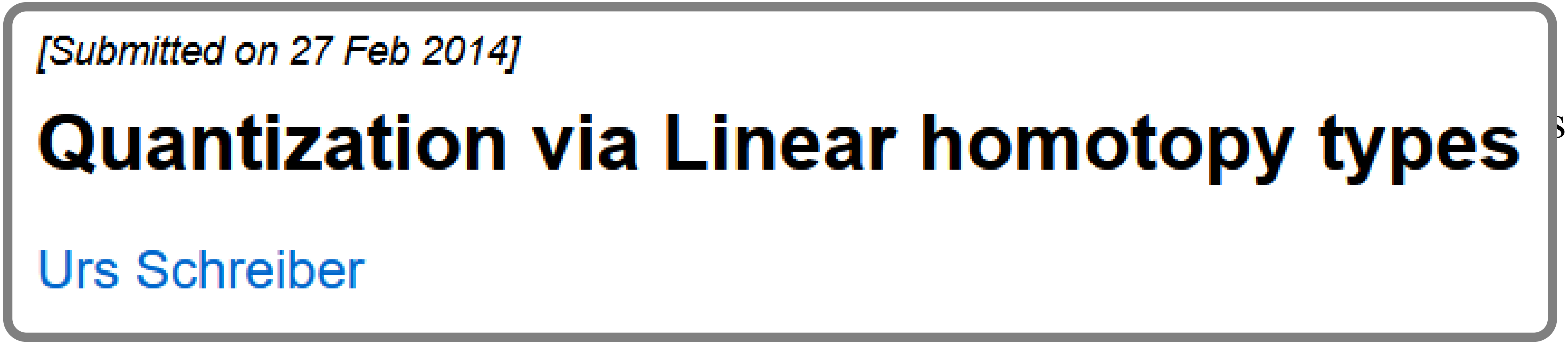
S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret
a modal, namely *infinitesimally cohesive* Homotopy Type Theory (`HoTT`), and
S. (Feb 2014) suggests that classical `HoTT` should have a linear extension with
multiplicative conjunction $\otimes$ interpreted as the (external) smash product of spectra.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \left(\mathcal{H} \otimes \mathcal{V}\right) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.



[Submitted on 27 Feb 2014]

Quantization via Linear homotopy types

Urs Schreiber

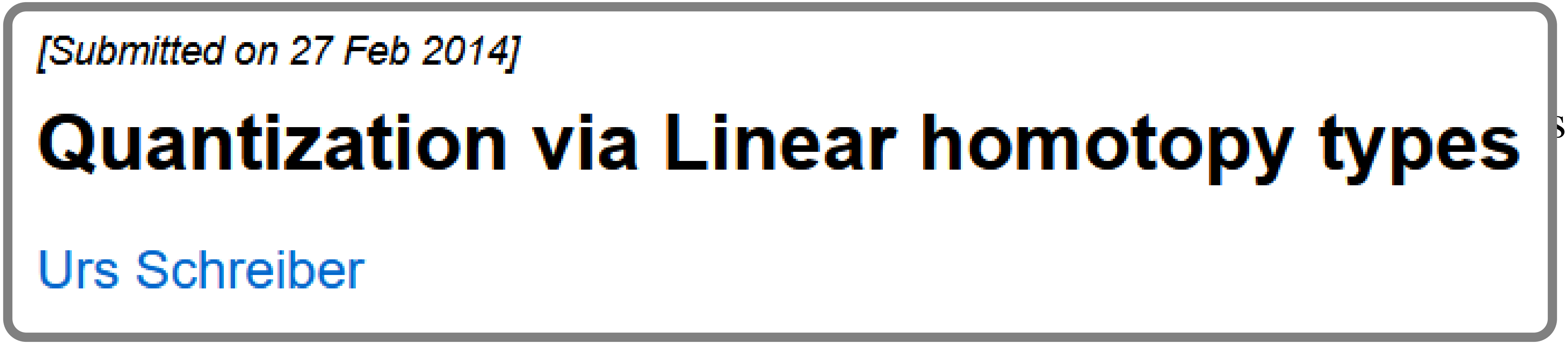
S. (Feb 2014) suggests that classical HoTT should have a linear extension with multiplicative conjunction $\otimes$ interpreted as the (external) smash product of spectra.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times (\mathcal{H} \otimes \mathcal{V}) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.



[Submitted on 27 Feb 2014]

Quantization via Linear homotopy types

Urs Schreiber

S. (Feb 2014) suggests that classical `HoTT` should have a linear extension with multiplicative conjunction $\otimes$ interpreted as the (external) smash product of spectra.
Demonstrates potential relevance as a quantum language – but gives no formal syntax.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret a modal, namely *infinitesimally cohesive* Homotopy Type Theory (`HoTT`), and S. (Feb 2014) suggests that classical `HoTT` should have a linear extension with multiplicative conjunction $\otimes$ interpreted as the (external) smash product of spectra. Demonstrates potential relevance as a quantum language – but gives no formal syntax.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret a modal, namely *infinitesimally cohesive* Homotopy Type Theory (`HoTT`), and S. (Feb 2014) suggests that classical `HoTT` should have a linear extension with multiplicative conjunction $\otimes$ interpreted as the (external) smash product of spectra. Demonstrates potential relevance as a quantum language – but gives no formal syntax.

Riley, Fister & Licata (2021) propose a syntax for such infinitesimally cohesive `HoTT`.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian $(\times)$ and "external" $(\otimes)$ tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joy ypes:
rem ses!

S. (ret
a m
S. (
mul ra.
Der ntax.


[Submitted on 8 Feb 2021]

Synthetic Spectra via a Monadic and Comonadic Modality

Mitchell Riley, Eric Finster, Daniel R. Licata

Riley, Fister & Licata (2021) propose a syntax for such infinitesimally cohesive `HoTT`.

NB: $\mathrm{Vect}_{\mathrm{Set}}$ carries *two* monoidal structures: cartesian ($\times$) and "external" ($\otimes$) tensor.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret a modal, namely *infinitesimally cohesive* Homotopy Type Theory (`HoTT`), and S. (Feb 2014) suggests that classical `HoTT` should have a linear extension with multiplicative conjunction $\otimes$ interpreted as the (external) smash product of spectra. Demonstrates potential relevance as a quantum language – but gives no formal syntax.

Riley, Fister & Licata (2021) propose a syntax for such infinitesimally cohesive `HoTT`.

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret a modal, namely *infinitesimally cohesive* Homotopy Type Theory (`HoTT`), and S. (Feb 2014) suggests that classical `HoTT` should have a linear extension with multiplicative conjunction ⊗ interpreted as the (external) smash product of spectra. Demonstrates potential relevance as a quantum language – but gives no formal syntax.

Riley, Fister & Licata (2021) propose a syntax for such infinitesimally cohesive `HoTT`.

Riley (2022) realizes that infinitesimal cohesion allows to fix the bunching of ⊗:

Hence full dependency requires handling nested contexts like $X \times \big(\mathcal{H} \otimes \mathcal{V}\big) \times \mathcal{W}$.

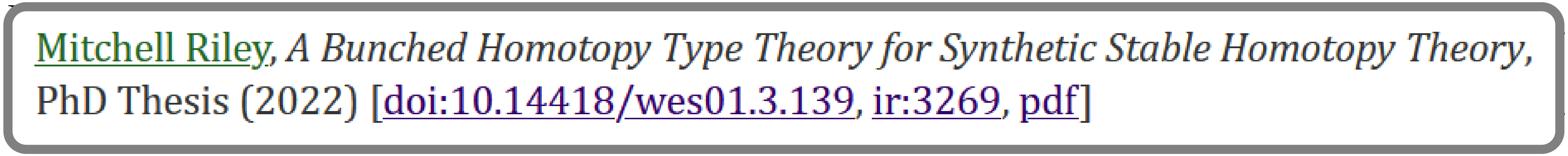
O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret a modal, namely *infinitesimally cohesive* Homotopy Type Theory (`HoTT`), and S. (Feb 2014) suggests that classical `HoTT` should have a linear extension with multiplicative conjunction ⊗ interpreted as the (external) smash product of spectra.



Mitchell Riley, *A Bunched Homotopy Type Theory for Synthetic Stable Homotopy Theory*, PhD Thesis (2022) [doi:10.14418/wes01.3.139, ir:3269, pdf]

Riley (2022) realizes that infinitesimal cohesion allows to fix the bunching of ⊗:

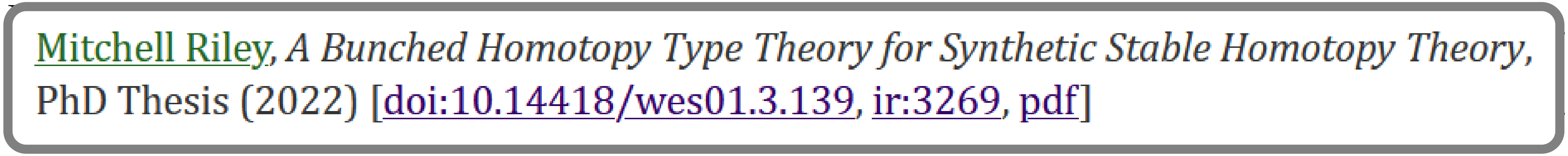
O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret a modal, namely *infinitesimally cohesive* Homotopy Type Theory (`HoTT`), and S. (Feb 2014) suggests that classical `HoTT` should have a linear extension with multiplicative conjunction $\otimes$ interpreted as the (external) smash product of spectra.



Mitchell Riley, *A Bunched Homotopy Type Theory for Synthetic Stable Homotopy Theory*, PhD Thesis (2022) [doi:10.14418/wes01.3.139, ir:3269, pdf]

Riley (2022) realizes that infinitesimal cohesion allows to fix the bunching of $\otimes$: the resulting bunched & infinitesimally cohesive homotopy type theory seems to finally deal with all the above technical issues.

O'Hearn & Pym (1999) already experimented with such "bunched" contexts, but ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret a modal, namely *infinitesimally cohesive* Homotopy Type Theory (`HoTT`), and S. (Feb 2014) suggests that classical `HoTT` should have a linear extension with multiplicative conjunction $\otimes$ interpreted as the (external) smash product of spectra. Demonstrates potential relevance as a quantum language – but gives no formal syntax.

Riley, Fister & Licata (2021) propose a syntax for such infinitesimally cohesive `HoTT`.

Riley (2022) realizes that infinitesimal cohesion allows to fix the bunching of $\otimes$: the resulting bunched & infinitesimally cohesive homotopy type theory seems to finally deal with all the above technical issues.

ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret a modal, namely *infinitesimally cohesive* Homotopy Type Theory (`HoTT`), and S. (Feb 2014) suggests that classical `HoTT` should have a linear extension with multiplicative conjunction $\otimes$ interpreted as the (external) smash product of spectra. Demonstrates potential relevance as a quantum language – but gives no formal syntax.

Riley, Fister & Licata (2021) propose a syntax for such infinitesimally cohesive `HoTT`.

Riley (2022) realizes that infinitesimal cohesion allows to fix the bunching of $\otimes$: the resulting bunched & infinitesimally cohesive homotopy type theory seems to finally deal with all the above technical issues.

CQTS (2023) understand this *Linear Homotopy Type Theory* (`LHoTT`) as the missing quantum certification language (close to `Quipper`)...

ran into technical problems, eg. unsound identification of classical and linear unit.

Schöpp & Stark (2004) make progress towards bunched dependent type theory
but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types:
remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret

Effective Quantum Certification via Linear Homotopy Types

David J. Myers djm10080@nyu.edu
Mitchell Riley mvr9774@nyu.edu
Hisham Sati hsati@nyu.edu
Urs Schreiber us13@nyu.edu

CENTER FOR QUANTUM & TOPOLOGICAL SYSTEMS
New York University
Abu Dhabi, UAE

CQTS (2023) understand this *Linear Homotopy Type Theory* (`LHoTT`)
as the missing quantum certification language (close to `Quipper`)...

Schöpp & Stark (2004) make progress towards bunched dependent type theory but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret

Effective Quantum Certification via Linear Homotopy Types

David J. Myers djm10080@nyu.edu
Mitchell Riley mvr9774@nyu.edu
Hisham Sati hsati@nyu.edu
Urs Schreiber us13@nyu.edu

CENTER FOR QUANTUM & TOPOLOGICAL SYSTEMS
New York University
Abu Dhabi, UAE

CQTS (2023) understand this *Linear Homotopy Type Theory* (`LHoTT`) as the missing quantum certification language (close to `Quipper`)...

but do not resolve these problems.

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret

Effective Quantum Certification via Linear Homotopy Types

David J. Myers
djm10080@nyu.edu

Mitchell Riley
mvr9774@nyu.edu

Hisham Sati
hsati@nyu.edu

Urs Schreiber
us13@nyu.edu

CENTER FOR
QUANTUM &
TOPOLOGICAL
SYSTEMS

New York University
Abu Dhabi, UAE

CQTS (2023) understand this *Linear Homotopy Type Theory* (`LHoTT`)
as the missing quantum certification language (close to `Quipper`)...

The key hint for how to progress came from developments in higher topos theory:

Joyal (2008), Hoyois (2016) show unification of classical and linear *homotopy*-types: remarkably: ∞-categories of *bundles* of linear homotopy types are again ∞-toposes!

S. (2013), §4.1.2; S. (May 2014) points out that bundles of *spectra* should interpret

Effective Quantum Certification via Linear Homotopy Types

David J. Myers djm10080@nyu.edu
Mitchell Riley mvr9774@nyu.edu
Hisham Sati hsati@nyu.edu
Urs Schreiber us13@nyu.edu

CENTER FOR QUANTUM & TOPOLOGICAL SYSTEMS
New York University
Abu Dhabi, UAE

CQTS (2023) understand this *Linear Homotopy Type Theory* (`LHoTT`)
as the missing quantum certification language (close to `Quipper`)...

Our Solution

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

$\exists$ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
via a bireflective modality ♮ exhibiting linear extension of classical types

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
via a bireflective modality ♮ exhibiting linear extension of classical types

Idea: Frobenius monad on type system carves out classical types

classical types

bireflective among

all types

$$\begin{tikzcd}
\mathrm{ClType} \arrow[r, hook, "\iota" description] \arrow[r, phantom, shift left=3, "\perp"] \arrow[r, phantom, shift right=3, "\perp"] & \mathrm{Type} \arrow[l, shift right=6, "\beta"'] \arrow[l, shift left=6, "\beta"] \arrow[loop right, "\natural"]
\end{tikzcd}$$

classical modality

such that:

$$\begin{tikzcd}
\natural \arrow[r, "\varepsilon^{\natural}"] \arrow[dr, equal] & \mathrm{id}_{\mathscr{B}} \arrow[d, "\eta^{\natural}"] \\
& \natural
\end{tikzcd}$$

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

$\exists$ classical & linear dependent type theory

conservative over classical *Homotopy Type Theory* (`HoTT`)

via a bireflective modality $\natural$ exhibiting linear extension of classical types

Linear Homotopy Type Theory (`LHoTT`)

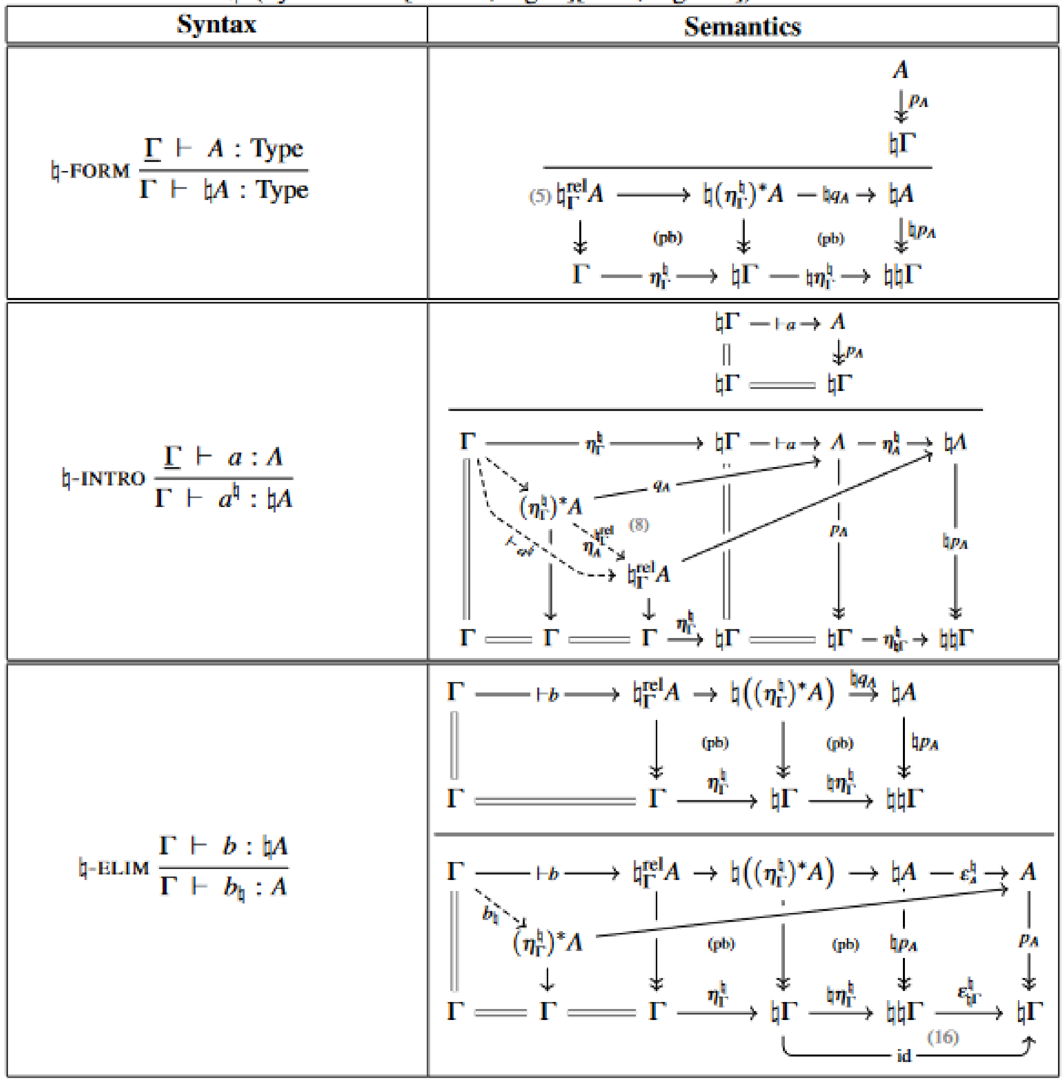
Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

$\exists$ classical & linear dependent type theory

conservative over classical *Homotopy Type Theory* (`HoTT`)

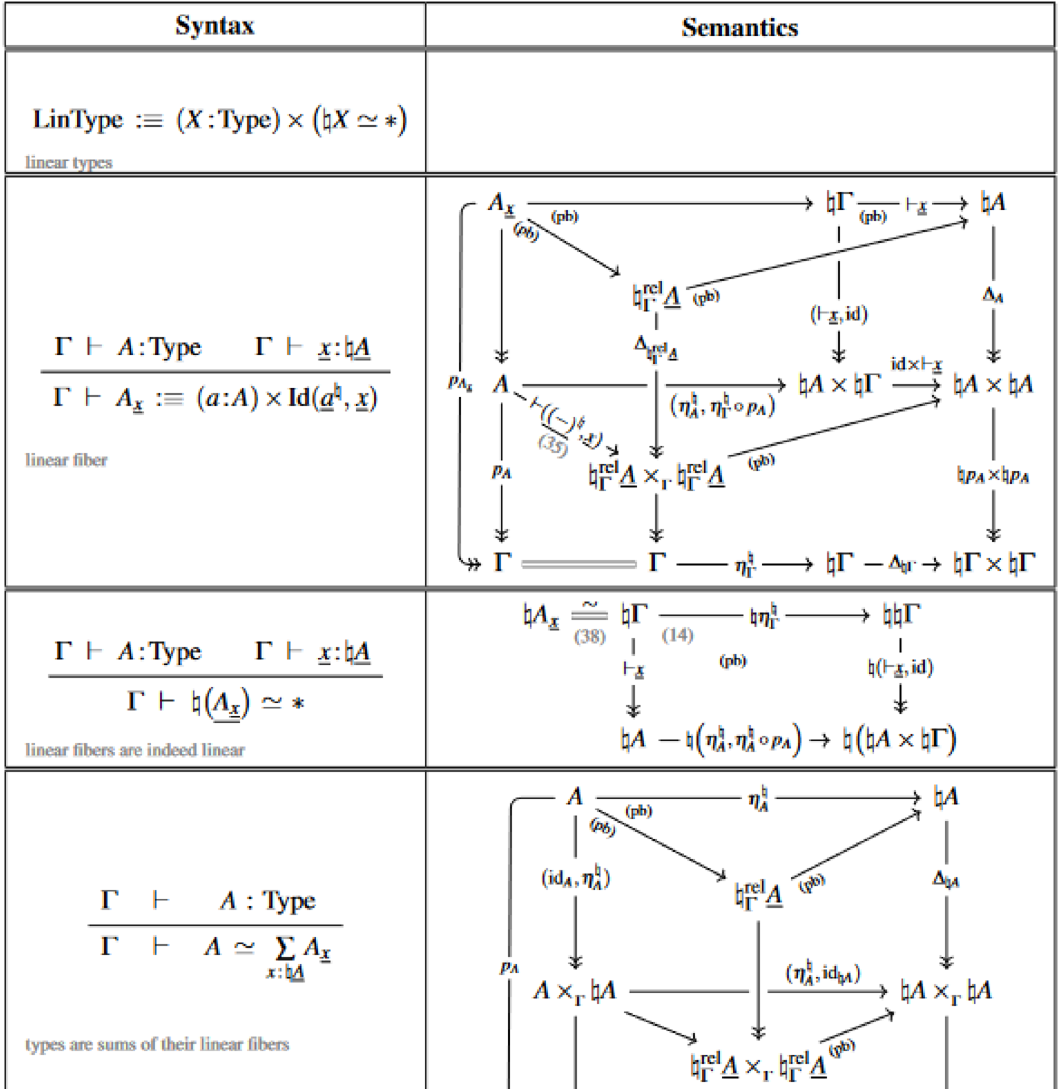
via a bireflective modality $\natural$ exhibiting linear extension of classical types

Inference rules for $\natural$. (Syntax from [RFL21, Fig. 2][Ri22, Fig. 1.2]).



Syntax	Semantics
$\natural$-FORM $\dfrac{\Gamma \vdash A : \mathrm{Type}}{\Gamma \vdash \natural A : \mathrm{Type}}$	$\begin{tikzcd} A \arrow[d, two heads, "p_A"] \\ \natural\Gamma \end{tikzcd}$ $\begin{tikzcd} (5)\ \natural^{\mathrm{rel}}_{\Gamma} A \arrow[r] \arrow[d, two heads] \arrow[dr, phantom, "(\mathrm{pb})"] & \natural(\eta^{\natural}_{\Gamma})^{*} A \arrow[r, "\natural q_A"] \arrow[d, two heads] \arrow[dr, phantom, "(\mathrm{pb})"] & \natural A \arrow[d, two heads, "\natural p_A"] \\ \Gamma \arrow[r, "\eta^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, "\natural\eta^{\natural}_{\Gamma}"] & \natural\natural\Gamma \end{tikzcd}$
$\natural$-INTRO $\dfrac{\Gamma \vdash a : A}{\Gamma \vdash a^{\natural} : \natural A}$	$\begin{tikzcd} \natural\Gamma \arrow[r, "\vdash a"] \arrow[d, equal] & A \arrow[d, two heads, "p_A"] \\ \natural\Gamma \arrow[r, equal] & \natural\Gamma \end{tikzcd}$ $\begin{tikzcd} \Gamma \arrow[rrr, "\eta^{\natural}_{\Gamma}"] \arrow[ddd, equal] \arrow[dr, dashed] \arrow[ddrr, dashed, bend right, "\vdash a^{\natural}"'] & & & \natural\Gamma \arrow[r, "\vdash a"] \arrow[ddd, equal] & A \arrow[r, "\eta^{\natural}_{A}"] \arrow[ddd, two heads, "p_A"] & \natural A \arrow[ddd, two heads, "\natural p_A"] \\ & (\eta^{\natural}_{\Gamma})^{*} A \arrow[urrr, "q_A"] \arrow[dr, dashed, "\eta^{\natural\mathrm{rel}}_{A}"] \arrow[dd] & (8) & & & \\ & & \natural^{\mathrm{rel}}_{\Gamma} A \arrow[uurrr] \arrow[d] & & & \\ \Gamma \arrow[r, equal] & \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\eta^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, equal] & \natural\Gamma \arrow[r, "\eta^{\natural}_{\natural\Gamma}"] & \natural\natural\Gamma \end{tikzcd}$
$\natural$-ELIM $\dfrac{\Gamma \vdash b : \natural A}{\Gamma \vdash b_{\natural} : A}$	$\begin{tikzcd} \Gamma \arrow[r, "\vdash b"] \arrow[d, equal] & \natural^{\mathrm{rel}}_{\Gamma} A \arrow[r] \arrow[d, two heads] \arrow[dr, phantom, "(\mathrm{pb})"] & \natural\big((\eta^{\natural}_{\Gamma})^{*} A\big) \arrow[r, "\natural q_A"] \arrow[d, two heads] \arrow[dr, phantom, "(\mathrm{pb})"] & \natural A \arrow[d, two heads, "\natural p_A"] \\ \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\eta^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, "\natural\eta^{\natural}_{\Gamma}"] & \natural\natural\Gamma \end{tikzcd}$ $\begin{tikzcd} \Gamma \arrow[rr, "\vdash b"] \arrow[dd, equal] \arrow[dr, dashed, "b_{\natural}"'] & & \natural^{\mathrm{rel}}_{\Gamma} A \arrow[r] \arrow[dd, two heads] \arrow[ddr, phantom, "(\mathrm{pb})"] & \natural\big((\eta^{\natural}_{\Gamma})^{*} A\big) \arrow[r] \arrow[dd, two heads] \arrow[ddr, phantom, "(\mathrm{pb})"] & \natural A \arrow[r, "\varepsilon^{\natural}_{A}"] \arrow[dd, two heads, "\natural p_A"] & A \arrow[dd, two heads, "p_A"] \\ & (\eta^{\natural}_{\Gamma})^{*} A \arrow[urrrr] \arrow[d] & & & & \\ \Gamma \arrow[r, equal] & \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\eta^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, "\natural\eta^{\natural}_{\Gamma}"] \arrow[rr, bend right, "\mathrm{id}"'] & \natural\natural\Gamma \arrow[r, "\varepsilon^{\natural}_{\natural\Gamma}"] \arrow[d, phantom, "(16)"] & \natural\Gamma \end{tikzcd}$

Linear types. (Syntax from [RFL21, pp. 24][Ri22, §2.1])



Syntax	Semantics
$\mathrm{LinType} :\equiv (X : \mathrm{Type}) \times (\natural X \simeq *)$ linear types	
$\dfrac{\Gamma \vdash A : \mathrm{Type} \qquad \Gamma \vdash \underline{x} : \natural A}{\Gamma \vdash A_{\underline{x}} :\equiv (a : A) \times \mathrm{Id}(\underline{a}^{\natural}, \underline{x})}$ linear fiber	$\begin{tikzcd} A_{\underline{x}} \arrow[rr, "(\mathrm{pb})"'] \arrow[dr, "(\mathrm{pb})"'] \arrow[dd, two heads] \arrow[dddd, two heads, bend right, "p_{A_{\underline{x}}}"'] & & \natural\Gamma \arrow[r, "\vdash \underline{x}", "(\mathrm{pb})"'] \arrow[dd, two heads, "(\vdash \underline{x}, \mathrm{id})"] & \natural A \arrow[dd, two heads, "\Delta_A"] \\ & \natural^{\mathrm{rel}}_{\Gamma} \underline{A} \arrow[urr, "(\mathrm{pb})"'] \arrow[dd, two heads, "\Delta_{\natural^{\mathrm{rel}}_{\Gamma} \underline{A}}"] & & \\ A \arrow[rr, "(\eta^{\natural}_{A}, \eta^{\natural}_{\Gamma} \circ p_A)"'] \arrow[dr, "\vdash((-)^{\natural}, \underline{x})", "(35)"'] \arrow[dd, two heads, "p_A"] & & \natural A \times \natural\Gamma \arrow[r, "\mathrm{id} \times \vdash \underline{x}"] & \natural A \times \natural A \arrow[dd, two heads, "\natural p_A \times \natural p_A"] \\ & \natural^{\mathrm{rel}}_{\Gamma} \underline{A} \times_{\Gamma} \natural^{\mathrm{rel}}_{\Gamma} \underline{A} \arrow[urr, "(\mathrm{pb})"'] \arrow[d, two heads] & & \\ \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\eta^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, "\Delta_{\natural\Gamma}"] & \natural\Gamma \times \natural\Gamma \end{tikzcd}$
$\dfrac{\Gamma \vdash A : \mathrm{Type} \qquad \Gamma \vdash \underline{x} : \natural A}{\Gamma \vdash \natural(A_{\underline{x}}) \simeq *}$ linear fibers are indeed linear	$\begin{tikzcd} \natural A_{\underline{x}} \arrow[r, equal, "\sim", "(38)"'] & \natural\Gamma \arrow[rr, "\natural\eta^{\natural}_{\Gamma}", "(14)"'] \arrow[d, two heads, "\vdash \underline{x}"] \arrow[drr, phantom, "(\mathrm{pb})"] & & \natural\natural\Gamma \arrow[d, two heads, "\natural(\vdash \underline{x}, \mathrm{id})"] \\ & \natural A \arrow[rr, "\natural(\eta^{\natural}_{A}, \eta^{\natural}_{A} \circ p_A)"] & & \natural\big(\natural A \times \natural\Gamma\big) \end{tikzcd}$
$\dfrac{\Gamma \vdash A : \mathrm{Type}}{\Gamma \vdash A \simeq \sum\limits_{\underline{x} : \natural A} A_{\underline{x}}}$ types are sums of their linear fibers	$\begin{tikzcd} A \arrow[rr, "\eta^{\natural}_{A}", "(\mathrm{pb})"'] \arrow[dr, "(\mathrm{pb})"'] \arrow[dd, two heads, "(\mathrm{id}_A, \eta^{\natural}_{A})"] & & \natural A \arrow[dd, two heads, "\Delta_{\natural A}"] \\ & \natural^{\mathrm{rel}}_{\Gamma} \underline{A} \arrow[ur, "(\mathrm{pb})"'] \arrow[d, two heads] & \\ A \times_{\Gamma} \natural A \arrow[rr, "(\eta^{\natural}_{A}, \mathrm{id}_{\natural A})"] \arrow[r] & \natural^{\mathrm{rel}}_{\Gamma} \underline{A} \times_{\Gamma} \natural^{\mathrm{rel}}_{\Gamma} \underline{A} \arrow[r, "(\mathrm{pb})"] & \natural A \times_{\Gamma} \natural A \end{tikzcd}$ p_A

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

$\exists$ classical & linear dependent type theory

conservative over classical *Homotopy Type Theory* (`HoTT`)

via a bireflective modality $\natural$ exhibiting linear extension of classical types

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

$\exists$ classical & linear dependent type theory

conservative over classical *Homotopy Type Theory* (`HoTT`)

and

verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]

(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

closed base type | B-dependent linear types

$$B : \mathrm{BType} \quad \vdash \quad \mathrm{LinType}_B \;:\equiv\; B \to \mathrm{LinType}$$

linear base change

closed function

$$f : B \to B' \quad \vdash \quad \begin{tikzcd} \mathrm{LinType}_B \arrow[r, shift left=3, "f_!"] \arrow[r, shift right=3, "f_*"'] & \mathrm{LinType}_{B'} \arrow[l, "f^*"'] \end{tikzcd}$$

$$E_{(-)} : \mathrm{LinType}_{B'} \quad \vdash \quad \big(f^* E_{(-)}\big) \;:\equiv\; b \mapsto E_{f(b)}$$

precomposition

$$E_{(-)} : \mathrm{LinType}_{B} \quad \vdash \quad \big(f_* E_{(-)}\big) \;:\equiv\; b' \mapsto \prod_{(b,p) : \mathrm{fib}_f(b')} E_b$$

dependent product

$$E_{(-)} : \mathrm{LinType}_{B} \quad \vdash \quad \big(f_! E_{(-)}\big) \;:\equiv\; b' \mapsto \bigvee_{(b,p) : \mathrm{fib}_f(b')} E_b$$

HIT cofiber of
zero inclusion in
dependent sum

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

$\exists$ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

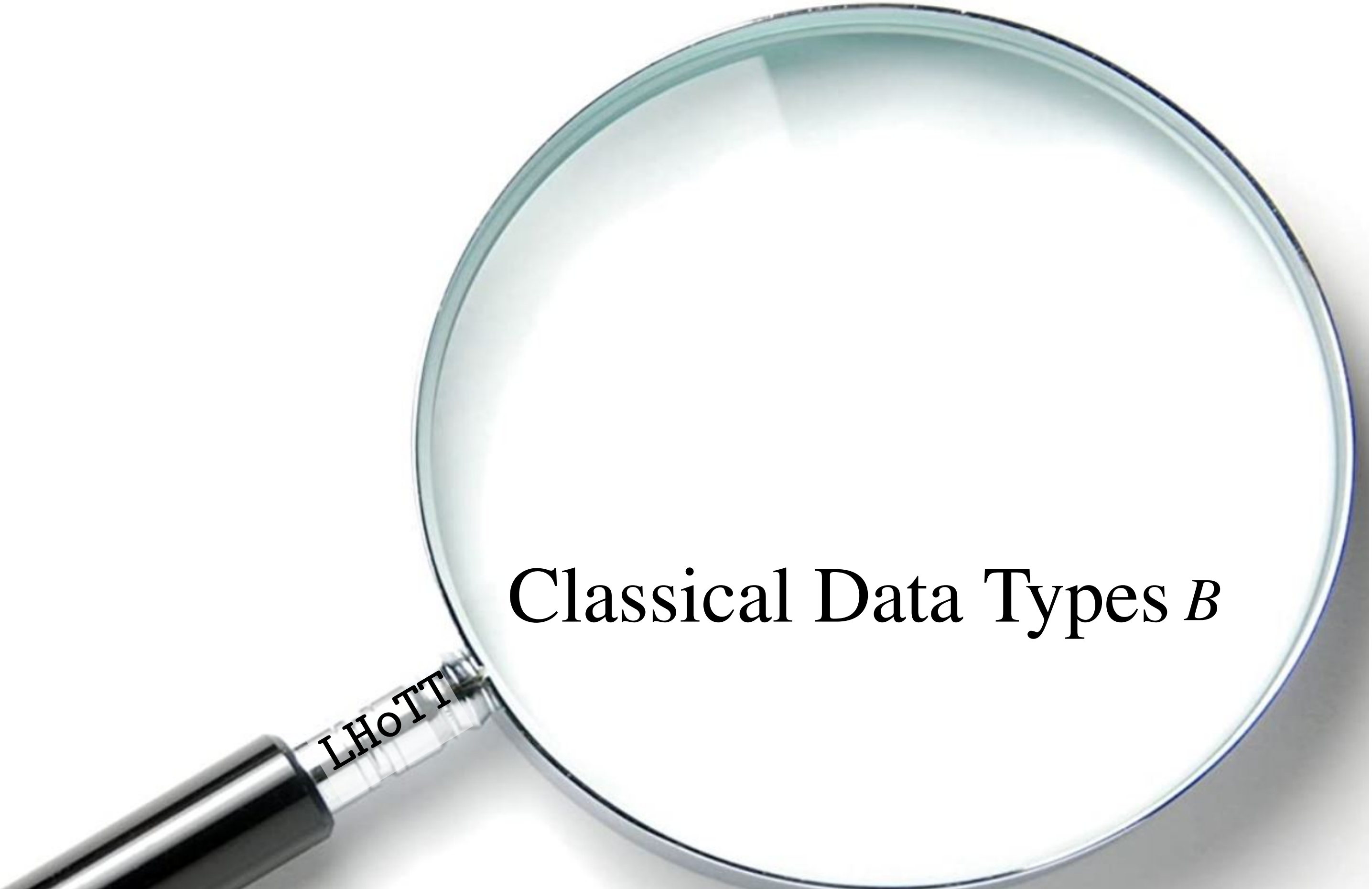
Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

`LHoTT` is like a quantum microscope for Classical Data Types B

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

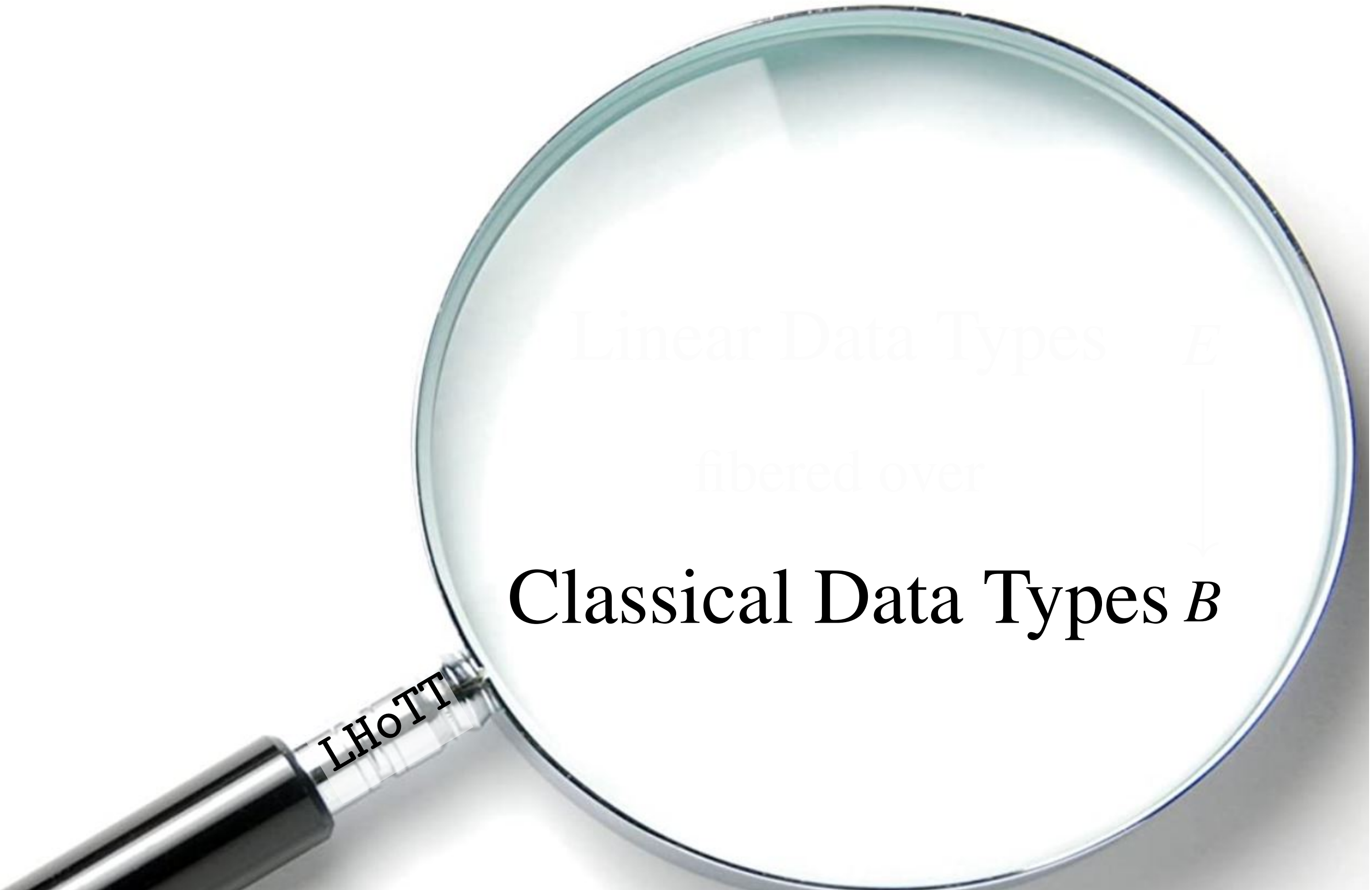
∃ classical & linear dependent type theory

conservative over classical *Homotopy Type Theory* (`HoTT`)

and

verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]

(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

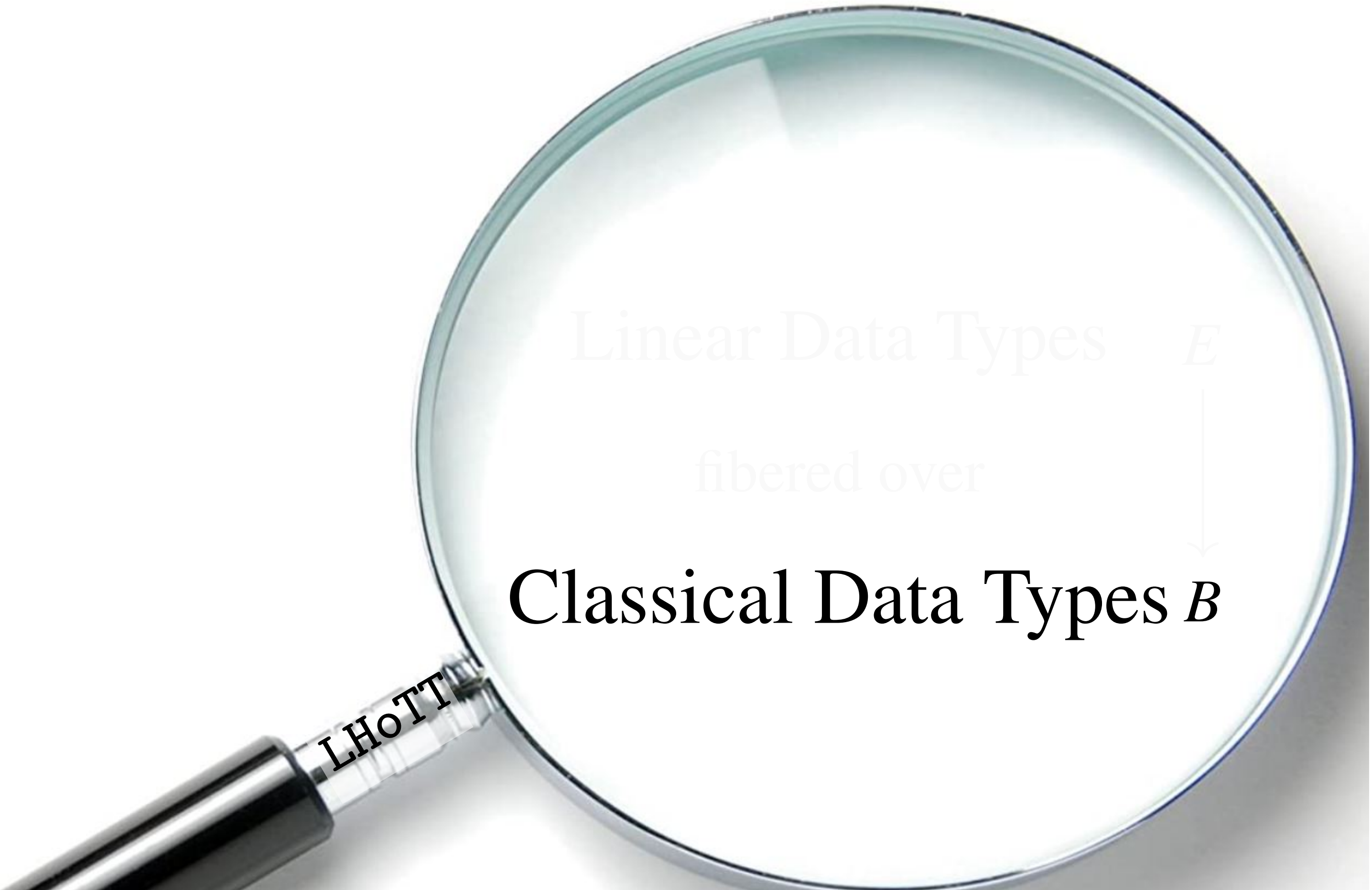
$\exists$ classical & linear dependent type theory

conservative over classical *Homotopy Type Theory* (`HoTT`)

and

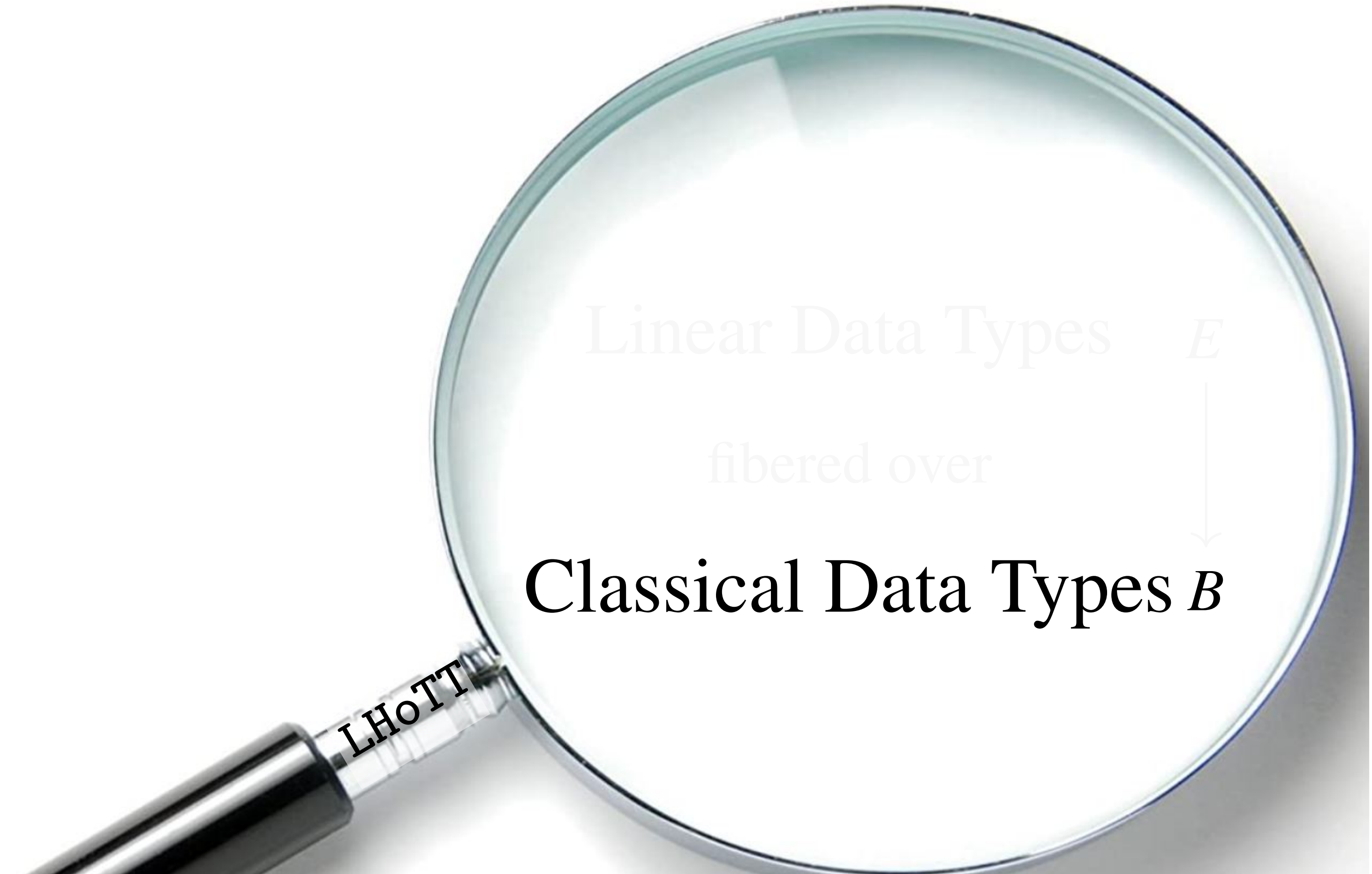
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]

(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



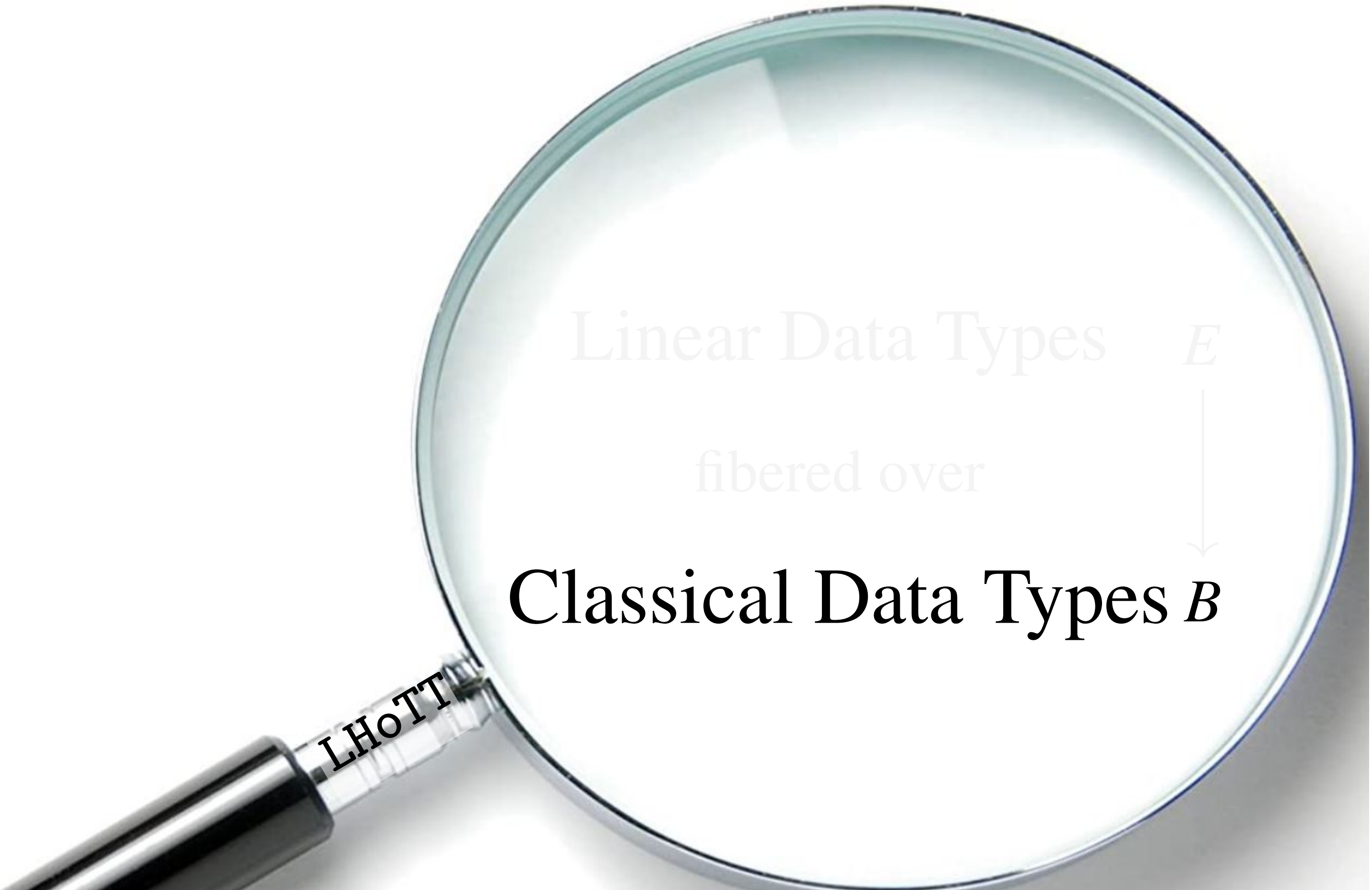
Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



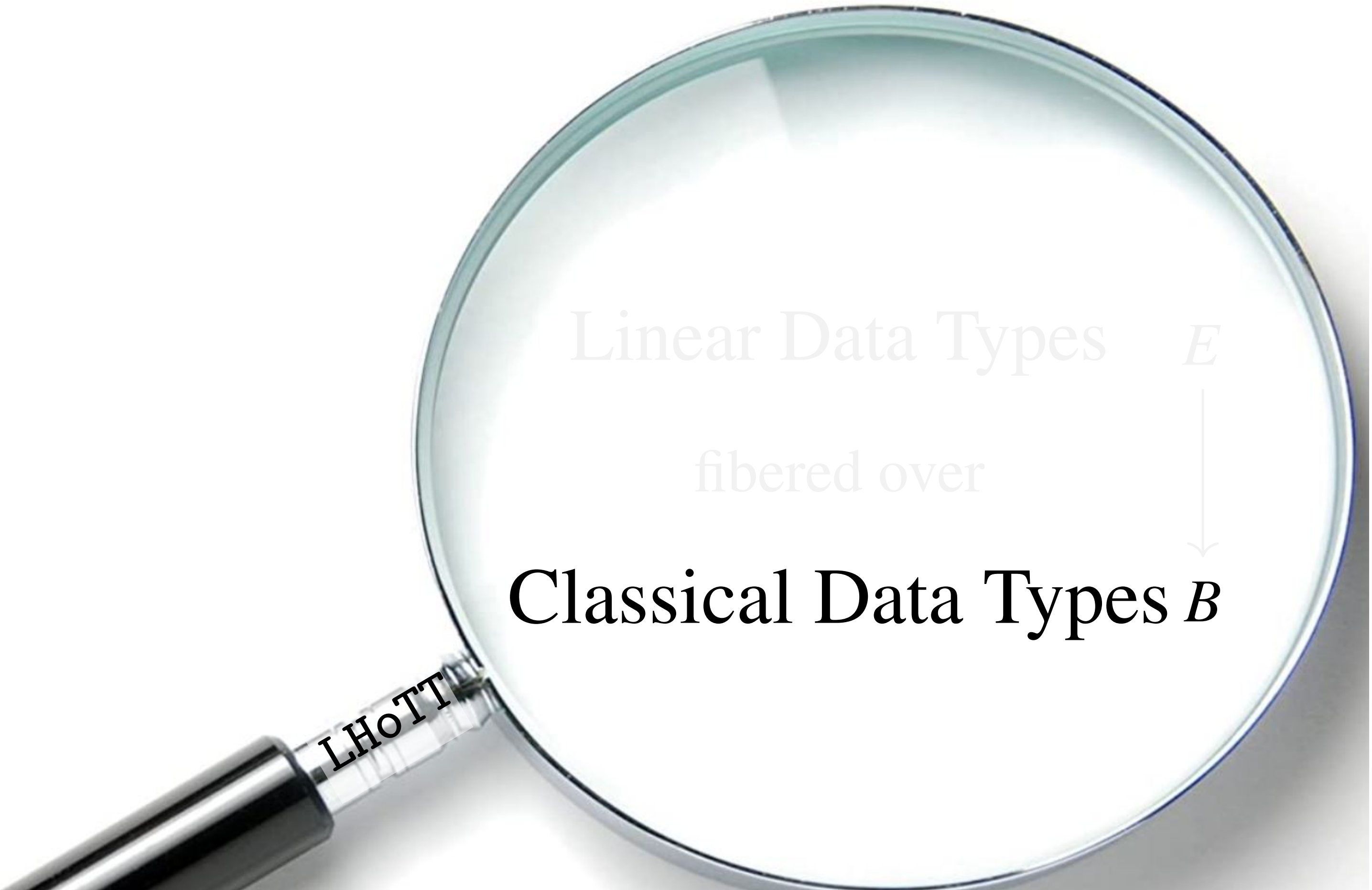
Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

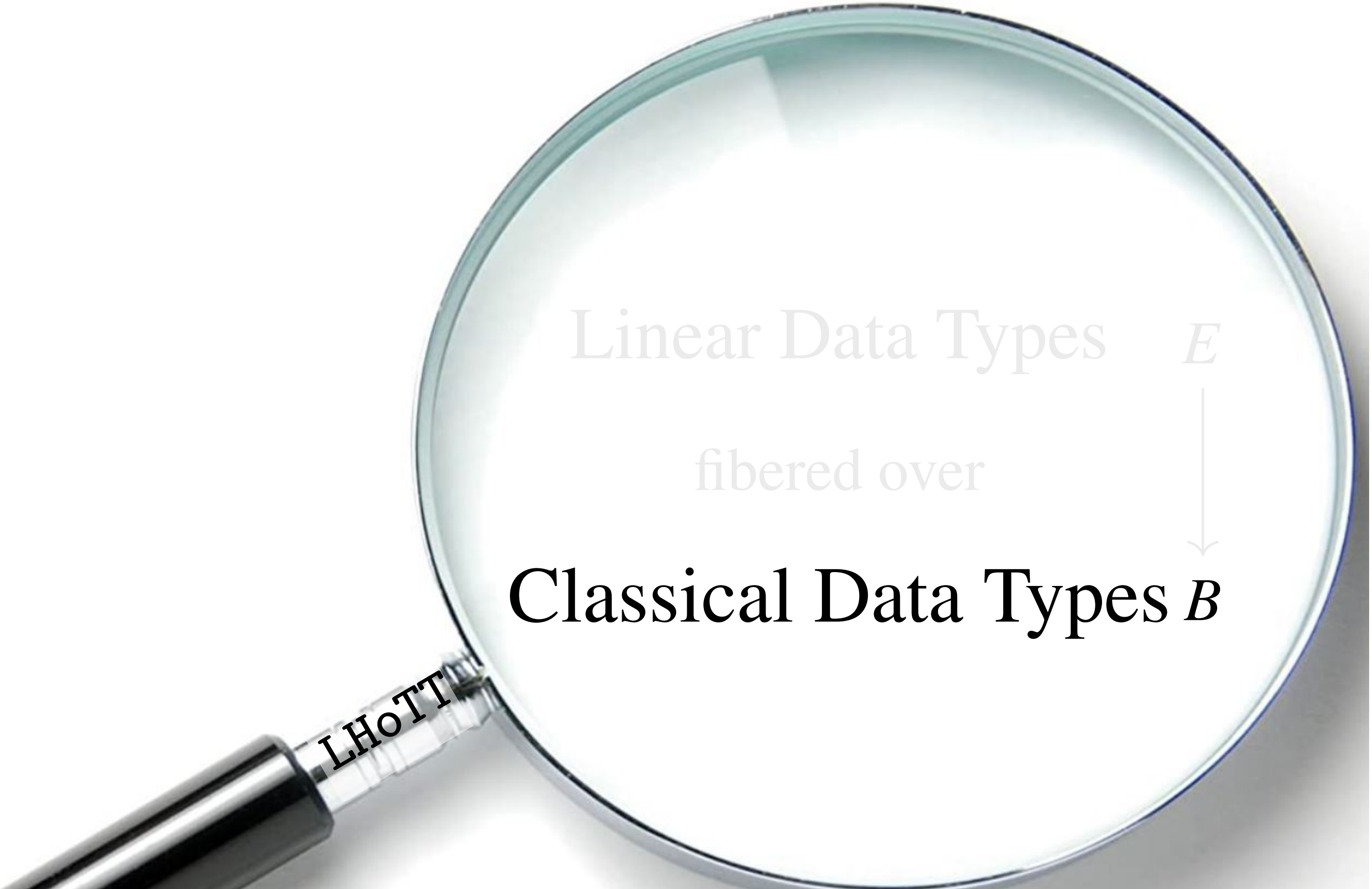


Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

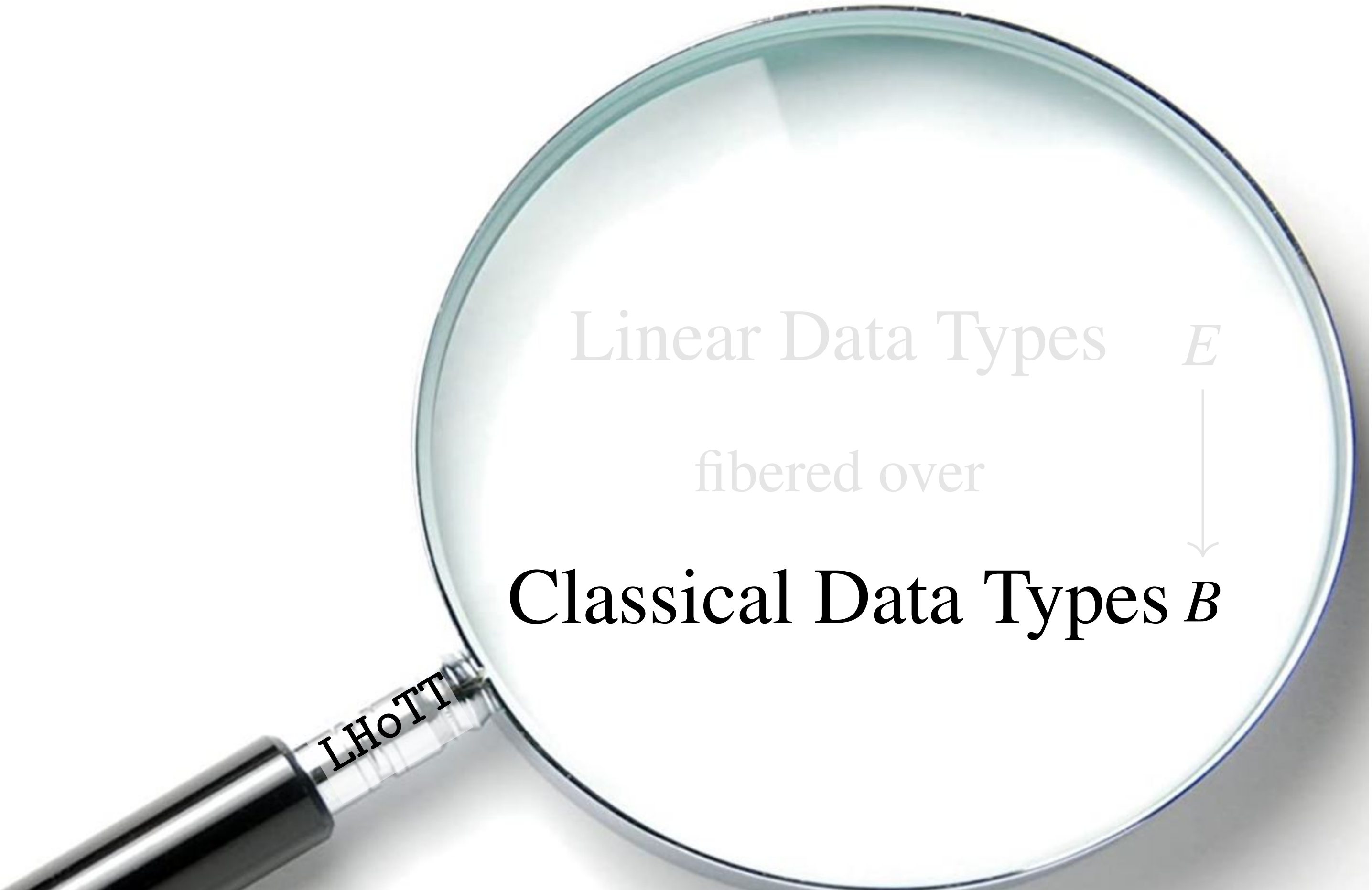
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]

(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

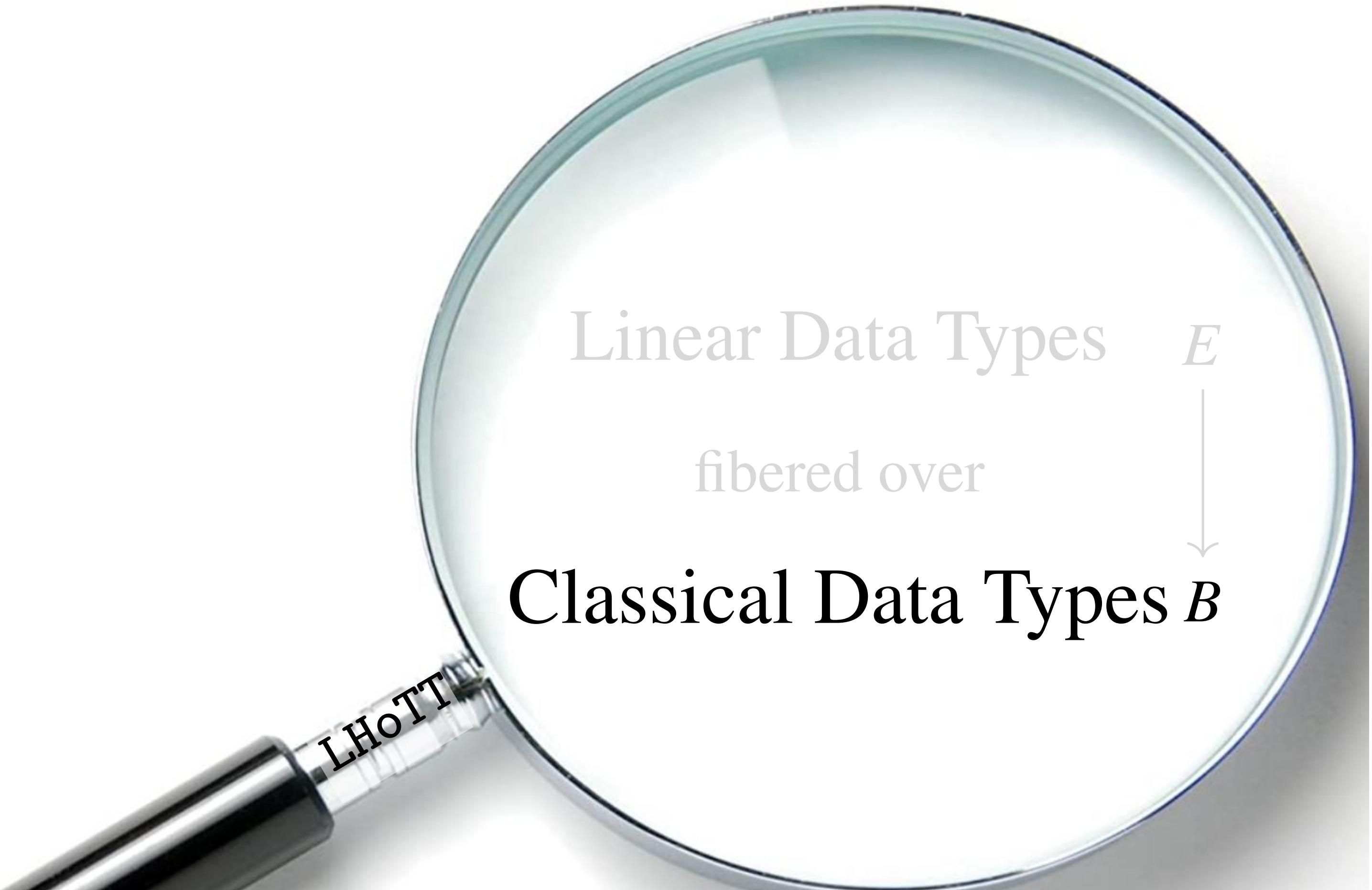


Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

$\exists$ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]

(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



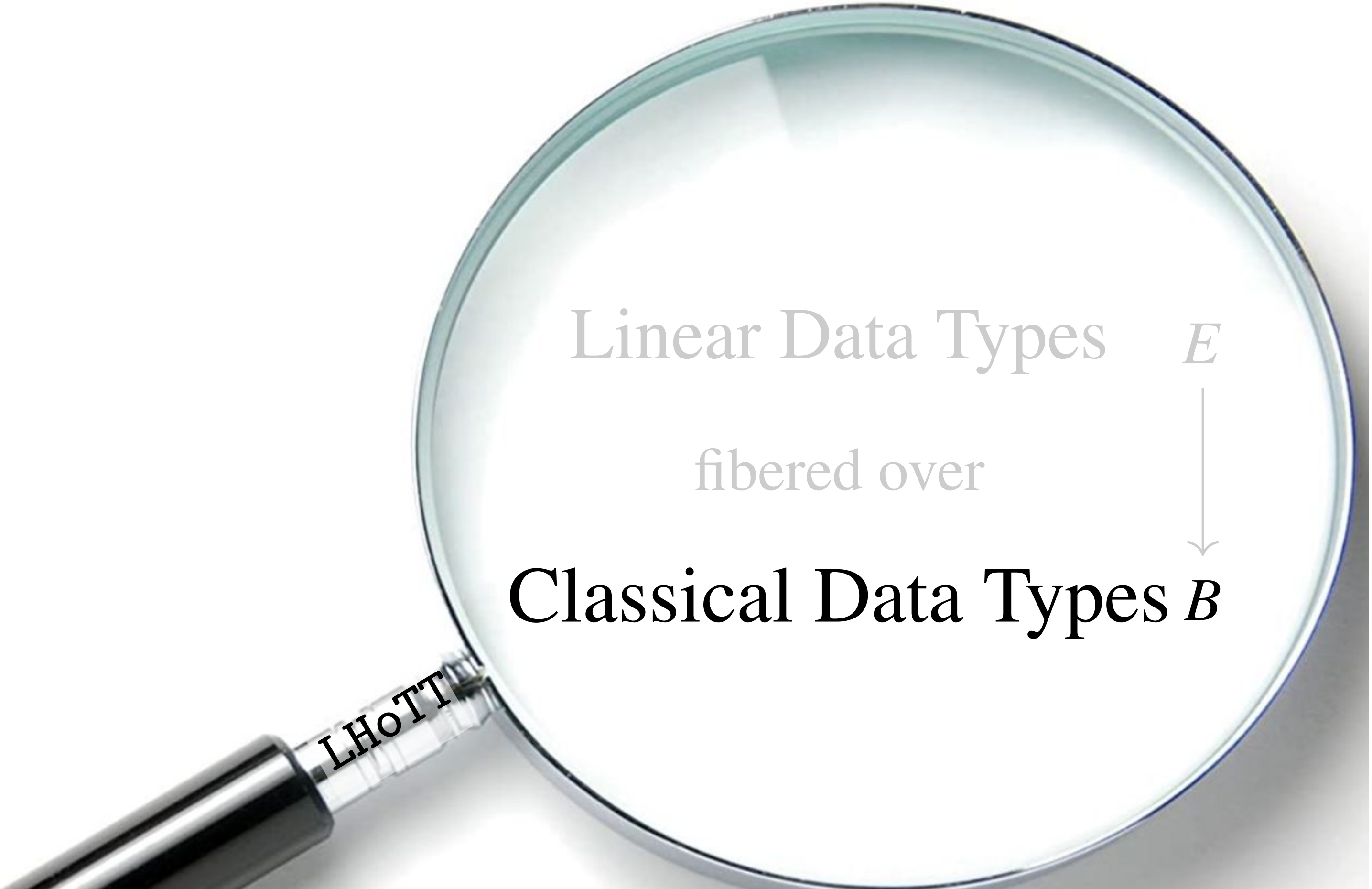
Linear Data Types E

fibered over

Classical Data Types B

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Data Types E

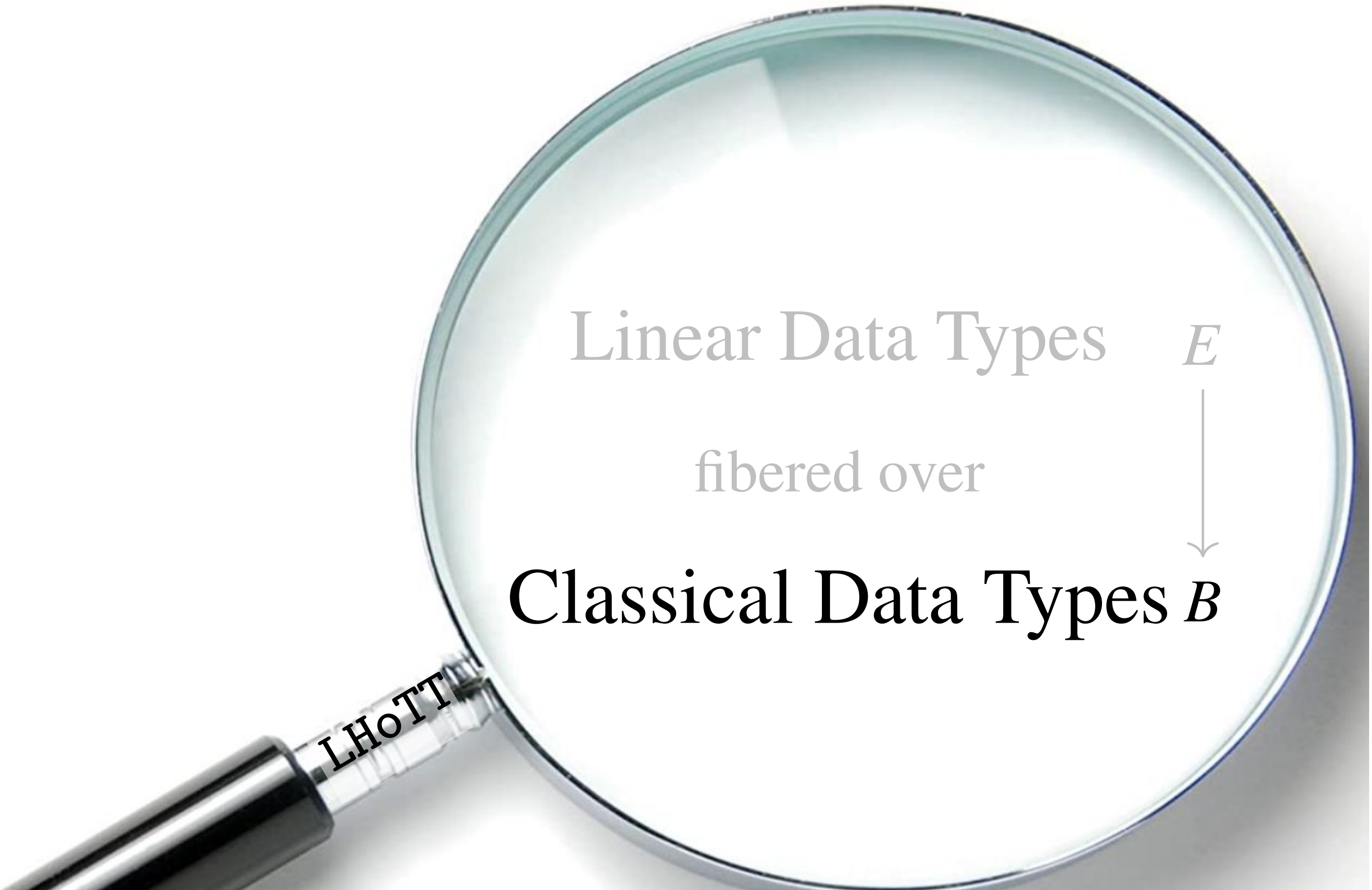
fibered over

Classical Data Types B

$$\begin{tikzcd} E \arrow[d] \\ B \end{tikzcd}$$

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Data Types

fibered over

Classical Data Types

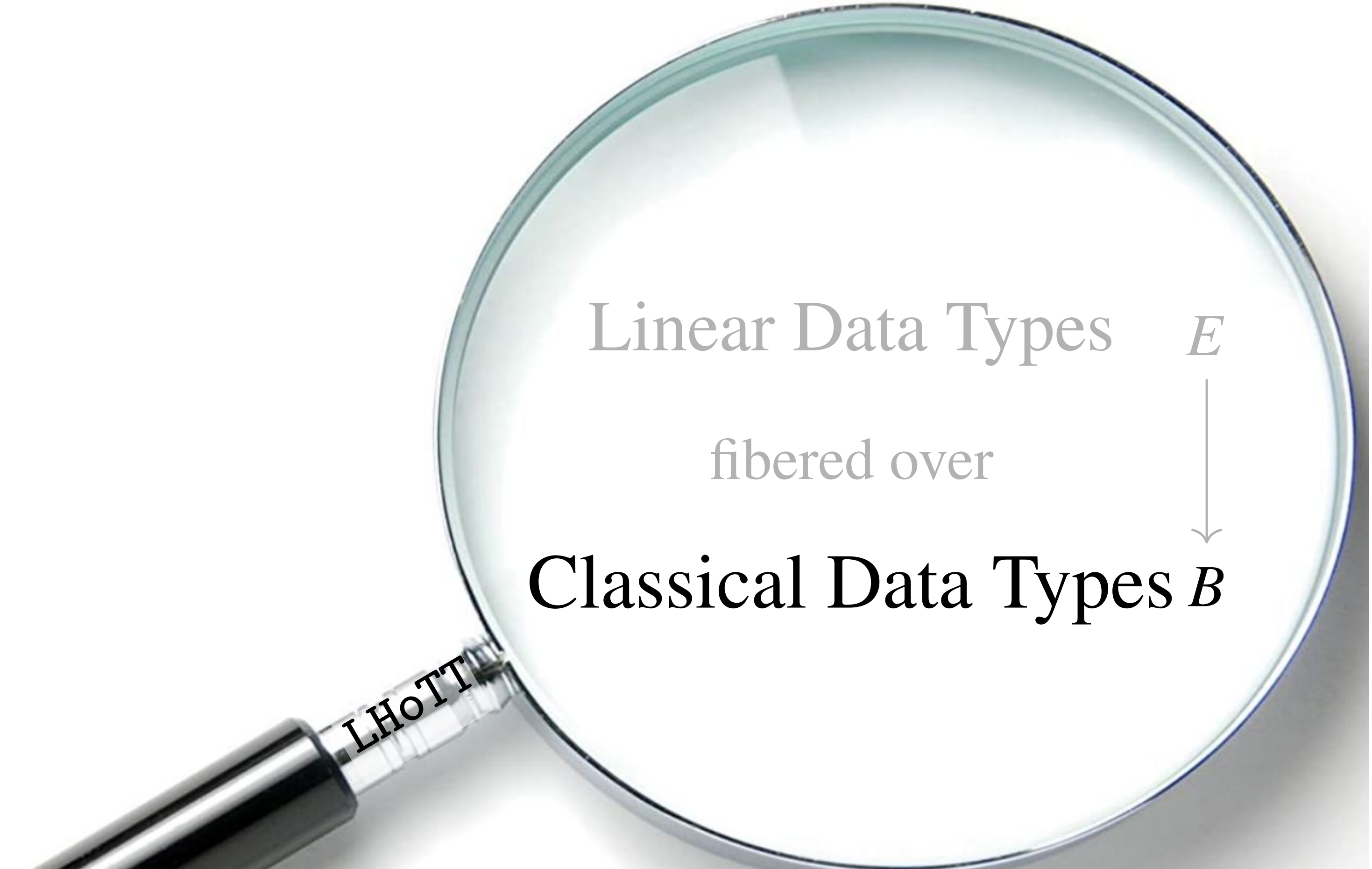
$$\begin{tikzcd} E \arrow[d] \\ B \end{tikzcd}$$

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]

(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Data Types

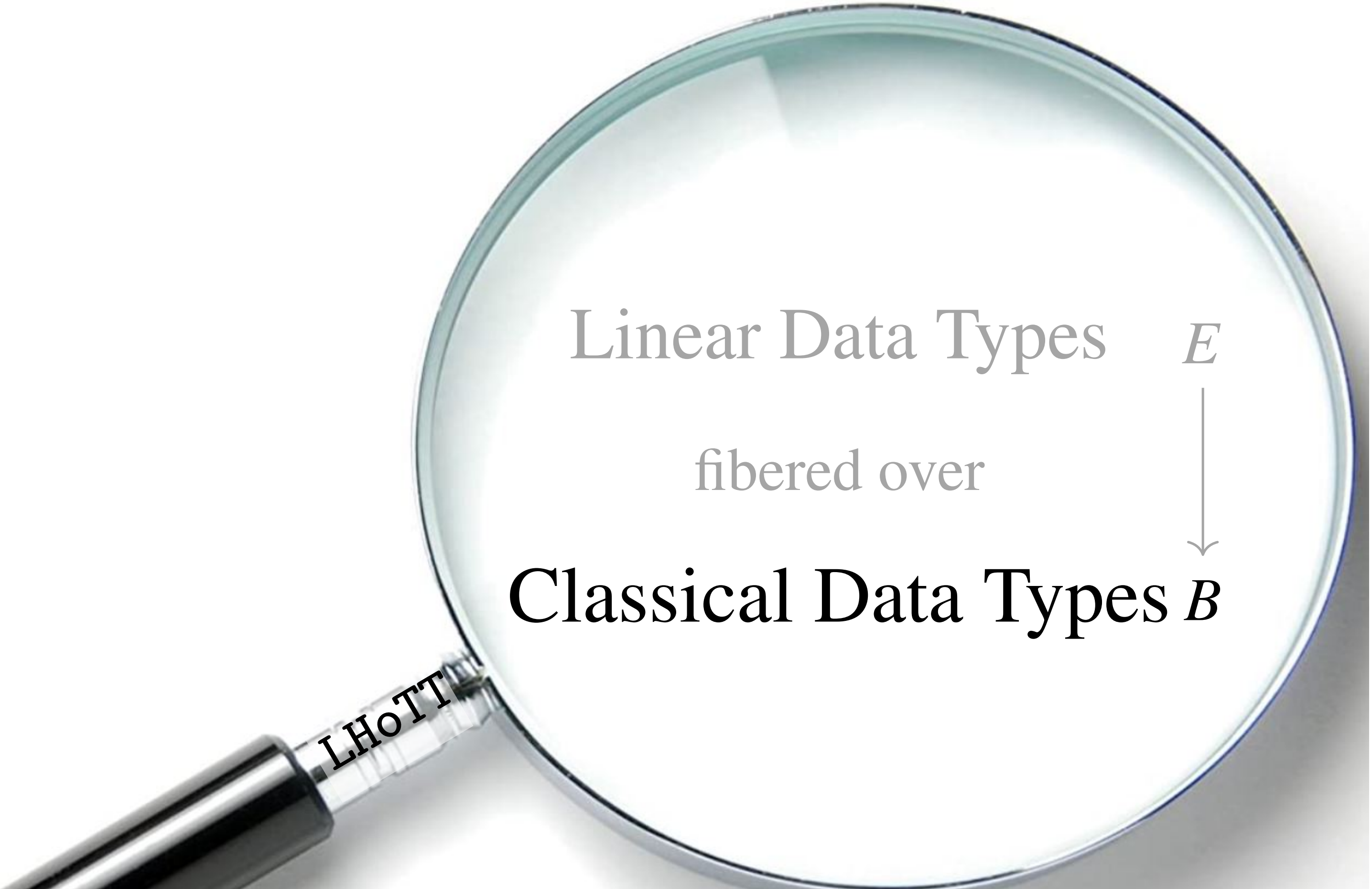
fibered over

Classical Data Types

$$\begin{tikzcd} E \arrow[d] \\ B \end{tikzcd}$$

Linear Homotopy Type Theory (`LHoTT`)

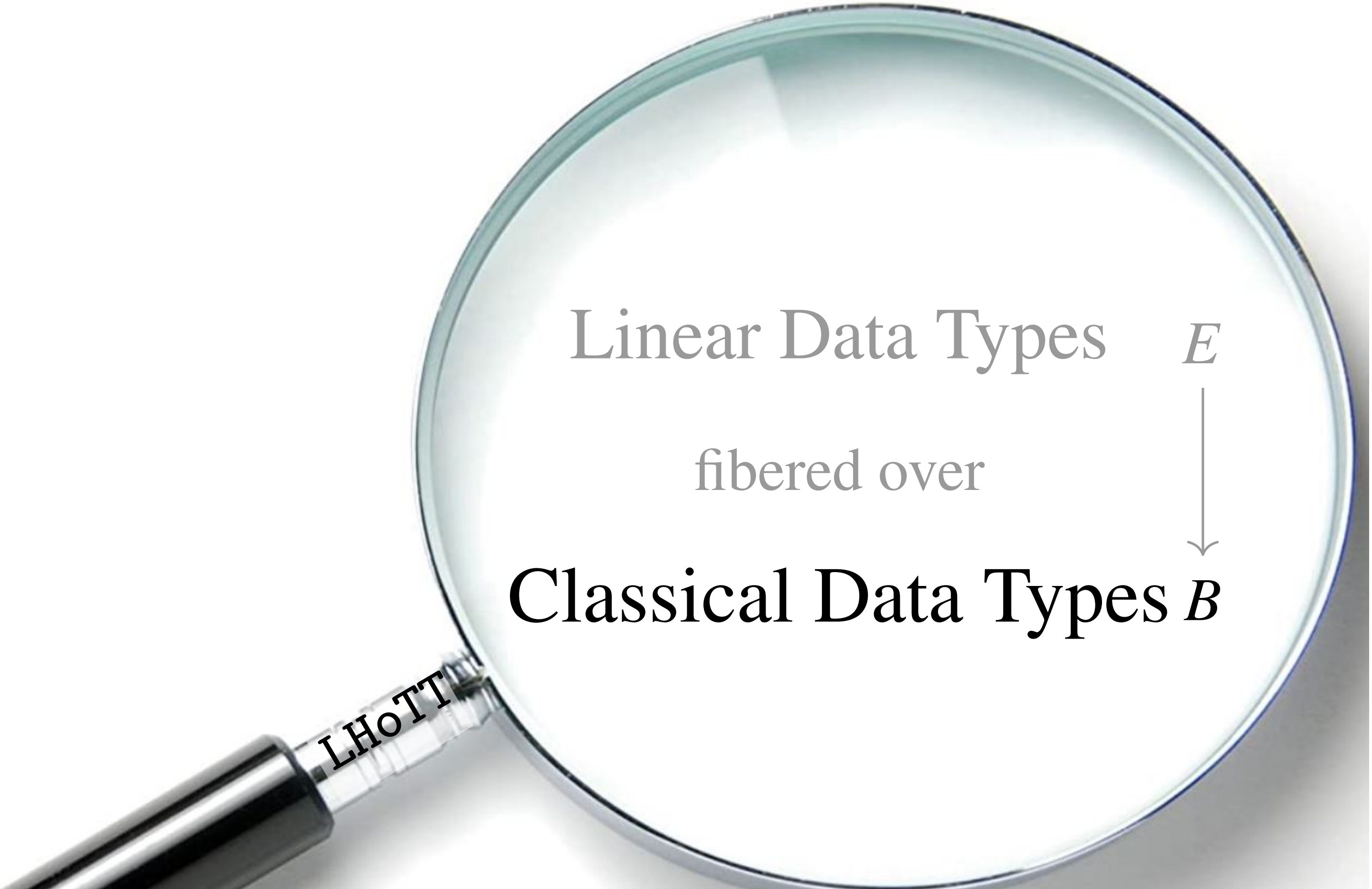
Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

$\exists$ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Data Types

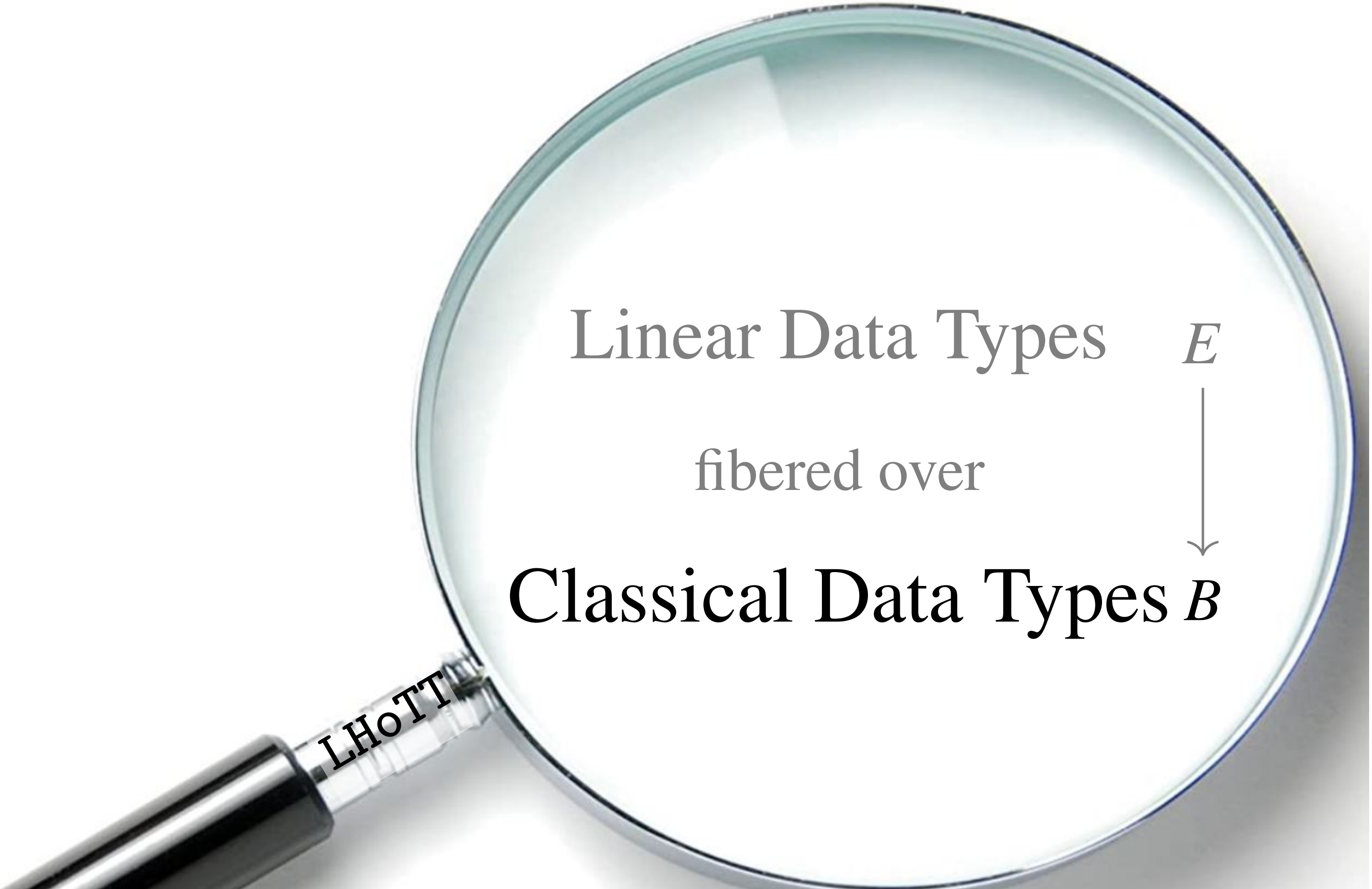
fibered over

Classical Data Types

$$\begin{tikzcd} E \arrow[d] \\ B \end{tikzcd}$$

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Data Types

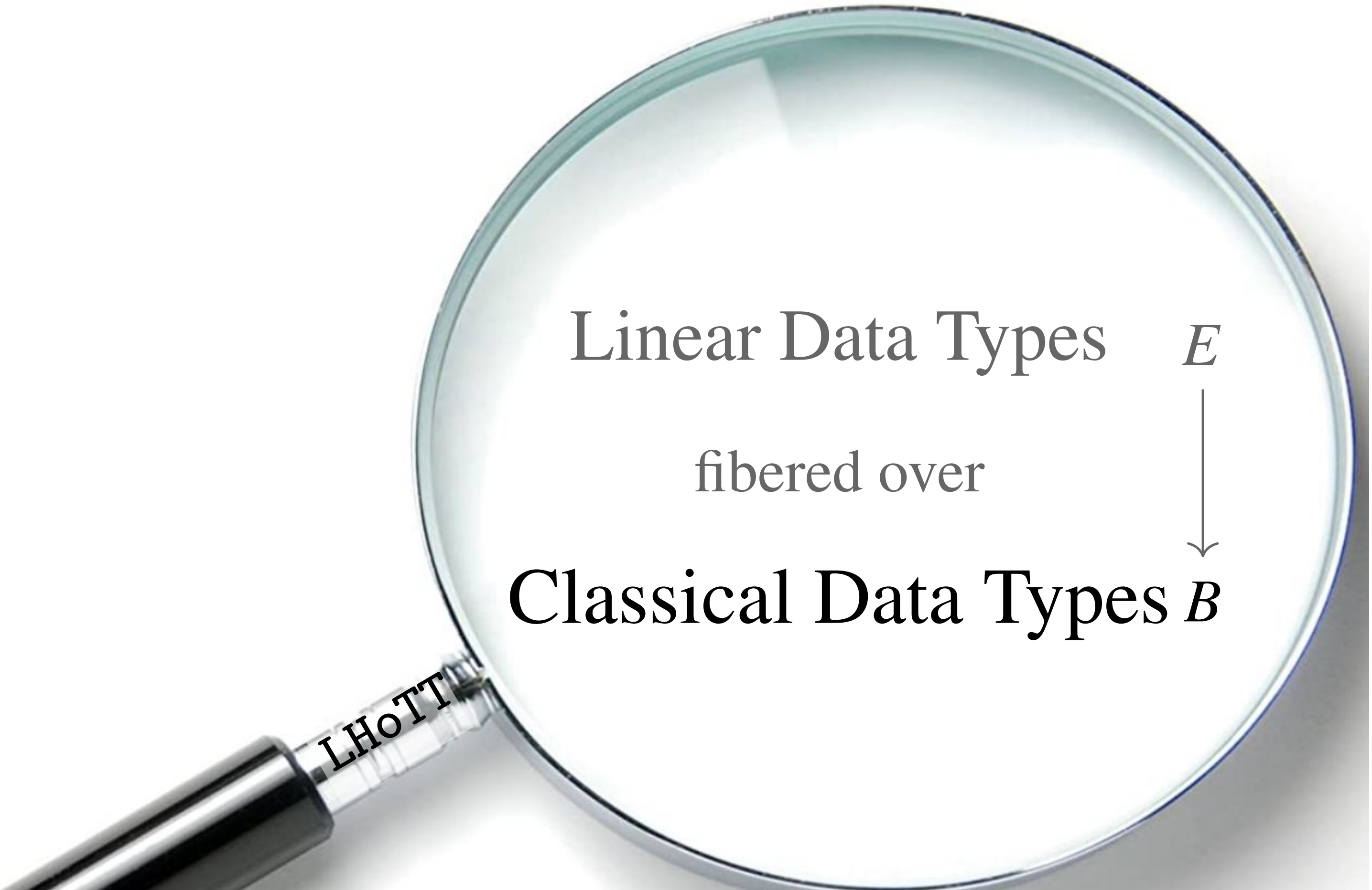
fibered over

Classical Data Types

$$\begin{tikzcd} E \arrow[d] \\ B \end{tikzcd}$$

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Data Types

fibered over

Classical Data Types

$$\begin{tikzcd} E \arrow[d] \\ B \end{tikzcd}$$

Linear Homotopy Type Theory (`LHoTT`)

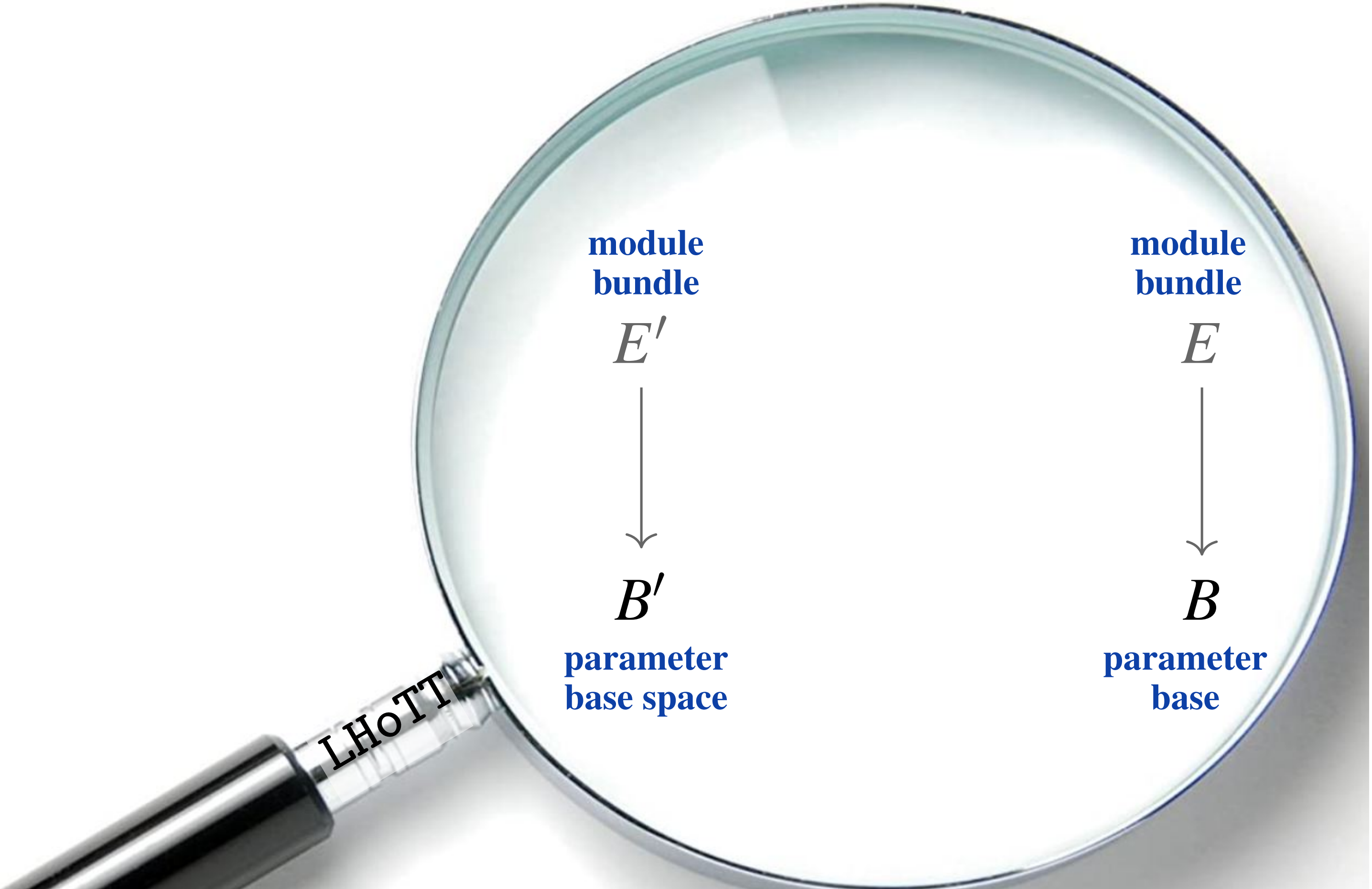
Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

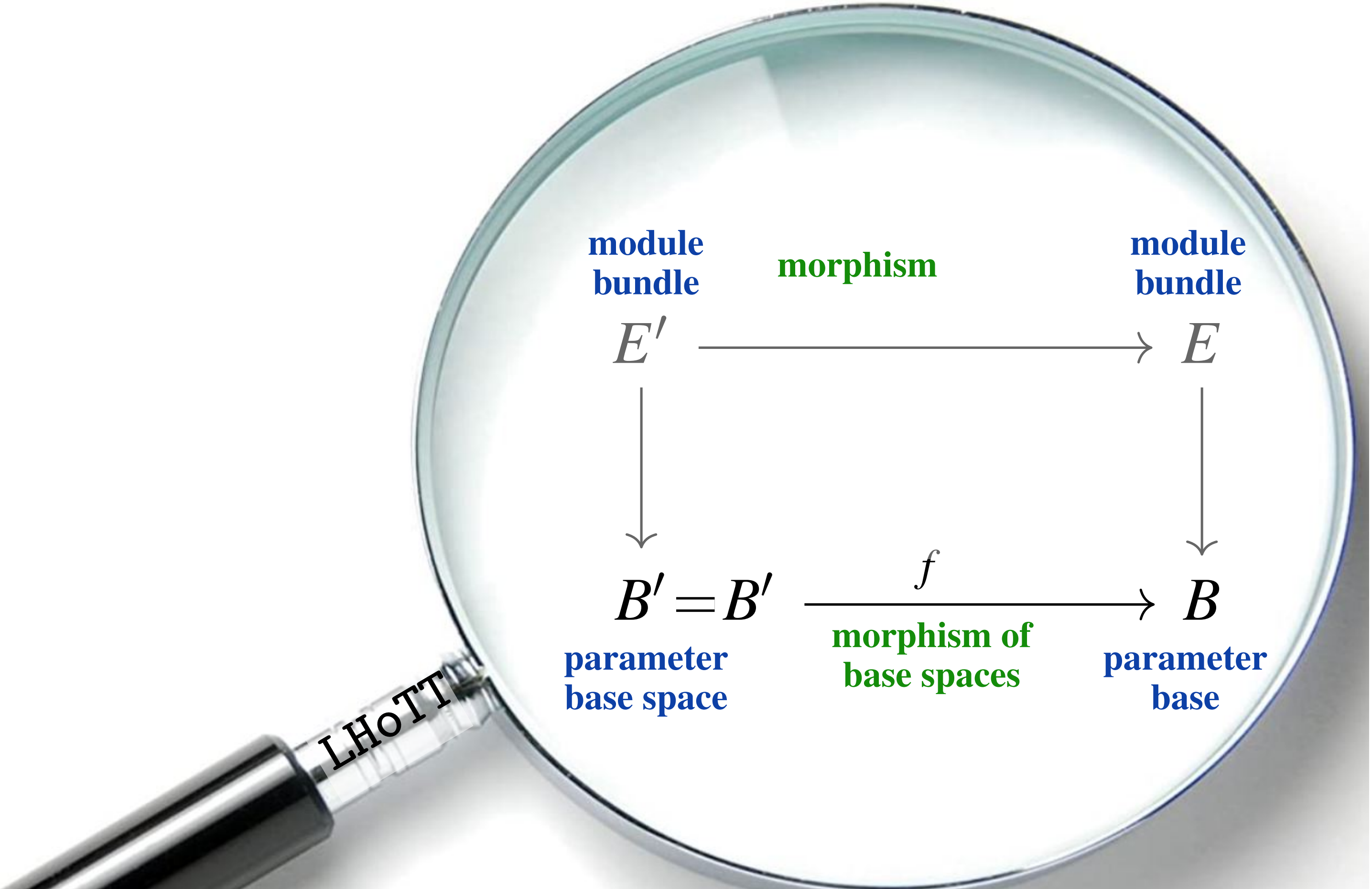
$\exists$ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]

(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Homotopy Type Theory (`LHoTT`)

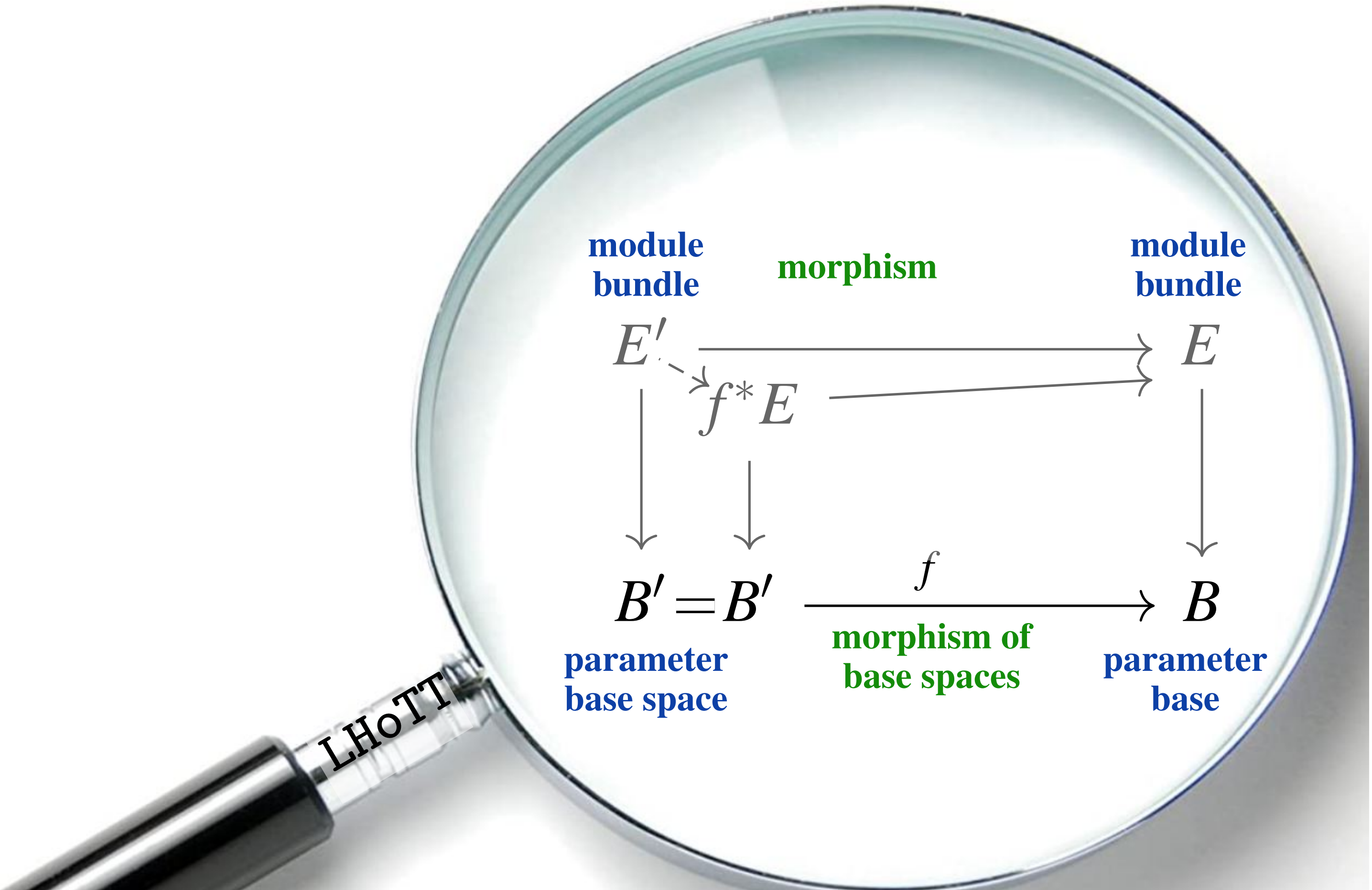
Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



$$\begin{tikzcd} E' \arrow[r] \arrow[d] & E \arrow[d] \\ B' = B' \arrow[r, "f"] & B \end{tikzcd}$$

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



module bundle **morphism** **module bundle**

$$\begin{tikzcd}
E' \arrow[rr] \arrow[dr, dashed] \arrow[dd] & & E \arrow[dd] \\
& f^*E \arrow[ur] \arrow[d] & \\
B' \arrow[r, equal] & B' \arrow[r, "f"] & B
\end{tikzcd}$$

parameter base space **morphism of base spaces** **parameter base**

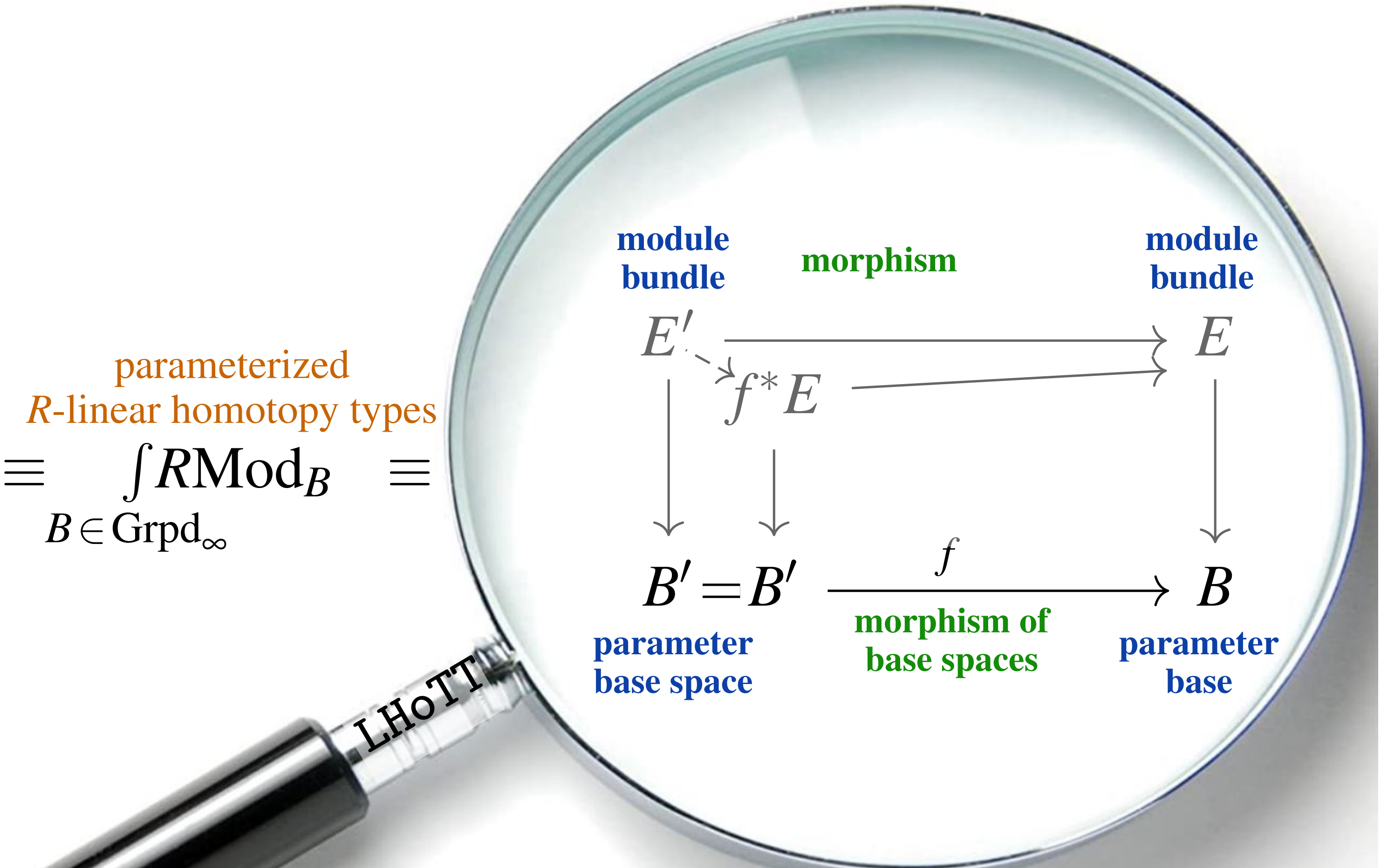
Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

tangent ∞-topos of

parameterized
R-linear homotopy types

$$T\mathrm{Grpd}_\infty \;:\equiv\; \int_{B \in \mathrm{Grpd}_\infty} R\mathrm{Mod}_B \;\equiv$$



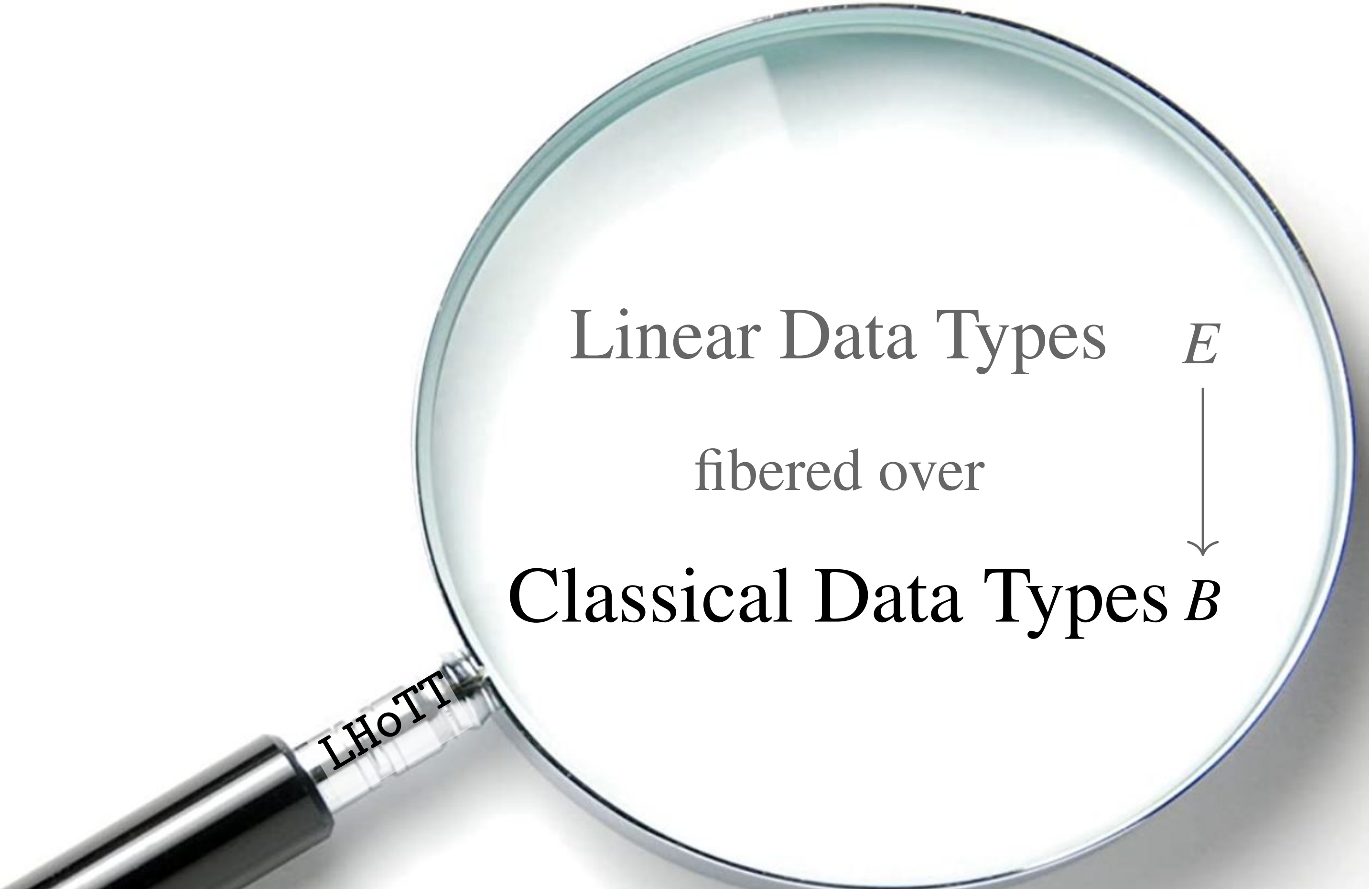
module bundle **morphism** **module bundle**

$$\begin{tikzcd} E' \arrow[rr] \arrow[dr, dashed] \arrow[dd] & & E \arrow[dd] \\ & f^*E \arrow[ur] \arrow[d] & \\ B' \arrow[r, equal] & B' \arrow[r, "f"] & B \end{tikzcd}$$

parameter base space **morphism of base spaces** **parameter base**

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)



Linear Data Types E

fibered over $\big\downarrow$

Classical Data Types B

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

∃ classical & linear dependent type theory

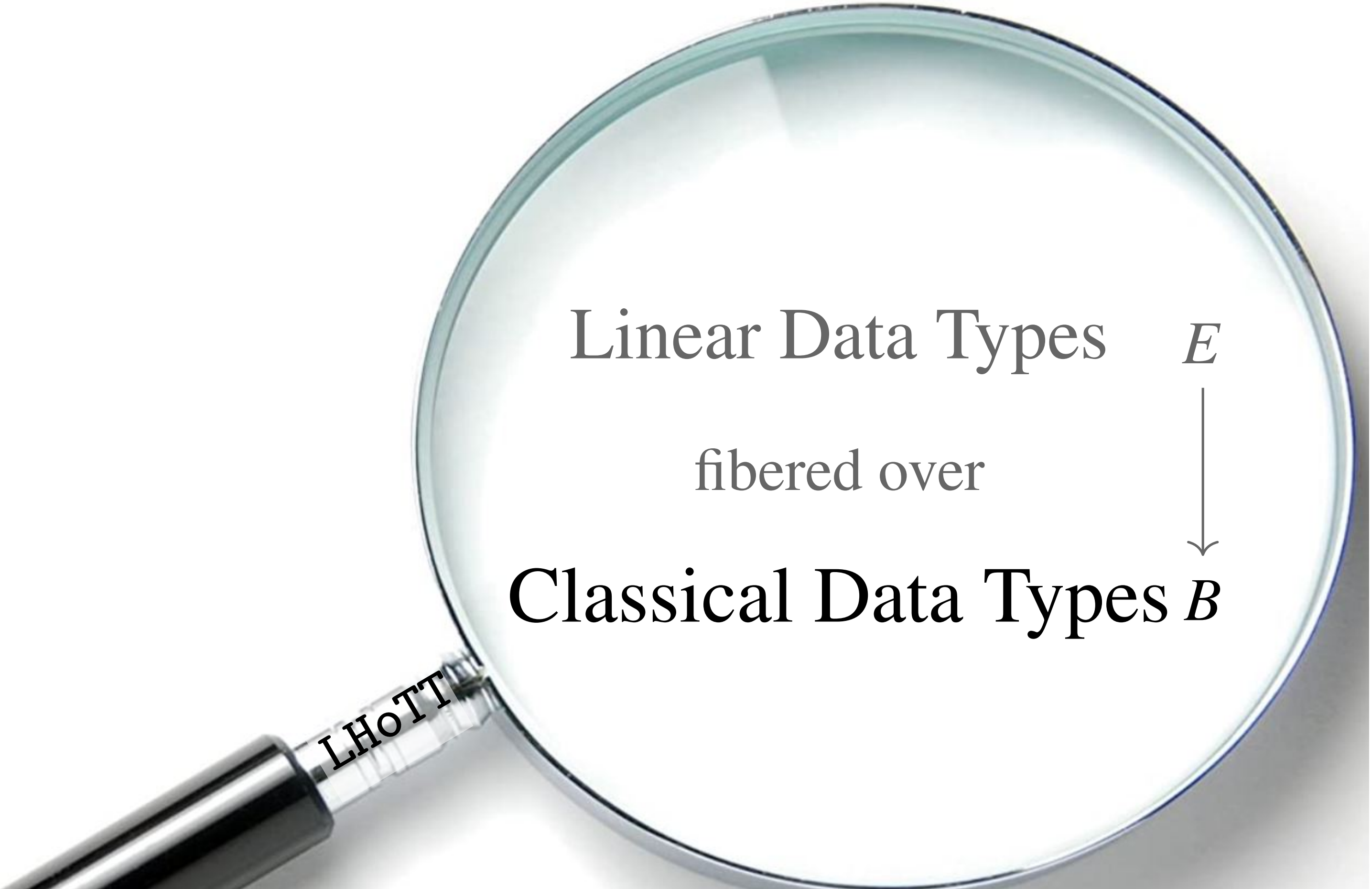
conservative over classical *Homotopy Type Theory* (`HoTT`)

and

verifying axiom scheme “Motivic Yoga” [Riley, §2.4, anticipated in S. (2014), §3.2]

(i.e. Grothendieck’s six operations *à la* Wirthmüller — more on all this below)

More technically:



Linear Data Types

fibered over

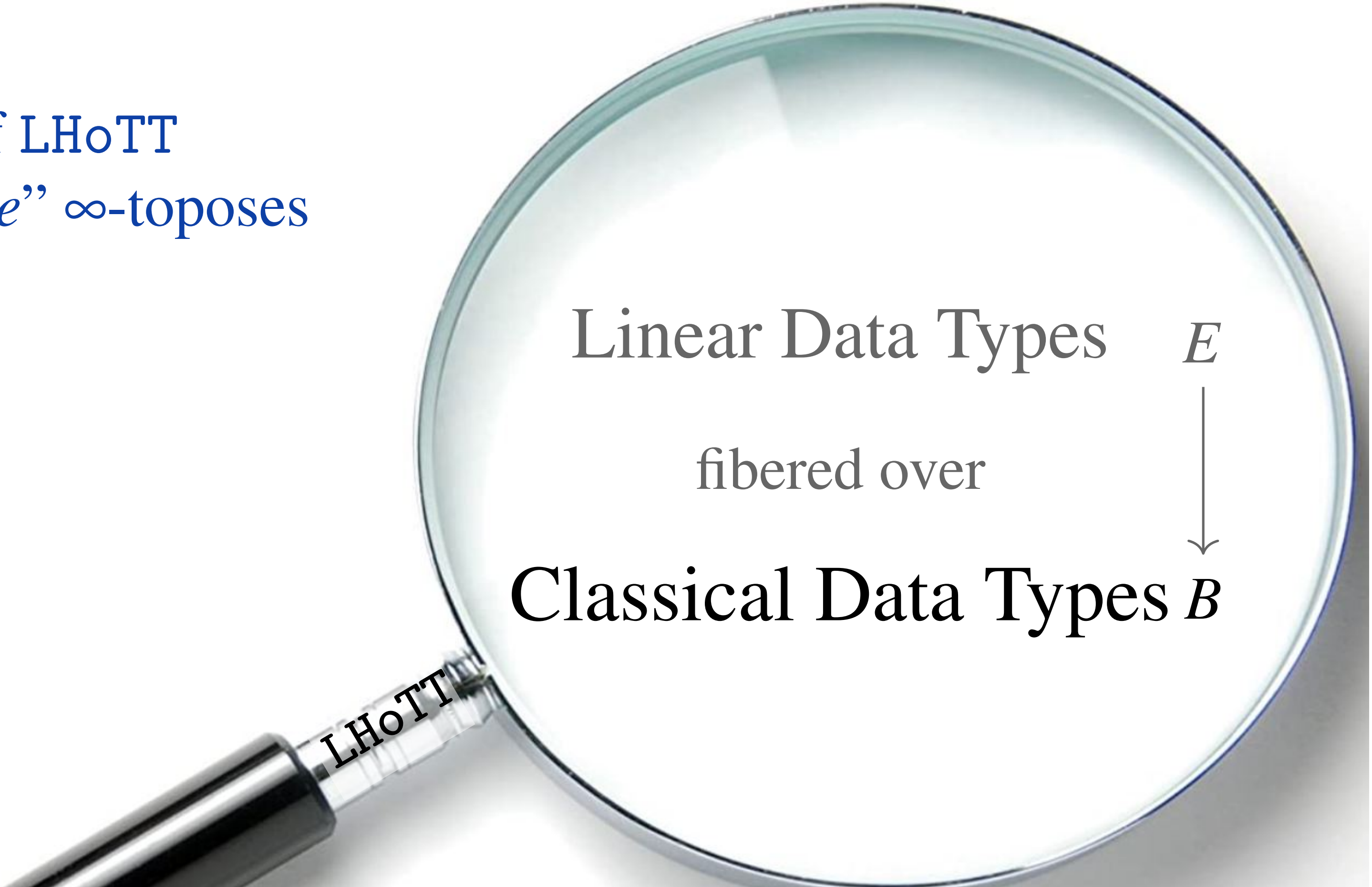
Classical Data Types

$$\begin{tikzcd} E \arrow[d] \\ B \end{tikzcd}$$

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

More technically:
The categorical semantics of `LHoTT`
is in "*infinitesimally cohesive*" ∞-toposes



Linear Data Types E

fibered over $\big\downarrow$

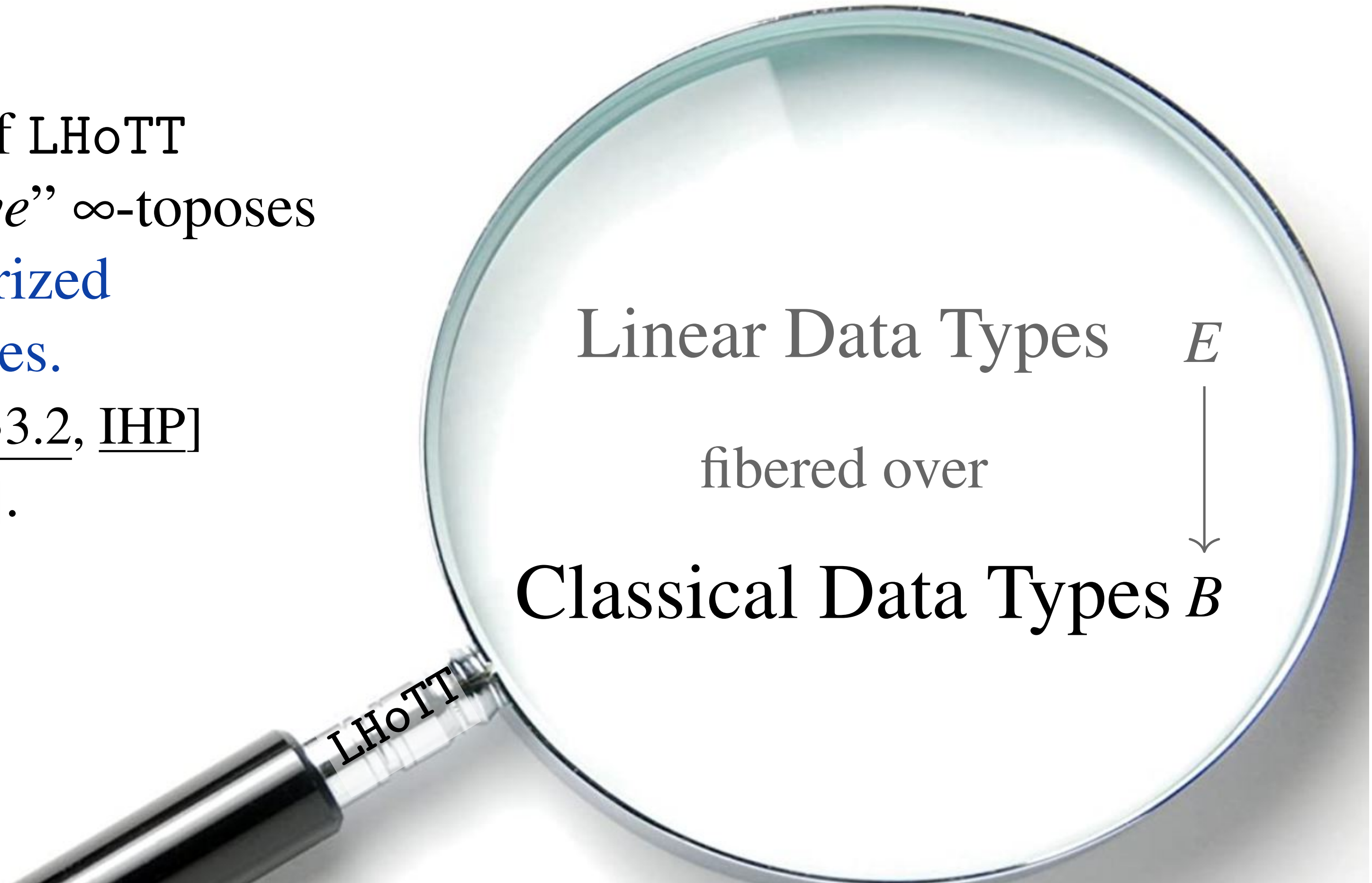
Classical Data Types B

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

More technically:

The categorical semantics of `LHoTT`
is in "*infinitesimally cohesive*" ∞-toposes
of module spectra parameterized
over classical homotopy types.
[S. (2013), §4.1.2] [S. (2014), §3.2, IHP]
[Riley, Finster & Licata (2021)].



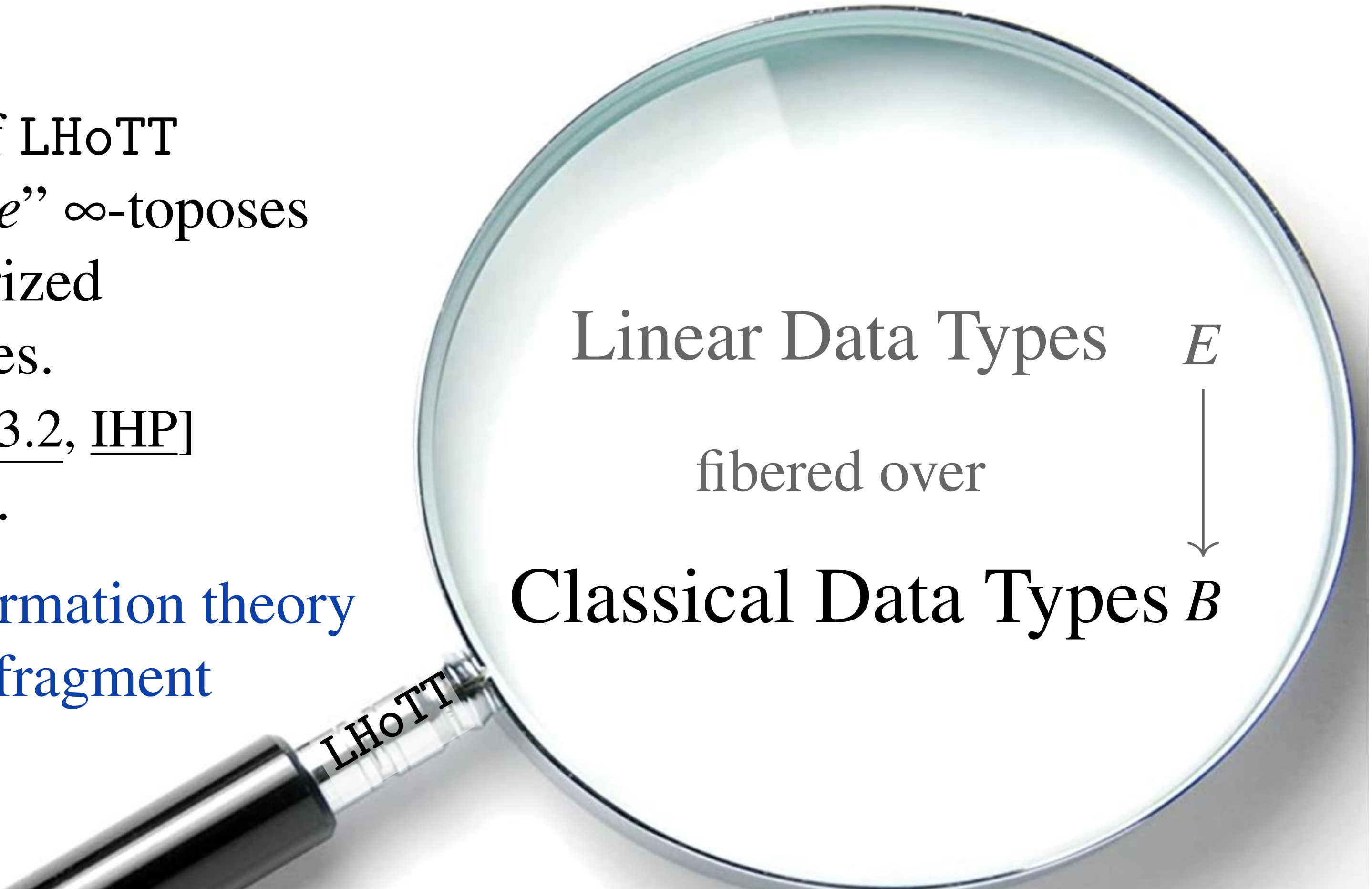
$$\begin{tikzcd} \text{Linear Data Types} & E \arrow[dd] \\ \text{fibered over} & \\ \text{Classical Data Types} & B \end{tikzcd}$$

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

More technically:
The categorical semantics of `LHoTT`
is in "*infinitesimally cohesive*" ∞-toposes
of module spectra parameterized
over classical homotopy types.
[S. (2013), §4.1.2] [S. (2014), §3.2, IHP]
[Riley, Finster & Licata (2021)].

For traditional quantum information theory
this faithfully subsumes the fragment



Linear Data Types

fibered over

Classical Data Types

$$\begin{tikzcd} E \arrow[d] \\ B \end{tikzcd}$$

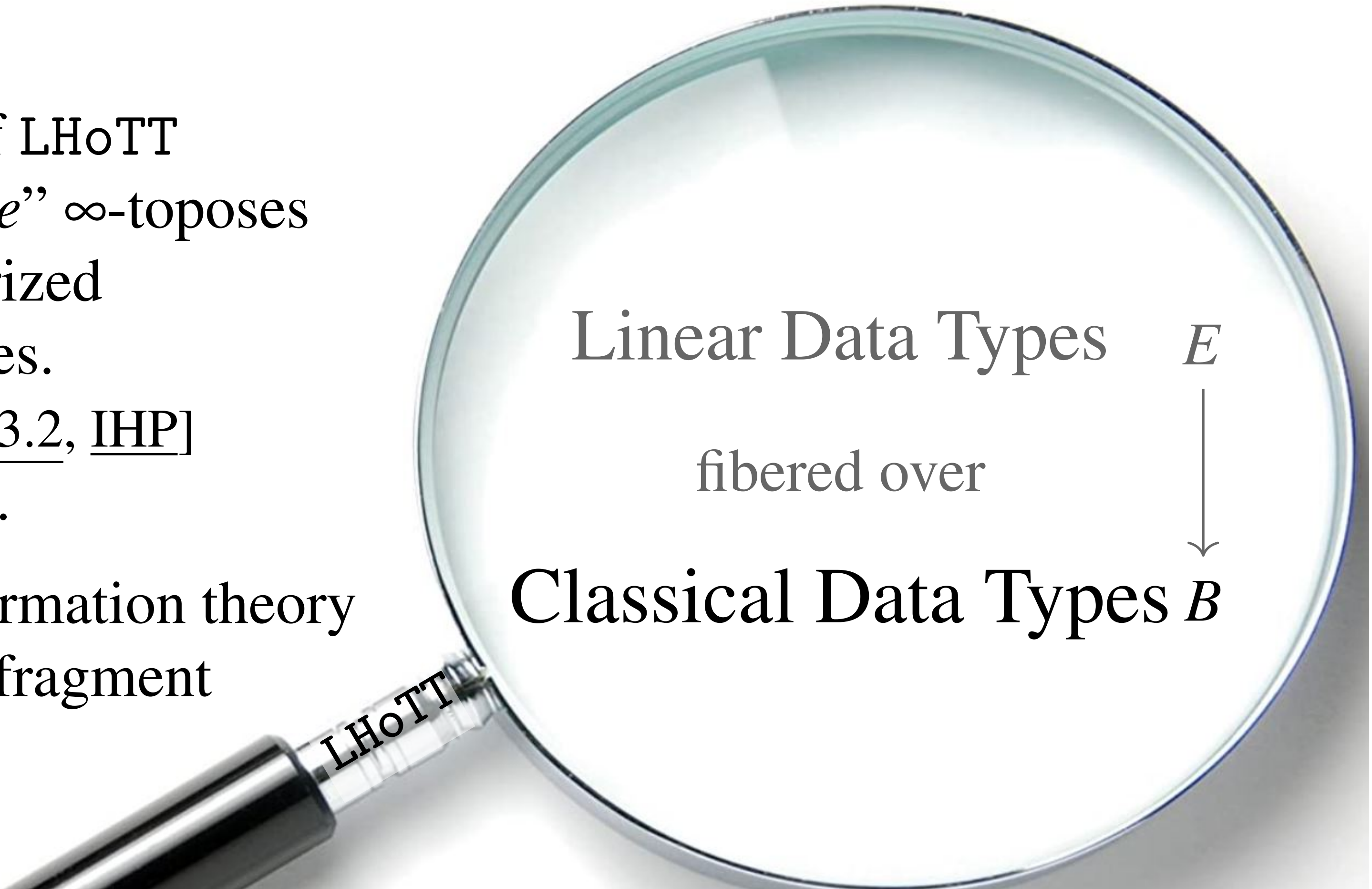
Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

More technically:

The categorical semantics of `LHoTT`
is in "*infinitesimally cohesive*" ∞-toposes
of module spectra parameterized
over classical homotopy types.
[S. (2013), §4.1.2] [S. (2014), §3.2, IHP]
[Riley, Finster & Licata (2021)].

For traditional quantum information theory
this faithfully subsumes the fragment
of complex vector bundles
over finite sets.



Linear Data Types E

fibered over $\big\downarrow$

Classical Data Types B

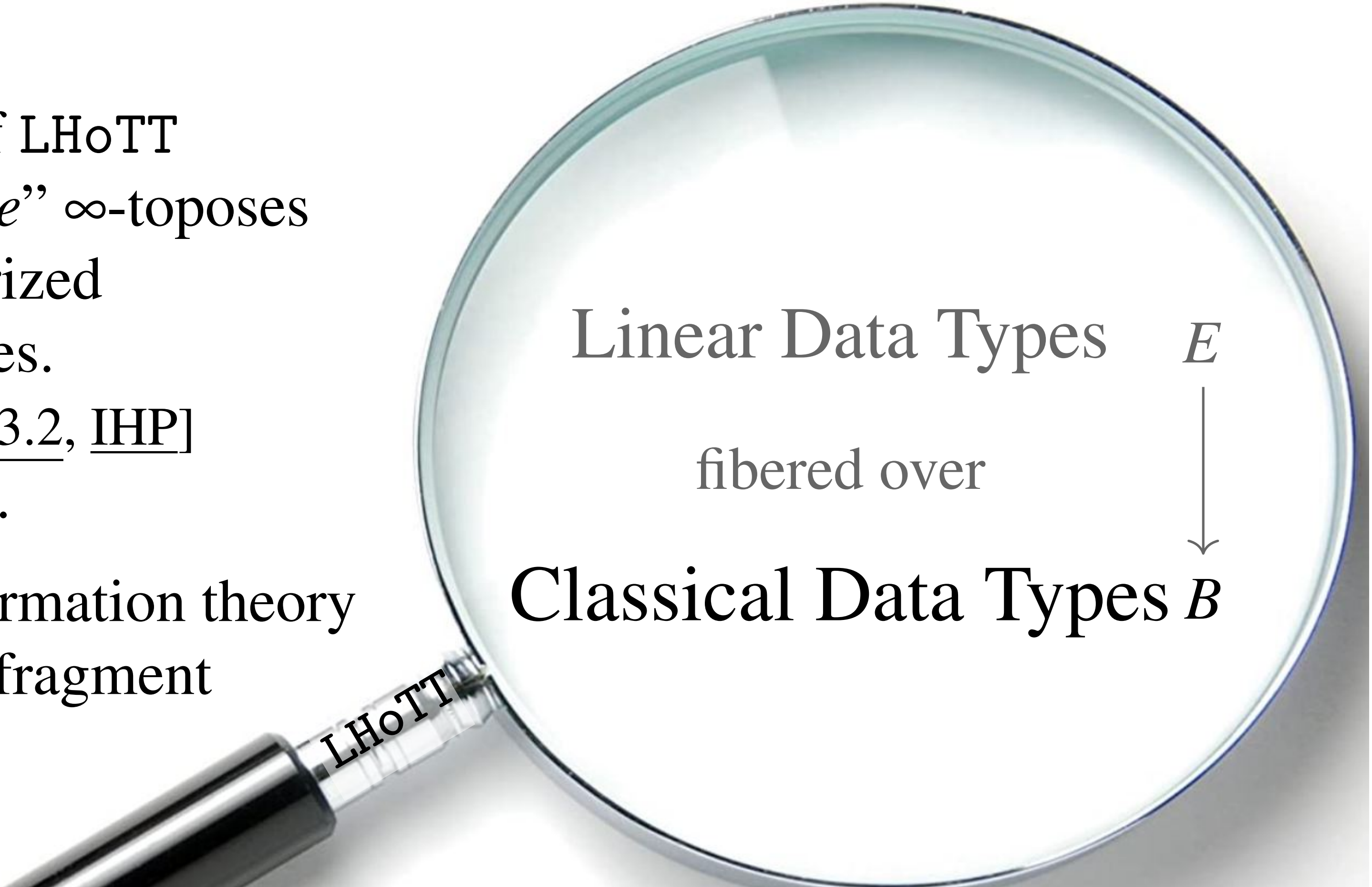
Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

More technically:

The categorical semantics of `LHoTT`
is in "*infinitesimally cohesive*" ∞-toposes
of module spectra parameterized
over classical homotopy types.
[S. (2013), §4.1.2] [S. (2014), §3.2, IHP]
[Riley, Finster & Licata (2021)].

For traditional quantum information theory
this faithfully subsumes the fragment
of complex vector bundles
over finite sets.



Linear Data Types

fibered over

Classical Data Types

$$\begin{tikzcd} E \arrow[d] \\ B \end{tikzcd}$$

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

$\exists$ classical & linear dependent type theory

conservative over classical *Homotopy Type Theory* (`HoTT`)

and

verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]

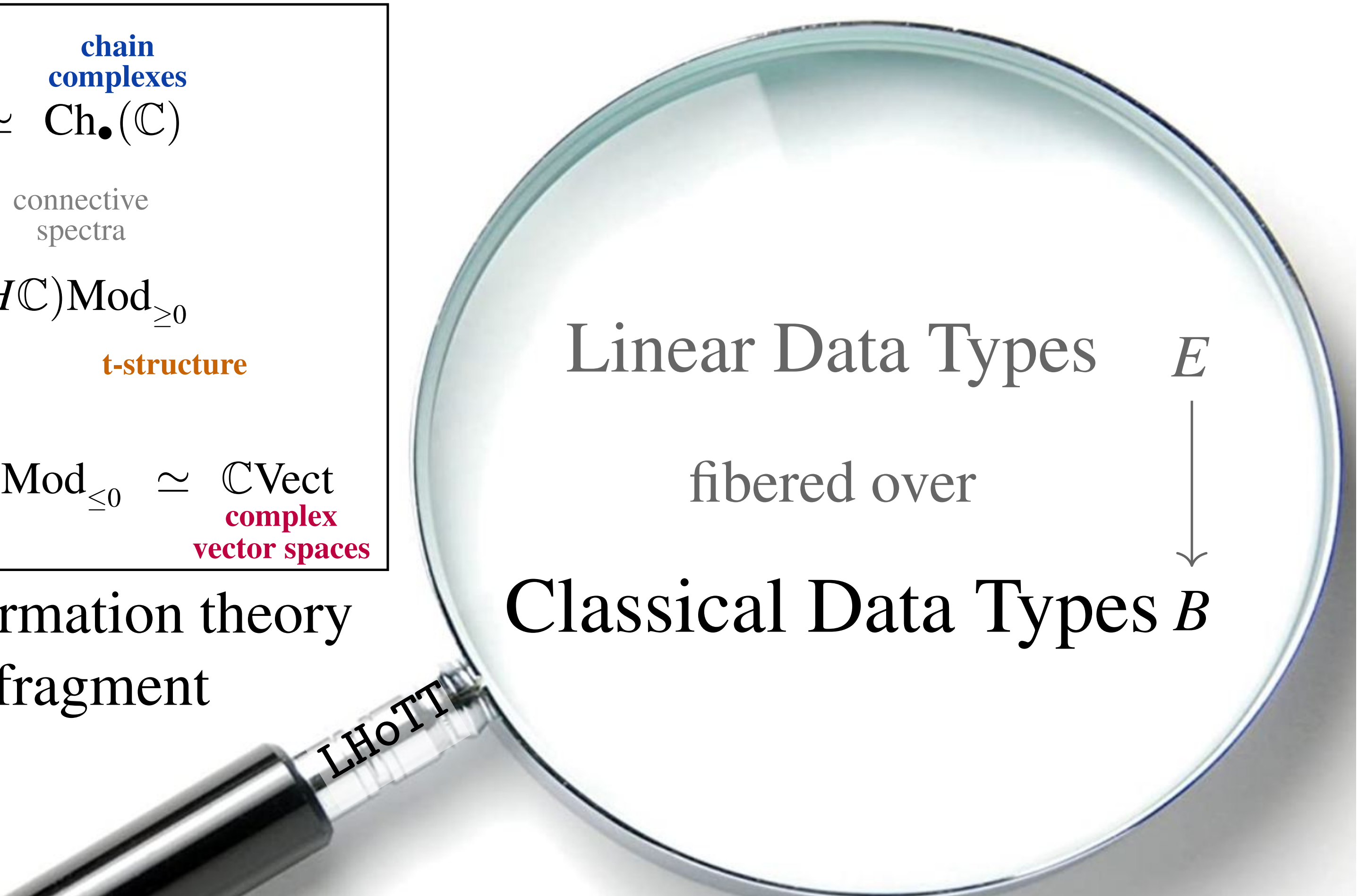
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

$$\overset{\text{complex EM-spectrum}}{R :\equiv H\mathbb{C}} \quad \Rightarrow \quad \overset{\text{EM-module spectra}}{(H\mathbb{C})\mathrm{Mod}} \;\simeq\; \overset{\text{chain complexes}}{\mathrm{Ch}_\bullet(\mathbb{C})}$$

$$\begin{tikzcd}
(H\mathbb{C})\mathrm{Mod} \arrow[loop left] \arrow[r, shift right=1.5ex, "\underset{\text{connective cover}}{\tau_{\geq 0}}"'] \arrow[r, phantom, "\perp"] & \overset{\text{connective spectra}}{\underset{\text{t-structure}}{(H\mathbb{C})\mathrm{Mod}_{\geq 0}}} \arrow[l, hook', shift right=1.5ex]
\end{tikzcd}$$

$$\underset{\text{the heart}}{(H\mathbb{C})\mathrm{Mod}^{\heartsuit}} \;:\equiv\; (H\mathbb{C})\mathrm{Mod}_{\geq 0} \cap (H\mathbb{C})\mathrm{Mod}_{\leq 0} \;\simeq\; \underset{\text{complex vector spaces}}{\mathbb{C}\mathrm{Vect}}$$

For traditional quantum information theory
this faithfully subsumes the fragment
of complex vector bundles
over finite sets.



Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

$\exists$ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

Theorem [CQTS (2022)]:

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

Theorem [CQTS (2022)]:
Motivic Yoga induces a system of monadic computational effects
constituting
linear modalities of actuality and potentiality

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

Theorem [CQTS (2022)]:
Motivic Yoga induces a system of monadic computational effects
constituting
linear modalities of actuality and potentiality
which happen to
know all about quantum information theory:

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

Theorem [CQTS (2022)]:
Motivic Yoga induces a system of monadic computational effects
constituting
linear modalities of actuality and potentiality
which happen to
know all about quantum information theory:

quantum measurement is handling of linear indefiniteness effects

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

Theorem [CQTS (2022)]:
Motivic Yoga induces a system of monadic computational effects
constituting
linear modalities of actuality and potentiality
which happen to
know all about quantum information theory:

quantum measurement	is	handling of linear indefiniteness effects
quantum state preparation	is	handling of linear randomness co-effects

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

Theorem [CQTS (2022)]:
Motivic Yoga induces a system of monadic computational effects
constituting
linear modalities of actuality and potentiality
which happen to
know all about quantum information theory:

quantum measurement	is	handling of linear indefiniteness effects
quantum state preparation	is	handling of linear randomness co-effects
quantum+classical circuits	are	the effectful string diagrams

Linear Homotopy Type Theory (`LHoTT`)

Theorem [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

Theorem [CQTS (2022)]:
Motivic Yoga induces a system of monadic computational effects
constituting
linear modalities of actuality and potentiality
which happen to
know all about quantum information theory:

quantum measurement	is	handling of linear indefiniteness effects
quantum state preparation	is	handling of linear randomness co-effects
quantum+classical circuits	are	the effectful string diagrams
quantum dynamic lifting	is	comparison functor of monadicity theorem

Linear Homotopy Type Theory (LHoTT)

Theorem [M. Riley (2022), doi:10.14418/wes01.3.139]:
∃ classical & linear dependent type theory
conservative over classical *Homotopy Type Theory* (HoTT)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à la* Wirthmüller — more on all this below)

Theorem [CQTS (2022)]:
Motivic Yoga induces a system of monadic computational effects
constituting
linear modalities of actuality and potentiality
which happen to
know all about quantum information theory:

quantum measurement	is	handling of linear indefiniteness effects
quantum state preparation	is	handling of linear randomness co-effects
quantum+classical circuits	are	the effectful string diagrams
quantum dynamic lifting	is	comparison functor of monadicity theorem

`QS` – Quantum Systems language @ CQTS

$\rightsquigarrow$ full-blown Quantum Systems language emerges embedded in `LHoTT`

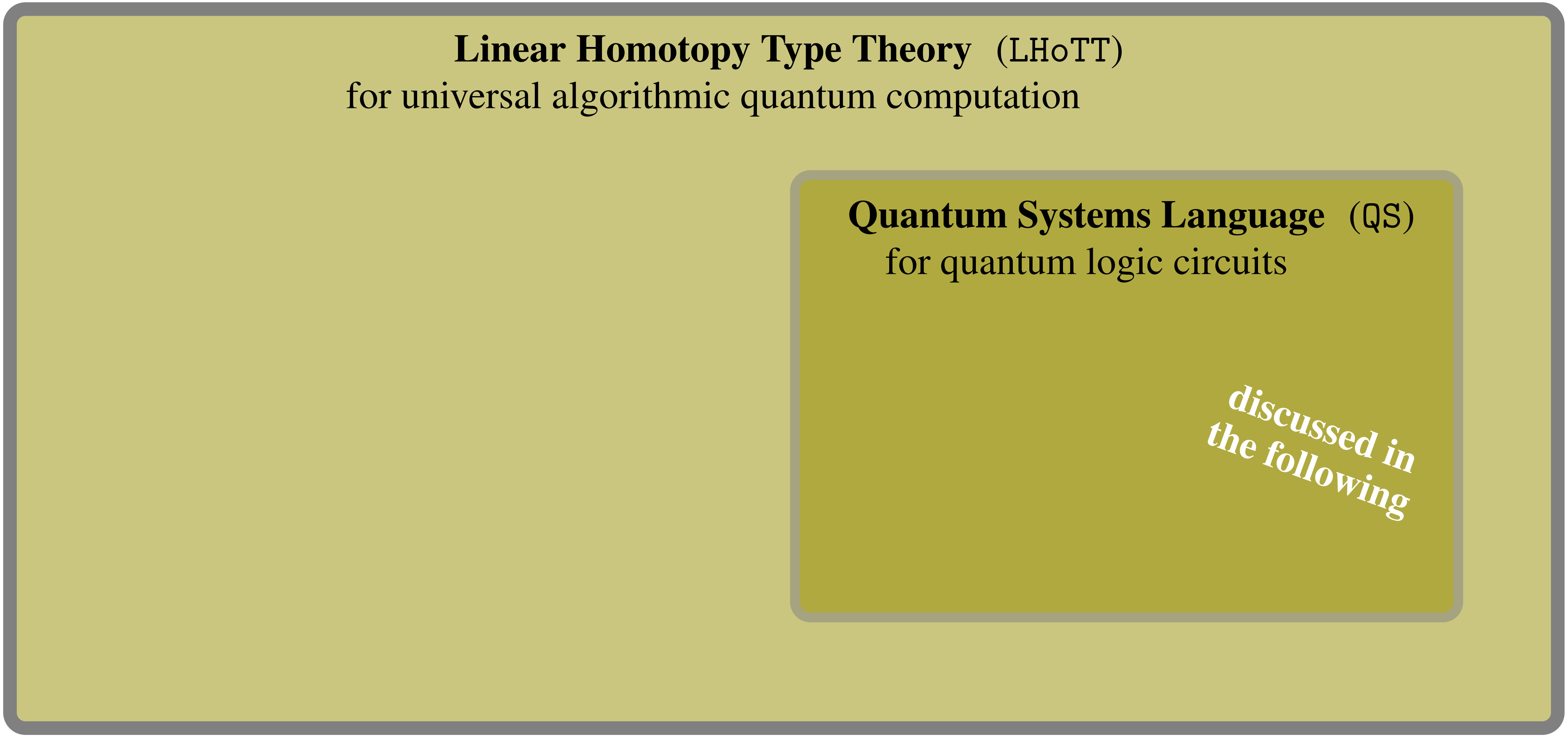
`QS` – Quantum Systems language @ CQTS

$\rightsquigarrow$ full-blown Quantum Systems language emerges embedded in `LHoTT`

Linear Homotopy Type Theory (`LHoTT`)
for universal algorithmic quantum computation

QS – Quantum Systems language @ CQTS

⇝ full-blown Quantum Systems language emerges embedded in LHoTT



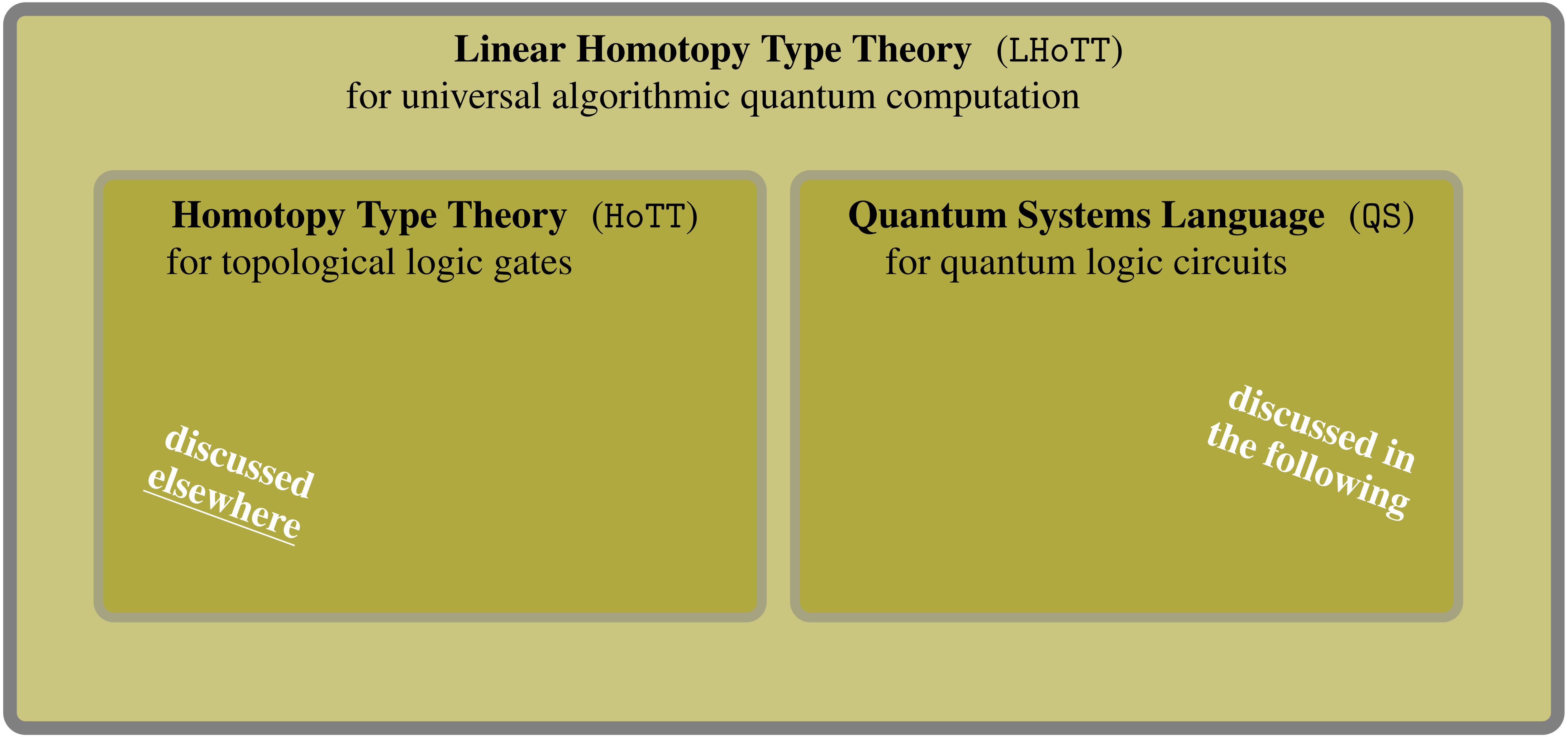
Linear Homotopy Type Theory (LHoTT)
for universal algorithmic quantum computation

Quantum Systems Language (QS)
for quantum logic circuits

discussed in the following

QS – Quantum Systems language @ CQTS

⇝ full-blown Quantum Systems language emerges embedded in LHoTT



Linear Homotopy Type Theory (LHoTT)
for universal algorithmic quantum computation

Homotopy Type Theory (HoTT)
for topological logic gates

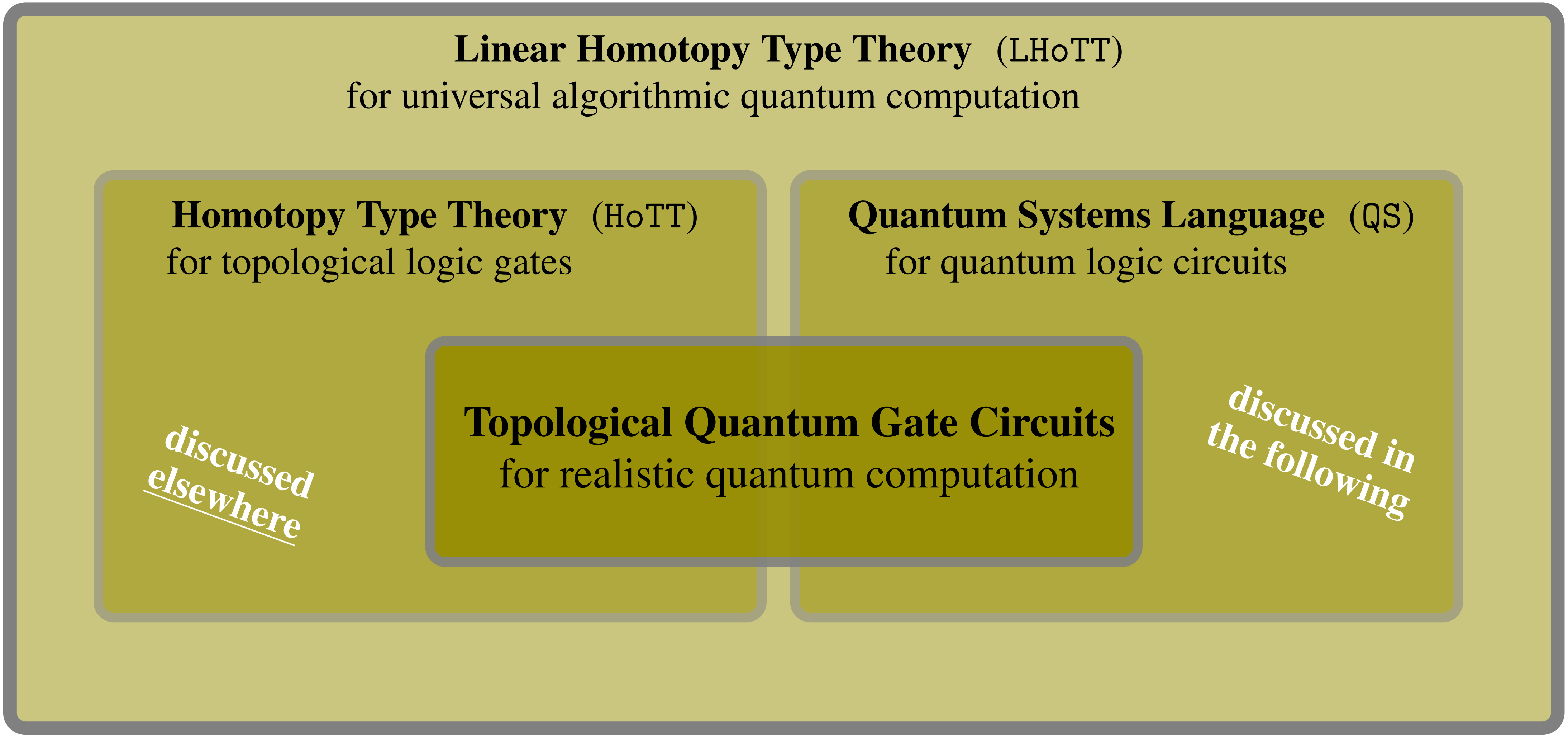
discussed elsewhere

Quantum Systems Language (QS)
for quantum logic circuits

discussed in the following

`QS` – Quantum Systems language @ CQTS

⇝ full-blown Quantum Systems language emerges embedded in `LHoTT`



Linear Homotopy Type Theory (`LHoTT`)
for universal algorithmic quantum computation

Homotopy Type Theory (`HoTT`)
for topological logic gates

Quantum Systems Language (`QS`)
for quantum logic circuits

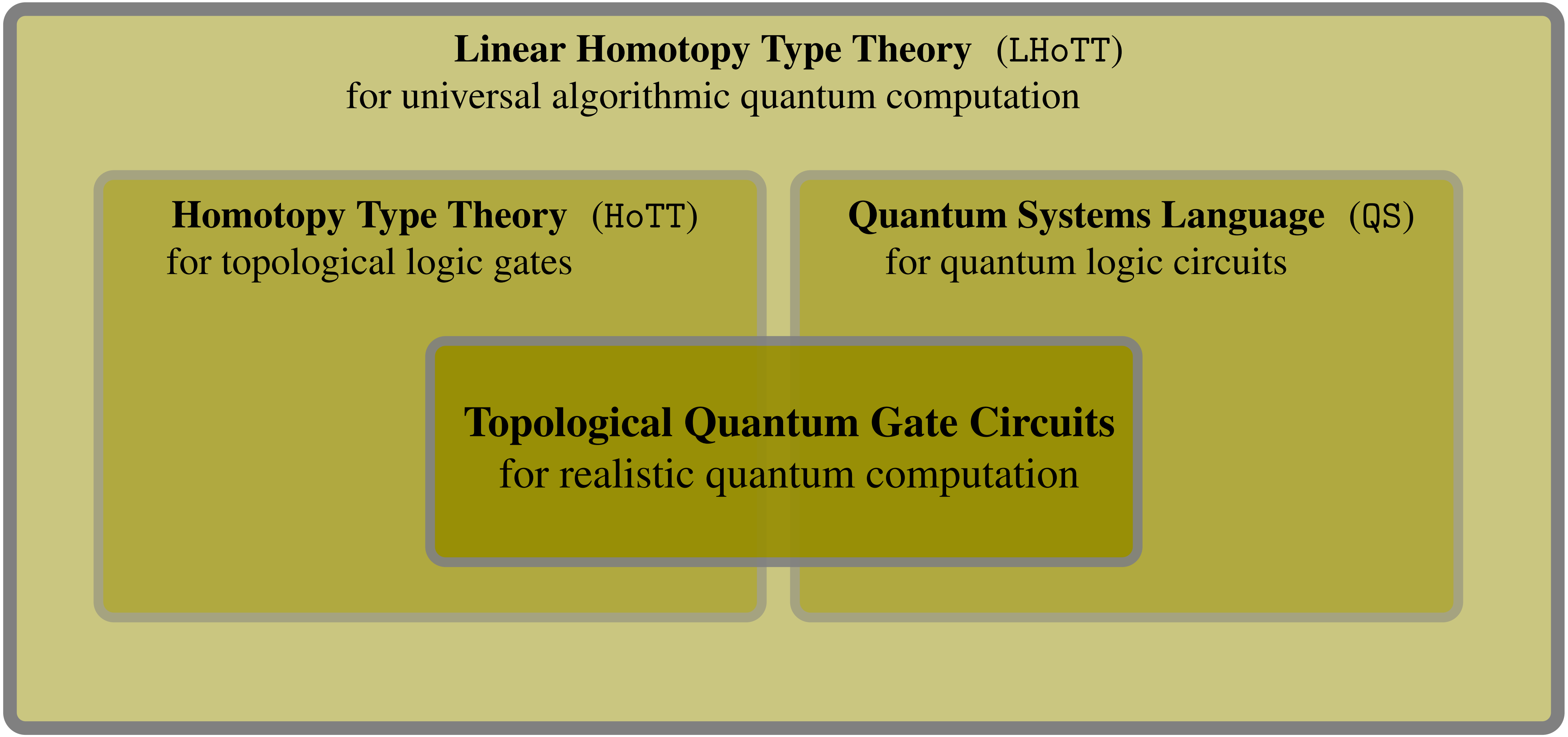
Topological Quantum Gate Circuits
for realistic quantum computation

discussed elsewhere

discussed in the following

QS – Quantum Systems language @ CQTS

⇝ full-blown Quantum Systems language emerges embedded in LHoTT



Linear Homotopy Type Theory (LHoTT)
for universal algorithmic quantum computation

Homotopy Type Theory (HoTT)
for topological logic gates

Quantum Systems Language (QS)
for quantum logic circuits

Topological Quantum Gate Circuits
for realistic quantum computation

ambient LHoTT	verifies	classically dependent quantum linear types
ambient HoTT	provides	specification of topological quantum gates
ambient dTT	provides	full verified classical control

Quantum Data Types

Linear/Quantum Data Types

<table>
<tr><td>Characteristic Property</td><td></td><td></td><td></td></tr>
<tr><td>Symbol</td><td></td><td></td><td></td></tr>
<tr><td>Formula
(for B : FinType)</td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">AlgTop Jargon</td><td rowspan="2"></td><td></td><td></td></tr>
<tr><td colspan="2"></td></tr>
<tr><td>Linear Logic</td><td></td><td></td><td></td></tr>
<tr><td>Physics Meaning</td><td></td><td></td><td></td></tr>
</table>

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:		
Symbol			
Formula (for B : FinType)			
AlgTop Jargon			
Linear Logic			
Physics Meaning			

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:		
Symbol	$\oplus$ direct sum		
Formula (for B : FinType)			
AlgTop Jargon			
Linear Logic			
Physics Meaning			

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	
Symbol	$\oplus$ direct sum		
Formula (for B : FinType)			
AlgTop Jargon			
Linear Logic			
Physics Meaning			

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	
Symbol	$\bigoplus$ direct sum	$\bigotimes$ tensor product	
Formula (for B : FinType)			
AlgTop Jargon			
Linear Logic			
Physics Meaning			

Linear/Quantum Data Types

<table>
<tr><th>Characteristic Property</th><th>1. their cartesian product blends into the co-product:</th><th>2. a tensor product appears & distributes over direct sum</th><th>3. a linear function type appears adjoint to tensor</th></tr>
<tr><td>Symbol</td><td>$\oplus$ direct sum</td><td>$\otimes$ tensor product</td><td></td></tr>
<tr><td>Formula
(for B : FinType)</td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">AlgTop Jargon</td><td rowspan="2"></td><td></td><td></td></tr>
<tr><td colspan="2"></td></tr>
<tr><td>Linear Logic</td><td></td><td></td><td></td></tr>
<tr><td>Physics Meaning</td><td></td><td></td><td></td></tr>
</table>

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)			
AlgTop Jargon			
Linear Logic			
Physics Meaning			

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$		
AlgTop Jargon			
Linear Logic			
Physics Meaning			

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\textbf{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\textbf{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\textbf{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} \big(\mathcal{V} \otimes \mathcal{H}_b\big)$	
AlgTop Jargon			
Linear Logic			
Physics Meaning			

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$
AlgTop Jargon			
Linear Logic			
Physics Meaning			

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\bigoplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$
AlgTop Jargon	biproduct, stability, ambidexterity	Frobenius reciprocity	mapping spectrum
Linear Logic			
Physics Meaning			

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\textbf{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\textbf{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\textbf{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$
AlgTop Jargon	biproduct, stability, ambidexterity	Frobenius reciprocity	mapping spectrum
		Grothendieck's Motivic Yoga of 6 oper. (Wirthmüller form)	
Linear Logic			
Physics Meaning			

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} \big(\mathcal{V} \otimes \mathcal{H}_b \big)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$
AlgTop Jargon	biproduct, stability, ambidexterity	Frobenius reciprocity	mapping spectrum
		Grothendieck's Motivic Yoga of 6 oper. (Wirthmüller form)	
Linear Logic	additive disjunction	multiplicative conjunction	linear implication
Physics Meaning			

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\textbf{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\textbf{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\textbf{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$
AlgTop Jargon	biproduct, stability, ambidexterity	Frobenius reciprocity	mapping spectrum
		Grothendieck's Motivic Yoga of 6 oper. (Wirthmüller form)	
Linear Logic	additive disjunction	multiplicative conjunction	linear implication
Physics Meaning	superselection sectors / quantum parallelism	compound quantum systems / quantum entanglement	QRAM systems

Linear/Quantum Data Types

<table>
<tr><td>Characteristic Property</td><td>1. their cartesian product blends into the co-product:</td><td>2. a tensor product appears & distributes over direct sum</td><td>3. a linear function type appears adjoint to tensor</td></tr>
<tr><td>Symbol</td><td>$\oplus$ direct sum</td><td>$\otimes$ tensor product</td><td>$\multimap$ linear function type</td></tr>
<tr><td>Formula
(for B : FinType)</td><td>$\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$</td><td>$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$</td><td>$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$
$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$</td></tr>
<tr><td rowspan="2">AlgTop Jargon</td><td rowspan="2">biproduct,
stability, ambidexterity</td><td>Frobenius reciprocity</td><td>mapping spectrum</td></tr>
<tr><td colspan="2">Grothendieck’s Motivic Yoga of 6 oper. (Wirthmüller form)</td></tr>
<tr><td>Linear Logic</td><td>additive disjunction</td><td>multiplicative conjunction</td><td>linear implication</td></tr>
<tr><td>Physics Meaning</td><td>superselection sectors
/ quantum parallelism</td><td>compound quantum systems
/ quantum entanglement</td><td>QRAM systems</td></tr>
</table>

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus\limits_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus\limits_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

Dependent linear Type Formers

finite classical context
(variables, parameters, ...)

reference context

$$\begin{tikzcd}
B \arrow[rr, "p_B"] & & \ast \\
\mathrm{ClType}_B & & \mathrm{ClType}
\end{tikzcd}$$

classical type system
dependent on context

classical
type system

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	**cart. product** **co-product** $\prod_B \mathcal{H}_b \simeq \bigoplus_B \mathcal{H}_b \simeq \coprod_B \mathcal{H}_b$ **direct sum**	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

Dependent linear Type Formers

finite classical context (variables, parameters, ...)

reference context

$$\begin{tikzcd} B \arrow[r, "p_B"] & * \end{tikzcd}$$

classical context extension

classical type system dependent on context

classical type system

$$\begin{tikzcd} \mathrm{ClType}_B & \mathrm{ClType} \arrow[l, "*_B \times"'] \end{tikzcd}$$

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	**cart. product** **co-product** $\prod_B \mathcal{H}_b \simeq \bigoplus_B \mathcal{H}_b \simeq \coprod_B \mathcal{H}_b$ **direct sum**	$\mathcal{V} \otimes \Big(\bigoplus\limits_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus\limits_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

Dependent linear Type Formers

finite classical context (variables, parameters, ...)

reference context

$$\begin{tikzcd} B \arrow[rr, "p_B"] & & * \end{tikzcd}$$

co-product

$$\begin{tikzcd} \mathrm{ClType}_B \arrow[rr, bend left, "\coprod_{b:B}"] \arrow[rr, phantom, "\perp"] & & \mathrm{ClType} \arrow[ll, bend left, "*_B \times"] \end{tikzcd}$$

classical type system dependent on context

classical type system

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\bigoplus$ direct sum	$\bigotimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	**cart. product** **co-product** $\prod_B \mathcal{H}_b \simeq \bigoplus_B \mathcal{H}_b \simeq \coprod_B \mathcal{H}_b$ **direct sum**	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

Dependent linear Type Formers

finite classical context (variables, parameters, ...)

reference context

$$\begin{tikzcd} B \arrow[r, "p_B"] & * \end{tikzcd}$$

co-product

classical type system dependent on context

$$\begin{tikzcd}[column sep=huge]
\mathrm{ClType}_B \arrow[r, shift left=5, "\coprod_{b:B}"] \arrow[r, shift right=5, "\prod_{b:B}"'] \arrow[r, phantom, shift left=2.5, "\perp"] \arrow[r, phantom, shift right=2.5, "\perp"] & \mathrm{ClType} \arrow[l, "*_B \times" description]
\end{tikzcd}$$

classical type system

product

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\bigoplus$ direct sum	$\bigotimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	**cart. product** **co-product** $\prod_B \mathcal{H}_b \simeq \bigoplus_B \mathcal{H}_b \simeq \coprod_B \mathcal{H}_b$ **direct sum**	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

Dependent linear Type Formers

finite classical context (variables, parameters, ...)

reference context

$$\begin{tikzcd} B \arrow[rr, "p_B"] & & * \end{tikzcd}$$

co-product

$$\begin{tikzcd} \mathrm{ClType}_B \arrow[rr, shift left=8, "\coprod_{b:B}"] \arrow[rr, phantom, shift left=4, "\perp"] & & \mathrm{ClType} \arrow[ll, "*_B \times" description] \arrow[ll, phantom, shift left=4, "\perp"] \\ \end{tikzcd}$$

$$\begin{tikzcd} \mathrm{ClType}_B \arrow[rr, "\prod_{b:B}"'] & & \mathrm{ClType} \end{tikzcd}$$

product

classical type system dependent on context

classical type system

classical base change / classical quantification

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\bigoplus$ direct sum	$\bigotimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

Dependent linear Type Formers

finite classical context (variables, parameters, ...)

reference context

$$\begin{tikzcd} B \arrow[rr, "p_B"] & & \ast \end{tikzcd}$$

classical type system dependent on context

co-product

$$\begin{tikzcd}[column sep=huge] \mathrm{ClType}_B \arrow[r, shift left=14pt, "\coprod_{b:B}"] \arrow[r, phantom, shift left=7pt, "\perp"] \arrow[r, phantom, shift right=7pt, "\perp"] \arrow[r, shift right=14pt, "\prod_{b:B}"'] & \mathrm{ClType} \arrow[l, "\ast_B \times"'] \end{tikzcd}$$

product

classical type system

classical base change /
classical quantification

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\bigoplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	**cart. product** **co-product** $\prod_B \mathcal{H}_b \simeq \bigoplus_B \mathcal{H}_b \simeq \coprod_B \mathcal{H}_b$ **direct sum**	$\mathcal{V} \otimes \Big(\bigoplus\limits_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus\limits_{b:B} \big(\mathcal{V} \otimes \mathcal{H}_b\big)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

Dependent linear Type Formers

finite classical context (variables, parameters, ...)

reference context

$$\begin{tikzcd}[column sep=huge] B \arrow[r, "p_B"] & * \end{tikzcd}$$

classical type system dependent on context

co-product

$$\begin{tikzcd}[column sep=huge, row sep=large] \mathrm{ClType}_B \arrow[r, shift left=5, "\coprod_{b:B}"] \arrow[r, phantom, shift left=2.5, "\perp"] \arrow[r, phantom, shift right=2.5, "\perp"] \arrow[r, shift right=5, "\prod_{b:B}"'] & \mathrm{ClType} \arrow[l, "*_B \times" description] \end{tikzcd}$$

product

classical type system

linear type system in classical context

$$\Big(\mathrm{LType}_B,\ \overset{\text{tensor}}{\otimes_B}\Big) \qquad\qquad \Big(\mathrm{LType},\ \overset{\text{tensor}}{\otimes}\Big)$$

linear type system

classical base change / classical quantification

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\bigoplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

Dependent linear Type Formers

finite classical context (variables, parameters, ...)

reference context

$$\begin{tikzcd}
B \arrow[r, "p_B"] & *
\end{tikzcd}$$

classical type system dependent on context

classical type system

$$\begin{tikzcd}[column sep=huge]
\mathrm{ClType}_B \arrow[r, shift left=10, "\overset{\text{co-product}}{\coprod_{b:B}}"] \arrow[r, phantom, shift left=5, "\perp"] \arrow[r, phantom, shift right=5, "\perp"] \arrow[r, shift right=10, "\underset{\text{product}}{\prod_{b:B}}"'] & \mathrm{ClType} \arrow[l, "{*_B \times}"']
\end{tikzcd}$$

classical base change / classical quantification

linear type system in classical context

linear type system

$$\begin{tikzcd}[column sep=huge]
\Big(\mathrm{LType}_B, \overset{\text{tensor}}{\otimes_B}\Big) & \Big(\mathrm{LType}, \overset{\text{tensor}}{\otimes}\Big) \arrow[l, "\overset{\textbf{linear context extension}}{\mathbb{1}_B \otimes}"']
\end{tikzcd}$$

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\bigoplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	**cart. product** **co-product** $\prod_B \mathcal{H}_b \simeq \bigoplus_B \mathcal{H}_b \simeq \coprod_B \mathcal{H}_b$ **direct sum**	$\mathcal{V} \otimes \Big(\bigoplus\limits_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus\limits_{b:B} \big(\mathcal{V} \otimes \mathcal{H}_b \big)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \;\; \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

Dependent linear Type Formers

finite classical context (variables, parameters, ...)

reference context

$$\begin{tikzcd}
B \arrow[r, "p_B"] & *
\end{tikzcd}$$

classical type system dependent on context

$$\begin{tikzcd}[column sep=huge]
\mathrm{ClType}_B \arrow[r, shift left=4, "\overset{\text{co-product}}{\coprod_{b:B}}"] \arrow[r, shift right=4, "\underset{\text{product}}{\prod_{b:B}}"'] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] & \mathrm{ClType} \arrow[l, "{*_B} \times" description]
\end{tikzcd}$$

classical type system

linear type system in classical context

$$\begin{tikzcd}[column sep=huge]
\Big(\mathrm{LType}_B, \overset{\text{tensor}}{\otimes_B} \Big) \arrow[r, shift left=4, "\overset{\text{direct sum}}{\bigoplus_{b:B}}"] \arrow[r, shift right=4, "\bigoplus_{b:B}"'] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] & \Big(\mathrm{LType}, \overset{\text{tensor}}{\otimes} \Big) \arrow[l, "\mathbb{1}_B \otimes" description]
\end{tikzcd}$$

linear type system

classical base change / classical quantification

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\textbf{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\textbf{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\textbf{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \; \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

Dependent linear Type Formers

finite classical context (variables, parameters, ...)

reference context

$$\begin{tikzcd} B \arrow[r, "p_B"] & \ast \end{tikzcd}$$

classical type system dependent on context

$$\begin{tikzcd}[column sep=huge] \mathrm{ClType}_B \arrow[r, shift left=4, "\overset{\text{co-product}}{\coprod_{b:B}}"] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] \arrow[r, shift right=4, "\underset{\text{product}}{\prod_{b:B}}"'] & \mathrm{ClType} \arrow[l, "\ast_B \times" description] \end{tikzcd}$$

classical type system

classical base change / classical quantification

linear type system in classical context

$$\begin{tikzcd}[column sep=huge] \big(\mathrm{LType}_B, \overset{\text{tensor}}{\otimes_B}\big) \arrow[r, shift left=4, "\overset{\text{direct sum}}{\bigoplus_{b:B}}"] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] \arrow[r, shift right=4, "\bigoplus_{b:B}"'] & \big(\mathrm{LType}, \overset{\text{tensor}}{\otimes}\big) \arrow[l, "\mathbb{1}_B \otimes" description] \end{tikzcd}$$

linear type system

quantum base change / Motivic Yoga

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

Dependent linear Type Formers

finite classical context (variables, parameters, ...)

reference context

$$\begin{tikzcd}[column sep=huge] B \arrow[r, "p_B"] & * \end{tikzcd}$$

classical type system dependent on context

$$\begin{tikzcd}[column sep=huge] \mathrm{ClType}_B \arrow[r, shift left=8, "\overset{\text{co-product}}{\coprod_{b:B}}"] \arrow[r, phantom, shift left=4, "\perp"] \arrow[r, phantom, shift right=4, "\perp"] \arrow[r, shift right=8, "\underset{\text{product}}{\prod_{b:B}}"'] & \mathrm{ClType} \arrow[l, "*_B \times"'] \end{tikzcd}$$

classical type system

classical base change / classical quantification

linear type system in classical context

$$\begin{tikzcd}[column sep=huge] \big(\mathrm{LType}_B, \overset{\text{tensor}}{\otimes_B}\big) \arrow[r, shift left=8, "\overset{\text{direct sum}}{\bigoplus_{b:B}}"] \arrow[r, phantom, shift left=4, "\perp"] \arrow[r, phantom, shift right=4, "\perp"] \arrow[r, shift right=8, "\bigoplus_{b:B}"'] & \big(\mathrm{LType}, \overset{\text{tensor}}{\otimes}\big) \arrow[l, "\mathbb{1}_B \otimes"'] \end{tikzcd}$$

linear type system

quantum base change / Motivic Yoga

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\oplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

all available in LHoTT

Dependent linear Type Formers

finite classical context (variables, parameters, ...)

reference context

$$\begin{tikzcd}[column sep=huge]
B \arrow[r, "p_B"] & *
\end{tikzcd}$$

classical type system dependent on context

$$\begin{tikzcd}[column sep=huge]
\mathrm{ClType}_B \arrow[r, shift left=18pt, "\overset{\text{co-product}}{\coprod_{b:B}}"] \arrow[r, shift right=18pt, "\underset{\text{product}}{\prod_{b:B}}"'] \arrow[r, phantom, shift left=9pt, "\perp"] \arrow[r, phantom, shift right=9pt, "\perp"] & \mathrm{ClType} \arrow[l, "{*_B} \times"']
\end{tikzcd}$$

classical type system

classical base change / classical quantification

linear type system in classical context

$$\begin{tikzcd}[column sep=huge]
\big(\mathrm{LType}_B, \overset{\text{tensor}}{\otimes_B}\big) \arrow[r, shift left=18pt, "\overset{\text{direct sum}}{\bigoplus_{b:B}}"] \arrow[r, shift right=18pt, "\bigoplus_{b:B}"'] \arrow[r, phantom, shift left=9pt, "\perp"] \arrow[r, phantom, shift right=9pt, "\perp"] & \big(\mathrm{LType}, \overset{\text{tensor}}{\otimes}\big) \arrow[l, "\mathbb{1}_B \otimes"']
\end{tikzcd}$$

linear type system

quantum base change / Motivic Yoga

Linear/Quantum Data Types

Characteristic Property	**1.** their cartesian product blends into the co-product:	**2.** a tensor product appears & distributes over direct sum	**3.** a linear function type appears adjoint to tensor
Symbol	$\bigoplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathcal{V} \otimes \Big(\bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} \big(\mathcal{V} \otimes \mathcal{H}_b \big)$	$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$ $\simeq \;\; \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$

Dependent linear Type Formers

all available in LHoTT

finite classical context (variables, parameters, ...)

reference context

$$B \xrightarrow{\;\;p_B\;\;} *$$

co-product

classical type syst
dependent on con

Observation: Induces quantum effects on linear types. ⟶

Type

classical type system

classical base change / classical quantification

linear type system in classical context

$$\begin{tikzcd}
\Big(\mathrm{LType}_B, \overset{\text{tensor}}{\otimes_B}\Big) \arrow[r, shift left=12, "\overset{\text{direct sum}}{\bigoplus_{b:B}}"] \arrow[r, phantom, shift left=6, "\perp"] \arrow[r, leftarrow, "\mathbb{1}_B \otimes" description] \arrow[r, phantom, shift right=6, "\perp"] \arrow[r, shift right=12, "\bigoplus_{b:B}" description] & \Big(\mathrm{LType}, \overset{\text{tensor}}{\otimes}\Big)
\end{tikzcd}$$

linear type system

quantum base change / Motivic Yoga

Quantum Effects

Recall: **Monadic computational effects.**

A *monad* $\mathscr{E}(-)$ on a data type system encodes *computational effects*:

effectful program

$$D_1 \xrightarrow{\mathrm{prog}_{12}} \mathscr{E}(D_2)$$

output data of nominal type D_2
causing effects of type $\mathscr{E}(-)$

Recall: **Monadic computational effects.**

A *monad* $\mathscr{E}(-)$ on a data type system encodes *computational effects*:

first program

$$D_1 \xrightarrow{\mathrm{prog}_{12}} \mathscr{E}(D_2)$$

output data of nominal type D_2
causing effects of type $\mathscr{E}(-)$

second program

$$D_2 \xrightarrow{\mathrm{prog}_{23}} \mathscr{E}(D_3)$$

input data of type D_2
causing effects of type $\mathscr{E}(-)$

Recall: **Monadic computational effects.**

A *monad* $\mathscr{E}(-)$ on a data type system encodes *computational effects*:

first program

second program

$$\begin{tikzcd}
D_1 \arrow[r, "\mathrm{prog}_{12}"] & \mathscr{E}(D_2) & D_2 \arrow[r, "\mathrm{prog}_{23}"] & \mathscr{E}(D_3) \arrow[d, mapsto, bend left, "\text{bind previous effects into second program}"] \\
D_1 \arrow[r, "\mathrm{prog}_{12}"] & \mathscr{E}(D_2) & \mathscr{E}(D_2) \arrow[r, "\mathrm{bind}^{\mathscr{E}}\,\mathrm{prog}_{23}"] & \mathscr{E}(D_3)
\end{tikzcd}$$

output data of nominal type D_2
causing effects of type $\mathscr{E}(-)$

input data of type D_2
causing effects of type $\mathscr{E}(-)$

bind previous effects
into second program

carry any previous $\mathscr{E}(-)$-effects along

Recall: **Monadic computational effects.**

A *monad* $\mathscr{E}(-)$ on a data type system encodes *computational effects*:

first program

second program

$$\begin{tikzcd}
D_1 \arrow[r, "\mathrm{prog}_{12}"] & \mathscr{E}(D_2) & D_2 \arrow[r, "\mathrm{prog}_{23}"] & \mathscr{E}(D_3) \arrow[d, mapsto, bend left, "\text{bind previous effects into second program}"] \\
D_1 \arrow[r, "\mathrm{prog}_{12}"] & \mathscr{E}(D_2) & \mathscr{E}(D_2) \arrow[r, "\mathrm{bind}^{\mathscr{E}}\,\mathrm{prog}_{23}"] & \mathscr{E}(D_3) \arrow[d, mapsto, bend left, "\text{compose}"] \\
D_1 \arrow[rrr, "\mathrm{bind}^{\mathscr{E}}\,\mathrm{prog}_{23}\circ\mathrm{prog}_{12}"] & & & \mathscr{E}(D_3)
\end{tikzcd}$$

output data of nominal type D_2
causing effects of type $\mathscr{E}(-)$

input data of type D_2
causing effects of type $\mathscr{E}(-)$

bind previous effects
into second program

carry any previous $\mathscr{E}(-)$-effects along

compose

$\mathscr{E}$-composite program

causing cumulative $\mathscr{E}(-)$-effects

Recall: **Monadic effect handlers.**

$$D_1 \xrightarrow[\textbf{in-effectful program}]{\text{prog}_{12}} D_2 \quad \textbf{data type to absorb } \mathscr{E}\textbf{-effects}$$

Recall: **Monadic effect handlers.**

$$\begin{tikzcd}
D_1 \arrow[r, "\mathrm{prog}_{12}", "\textbf{in-effectful program}"'] & D_2 & {} \arrow[d, mapsto, bend left, "\substack{\textbf{incorporate handling}\\ \textbf{of } \mathscr{E}(-)\textbf{-effects}}"] \\
\mathscr{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathscr{E}}_{D_2}\,\mathrm{prog}_{12}", "\substack{\textbf{in-effectful program}\\ \textbf{handling effects of type } \mathscr{E}(-)}"'] & D_2 & {}
\end{tikzcd}$$

Recall: **Monadic effect handlers.**

$$\begin{tikzcd}
D_1 \arrow[r, "\mathrm{prog}_{12}", "\textbf{in-effectful program}"'] & D_2 \arrow[d, mapsto, bend left, "\substack{\textbf{incorporate handling}\\ \textbf{of } \mathscr{E}(-)\textbf{-effects}}"] \\
\mathscr{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathscr{E}}_{D_2}\mathrm{prog}_{12}", "\substack{\textbf{in-effectful program}\\ \textbf{handling effects of type } \mathscr{E}(-)}"'] & D_2
\end{tikzcd}$$

$$\begin{tikzcd}
D_1 \arrow[r, "\mathrm{ret}^{\mathscr{E}}_{D_1}", "\substack{\textbf{produce}\\ \textbf{trivial effect}}"'] \arrow[rr, bend right=40, "\substack{\mathrm{prog}_{12}\\ \textbf{no effect}}"'] & \mathscr{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathscr{E}}_{D_2}\mathrm{prog}_{12}", "\substack{\textbf{handle effects}\\ \textbf{running program}}"'] & D_2
\end{tikzcd}$$

consistency conditions

Recall: **Monadic effect handlers.**

$$\begin{tikzcd}[column sep=huge] D_1 \arrow[r, "\mathrm{prog}_{12}", "\text{in-effectful program}"'] & D_2 \end{tikzcd}$$

$\mapsto$ incorporate handling of $\mathscr{E}(-)$-effects

$$\begin{tikzcd}[column sep=huge] \mathscr{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathscr{E}}_{D_2}\mathrm{prog}_{12}", "\text{in-effectful program handling effects of type } \mathscr{E}(-)"'] & D_2 \end{tikzcd}$$

$$\begin{tikzcd}[column sep=huge] D_1 \arrow[r, "\mathrm{ret}^{\mathscr{E}}_{D_1}", "\text{produce trivial effect}"'] \arrow[rr, bend right, "\mathrm{prog}_{12}"', "\text{no effect}"] & \mathscr{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathscr{E}}_{D_2}\mathrm{prog}_{12}", "\text{handle effects running program}"'] & D_2 \end{tikzcd}$$

consistency conditions

$$\begin{tikzcd}[column sep=huge] \mathscr{E}(D_0) \arrow[r, "\mathrm{bind}^{\mathscr{E}}\mathrm{prog}_{01}", "\text{carry effects along}"'] \arrow[rr, bend right] & \mathscr{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathscr{E}}_{D_2}\mathrm{prog}_{12}", "\text{handle cumulative effects}"'] & D_2 \end{tikzcd}$$

$$\mathrm{hndl}^{\mathscr{E}}_{D_2}\Big(D_0 \xrightarrow{\mathrm{prog}_{01}} \mathscr{E}(D_1) \xrightarrow{\mathrm{hndl}^{\mathscr{E}}_{D_2}\mathrm{prog}_{12}} D_2\Big)$$

handle effects... consecutively

Recall: **Data type system of Monadic effect handlers.**

$$\begin{tikzcd}[column sep=huge, row sep=huge]
D_1 \arrow[r, "\mathrm{prog}_{12}", "\textbf{in-effectful program}"'] & D_2 \arrow[d, mapsto, bend left, "\textbf{incorporate handling of } \mathscr{E}(-)\textbf{-effects}"] \\
\mathscr{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathscr{E}}_{D_2}\mathrm{prog}_{12}", "\substack{\textbf{in-effectful program} \\ \textbf{handling effects of type } \mathscr{E}(-)}"'] & D_2 \quad \text{“}\mathscr{E}\textbf{-modal data type}\text{”}
\end{tikzcd}$$

Monadicity:

$\mathscr{E}$-modales in Type
(“EM-category”)

$\mathrm{Type}^{\mathscr{E}}$

Recall: **Data type system of Monadic effect handlers.**

$$\begin{tikzcd}
D_1 \arrow[r, "\mathrm{prog}_{12}"] & D_2 \arrow[d, mapsto, bend left, "\text{incorporate handling of } \mathscr{E}(-)\text{-effects}"] \\
\mathscr{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathscr{E}}_{D_2}\mathrm{prog}_{12}"] & D_2
\end{tikzcd}$$

in-effectful program

incorporate handling of $\mathscr{E}(-)$-effects

“$\mathscr{E}$-modal data type”

in-effectful program handling effects of type $\mathscr{E}(-)$

Monadicity:

$$\begin{tikzcd}
\mathrm{Type} \arrow[loop right, "\mathscr{E}"]
\end{tikzcd}$$

monad

$\mathscr{E}$-modales in Type (“EM-category”) $\mathrm{Type}^{\mathscr{E}}$

Recall: **Data type system of Monadic effect handlers.**

$$\begin{tikzcd}
D_1 \arrow[r, "\mathrm{prog}_{12}", "\text{in-effectful program}"'] & D_2 \arrow[d, mapsto, bend left, "\text{incorporate handling of } \mathscr{E}(-)\text{-effects}"] & \\
\mathscr{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathscr{E}}_{D_2}\,\mathrm{prog}_{12}", "\text{in-effectful program handling effects of type } \mathscr{E}(-)"'] & D_2 & \text{``}\mathscr{E}\text{-modal data type''}
\end{tikzcd}$$

Monadicity:

$$\begin{tikzcd}
 & \mathrm{Type} \arrow[loop right, "\mathscr{E}"] \arrow[dl, shift right=2, "F^{\mathscr{E}}"'] \arrow[dl, phantom, "\perp"] & \text{induced monad} \\
\text{$\mathscr{E}$-modales in Type (``EM-category'')} \quad \mathrm{Type}^{\mathscr{E}} \arrow[ur, shift right=2, "U^{\mathscr{E}}"', "\text{monadic adjunction}"'] & &
\end{tikzcd}$$

Recall: **Data type system of Monadic effect handlers.**

$$\begin{tikzcd}
D_1 \arrow[r, "\mathrm{prog}_{12}", "\text{in-effectful program}"'] & D_2 \arrow[d, maps to, bend left, "\text{incorporate handling of } \mathscr{E}(-)\text{-effects}"] \\
\mathscr{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathscr{E}}_{D_2}\mathrm{prog}_{12}", "\text{in-effectful program handling effects of type } \mathscr{E}(-)"'] & D_2 \;\; \text{“}\mathscr{E}\text{-modal data type”}
\end{tikzcd}$$

Monadicity:

$$\begin{tikzcd}
\text{any adjunction on type system} & \mathrm{Type}' \arrow[rr, shift right, "U"'] \arrow[rr, phantom, "\perp"] \arrow[d, "K^{UF}", "\text{terminal comparison functor}"'] & & \mathrm{Type} \arrow[ll, shift right, "F"'] \arrow[dll, shift right, "F^{\mathscr{E}}"'] \arrow[loop right, "\mathscr{E}"] & \text{induced monad} \\
\mathscr{E}\text{-modales in } \mathrm{Type} \; (\text{“EM-category”}) & \mathrm{Type}^{\mathscr{E}} \arrow[urr, shift right, "U^{\mathscr{E}}"'] \arrow[urr, phantom, "\perp"] \arrow[urr, phantom, "\text{monadic adjunction}"', shift right=3ex] & & &
\end{tikzcd}$$

Recall: **Data type system of Monadic effect handlers.**

$$\begin{tikzcd}
D_1 \arrow[r, "\mathrm{prog}_{12}", "\text{in-effectful program}"'] & D_2 \arrow[d, mapsto, bend left, "\begin{array}{c}\text{incorporate handling}\\ \text{of } \mathscr{E}(-)\text{-effects}\end{array}"] \\
\mathscr{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathscr{E}}_{D_2}\mathrm{prog}_{12}", "\begin{array}{c}\text{in-effectful program}\\ \text{handling effects of type } \mathscr{E}(-)\end{array}"'] & D_2 \quad \text{“}\mathscr{E}\text{-modal data type”}
\end{tikzcd}$$

Monadicity:

$$\begin{tikzcd}
\begin{array}{r}\text{free } \mathscr{E}\text{-modales in Type}\\ \text{(“Kleisli category”)}\end{array} & \mathrm{Type}_{\mathscr{E}} \arrow[d, hook, "K_{UF}", "\begin{array}{c}\text{initial}\\ \text{comparison}\\ \text{functor}\end{array}"'] \arrow[dr, shift right, "U_{\mathscr{E}}"'] \arrow[dr, phantom, "\perp"] & \\
\begin{array}{r}\text{any adjunction}\\ \text{on type system}\end{array} & \mathrm{Type}' \arrow[d, "K^{UF}", "\begin{array}{c}\text{terminal}\\ \text{comparison}\\ \text{functor}\end{array}"'] \arrow[r, shift right, "U"'] \arrow[r, phantom, "\perp"] & \mathrm{Type} \arrow[l, shift right, "F"'] \arrow[ul, two heads, shift right, "F^{\mathscr{E}}"'] \arrow[dl, shift right, "F^{\mathscr{E}}"'] \arrow[loop right, "\mathscr{E}"] \quad \text{induced monad} \\
\begin{array}{r}\mathscr{E}\text{-modales in Type}\\ \text{(“EM-category”)}\end{array} & \mathrm{Type}^{\mathscr{E}} \arrow[ur, shift right, "U^{\mathscr{E}}"', "\text{monadic adjunction}"] \arrow[ur, phantom, "\perp"] &
\end{tikzcd}$$

Recall: **Data type system of Monadic effect handlers.**

$$\begin{tikzcd}
D_1 \arrow[r, "\mathrm{prog}_{12}"] & D_2 \arrow[d, mapsto, bend left] \\
\mathscr{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathscr{E}}_{D_2}\mathrm{prog}_{12}"] & D_2
\end{tikzcd}$$

in-effectful program

incorporate handling of $\mathscr{E}(-)$-effects

"$\mathscr{E}$-modal data type"

in-effectful program handling effects of type $\mathscr{E}(-)$

Monadicity:

$$\begin{tikzcd}
\mathrm{Type}_{\mathscr{E}} \arrow[d, hook, "K_{UF}"] \arrow[drr, shift left, "U_{\mathscr{E}}"] & & \\
\mathrm{Type}' \arrow[d, "K^{UF}"] \arrow[rr, shift right, "U"'] & & \mathrm{Type} \arrow[ll, shift right, "F"'] \arrow[ull, two heads, shift left, "F_{\mathscr{E}}"] \arrow[ull, phantom, "\perp"] \arrow[ll, phantom, "\perp"] \arrow[dll, shift right, "F^{\mathscr{E}}"'] \arrow[dll, phantom, "\perp"] \arrow[loop right, "\mathscr{E}"] \\
\mathrm{Type}^{\mathscr{E}} \arrow[urr, shift right, "U^{\mathscr{E}}"'] & &
\end{tikzcd}$$

free $\mathscr{E}$-modales in Type
("Kleisli category")

initial comparison functor

any adjunction
on type system

terminal comparison functor

$\mathscr{E}$-modales in Type
("EM-category")

monadic adjunction

induced monad

Now just to work this out
for the **effects induced by
dependent data type formers**
in LHoTT →

Given B : ClType of possible measurement outcomes (“possible worlds”) **the monadic effects of B-dependent data type formers** constitute modalities of **actual and potential B-measurements**:

classical data types

possibly **randomly**

$$\begin{tikzcd}
\textbf{actual data} & \perp & \mathrm{Type}_B \arrow[loop above, "\Diamond_B"] \arrow[loop below, "\Box_B"'] \arrow[rr, shift left=4, "\coprod_B"] \arrow[rr, phantom, shift left=2, "\perp"] \arrow[rr, phantom, shift right=2, "\perp"] \arrow[rr, shift right=4, "\prod_B"'] & & \mathrm{Type} \arrow[ll, "\times B"] \arrow[loop above, "\star_B"'] \arrow[loop below, "\bigcirc_B"] & \perp & \textbf{potential data}
\end{tikzcd}$$

necessarily **indefinitely**

Given B : ClType of possible measurement outcomes (“possible worlds”) **the monadic effects of B-dependent data type formers** constitute modalities of **actual and potential B-measurements**:

classical data types

possibly **randomly**

actual data **potential data**

$$\begin{tikzcd}
\mathrm{Type}_B \arrow[loop above, "\Diamond_B"] \arrow[loop below, "\Box_B"'] \arrow[r, shift left=6, "\coprod_B"] \arrow[r, phantom, shift left=3, "\perp"] \arrow[r, phantom, shift right=3, "\perp"] \arrow[r, shift right=6, "\prod_B"'] & \mathrm{Type} \arrow[l, "\times B"'] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"']
\end{tikzcd}$$

$\Diamond_B \dashv \Box_B$ $\quad$ $\star_B \dashv \bigcirc_B$

necessarily **indefinitely**

---

necessarily $P_\bullet$

$$\Box_B P_\bullet$$

$$b : B \;\vdash\; \prod_{b':B} P_{b'}$$

Given $B : \mathrm{ClType}$ of possible measurement outcomes (“possible worlds”) **the monadic effects of B-dependent data type formers** constitute modalities of **actual and potential B-measurements**:

classical data types

possibly **randomly**

actual data **potential data**

$$\begin{tikzcd}
\mathrm{Type}_B \arrow[loop above, "\lozenge_B"] \arrow[loop below, "\square_B"] \arrow[r, shift left=4, "\coprod_B"] \arrow[r, shift right=4, "\prod_B"'] & \mathrm{Type} \arrow[l, "\times B" description] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"]
\end{tikzcd}$$

$\lozenge_B \dashv \square_B$, $\coprod_B \dashv \times B \dashv \prod_B$, $\star_B \dashv \bigcirc_B$

necessarily **indefinitely**

---

necessarily $P_\bullet$ **entails** **actually $P_\bullet$**

$$\square_B P_\bullet \xrightarrow{\;\epsilon^{\square_B}_{P_\bullet}\;} P_\bullet$$

$$b : B \;\vdash\; \prod_{b':B} P_{b'} \xrightarrow{(p_{b'})_{b':B} \,\mapsto\, p_b} P_b$$

Given $B : \mathrm{ClType}$ of possible measurement outcomes (“possible worlds”) **the monadic effects of B-dependent data type formers** constitute modalities of **actual and potential B-measurements**:

classical data types

possibly **randomly**

actual data **potential data**

$$\begin{tikzcd}
\mathrm{Type}_B \arrow[loop above, "\lozenge_B"] \arrow[loop below, "\Box_B"'] \arrow[r, shift left=3, "\coprod_B"] \arrow[r, phantom, shift left=1.5, "\perp"] \arrow[r, phantom, shift right=1.5, "\perp"] \arrow[r, shift right=3, "\prod_B"'] & \mathrm{Type} \arrow[l, "\times B" description] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"']
\end{tikzcd}$$

$\lozenge_B \perp \Box_B$ $\qquad$ $\star_B \perp \bigcirc_B$

necessarily **indefinitely**

necessarily $P_\bullet$ **entails** **actually $P_\bullet$** **entails** **possibly $P_\bullet$**

$$\begin{tikzcd}
\Box_B P_\bullet \arrow[r, "\epsilon^{\Box_B}_{P_\bullet}" description] & P_\bullet \arrow[r, "\eta^{\lozenge_B}_{P_\bullet}" description] & \lozenge_B P_\bullet
\end{tikzcd}$$

$$b : B \;\vdash\; \begin{tikzcd}
\prod\limits_{b':B} P_{b'} \arrow[r, "(p_{b'})_{b':B} \mapsto p_b"] & P_b \arrow[r, "p_b \mapsto (b, p_b)"] & \coprod\limits_{b':B} P_{b'}
\end{tikzcd}$$

Given B : ClType of possible measurement outcomes (“possible worlds”) **the monadic effects of B-dependent data type formers** constitute modalities of **actual and potential B-measurements**:

classical data types

possibly **randomly**

$$\begin{tikzcd}
\textbf{actual data} & \perp & \mathrm{Type}_B \arrow[loop above, "\Diamond_B"] \arrow[loop below, "\Box_B"'] \arrow[rr, shift left=3, "\coprod_B"] \arrow[rr, shift right=3, "\prod_B"'] & \begin{array}{c}\perp\\ \\ \perp\end{array} & \mathrm{Type} \arrow[ll, "\times B" description] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"'] & \perp & \textbf{potential data}
\end{tikzcd}$$

necessarily **indefinitely**

---

necessarily $P_\bullet$ **entails** **actually $P_\bullet$** **entails** **possibly $P_\bullet$**

$$\begin{tikzcd}
\Box_B P_\bullet \arrow[r, "\epsilon^{\Box_B}_{P_\bullet}"] & P_\bullet \arrow[r, "\eta^{\Diamond_B}_{P_\bullet}"] & \Diamond_B P_\bullet
\end{tikzcd}$$

$$\begin{tikzcd}
b : B \;\vdash\; \prod\limits_{b':B} P_{b'} \arrow[r, "(p_{b'})_{b':B} \mapsto p_b"] & P_b \arrow[r, "p_b \mapsto (b, p_b)"] & \coprod\limits_{b':B} P_{b'}
\end{tikzcd}$$

randomly P

$$\star_B P$$

$$\coprod_{b:B} P$$

Given B : ClType of possible measurement outcomes ("possible worlds") **the monadic effects of B-dependent data type formers** constitute modalities of **actual and potential B-measurements**:

classical data types

possibly **randomly**

actual data $\perp$ **potential data** $\perp$

necessarily **indefinitely**

$$\begin{tikzcd}
\mathrm{Type}_B \arrow[loop, out=150, in=90, "\Diamond_B"] \arrow[loop, out=270, in=210, "\Box_B"'] \arrow[r, shift left=5, "\coprod_B"] \arrow[r, phantom, shift left=2.5, "\perp"] \arrow[r, phantom, shift right=2.5, "\perp"] \arrow[r, shift right=5, "\prod_B"'] & \mathrm{Type} \arrow[l, "\times B"'] \arrow[loop, out=90, in=30, "\star_B"] \arrow[loop, out=330, in=270, "\bigcirc_B"']
\end{tikzcd}$$

necessarily $P_\bullet$ **entails** **actually $P_\bullet$** **entails** **possibly $P_\bullet$**

$$\Box_B P_\bullet \xrightarrow{\epsilon^{\Box_B}_{P_\bullet}} P_\bullet \xrightarrow{\eta^{\Diamond_B}_{P_\bullet}} \Diamond_B P_\bullet$$

$$b : B \;\vdash\; \prod_{b':B} P_{b'} \xrightarrow{(p_{b'})_{b':B} \mapsto p_b} P_b \xrightarrow{p_b \mapsto (b, p_b)} \coprod_{b':B} P_{b'}$$

randomly P **entails** **potentially P**

$$\star_B P \xrightarrow{\epsilon^{\star_B}_{P}} P$$

$$\coprod_{b:B} P \xrightarrow{(b,p) \mapsto p} P$$

Given $B : \mathrm{ClType}$ of possible measurement outcomes (“possible worlds”) **the monadic effects of B-dependent data type formers** constitute modalities of **actual and potential B-measurements**:

classical data types

possibly **randomly**

actual data **potential data**

$$\begin{tikzcd}[column sep=huge]
\mathrm{Type}_B \arrow[loop above, "\Diamond_B"] \arrow[loop below, "\Box_B"'] \arrow[r, shift left=6, "\coprod_B"] \arrow[r, shift right=6, "\prod_B"'] & \mathrm{Type} \arrow[l, "\times B" description] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"']
\end{tikzcd}$$

$$\Diamond_B \dashv \Box_B \qquad \coprod_B \dashv \times B \dashv \prod_B \qquad \star_B \dashv \bigcirc_B$$

necessarily **indefinitely**

necessarily $P_\bullet$ **entails** **actually $P_\bullet$** **entails** **possibly $P_\bullet$**

$$\Box_B P_\bullet \xrightarrow{\;\epsilon^{\Box_B}_{P_\bullet}\;} P_\bullet \xrightarrow{\;\eta^{\Diamond_B}_{P_\bullet}\;} \Diamond_B P_\bullet$$

$$b : B \;\vdash\; \prod_{b':B} P_{b'} \xrightarrow{(p_{b'})_{b':B} \,\mapsto\, p_b} P_b \xrightarrow{p_b \,\mapsto\, (b, p_b)} \coprod_{b':B} P_{b'}$$

randomly P **entails** **potentially P** **entails** **indefinitely P**

$$\star_B P \xrightarrow{\;\epsilon^{\star_B}_{P}\;} P \xrightarrow{\;\eta^{\bigcirc_B}_{P}\;} \bigcirc_B P$$

$$\coprod_{b:B} P \xrightarrow{(b,p) \,\mapsto\, p} P \xrightarrow{p \,\mapsto\, (p)_{b:B}} \prod_{b:B} P$$

Given B : ClType of possible measurement outcomes (“possible worlds”) **the monadic effects of B-dependent linear data type formers** constitute modalities of **actual and potential quantum B-measurements**.

quantum data types

possibly **randomly**

actual data $\perp$ **potential data** $\perp$

$$\begin{tikzcd}
 & (\mathrm{LType}_B)_{\Diamond_B} \arrow[r, hook, "\sim"] \arrow[d, phantom, "\perp"] & (\mathrm{LType}_B)^{\Diamond_B} \arrow[dr, dash, "\sim"] & \\
\mathrm{LType}_B \simeq \mathrm{LType}^{\bigcirc_B} \arrow[loop above, "\Diamond_B"] \arrow[loop below, "\Box_B"'] \arrow[ur, two heads, "\oplus_B"] \arrow[dr, two heads, "\oplus_B"'] & \mathrm{LType}_{\bigcirc_B} \arrow[l, hook'] \arrow[d, phantom, "\perp"] & & \mathrm{LType} \arrow[ll, two heads, "\mathbb{1}_B \otimes"'] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"'] \\
 & (\mathrm{LType}_B)_{\Box_B} \arrow[r, hook, "\sim"'] & (\mathrm{LType}_B)^{\Box_B} \arrow[ur, dash, "\sim"'] &
\end{tikzcd}$$

necessarily **indefinitely**

Given B : ClType of possible measurement outcomes ("possible worlds") **the monadic effects of B-dependent linear data type formers** constitute modalities of **actual and potential quantum B-measurements**.

quantum data types

$$\begin{tikzcd}
 & & & (\mathrm{LType}_B)_{\lozenge_B} \arrow[r, hook, "\sim"] \arrow[d, phantom, "\perp"] & (\mathrm{LType}_B)^{\lozenge_B} \arrow[dr, phantom, "\simeq" sloped] & & \\
\textbf{actual data} \perp & \mathrm{LType}_B \simeq \mathrm{LType}^{\bigcirc_B} \arrow[loop above, "\overset{\textbf{possibly}}{\lozenge_B}"] \arrow[loop below, "\underset{\textbf{necessarily}}{\Box_B}"'] \arrow[urr, two heads, "\oplus_B"] \arrow[drr, two heads, "\oplus_B"'] & & \mathrm{LType}_{\bigcirc_B} \arrow[ll, hook'] \arrow[d, phantom, "\perp"] & & \mathrm{LType} \arrow[ll, two heads, "\mathbb{1}_B \otimes"'] \arrow[loop above, "\overset{\textbf{randomly}}{\star_B}"] \arrow[loop below, "\underset{\textbf{indefinitely}}{\bigcirc_B}"'] & \perp \textbf{potential data} \\
 & & & (\mathrm{LType}_B)_{\Box_B} \arrow[r, hook, "\sim"'] & (\mathrm{LType}_B)^{\Box_B} \arrow[ur, phantom, "\simeq" sloped] & &
\end{tikzcd}$$

necessarily $\mathcal{H}_\bullet$

$\Box_B \mathcal{H}_\bullet$

Given... **obtain...**

$b : B \quad \vdash \quad \mathcal{H}$

measurement result

where $\mathcal{H} := \bigoplus_{b':B} \mathcal{H}_{b'}$

Given B : ClType of possible measurement outcomes (“possible worlds”) **the monadic effects of B-dependent linear data type formers** constitute modalities of **actual and potential quantum B-measurements**.

quantum data types

possibly **randomly**

actual data $\perp$ $\quad\quad$ $\perp$ **potential data**

$$\begin{tikzcd}
 & & (\mathrm{LType}_B)_{\Diamond_B} \arrow[r, hook, "\sim"] \arrow[d, phantom, "\perp"] & (\mathrm{LType}_B)^{\Diamond_B} \arrow[dr, phantom, "\simeq"] & \\
\mathrm{LType}_B \arrow[loop above, "\Diamond_B"] \arrow[loop below, "\Box_B"'] \arrow[urr, ->>, "\oplus_B"] \arrow[drr, ->>, "\oplus_B"'] & \simeq \mathrm{LType}^{\bigcirc_B} & \mathrm{LType}_{\bigcirc_B} \arrow[l, hook'] \arrow[d, phantom, "\perp"] & & \mathrm{LType} \arrow[ll, ->>, "\mathbb{1}_B\otimes"'] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"'] \\
 & & (\mathrm{LType}_B)_{\Box_B} \arrow[r, hook, "\sim"'] & (\mathrm{LType}_B)^{\Box_B} \arrow[ur, phantom, "\simeq"] &
\end{tikzcd}$$

necessarily **indefinitely**

necessarily $\mathcal{H}_\bullet$ $\quad$ **entails** $\quad$ **actually $\mathcal{H}_\bullet$**

$$\Box_B\,\mathcal{H}_\bullet \xrightarrow{\quad \varepsilon^{\Box_B}_{\mathcal{H}_\bullet} \quad} \mathcal{H}_\bullet$$

Given... $\quad$ **obtain...**

$$b : B \;\vdash\; \mathcal{H} \xrightarrow[\text{measurement collapse}]{\sum_{b'} |\psi_{b'}\rangle \;\mapsto\; |\psi_b\rangle} \mathcal{H}_b \qquad \text{where } \mathcal{H} := \bigoplus_{b':B} \mathcal{H}_{b'}$$

measurement result

Given B : ClType of possible measurement outcomes (“possible worlds”) **the monadic effects of B-dependent linear data type formers** constitute modalities of **actual and potential quantum B-measurements**.

quantum data types

$$\begin{tikzcd}
 & & (\mathrm{LType}_B)_{\Diamond_B} \arrow[r, hook, "\sim"] \arrow[d, phantom, "\perp"] & (\mathrm{LType}_B)^{\Diamond_B} \arrow[dr, phantom, "\wr"] & & \\
\textbf{actual data} \perp & \mathrm{LType}_B \simeq \mathrm{LType}^{\bigcirc_B} \arrow[loop above, "\Diamond_B", "\textbf{possibly}"'] \arrow[loop below, "\Box_B"', "\textbf{necessarily}"] \arrow[ur, two heads, "\oplus_B"] \arrow[dr, two heads, "\oplus_B"'] & \mathrm{LType}_{\bigcirc_B} \arrow[l, hook'] \arrow[d, phantom, "\perp"] & & \mathrm{LType} \arrow[ll, two heads, "\mathbb{1}_B \otimes"'] \arrow[loop above, "\star_B", "\textbf{randomly}"'] \arrow[loop below, "\bigcirc_B"', "\textbf{indefinitely}"] & \perp \textbf{ potential data} \\
 & & (\mathrm{LType}_B)_{\Box_B} \arrow[r, hook, "\sim"'] & (\mathrm{LType}_B)^{\Box_B} \arrow[ur, phantom, "\wr"] & &
\end{tikzcd}$$

$$\begin{tikzcd}
\textbf{necessarily } \mathcal{H}_\bullet & \textbf{entails} & \textbf{actually } \mathcal{H}_\bullet & \textbf{entails} & \textbf{possibly } \mathcal{H}_\bullet \\
\Box_B \mathcal{H}_\bullet \arrow[rr, "\epsilon^{\Box_B}_{\mathcal{H}_\bullet}"] & & \mathcal{H}_\bullet \arrow[rr, "\eta^{\Diamond_B}_{\mathcal{H}_\bullet}"] & & \Diamond_B \mathcal{H}_\bullet
\end{tikzcd}$$

$$\begin{tikzcd}
\textbf{Given...} & \textbf{obtain...} & & & \\
b : B & \vdash & \mathcal{H} \arrow[r, two heads, "\Sigma_{b'} \lvert\psi_{b'}\rangle \;\mapsto\; \lvert\psi_b\rangle", "\textbf{measurement collapse}"'] \arrow[rr, bend right, "\textbf{linear projector onto sub-Hilbert space } \mathcal{H}_b"'] & \mathcal{H}_b \arrow[r, hook, "\lvert\psi_b\rangle \mapsto \oplus_{b'} \begin{cases} \lvert\psi_b\rangle \text{ if } b'{=}b \\ 0 \quad \text{else} \end{cases}", "\textbf{state preparation}"'] & \mathcal{H}, \\
\textbf{measurement result} & & & &
\end{tikzcd}$$

where $\mathcal{H} := \bigoplus\limits_{b':B} \mathcal{H}_{b'}$

Given B : ClType of possible measurement outcomes ("possible worlds") **the monadic effects of B-dependent linear data type formers** constitute modalities of **actual and potential quantum B-measurements**.

quantum
data types

possibly **randomly**

actual data $\perp$ **potential data**

necessarily **indefinitely**

$$\begin{tikzcd}
 & (\mathrm{LType}_B)_{\lozenge_B} \arrow[r, hook, "\sim"] \arrow[d, phantom, "\perp"] & (\mathrm{LType}_B)^{\lozenge_B} \arrow[dr, phantom, "\simeq"] & \\
\mathrm{LType}_B \simeq \mathrm{LType}^{\bigcirc_B} \arrow[loop above, "\lozenge_B"] \arrow[loop below, "\Box_B"'] \arrow[ur, two heads, "\oplus_B"] \arrow[dr, two heads, "\oplus_B"'] & \mathrm{LType}_{\bigcirc_B} \arrow[l, hook'] \arrow[d, phantom, "\perp"] & & \mathrm{LType} \arrow[ll, two heads, "\mathbb{1}_B \otimes"'] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"'] \\
 & (\mathrm{LType}_B)_{\Box_B} \arrow[r, hook, "\sim"'] & (\mathrm{LType}_B)^{\Box_B} \arrow[ur, phantom, "\simeq"] &
\end{tikzcd}$$

---

principle of quantum compulsion:

necessarily $\mathcal{H}_\bullet$ **entails** **actually $\mathcal{H}_\bullet$** **entails** **possibly $\mathcal{H}_\bullet$** **is** **necessarily $\mathcal{H}_\bullet$**

$$\begin{tikzcd}
\Box_B \mathcal{H}_\bullet \arrow[r, "\epsilon^{\Box_B}_{\mathcal{H}_\bullet}"] & \mathcal{H}_\bullet \arrow[r, "\eta^{\lozenge_B}_{\mathcal{H}_\bullet}"] & \lozenge_B \mathcal{H}_\bullet \arrow[r, phantom, "\simeq"] & \Box_B \mathcal{H}_\bullet
\end{tikzcd}$$

ambidexterity

Given... **obtain...**

$$b : B \quad \vdash \quad \begin{tikzcd}
\mathcal{H} \arrow[r, two heads, "\sum_{b'} |\psi_{b'}\rangle \mapsto |\psi_b\rangle", "\text{measurement collapse}"'] \arrow[rr, bend right, "\text{linear projector onto sub-Hilbert space } \mathcal{H}_b"'] & \mathcal{H}_b \arrow[r, hook, "|\psi_b\rangle \mapsto \oplus_{b'} \begin{cases} |\psi_b\rangle & \text{if } b'=b \\ 0 & \text{else} \end{cases}", "\text{state preparation}"'] & \mathcal{H},
\end{tikzcd} \qquad \text{where } \mathcal{H} := \bigoplus_{b':B} \mathcal{H}_{b'}$$

measurement result

Given B : ClType of possible measurement outcomes (“possible worlds”) **the monadic effects of B-dependent linear data type formers** constitute modalities of **actual and potential quantum B-measurements**.

quantum data types

possibly **randomly**

actual data $\perp$ $\perp$ **potential data**

$$\begin{tikzcd}
 & (\mathrm{LType}_B)_{\lozenge_B} \arrow[r, hook, "\sim"] \arrow[d, phantom, "\perp"] & (\mathrm{LType}_B)^{\lozenge_B} \arrow[dr, phantom, "\simeq"] & \\
\mathrm{LType}_B \simeq \mathrm{LType}^{\bigcirc_B} \arrow[loop above, "\lozenge_B"] \arrow[loop below, "\square_B"] \arrow[ur, two heads, "\oplus_B"] \arrow[dr, two heads, "\oplus_B"'] & \mathrm{LType}_{\bigcirc_B} \arrow[l, hook'] \arrow[d, phantom, "\perp"] & & \mathrm{LType} \arrow[ll, two heads, "\mathbb{1}_B \otimes"'] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"] \\
 & (\mathrm{LType}_B)_{\square_B} \arrow[r, hook, "\sim"'] & (\mathrm{LType}_B)^{\square_B} \arrow[ur, phantom, "\simeq"] &
\end{tikzcd}$$

necessarily **indefinitely**

---

principle of quantum compulsion:

necessarily $\mathcal{H}_\bullet$ **entails** **actually $\mathcal{H}_\bullet$** **entails** **possibly $\mathcal{H}_\bullet$** **is** **necessarily $\mathcal{H}_\bullet$**

$$\begin{tikzcd}
\square_B\, \mathcal{H}_\bullet \arrow[r, "\varepsilon^{\square_B}_{\mathcal{H}_\bullet}"] & \mathcal{H}_\bullet \arrow[r, "\eta^{\lozenge_B}_{\mathcal{H}_\bullet}"] & \lozenge_B\, \mathcal{H}_\bullet \arrow[r, phantom, "\simeq"] & \square_B\, \mathcal{H}_\bullet
\end{tikzcd}$$

ambidexterity

Given... **obtain...**

$b : B \;\vdash$

measurement result

$$\begin{tikzcd}
\mathcal{H} \arrow[r, two heads, "\sum_{b'} |\psi_{b'}\rangle \;\mapsto\; |\psi_b\rangle", "\text{measurement collapse}"'] \arrow[rr, bend right, "\text{linear projector onto sub-Hilbert space } \mathcal{H}_b"'] & \mathcal{H}_b \arrow[r, hook, "|\psi_b\rangle \mapsto \oplus_{b'} \begin{cases} |\psi_b\rangle & \text{if } b'=b \\ 0 & \text{else} \end{cases}", "\text{state preparation}"'] & \mathcal{H},
\end{tikzcd}$$

where $\mathcal{H} := \bigoplus\limits_{b':B} \mathcal{H}_{b'}$

---

randomly $\mathcal{H}$

$\star_B \mathcal{H}$

$\bigoplus\limits_{b:B} \mathcal{H}$

Given $B : \mathrm{ClType}$ of possible measurement outcomes (“possible worlds”)
the monadic effects of B-dependent linear data type formers constitute
modalities of **actual and potential quantum B-measurements**.

quantum
data types

possibly **randomly**

actual data $\perp$ $\perp$ **potential data**

$$\begin{tikzcd}
& (\mathrm{LType}_B)_{\Diamond_B} \arrow[r, hook, "\sim"] \arrow[d, phantom, "\perp"] & (\mathrm{LType}_B)^{\Diamond_B} \\
\mathrm{LType}_B \simeq \mathrm{LType}^{\bigcirc_B} \arrow[loop above, "\Diamond_B"] \arrow[loop below, "\Box_B"'] \arrow[ur, two heads, bend left, "\oplus_B"] \arrow[dr, two heads, bend right, "\oplus_B"'] & \mathrm{LType}_{\bigcirc_B} \arrow[l, hook'] \arrow[d, phantom, "\perp"] & \mathrm{LType} \arrow[l, two heads, "\mathbb{1}_B \otimes"'] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"'] \arrow[u, phantom, "\simeq" sloped] \arrow[d, phantom, "\simeq" sloped] \\
& (\mathrm{LType}_B)_{\Box_B} \arrow[r, hook, "\sim"'] & (\mathrm{LType}_B)^{\Box_B}
\end{tikzcd}$$

necessarily **indefinitely**

principle of quantum compulsion:

necessarily $\mathcal{H}_\bullet$ **entails** **actually $\mathcal{H}_\bullet$** **entails** **possibly $\mathcal{H}_\bullet$** **is** **necessarily $\mathcal{H}_\bullet$**

$$\Box_B \mathcal{H}_\bullet \xrightarrow{\;\epsilon^{\Box_B}_{\mathcal{H}_\bullet}\;} \mathcal{H}_\bullet \xrightarrow{\;\eta^{\Diamond_B}_{\mathcal{H}_\bullet}\;} \Diamond_B \mathcal{H}_\bullet \underset{\text{ambidexterity}}{\simeq} \Box_B \mathcal{H}_\bullet$$

Given... **obtain...**

$b : B \;\vdash$

measurement result

$$\begin{tikzcd}
\mathcal{H} \arrow[r, two heads, "\Sigma_{b'} |\psi_{b'}\rangle \;\mapsto\; |\psi_b\rangle", "\text{measurement collapse}"'] \arrow[rr, bend right, "\text{linear projector onto sub-Hilbert space } \mathcal{H}_b"'] & \mathcal{H}_b \arrow[r, hook, "|\psi_b\rangle \mapsto \oplus_{b'} \begin{cases} |\psi_b\rangle & \text{if } b' = b \\ 0 & \text{else} \end{cases}", "\text{state preparation}"'] & \mathcal{H},
\end{tikzcd}$$

where $\mathcal{H} := \bigoplus\limits_{b' : B} \mathcal{H}_{b'}$

randomly $\mathcal{H}$ **entails** **potentially $\mathcal{H}$**

$$\star_B \mathcal{H} \xrightarrow{\;\epsilon^{\star_B}_{\mathcal{H}}\;} \mathcal{H}$$

$$\bigoplus_{b:B} \mathcal{H} \xrightarrow[\text{quantum superposition}]{\oplus_b |\psi_b\rangle \mapsto \Sigma_b |\psi_b\rangle} \mathcal{H}$$

Given B : ClType of possible measurement outcomes ("possible worlds") **the monadic effects of B-dependent linear data type formers** constitute modalities of **actual and potential quantum B-measurements**.

quantum
data types

possibly **randomly**

actual data $\perp$ $\perp$ **potential data**

$$\begin{tikzcd}
 & (\mathrm{LType}_B)_{\Diamond_B} \arrow[r, hook, "\sim"] \arrow[d, phantom, "\perp"] & (\mathrm{LType}_B)^{\Diamond_B} \arrow[dr, phantom, "\sim" sloped] & \\
\mathrm{LType}_B \simeq \mathrm{LType}^{\bigcirc_B} \arrow[ur, two heads, "\oplus_B"] \arrow[dr, two heads, "\oplus_B"'] \arrow[loop above, "\Diamond_B"] \arrow[loop below, "\Box_B"'] & \mathrm{LType}_{\bigcirc_B} \arrow[l, hook'] \arrow[d, phantom, "\perp"] & & \mathrm{LType} \arrow[ll, two heads, "\mathbb{1}_B \otimes"'] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"'] \\
 & (\mathrm{LType}_B)_{\Box_B} \arrow[r, hook, "\sim"'] & (\mathrm{LType}_B)^{\Box_B} \arrow[ur, phantom, "\sim" sloped] &
\end{tikzcd}$$

necessarily **indefinitely**

---

principle of quantum compulsion:

necessarily $\mathcal{H}_\bullet$ **entails** **actually $\mathcal{H}_\bullet$** **entails** **possibly $\mathcal{H}_\bullet$** **is** **necessarily $\mathcal{H}_\bullet$**

$$\begin{tikzcd}
\Box_B \mathcal{H}_\bullet \arrow[r, "\epsilon^{\Box_B}_{\mathcal{H}_\bullet}"] & \mathcal{H}_\bullet \arrow[r, "\eta^{\Diamond_B}_{\mathcal{H}_\bullet}"] & \Diamond_B \mathcal{H}_\bullet \arrow[r, phantom, "\simeq"] & \Box_B \mathcal{H}_\bullet
\end{tikzcd}$$

ambidexterity

Given... **obtain...**

$$\begin{tikzcd}
b : B \vdash & \mathcal{H} \arrow[r, two heads, "\Sigma_{b'} |\psi_{b'}\rangle \;\mapsto\; |\psi_b\rangle", "\text{measurement collapse}"'] \arrow[rr, bend right, "\text{linear projector onto sub-Hilbert space } \mathcal{H}_b"'] & \mathcal{H}_b \arrow[r, hook, "|\psi_b\rangle \mapsto \oplus_{b'} \begin{cases} |\psi_b\rangle & \text{if } b'=b \\ 0 & \text{else} \end{cases}", "\text{state preparation}"'] & \mathcal{H},
\end{tikzcd}$$

measurement result

where $\mathcal{H} := \bigoplus\limits_{b':B} \mathcal{H}_{b'}$

- - -

randomly $\mathcal{H}$ **entails** **potentially $\mathcal{H}$** **entails** **indefinitely $\mathcal{H}$**

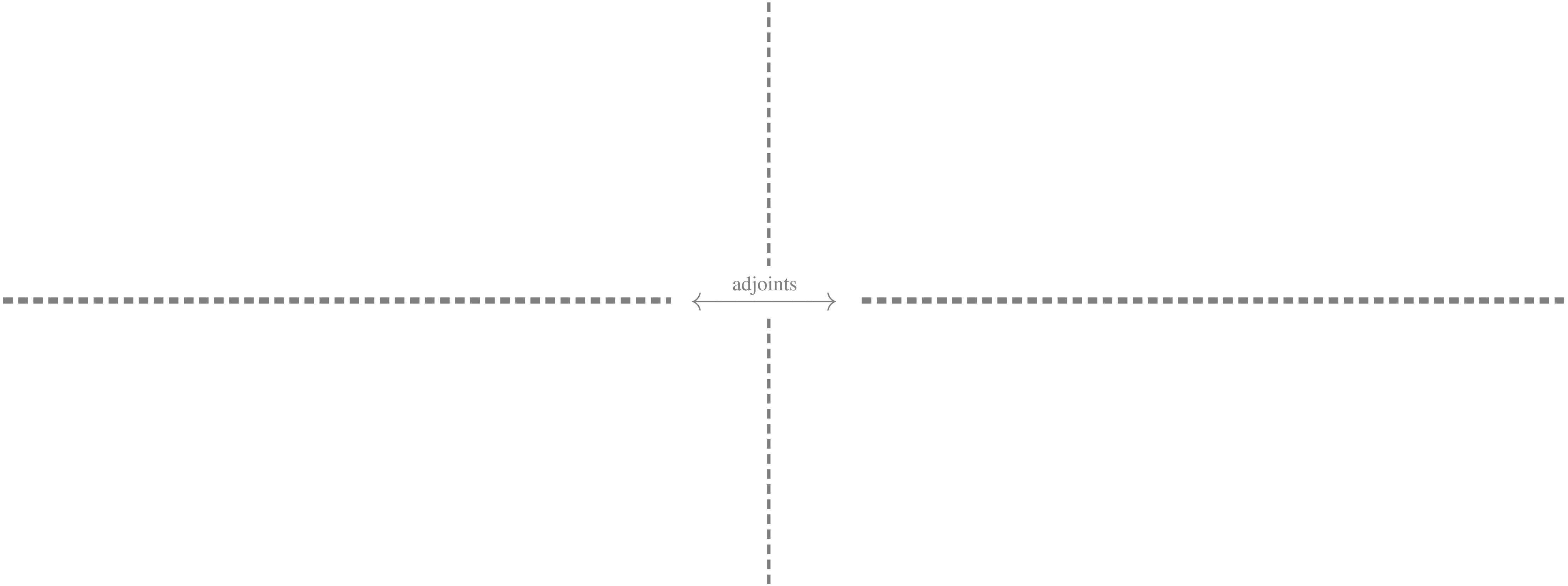
$$\begin{tikzcd}
\star_B \mathcal{H} \arrow[r, "\epsilon^{\star_B}_{\mathcal{H}}"] & \mathcal{H} \arrow[r, "\eta^{\bigcirc_B}_{\mathcal{H}}"] & \bigcirc_B \mathcal{H}
\end{tikzcd}$$

$$\begin{tikzcd}
\bigoplus\limits_{b:B} \mathcal{H} \arrow[r, "\oplus_b |\psi_b\rangle \mapsto \Sigma_b |\psi_b\rangle", "\text{quantum superposition}"'] & \mathcal{H} \arrow[r, "|\psi\rangle \mapsto \oplus_b |\psi\rangle_b", "\text{quantum parallelization}"'] & \bigoplus\limits_{b:B} \mathcal{H}
\end{tikzcd}$$

The pure effects of these modalities of dependent linear data type formation

are remarkable in their sheer quantum information-theoretic content.

To repeat:



The pure effects of these modalities of dependent linear data type formation

are remarkable in their sheer quantum information-theoretic content.

To repeat:

$$\overbrace{(p_B)^*(p_B)_*}^{\Box_B}\mathcal{H}_\bullet \xrightarrow[\textbf{necessity counit}]{\varepsilon^{\Box_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet$$

$$b : B \vdash \bigoplus_{b' : B} \mathcal{H}_{b'} \xrightarrow[\text{quantum } \textbf{measurement}]{\oplus_{b'}|\psi_{b'}\rangle \;\mapsto\; |\psi_b\rangle} \mathcal{H}_b$$

“ the necessary becomes actual ”

adjoints

$\longleftrightarrow$

The pure effects of these modalities of dependent linear data type formation

are remarkable in their sheer quantum information-theoretic content.

To repeat:

$$\overbrace{(p_B)^*(p_B)_*}^{\Box_B}\mathcal{H}_\bullet \xrightarrow[\textbf{necessity counit}]{\epsilon^{\Box_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet$$

$$b : B \vdash \bigoplus_{b':B} \mathcal{H}_{b'} \xrightarrow[\text{quantum }\textbf{measurement}]{\oplus_{b'}|\psi_{b'}\rangle \;\mapsto\; |\psi_b\rangle} \mathcal{H}_b$$

“ the necessary becomes actual ”

adjoints

$$\longleftrightarrow$$

$$\mathcal{H}_\bullet \xrightarrow[\textbf{possibility unit}]{\eta^{\Diamond_B}_{\mathcal{H}_\bullet}} \overbrace{(p_B)^*(p_B)_!}^{\Diamond_B}\mathcal{H}_\bullet$$

$$b : B \vdash \mathcal{H}_b \xrightarrow[\text{quantum }\textbf{state preparation}]{|\psi_b\rangle \;\mapsto\; \oplus_{b'}\left\{\begin{array}{ll}|\psi_b\rangle & \text{if } b'{=}b \\ 0 & \text{else}\end{array}\right.} \bigoplus_{b':B} \mathcal{H}_{b'}$$

“ the actual is possible ”

The pure effects of these modalities of dependent linear data type formation

are remarkable in their sheer quantum information-theoretic content.

To repeat:

$$\overbrace{(p_B)^*(p_B)_*}^{\Box_B}\mathcal{H}_\bullet \xrightarrow[\textbf{necessity counit}]{\epsilon^{\Box_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet$$

$$b : B \vdash \bigoplus_{b':B}\mathcal{H}_{b'} \xrightarrow[\text{quantum } \textbf{measurement}]{\oplus_{b'}|\psi_{b'}\rangle \;\mapsto\; |\psi_b\rangle} \mathcal{H}_b$$

“ the necessary becomes actual ”

$$\mathcal{H}_\bullet \xrightarrow[\textbf{possibility unit}]{\eta^{\Diamond_B}_{\mathcal{H}_\bullet}} \overbrace{(p_B)^*(p_B)_!}^{\Diamond_B}\mathcal{H}_\bullet$$

$$b : B \vdash \mathcal{H}_b \xrightarrow[\text{quantum } \textbf{state preparation}]{|\psi_b\rangle \;\mapsto\; \oplus_{b'}\begin{cases}|\psi_b\rangle & \text{if } b'=b\\ 0 & \text{else}\end{cases}} \bigoplus_{b':B}\mathcal{H}_{b'}$$

“ the actual is possible ”

$$\xleftrightarrow{\text{adjoints}}$$

“ the random becomes potential ”

$$\overbrace{(p_B)_!(p_B)^*}^{\star_B}\mathcal{H} \xhookrightarrow[\textbf{randomness counit}]{\epsilon^{\star_B}_{\mathcal{H}}} \mathcal{H}$$

$$\bigoplus_{b:B}\mathcal{H} \xrightarrow[\text{quantum } \textbf{superposition}]{\oplus_b|\psi_b\rangle \;\mapsto\; \Sigma_b\,|\psi_b\rangle} \mathcal{H}$$

The pure effects of these modalities of dependent linear data type formation

are remarkable in their sheer quantum information-theoretic content.

To repeat:

$$\overbrace{(p_B)^*(p_B)_*}^{\Box_B}\mathcal{H}_\bullet \xrightarrow[\textbf{necessity counit}]{\varepsilon^{\Box_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet$$

$$b : B \vdash \bigoplus_{b':B} \mathcal{H}_{b'} \xrightarrow[\text{quantum }\textbf{measurement}]{\oplus_{b'}|\psi_{b'}\rangle \;\mapsto\; |\psi_b\rangle} \mathcal{H}_b$$

“ the necessary becomes actual ”

$$\mathcal{H}_\bullet \xrightarrow[\textbf{possibility unit}]{\eta^{\Diamond_B}_{\mathcal{H}_\bullet}} \overbrace{(p_B)^*(p_B)_!}^{\Diamond_B}\mathcal{H}_\bullet$$

$$b : B \vdash \mathcal{H}_b \xrightarrow[\text{quantum }\textbf{state preparation}]{|\psi_b\rangle \;\mapsto\; \oplus_{b'}\begin{cases}|\psi_b\rangle & \text{if } b'=b\\ 0 & \text{else}\end{cases}} \bigoplus_{b':B}\mathcal{H}_{b'}$$

“ the actual is possible ”

$$\xleftrightarrow{\text{adjoints}}$$

“ the random becomes potential ”

$$\overbrace{(p_B)_!(p_B)^*}^{\star_B}\mathcal{H} \xhookrightarrow[\textbf{randomness counit}]{\varepsilon^{\star_B}_{\mathcal{H}}} \mathcal{H}$$

$$\bigoplus_{b:B}\mathcal{H} \xrightarrow[\text{quantum }\textbf{superposition}]{\oplus_b|\psi_b\rangle \;\mapsto\; \Sigma_b|\psi_b\rangle} \mathcal{H}$$

“ the potential is indeterminate ”

$$\mathcal{H} \xrightarrow[\textbf{indefiniteness unit}]{\eta^{\bigcirc_B}_{\mathcal{H}}} \overbrace{(p_B)_*(p_B)^*}^{\bigcirc_B}\mathcal{H}$$

$$\mathcal{H} \xrightarrow[\text{quantum }\textbf{parallelism}]{|\psi\rangle \;\mapsto\; \oplus_b|\psi\rangle_b} \bigoplus_{b:B}\mathcal{H}$$

Q-bits are the free linear indefiniteness-effect handlers over $\mathrm{Bit} = \{0, 1\}$

Coherent q-bits:

$$\begin{tikzcd} \mathrm{QBit} \arrow[d, phantom, "\rotatebox{90}{$=:$}"] & : & \mathrm{LType} \arrow[r, hook, "\mathbb{1}_{\mathrm{Bit}}\otimes"] & \mathrm{LType}_{\mathrm{Bit}} \arrow[r, "\bigoplus_{\mathrm{Bit}}", "\sim"'] & \mathrm{LType}^{\bigcirc_B} \\ \bigcirc_{\mathrm{Bit}}\mathbb{1} \end{tikzcd}$$

Quantum gate with q-bit output:

De-cohered (measured) q-bits:

Q-bits are the free linear indefiniteness-effect handlers over $\mathrm{Bit} = \{0, 1\}$

Coherent q-bits:

$$\mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}\otimes} \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_B}$$

$$\overset{\cdot\,\cdot}{\Vert}$$

$$\bigcirc_{\mathrm{Bit}}\mathbb{1} \;=\; \bigoplus_{\{0,1\}}\mathbb{C} \;=\; \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle$$

De-cohered (measured) q-bits:

Quantum gate with q-bit output:

Q-bits are the free linear indefiniteness-effect handlers over $\mathrm{Bit} = \{0, 1\}$

Coherent q-bits:

$$\mathrm{QBit} : \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}{}^{\otimes}} \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_B}$$

$$\overset{\cdot\cdot}{\|}$$

$$\bigcirc_{\mathrm{Bit}} \mathbb{1} = \bigoplus_{\{0,1\}} \mathbb{C} = \mathbb{C} \cdot |0\rangle \oplus \mathbb{C} \cdot |1\rangle$$

De-cohered (measured) q-bits:

Quantum gate with q-bit output:

Q-bits are the free linear indefiniteness-effect handlers over $\mathrm{Bit} = \{0, 1\}$

Coherent q-bits:

$$\begin{array}{l}
\mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}{}^{\otimes}} \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_B} \\
\overset{\cdot\cdot}{\|} \\
\bigcirc_{\mathrm{Bit}}\mathbb{1} \;=\; \bigoplus_{\{0,1\}}\mathbb{C} \;=\; \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle
\end{array}$$

$$\begin{array}{l}
\mathrm{QBit} \\
\otimes \\
\mathcal{H} \\
\| \\
\bigcirc_{\mathrm{Bit}}\mathcal{H} \;=\; \bigoplus_{\{0,1\}}\mathcal{H} \;=\; \mathcal{H}\otimes|0\rangle \;\oplus\; \mathcal{H}\otimes|0\rangle
\end{array}$$

Quantum gate with q-bit output:

De-cohered (measured) q-bits:

Q-bits are the free linear indefiniteness-effect handlers over $\mathrm{Bit} = \{0,1\}$

Coherent q-bits:

━━━ $\mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}\otimes} \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_B}$

$\overset{\cdot\cdot}{\|}$

$\bigcirc_{\mathrm{Bit}}\mathbb{1} \;=\; \bigoplus_{\{0,1\}}\mathbb{C} \;=\; \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle$

━━━ QBit

$\otimes$

━━━ $\mathcal{H}$

$\|$

$\bigcirc_{\mathrm{Bit}}\mathcal{H} \;=\; \bigoplus_{\{0,1\}}\mathcal{H} \;=\; \mathcal{H}\otimes|0\rangle \;\oplus\; \mathcal{H}\otimes|0\rangle$

De-cohered (measured) q-bits:

═══ $\mathbb{1}_{\mathrm{Bit}} \;:\; \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_{\mathrm{Bit}}}$

$b : \mathrm{Bit} \quad\vdash\quad \mathbb{C}\cdot|b\rangle \;:\; \mathrm{LType}$

Quantum gate with q-bit output:

Q-bits are the free linear indefiniteness-effect handlers over $\mathrm{Bit} = \{0,1\}$

Coherent q-bits:

$$\mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}\otimes} \mathrm{LType}_{\mathrm{Bit}} \underset{\sim}{\xrightarrow{\bigoplus_{\mathrm{Bit}}}} \mathrm{LType}^{\bigcirc_B}$$

$$\overset{..}{\parallel}$$

$$\bigcirc_{\mathrm{Bit}} \mathbb{1} \;=\; \bigoplus_{\{0,1\}} \mathbb{C} \;=\; \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle$$

$$\mathrm{QBit}$$

$$\otimes$$

$$\mathcal{H}$$

$$\parallel$$

$$\bigcirc_{\mathrm{Bit}} \mathcal{H} \;=\; \bigoplus_{\{0,1\}} \mathcal{H} \;=\; \mathcal{H}\otimes|0\rangle \;\oplus\; \mathcal{H}\otimes|0\rangle$$

De-cohered (measured) q-bits:

$$\mathbb{1}_{\mathrm{Bit}} \;:\; \mathrm{LType}_{\mathrm{Bit}} \underset{\sim}{\xrightarrow{\bigoplus_{\mathrm{Bit}}}} \mathrm{LType}^{\bigcirc_{\mathrm{Bit}}}$$

$$b : \mathrm{Bit} \quad\vdash\quad \mathbb{C}\cdot|b\rangle \;:\; \mathrm{LType}$$

$$\mathbb{1}_{\mathrm{Bit}}$$

$$\otimes \qquad b : \mathrm{Bit} \quad\vdash\quad \mathcal{H}\otimes|b\rangle \;:\; \mathrm{LType}$$

$$\mathcal{H}$$

Quantum gate with q-bit output:

Q-bits are the free linear indefiniteness-effect handlers over $\mathrm{Bit} = \{0,1\}$

Coherent q-bits:

$$\mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}\otimes} \mathrm{LType}_{\mathrm{Bit}} \underset{\sim}{\xrightarrow{\bigoplus_{\mathrm{Bit}}}} \mathrm{LType}^{\bigcirc_B}$$

$$\overset{\cdot\cdot}{\|}$$

$$\bigcirc_{\mathrm{Bit}}\mathbb{1} \;=\; \bigoplus_{\{0,1\}}\mathbb{C} \;=\; \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle$$

$$\mathrm{QBit}$$

$$\otimes$$

$$\mathcal{H}$$

$$\|$$

$$\bigcirc_{\mathrm{Bit}}\mathcal{H} \;=\; \bigoplus_{\{0,1\}}\mathcal{H} \;=\; \mathcal{H}\otimes|0\rangle \;\oplus\; \mathcal{H}\otimes|0\rangle$$

De-cohered (measured) q-bits:

$$\mathbb{1}_{\mathrm{Bit}} \;:\; \mathrm{LType}_{\mathrm{Bit}} \underset{\sim}{\xrightarrow{\bigoplus_{\mathrm{Bit}}}} \mathrm{LType}^{\bigcirc_{\mathrm{Bit}}}$$

$$b : \mathrm{Bit} \quad\vdash\quad \mathbb{C}\cdot|b\rangle \;:\; \mathrm{LType}$$

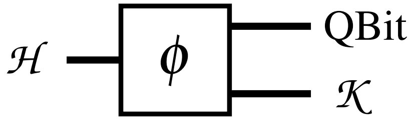
$$\mathbb{1}_{\mathrm{Bit}}$$

$$\otimes \qquad b : \mathrm{Bit} \quad\vdash\quad \mathcal{H}\otimes|b\rangle \;:\; \mathrm{LType}$$

$$\mathcal{H}$$

Quantum gate with q-bit output:

A quantum gate which
may handle $\bigcirc_{\mathrm{Bit}}$-effects
is one with a QBit-output:



$$\mathcal{H} \xrightarrow{\phi} \mathrm{QBit}\otimes\mathcal{K} \simeq \bigcirc_{\mathrm{Bit}}\mathcal{K}$$

Q-bits are the free linear indefiniteness-effect handlers over $\mathrm{Bit} = \{0,1\}$

Coherent q-bits:

$$\mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}\otimes} \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_B}$$

$$\overset{\cdot\cdot}{\|}$$

$$\bigcirc_{\mathrm{Bit}}\mathbb{1} \;=\; \bigoplus_{\{0,1\}}\mathbb{C} \;=\; \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle$$

$$\mathrm{QBit}$$

$$\otimes$$

$$\mathcal{H}$$

$$\|$$

$$\bigcirc_{\mathrm{Bit}}\mathcal{H} \;=\; \bigoplus_{\{0,1\}}\mathcal{H} \;=\; \mathcal{H}\otimes|0\rangle \;\oplus\; \mathcal{H}\otimes|0\rangle$$

De-cohered (measured) q-bits:

$$\mathbb{1}_{\mathrm{Bit}} \;:\; \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_{\mathrm{Bit}}}$$

$$b : \mathrm{Bit} \quad\vdash\quad \mathbb{C}\cdot|b\rangle \;:\; \mathrm{LType}$$

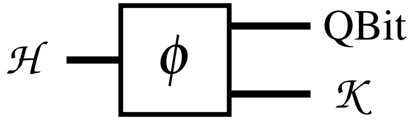
$$\mathbb{1}_{\mathrm{Bit}}$$

$$\otimes \qquad b : \mathrm{Bit} \quad\vdash\quad \mathcal{H}\otimes|b\rangle \;:\; \mathrm{LType}$$

$$\mathcal{H}$$

Quantum gate with q-bit output:

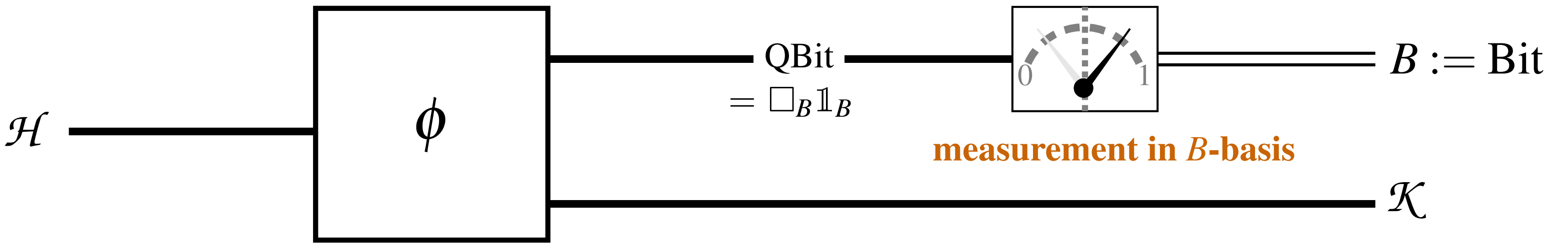
A quantum gate which may handle $\bigcirc_{\mathrm{Bit}}$-effects is one with a QBit-output:



$$\mathcal{H} \xrightarrow{\phi} \mathrm{QBit}\otimes\mathcal{K} \simeq \bigcirc_{\mathrm{Bit}}\mathcal{K}$$

Quantum measurement is Linear indefiniteness-effect handling.

quantum circuit



quantum gate

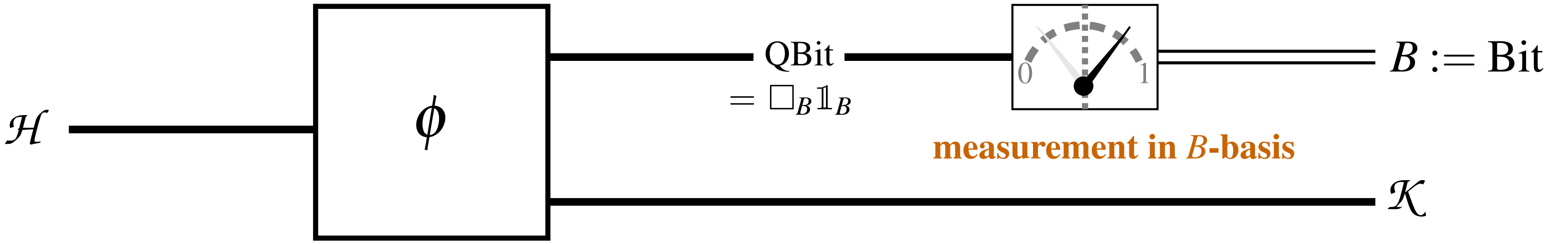
$$\mathcal{H} \xrightarrow{\phi} \mathrm{QBit} \otimes \mathcal{K} \simeq \bigcirc_B \mathcal{K}$$

$\bigcirc_B$-effect handling

Quantum measurement is Linear indefiniteness-effect handling.

$$\begin{tikzcd}
\text{quantum circuit} \arrow[d, squiggly, "\text{formalization}"] \\
\bigcirc_B\text{-modal linear types}
\end{tikzcd}$$

$$\mathrm{LType}_{\bigcirc_B}$$



measurement in B-basis

quantum gate

$\bigcirc_B$-effect handling

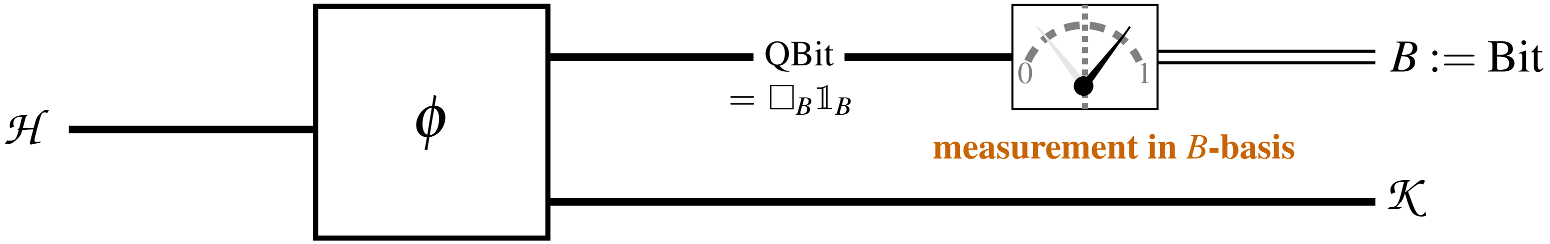
$$\mathcal{H} \xrightarrow{\phi} \mathrm{QBit} \otimes \mathcal{K} \simeq \bigcirc_B \mathcal{K}$$

Quantum measurement is Linear indefiniteness-effect handling.

$$\begin{tikzcd}
\textbf{quantum circuit} \arrow[d, squiggly, "\text{formalization}"] \\
\mathrm{LType}_{\bigcirc_B} \arrow[d, hook, "K_{(p_B)^*(p_B)_*}"', "\text{comparison functor}"] \\
\mathrm{LType}_B
\end{tikzcd}$$

$\bigcirc_B$-modal linear types $\mathrm{LType}_{\bigcirc_B}$

LType_B **B-dependent linear types**



subjection to $\bigcirc_B$-effects

quantum gate

$$\begin{tikzcd}
\bigcirc_B \mathcal{H} \arrow[loop above, "\bigcirc_B"] \arrow[r, "\bigcirc_B\left(\mathcal{H} \xrightarrow{\phi} \mathrm{QBit} \otimes \mathcal{K} \simeq \bigcirc_B \mathcal{K}\right)"] & \bigcirc_B \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"] \arrow[r, "\mathrm{hndl}^{\bigcirc_B}_{\bigcirc_B \mathcal{K}}", "\bigcirc_B\text{-effect handling}"'] & \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"]
\end{tikzcd}$$

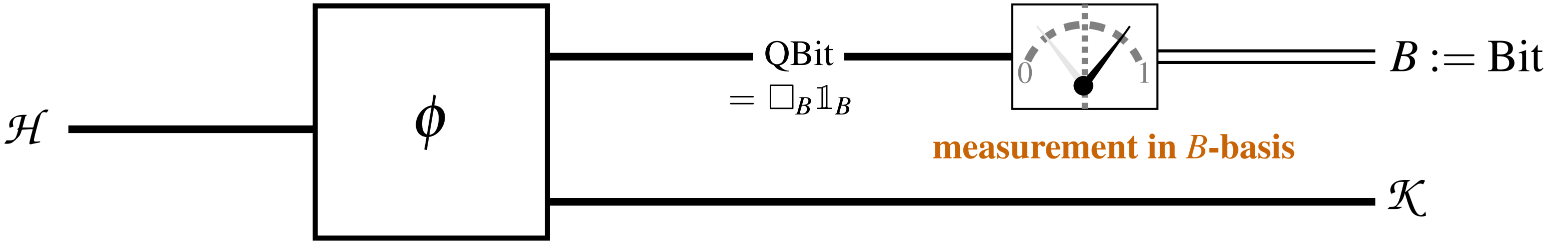
Quantum measurement is Linear indefiniteness-effect handling.

quantum circuit

$$\begin{tikzcd}
\text{quantum circuit} \arrow[d, squiggly, "\text{formalization}"] \\
\mathrm{LType}_{\bigcirc_B} \arrow[d, "K_{(p_B)^*(p_B)_*}"', "\text{comparison functor}"] \\
\mathrm{LType}_B
\end{tikzcd}$$

$\bigcirc_B$-modal linear types

B-dependent linear types



$$\begin{tikzcd}
\bigcirc_B \mathcal{H} \arrow[loop above, "\bigcirc_B"] \arrow[rr, "{\substack{\text{quantum gate} \\ \bigcirc_B\left(\mathcal{H} \xrightarrow{\phi} \mathrm{QBit} \otimes \mathcal{K} \simeq \bigcirc_B \mathcal{K}\right)}}"] & & \bigcirc_B \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"] \arrow[rr, "\mathrm{hndl}^{\bigcirc_B}_{\bigcirc_B \mathcal{K}}", "\bigcirc_B\text{-effect handling}"'] & & \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"]
\end{tikzcd}$$

subjection to $\bigcirc_B$-effects

$$\begin{tikzcd}
\mathbb{1}_B \otimes \mathcal{H} \arrow[rr, "\mathbb{1}_B \otimes \phi"] & & \Box_B \mathbb{1}_B \otimes \mathcal{K} \arrow[rr, "\epsilon^{\Box_B}_{\mathbb{1}_B} \otimes \mathcal{K}", "\Box_B\text{-counit}"'] & & \mathbb{1}_B \otimes \mathcal{K}
\end{tikzcd}$$

$$\begin{tikzcd}
b : B \vdash & \mathcal{H} \arrow[rr, "\phi"] & & \mathrm{QBit} \otimes \mathcal{K} \arrow[rr, "\text{quantum state collapse}"] & & \mathcal{K}
\end{tikzcd}$$

measurement result

$$\begin{array}{l} |0\rangle \otimes |\psi_0\rangle \\ + |1\rangle \otimes |\psi_1\rangle \end{array} \mapsto |\psi_b\rangle$$

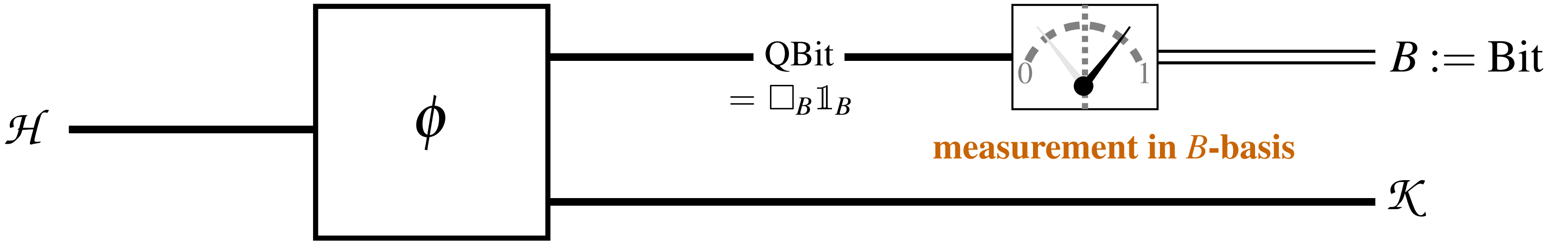
Quantum measurement is Linear indefiniteness-effect handling.

quantum circuit

$$\begin{tikzcd} \text{quantum circuit} \arrow[d, squiggly, "\text{formalization}"] \\ \mathrm{LType}_{\bigcirc_B} \arrow[d, "\text{comparison functor}", "K_{(p_B)^*(p_B)_*}"'] \\ \mathrm{LType}_B \end{tikzcd}$$

$\bigcirc_B$-modal linear types

B-dependent linear types



subjection to $\bigcirc_B$-effects **quantum gate**

$$\begin{tikzcd}[column sep=huge] \bigcirc_B \mathcal{H} \arrow[loop above, "\bigcirc_B"] \arrow[r, "\bigcirc_B\left(\mathcal{H} \xrightarrow{\phi} \mathrm{QBit} \otimes \mathcal{K} \simeq \bigcirc_B \mathcal{K}\right)"] & \bigcirc_B \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"] \arrow[r, "\mathrm{hndl}^{\bigcirc_B}_{\bigcirc_B \mathcal{K}}", "\bigcirc_B\text{-effect handling}"'] & \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"] \end{tikzcd}$$

$$\begin{tikzcd}[column sep=huge] \mathbb{1}_B \otimes \mathcal{H} \arrow[r, "\mathbb{1}_B \otimes \phi"] & \square_B \mathbb{1}_B \otimes \mathcal{K} \arrow[r, "\varepsilon^{\square_B}_{\mathbb{1}_B} \otimes \mathcal{K}", "\square_B\text{-counit}"'] & \mathbb{1}_B \otimes \mathcal{K} \end{tikzcd}$$

$$\begin{tikzcd}[column sep=huge] b : B \vdash \mathcal{H} \arrow[r, "\phi"] & \mathrm{QBit} \otimes \mathcal{K} \arrow[r, "\text{quantum state collapse}"] & \mathcal{K} \end{tikzcd}$$

measurement result

$$\begin{array}{l} |0\rangle \otimes |\psi_0\rangle \\ + |1\rangle \otimes |\psi_1\rangle \end{array} \mapsto |\psi_b\rangle$$

full linearly-typed detail of quantum measurement logic is emergent effect in LHoTT

Quantum measurement is Linear indefiniteness-effect handling.

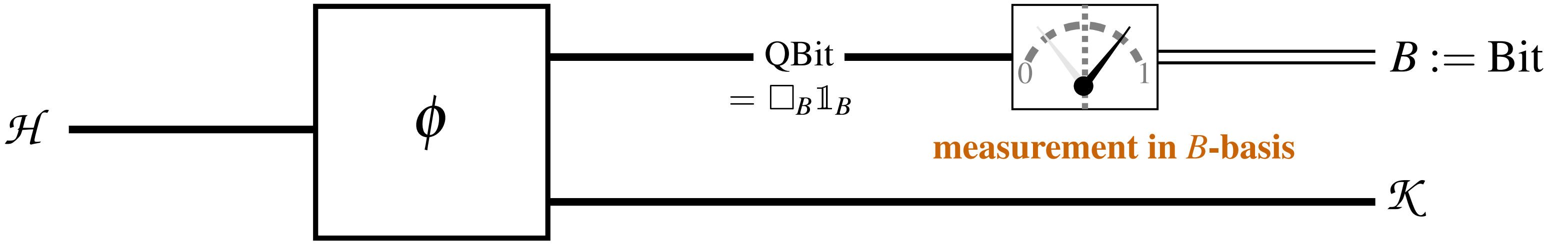
quantum circuit

$$\begin{tikzcd} \text{quantum circuit} \arrow[d, squiggly, "\text{formalization}"] \\ \bigcirc_B\text{-modal linear types} \end{tikzcd}$$

$\bigcirc_B$-modal linear types

$$\begin{tikzcd} \mathrm{LType}_{\bigcirc_B} \arrow[d, "K_{(p_B)^*(p_B)_*}"', "\text{comparison functor}"] \\ \mathrm{LType}_B \end{tikzcd}$$

B-dependent linear types



subjection to $\bigcirc_B$-effects

quantum gate

$$\begin{tikzcd} \bigcirc_B \mathcal{H} \arrow[loop above, "\bigcirc_B"] \arrow[r, "\bigcirc_B\left(\mathcal{H}\xrightarrow{\phi}\mathrm{QBit}\otimes\mathcal{K}\simeq\bigcirc_B\mathcal{K}\right)"] & \bigcirc_B\bigcirc_B\mathcal{K} \arrow[loop above, "\bigcirc_B"] \arrow[r, "\mathrm{hndl}^{\bigcirc_B}_{\bigcirc_B\mathcal{K}}", "\bigcirc_B\text{-effect handling}"'] & \bigcirc_B\mathcal{K} \arrow[loop above, "\bigcirc_B"] \end{tikzcd}$$

$$\begin{tikzcd} \mathbb{1}_B \otimes \mathcal{H} \arrow[r, "\mathbb{1}_B\otimes\phi"] & \Box_B\mathbb{1}_B \otimes \mathcal{K} \arrow[r, "\varepsilon^{\Box_B}_{\mathbb{1}_B}\otimes\mathcal{K}", "\Box_B\text{-counit}"'] & \mathbb{1}_B \otimes \mathcal{K} \end{tikzcd}$$

$$\begin{tikzcd} b : B \vdash \mathcal{H} \arrow[r, "\phi"] & \mathrm{QBit}\otimes\mathcal{K} \arrow[r, "\text{quantum state collapse}"] & \mathcal{K} \end{tikzcd}$$

measurement result

$$\begin{array}{l} |0\rangle\otimes|\psi_0\rangle \\ +|1\rangle\otimes|\psi_1\rangle \end{array} \mapsto |\psi_b\rangle$$

full linearly-typed detail of
quantum measurement logic
is emergent effect in LHoTT

Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

(see nLab:quantum+reader+monad)

$$
\begin{array}{c}
\bigcirc_B \\
\rotatebox[origin=c]{90}{$\in$} \\
\mathrm{Monad}\big(\mathrm{LType}\big)
\end{array}
$$

Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

(see nLab:quantum+reader+monad)

$$
\begin{array}{c}
\bigcirc_B \\
\rotatebox{-90}{$=:$} \\
B\text{-Reader} \\
\\
\\
\rotatebox{-90}{$\in$} \\
\mathrm{Monad}\big(\mathrm{LType}\big)
\end{array}
$$

Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

(see nLab:quantum+reader+monad)

$$B : \mathrm{FinType} \;\vdash\; \begin{tikzcd} \bigcirc_B \arrow[d, phantom, sloped, ":="] \\ B\text{-Reader} \arrow[d, phantom, sloped, "\simeq"] \\ \mathbb{1}^B\text{-Writer} \arrow[d, phantom, sloped, "\in"] \\ \mathrm{Monad}\big(\mathrm{LType}\big) \end{tikzcd}$$

$$\mathbb{1}^B\text{-Writer}(D) := \mathbb{1}^B \oplus D$$

$$\mathrm{bind}^{\mathbb{1}^B\mathrm{Writer}}\Big(D_1 \xrightarrow{\mathrm{prog}} \mathbb{1}^B \oplus D_2\Big) := \mathbb{1}^B \oplus D_1 \xrightarrow{\mathbb{1}^B \oplus \mathrm{prog}} \mathbb{1}^B \oplus \mathbb{1}^B \oplus D_2 \xrightarrow{\mu \oplus \mathrm{id}_{D_2}} \mathbb{1}^B \oplus D_2$$

Where $\mathbb{1}^B = \bigoplus\limits_{b:B} \mathbb{C} \cdot P_b \;\in\; \mathrm{CMon}\big(\mathrm{LType}\big)$ is algebra of B-projection operators :

unit

$$\mathbb{1} \xrightarrow[\;1 \mapsto \sum\limits_{b:B} P_b\;]{\eta} \mathbb{1}^B$$

product

$$\mathbb{1}^B \otimes \mathbb{1}^B \xrightarrow[\;P_b \otimes P_{b'} \mapsto \begin{cases} P_b & \text{if } b=b' \\ 0 & \text{else} \end{cases}\;]{\mu} \mathbb{1}^B$$

Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

(see nLab:quantum+reader+monad)

$$\begin{tikzcd}
 & & \bigcirc_B \arrow[d, phantom, sloped, "=:"] \\
 & & B\text{-Reader} \arrow[d, phantom, sloped, "\simeq"] \\
 & & \mathbb{1}^B\text{-Writer} \arrow[dl, phantom, sloped, "\simeq"] \\
B : \text{FinType} \;\; \vdash & \mathbb{1}^B\text{-DualWriter} \arrow[dr, phantom, sloped, "\subset"] & \\
 & & \text{Monad}(\text{LType}) \\
 & \text{FrobMonad}(\text{LType}) \arrow[ur] &
\end{tikzcd}$$

$$\mathbb{1}^B\text{-Writer}(D) :\equiv \mathbb{1}^B \oplus D$$

$$\text{bind}^{\mathbb{1}^B\text{Writer}}\Big(D_1 \xrightarrow{\text{prog}} \mathbb{1}^B \oplus D_2\Big) :\equiv$$

$$\mathbb{1}^B \oplus D_1 \xrightarrow{\mathbb{1}^B \oplus \text{prog}} \mathbb{1}^B \oplus \mathbb{1}^B \oplus D_2 \xrightarrow{\mu \oplus \text{id}_{D_2}} \mathbb{1}^B \oplus D_2$$

Where $\mathbb{1}^B = \bigoplus_{b:B} \mathbb{C} \cdot P_b \;\in\; \text{CMon}(\text{LType})$ is Frobenius algebra of B-projection operators :

$$\begin{tikzcd}[column sep=huge]
\mathbb{1} \arrow[r, "\overset{\textbf{unit}}{\eta}", "1 \mapsto \sum_{b:B} P_b"'] & \mathbb{1}^B \arrow[r, "\overset{\textbf{co-product}}{\delta}", "P_b \mapsto P_b \otimes P_b"'] & \mathbb{1}^B \otimes \mathbb{1}^B \arrow[r, "\overset{\textbf{product}}{\mu}", "P_b \otimes P_{b'} \mapsto \begin{cases} P_b & \text{if } b=b' \\ 0 & \text{else} \end{cases}"'] & \mathbb{1}^B \arrow[r, "\overset{\textbf{co-unit}}{\varepsilon}", "P_b \mapsto 1"'] & \mathbb{1}
\end{tikzcd}$$

Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

(see nLab:quantum+reader+monad)

$$\begin{tikzcd}
 & & \bigcirc_B \arrow[d, phantom, "\shortparallel"] \\
 & & B\text{-Reader} \arrow[d, phantom, "\wr"] \\
 & \mathbb{1}^B\text{-CoWriter} & & \mathbb{1}^B\text{-Writer} \\
B : \text{FinType} \;\; \vdash & & \mathbb{1}^B\text{-DualWriter} \arrow[ul, phantom, "\simeq"] \arrow[ur, phantom, "\simeq"] \arrow[d, phantom, "\Cap"] & \\
 & \text{CoMonad}\big(\text{LType}\big) & & \text{Monad}\big(\text{LType}\big) \\
 & & \text{FrobMonad}\big(\text{LType}\big) \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$$\mathbb{1}^B\text{-Writer}(D) :\equiv \mathbb{1}^B \oplus D$$

$$\text{bind}^{\mathbb{1}^B\text{Writer}}\Big(D_1 \xrightarrow{\text{prog}} \mathbb{1}^B \oplus D_2\Big) :\equiv$$

$$\mathbb{1}^B \oplus D_1 \xrightarrow{\mathbb{1}^B \oplus \text{prog}} \mathbb{1}^B \oplus \mathbb{1}^B \oplus D_2 \xrightarrow{\mu \oplus \text{id}_{D_2}} \mathbb{1}^B \oplus D_2$$

Where $\mathbb{1}^B = \bigoplus_{b:B} \mathbb{C} \cdot P_b \;\in\; \text{CMon}(\text{LType})$ is Frobenius algebra of B-projection operators :

$$\mathbb{1} \xrightarrow[1 \mapsto \sum_{b:B} P_b]{\overset{\textbf{unit}}{\eta}} \mathbb{1}^B \xrightarrow[P_b \mapsto P_b \otimes P_b]{\overset{\textbf{co-product}}{\delta}} \mathbb{1}^B \otimes \mathbb{1}^B \xrightarrow[P_b \otimes P_{b'} \mapsto \begin{cases} P_b \text{ if } b=b' \\ 0 \text{ else} \end{cases}]{\overset{\textbf{product}}{\mu}} \mathbb{1}^B \xrightarrow[P_b \mapsto 1]{\overset{\textbf{co-unit}}{\varepsilon}} \mathbb{1}$$

Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

(see nLab:quantum+reader+monad)

$$\begin{tikzcd}
 & & & \bigcirc_B \arrow[d, phantom, "\rotatebox{90}{$=:$}"] \\
 & B\text{-CoReader} \arrow[d, phantom, "\rotatebox{90}{$\simeq$}"] & & B\text{-Reader} \arrow[d, phantom, "\rotatebox{90}{$\simeq$}"] \\
 & \mathbb{1}^B\text{-CoWriter} & & \mathbb{1}^B\text{-Writer} \\
B : \mathrm{FinType} \;\vdash & & \mathbb{1}^B\text{-DualWriter} \arrow[ul, phantom, sloped, "\simeq"] \arrow[ur, phantom, sloped, "\simeq"] & \\
 & \mathrm{CoMonad}\big(\mathrm{LType}\big) & \rotatebox{-90}{$\in$} & \mathrm{Monad}\big(\mathrm{LType}\big) \\
 & & \mathrm{FrobMonad}\big(\mathrm{LType}\big) \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$$\mathbb{1}^B\text{-Writer}(D) := \mathbb{1}^B \oplus D$$

$$\mathrm{bind}^{\mathbb{1}^B\mathrm{Writer}}\Big(D_1 \xrightarrow{\mathrm{prog}} \mathbb{1}^B \oplus D_2\Big) := \mathbb{1}^B \oplus D_1 \xrightarrow{\mathbb{1}^B \oplus \mathrm{prog}} \mathbb{1}^B \oplus \mathbb{1}^B \oplus D_2 \xrightarrow{\mu \oplus \mathrm{id}_{D_2}} \mathbb{1}^B \oplus D_2$$

Where $\mathbb{1}^B = \bigoplus_{b:B} \mathbb{C} \cdot P_b \in \mathrm{CMon}(\mathrm{LType})$ is Frobenius algebra of B-projection operators :

$$\mathbb{1} \xrightarrow[1 \mapsto \sum_{b:B} P_b]{\overset{\textbf{unit}}{\eta}} \mathbb{1}^B \xrightarrow[P_b \mapsto P_b \otimes P_b]{\overset{\textbf{co-product}}{\delta}} \mathbb{1}^B \otimes \mathbb{1}^B \xrightarrow[P_b \otimes P_{b'} \mapsto \begin{cases} P_b \text{ if } b = b' \\ 0 \text{ else} \end{cases}]{\overset{\textbf{product}}{\mu}} \mathbb{1}^B \xrightarrow[P_b \mapsto 1]{\overset{\textbf{co-unit}}{\varepsilon}} \mathbb{1}$$

Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

(see nLab:quantum+reader+monad)

$B : \mathrm{FinType} \;\; \vdash$

$$\begin{tikzcd}
\star_B \arrow[d, phantom, sloped, "\eqqcolon"] & & \bigcirc_B \arrow[d, phantom, sloped, "\eqqcolon"] \\
B\text{-CoReader} \arrow[d, phantom, sloped, "\simeq"] & & B\text{-Reader} \arrow[d, phantom, sloped, "\simeq"] \\
\mathbb{1}^B\text{-CoWriter} \arrow[dr, phantom, sloped, "\simeq"] & & \mathbb{1}^B\text{-Writer} \arrow[dl, phantom, sloped, "\simeq"] \\
& \mathbb{1}^B\text{-DualWriter} \arrow[d, phantom, sloped, "\in"] & \\
\mathrm{CoMonad}\big(\mathrm{LType}\big) & & \mathrm{Monad}\big(\mathrm{LType}\big) \\
& \mathrm{FrobMonad}\big(\mathrm{LType}\big) \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$\mathbb{1}^B\text{-Writer}(D) := \mathbb{1}^B \oplus D$

$\mathrm{bind}^{\mathbb{1}^B\mathrm{Writer}}\Big(D_1 \xrightarrow{\mathrm{prog}} \mathbb{1}^B \oplus D_2\Big) :=$

$\mathbb{1}^B \oplus D_1 \xrightarrow{\mathbb{1}^B \oplus \mathrm{prog}} \mathbb{1}^B \oplus \mathbb{1}^B \oplus D_2 \xrightarrow{\mu \oplus \mathrm{id}_{D_2}} \mathbb{1}^B \oplus D_2$

Where $\mathbb{1}^B = \bigoplus_{b:B} \mathbb{C} \cdot P_b \;\in\; \mathrm{CMon}\big(\mathrm{LType}\big)$ is Frobenius algebra of B-projection operators :

$$\mathbb{1} \xrightarrow[1 \,\mapsto\, \sum\limits_{b:B} P_b]{\overset{\textbf{unit}}{\eta}} \mathbb{1}^B \xrightarrow[P_b \,\mapsto\, P_b \otimes P_b]{\overset{\textbf{co-product}}{\delta}} \mathbb{1}^B \otimes \mathbb{1}^B \xrightarrow[P_b \otimes P_{b'} \,\mapsto\, \begin{cases} P_b & \text{if } b = b' \\ 0 & \text{else} \end{cases}]{\overset{\textbf{product}}{\mu}} \mathbb{1}^B \xrightarrow[P_b \,\mapsto\, 1]{\overset{\textbf{co-unit}}{\varepsilon}} \mathbb{1}$$

Aside: **Linear indefiniteness monad recovers Coecke’s “classical structures”.**

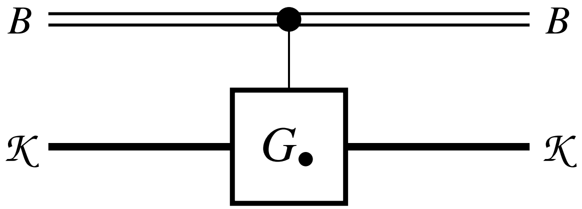
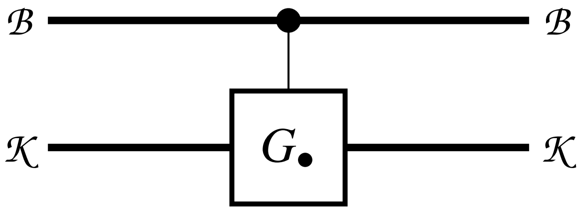
(see nLab:quantum+reader+monad)

$$\begin{tikzcd}
 & & \star_B \arrow[rr, equal] \arrow[dr, equal] & & \bigcirc_B \arrow[dl, equal] \\
B : \mathrm{FinType} \;\vdash & & & \mathbb{1}^B\text{-DualWriter} & \\
 & & \mathrm{CoMonad}\big(\mathrm{LType}\big) & \rotatebox{90}{$\in$} & \mathrm{Monad}\big(\mathrm{LType}\big) \\
 & & & \mathrm{FrobMonad}\big(\mathrm{LType}\big) \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Where $\mathbb{1}^B = \bigoplus_{b:B} \mathbb{C} \cdot P_b \;\in\; \mathrm{CMon}(\mathrm{LType})$ is Frobenius algebra of B-projection operators :

$$\begin{tikzcd}[column sep=huge]
\mathbb{1} \arrow[r, "\overset{\textbf{unit}}{\eta}", "1 \mapsto \sum\limits_{b:B} P_b"'] & \mathbb{1}^B \arrow[r, "\overset{\textbf{co-product}}{\delta}", "P_b \mapsto P_b \otimes P_b"'] & \mathbb{1}^B \otimes \mathbb{1}^B \arrow[r, "\overset{\textbf{product}}{\mu}", "P_b \otimes P_{b'} \mapsto \begin{cases} P_b \text{ if } b = b' \\ 0 \text{ else} \end{cases}"'] & \mathbb{1}^B \arrow[r, "\overset{\textbf{co-unit}}{\varepsilon}", "P_b \mapsto 1"'] & \mathbb{1}
\end{tikzcd}$$

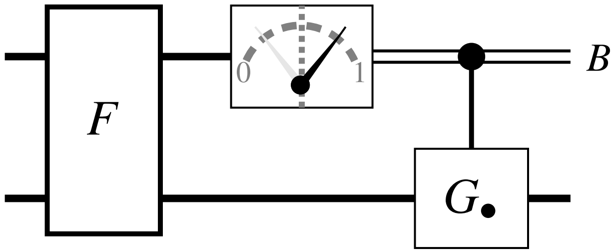
Exmp: Deferred measurement principle – Proven by monadic effect logic.

classically controlled gate	**quantumly controlled gate**
 $\mathcal{B}_\bullet \boxtimes \mathcal{K} \xrightarrow{G_\bullet} \mathcal{B}_\bullet \boxtimes \mathcal{K}$ $b : B \;\vdash\; \mathcal{K} \xrightarrow{G_b} \mathcal{K}$	 $\Box_B \mathcal{B}_\bullet \boxtimes \mathcal{K} \xrightarrow{\Box_B G_\bullet} \Box_B \mathcal{B}_\bullet \boxtimes \mathcal{K}$ $b : B \;\vdash\; \bigoplus_{b' B} \mathcal{K} \xrightarrow{\bigoplus_{b' : B} G_{b'}} \bigoplus_{b' B} \mathcal{K}$

Exmp: Deferred measurement principle – Proven by monadic effect logic.

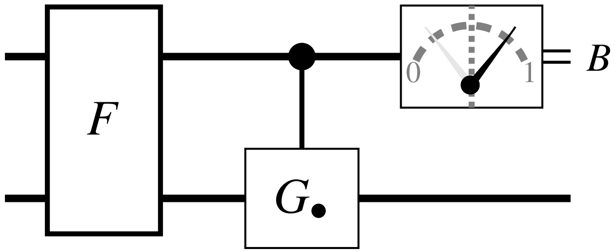
$$\Box_B \mathcal{H}_\bullet \xrightarrow{F} \Box_B \mathcal{H}_\bullet \xrightarrow{\varepsilon^{\Box_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet \xrightarrow{G_\bullet} \mathcal{H}_\bullet \quad \mapsto \quad \Box_B \mathcal{H}_\bullet \xrightarrow{\mathrm{diag}_B(G_\bullet)\circ F} \Box_B \mathcal{H}_\bullet \quad \mapsto \quad \Box_B \mathcal{H}_\bullet \xrightarrow{\mathrm{diag}_B(G_\bullet)\circ F} \Box_B \mathcal{H}_\bullet \xrightarrow{\varepsilon^{\Box_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet$$

measurement-controlled quantum gate **quantum-controlled quantum gate...** **...followed by measurement**

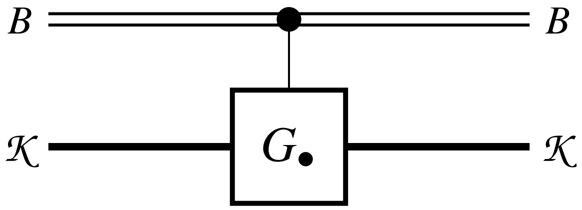


Deferred Measurement Principle

$\longleftrightarrow$



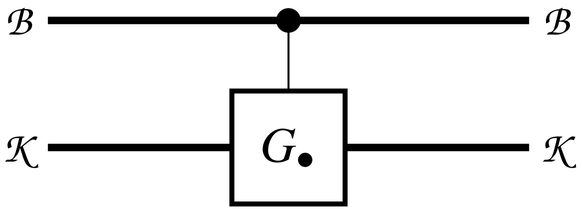
classically controlled gate



$$\mathcal{B}_\bullet \boxtimes \mathcal{K} \xrightarrow{\quad G_\bullet \quad} \mathcal{B}_\bullet \boxtimes \mathcal{K}$$

$$b : B \;\vdash\; \mathcal{K} \xrightarrow{\quad G_b \quad} \mathcal{K}$$

quantumly controlled gate



$$\Box_B \mathcal{B}_\bullet \boxtimes \mathcal{K} \xrightarrow{\quad \Box_B G_\bullet \quad} \Box_B \mathcal{B}_\bullet \boxtimes \mathcal{K}$$

$$b : B \;\vdash\; \bigoplus_{b' B} \mathcal{K} \xrightarrow{\quad \bigoplus_{b' : B} G_{b'} \quad} \bigoplus_{b' B} \mathcal{K}$$

Exmp: Deferred measurement principle – Proven by monadic effect logic.

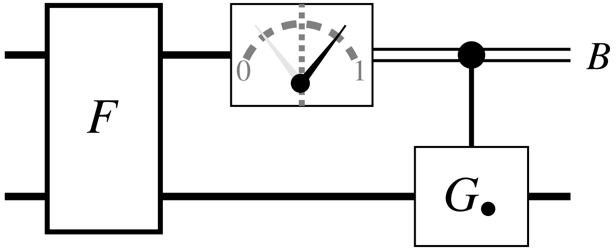
$$\begin{tikzcd} \mathrm{Kl}(\square_B) \arrow[r, "\sim", "\delta^B \circ \square_B(-)"'] \arrow[rr, bend left, "\mathrm{id}"] & \mathrm{LType}_{B\square_B} \arrow[r, "\sim", "\varepsilon^{\square_B} \circ (-)"'] & \mathrm{Kl}(\square_B) \end{tikzcd}$$

Kleisli equivalence

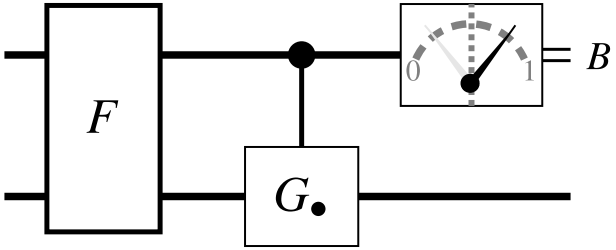
$\square_B$-Kleisli morphisms **$\square_B$-coalgebra homomorphisms** **$\square_B$-Kleisli morphisms**

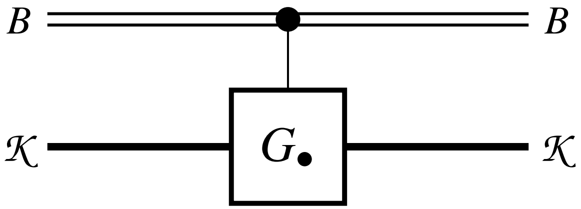
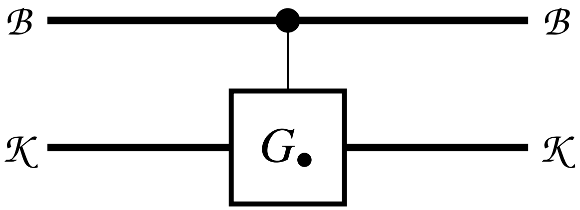
$$\square_B \mathcal{H}_\bullet \xrightarrow{F} \square_B \mathcal{H}_\bullet \xrightarrow{\varepsilon^{\square_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet \xrightarrow{G_\bullet} \mathcal{H}_\bullet \quad \mapsto \quad \square_B \mathcal{H}_\bullet \xrightarrow{\mathrm{diag}_B(G_\bullet) \circ F} \square_B \mathcal{H}_\bullet \quad \mapsto \quad \square_B \mathcal{H}_\bullet \xrightarrow{\mathrm{diag}_B(G_\bullet) \circ F} \square_B \mathcal{H}_\bullet \xrightarrow{\varepsilon^{\square_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet$$

measurement-controlled quantum gate **quantum-controlled quantum gate...** **...followed by measurement**



Deferred Measurement Principle ⟷



classically controlled gate	**quantumly controlled gate**
 $\mathcal{B}_\bullet \boxtimes \mathcal{K} \xrightarrow{G_\bullet} \mathcal{B}_\bullet \boxtimes \mathcal{K}$ $b : B \vdash \quad \mathcal{K} \xrightarrow{G_b} \mathcal{K}$	 $\square_B \mathcal{B}_\bullet \boxtimes \mathcal{K} \xrightarrow{\square_B G_\bullet} \square_B \mathcal{B}_\bullet \boxtimes \mathcal{K}$ $b : B \vdash \quad \bigoplus_{b' B} \mathcal{K} \xrightarrow{\bigoplus_{b' : B} G_{b'}} \bigoplus_{b' B} \mathcal{K}$

The Quantum modality.

Also the *exponential modality* traditionally postulated in linear logic is an emergent effect in `LHoTT`,

linear randomization
aka: stabilization/motivization

$$\begin{tikzcd} \mathrm{Type} \arrow[r, shift left=2, "\Sigma^\infty_+ \,:\, B \,\mapsto\, \star_B \mathbb{1}"] \arrow[r, phantom, "\perp"] & \mathrm{LType} \arrow[l, shift left=2, "\Omega^\infty"] \arrow[loop right, "!"] \end{tikzcd}$$

exponential modality

classical types

linear types

The Quantum modality.

Also the *exponential modality* traditionally postulated in linear logic
is an emergent effect in `LHoTT`,
as is the crucial *Quantum Modality*, not considered before:

linear randomization

aka: stabilization/motivization

quantum modality

$$\begin{tikzcd} \mathrm{Type} \arrow[loop left, "Q"] \arrow[r, shift left=2, "\Sigma^\infty_+ \,:\, B \,\mapsto\, \star_B \mathbb{1}"] \arrow[r, phantom, "\perp"] & \mathrm{LType} \arrow[l, shift left=2, "\Omega^\infty"] \arrow[loop right, "!"] \end{tikzcd}$$

exponential modality

classical types

linear types

The Quantum modality.

Also the *exponential modality* traditionally postulated in linear logic
is an emergent effect in `LHoTT`,
as is the crucial *Quantum Modality*, not considered before:

linear randomization

aka: stabilization/motivization

quantum modality

$$\begin{tikzcd}
\mathrm{Type} \arrow[loop left, "Q"] \arrow[r, shift left=2, "\Sigma^{\infty}_{+} \,:\, B \,\mapsto\, \star_{B} \mathbb{1}"] \arrow[r, phantom, "\perp"] & \mathrm{LType} \arrow[l, shift left=2, "\Omega^{\infty}"] \arrow[loop right, "!"]
\end{tikzcd}$$

exponential modality

classical types

linear types

The Q-monad plays a crucial role in the full formulation of the `QS`-language.
It is the secret actor behind QBit $=$ Q(Bit)...

The Quantum modality.

Also the *exponential modality* traditionally postulated in linear logic
is an emergent effect in `LHoTT`,
as is the crucial *Quantum Modality*, not considered before:

linear randomization

aka: stabilization/motivization

quantum modality

$$\begin{tikzcd} \mathrm{Type} \arrow[loop left, "\mathrm{Q}"] \arrow[r, shift left=2, "\Sigma^\infty_+ \,:\, B \,\mapsto\, \star_B \mathbb{1}"] \arrow[r, phantom, "\perp"] & \mathrm{LType} \arrow[l, shift left=2, "\Omega^\infty"] \arrow[loop right, "!"] \end{tikzcd}$$

exponential modality

classical types

linear types

The Q-monad plays a crucial role in the full formulation of the `QS`-language.
It is the secret actor behind QBit $=$ Q(Bit)...

Quantum Circuits

Quantum effects are compatible with tensor product.

Linear Randomness and Indefiniteness are “very strong” effects, in that:

$$\bigcirc_B\big(D \otimes D'\big) \;\simeq\; \big(\bigcirc_B D\big) \otimes D' \,, \qquad \star_B\big(D \otimes D'\big) \;\simeq\; \big(\star_B D\big) \otimes D'$$

There is a whole system of them:

$$\bigcirc_B \bigcirc_{B'} \;\simeq\; \bigcirc_{B\times B'} , \quad \text{NB: } \bigcirc_B \bigcirc'_B \;\simeq\; \bigcirc_B \mathbb{1} \otimes \bigcirc'_B$$

which under dynamic lifting (monadicity comparison functor)
gives the external tensor product of dependent linear types:

free $\bigcirc_B$-effect handlers
in linear data types

comparison
functor

B-dependent
linear data types

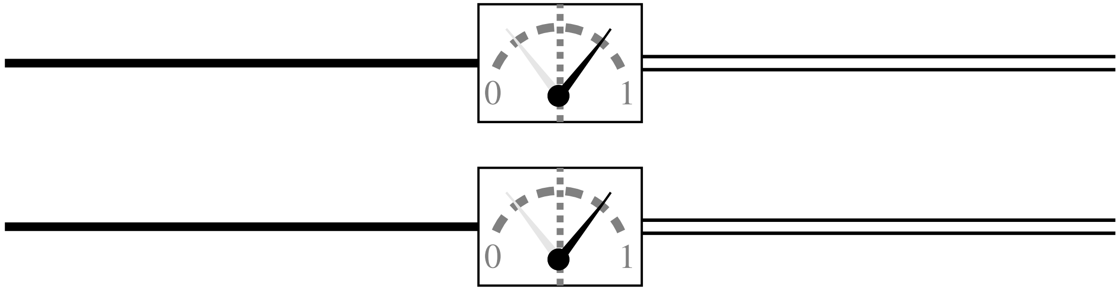
$$\begin{tikzcd}
\mathrm{LType}_{\bigcirc_B} \arrow[d, "K_{\oplus_B \mathbb{1}_B \otimes}"] & \overset{\curvearrowright\;\bigcirc_{B\times B'}}{\bigcirc_{B\times B'} \bigcirc_{B\times B'} \mathcal{H}} \arrow[d, mapsto] \\
\mathrm{LType}_B & \big(\Box_B \mathbb{1}_B\big) \boxtimes \big(\Box_{B'} \mathbb{1}_{B'}\big) \otimes \mathcal{H}
\end{tikzcd}$$

external
tensor product

Quantum circuits with classical control & effects

are the *effectful* string diagrams in the linear type system

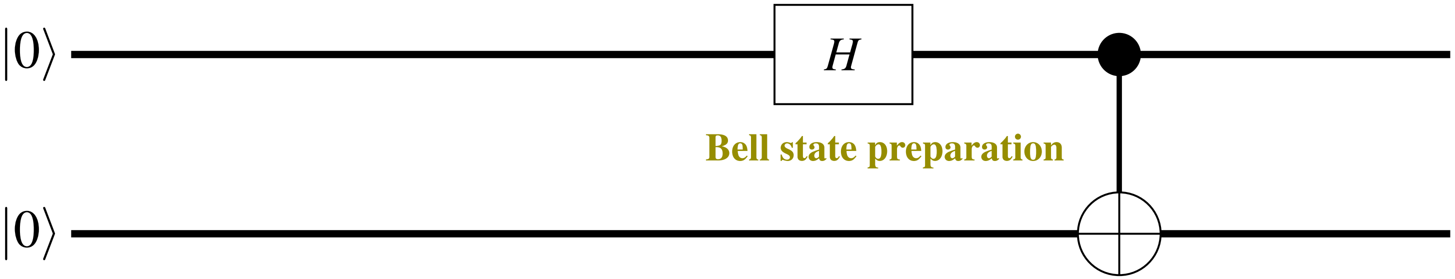
E.g.
The dependent linear type of a measurement on a pair of qbits:



$$\overset{\text{type of a pair of coherent qbits}}{\Box_{\mathrm{Bit}^2}\big(\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet\big)} \xrightarrow{\overset{\text{pair of measurements}}{\varepsilon_{\mathrm{Bit}^2}(\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet)}} \overset{\text{type of collapsed qbits dependent on measured bits } b, b'}{\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet}$$

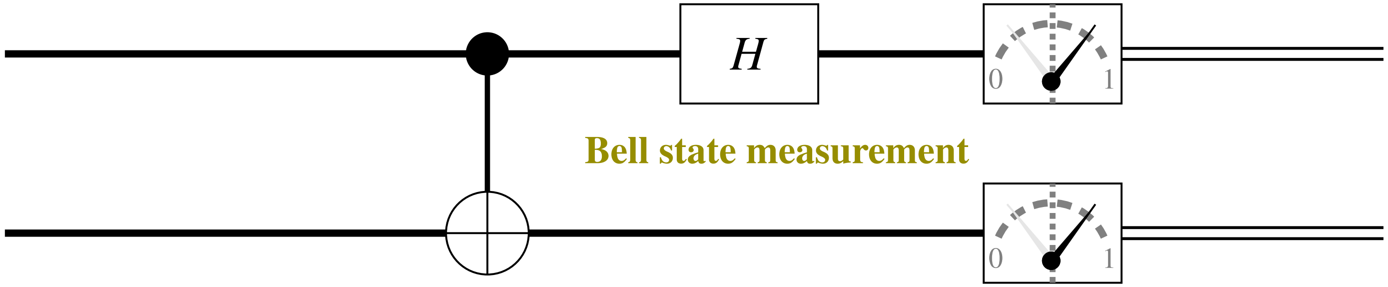
$$\overset{\text{measured bits}}{(b,b') : \mathrm{Bit}^2} \;\;\vdash\;\; \Box_{\mathrm{Bit}^2}\big(\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet\big)_{(b,b')} \simeq \mathbb{C}^2 \otimes \mathbb{C}^2 \xrightarrow[\text{collapse of the quantum state}]{\sum_{d,d'} q_{dd'}\;|d\rangle\otimes|d'\rangle \;\mapsto\; q_{bb'}\;|b\rangle\otimes|b'\rangle} \mathbb{C}\,.$$

Example: Bell states of q-bits are typed as follows (regarded in $\mathrm{LType}_{\mathrm{Bit}\times\mathrm{Bit}}$):



$$\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet \longrightarrow \big(\lozenge_{\mathrm{Bit}}\mathrm{QBit}_\bullet\big) \boxtimes \big(\lozenge_{\mathrm{Bit}}\mathrm{QBit}_\bullet\big) \simeq \square_{\mathrm{Bit}^2}\big(\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet\big) \longrightarrow \square_{\mathrm{Bit}^2}\big(\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet\big)$$

$$b, b' : \mathrm{Bit} \vdash \;\; \mathbb{C} \xrightarrow{\;1 \;\mapsto\; |0\rangle\otimes|0\rangle \;\mapsto\; \frac{1}{\sqrt{2}}\big(|0\rangle+|1\rangle\big)\otimes|0\rangle \;\mapsto\; \frac{1}{\sqrt{2}}\big(|0\rangle\otimes|0\rangle+|1\rangle\otimes|1\rangle\big)\;} \mathbb{C}^2\otimes\mathbb{C}^2$$

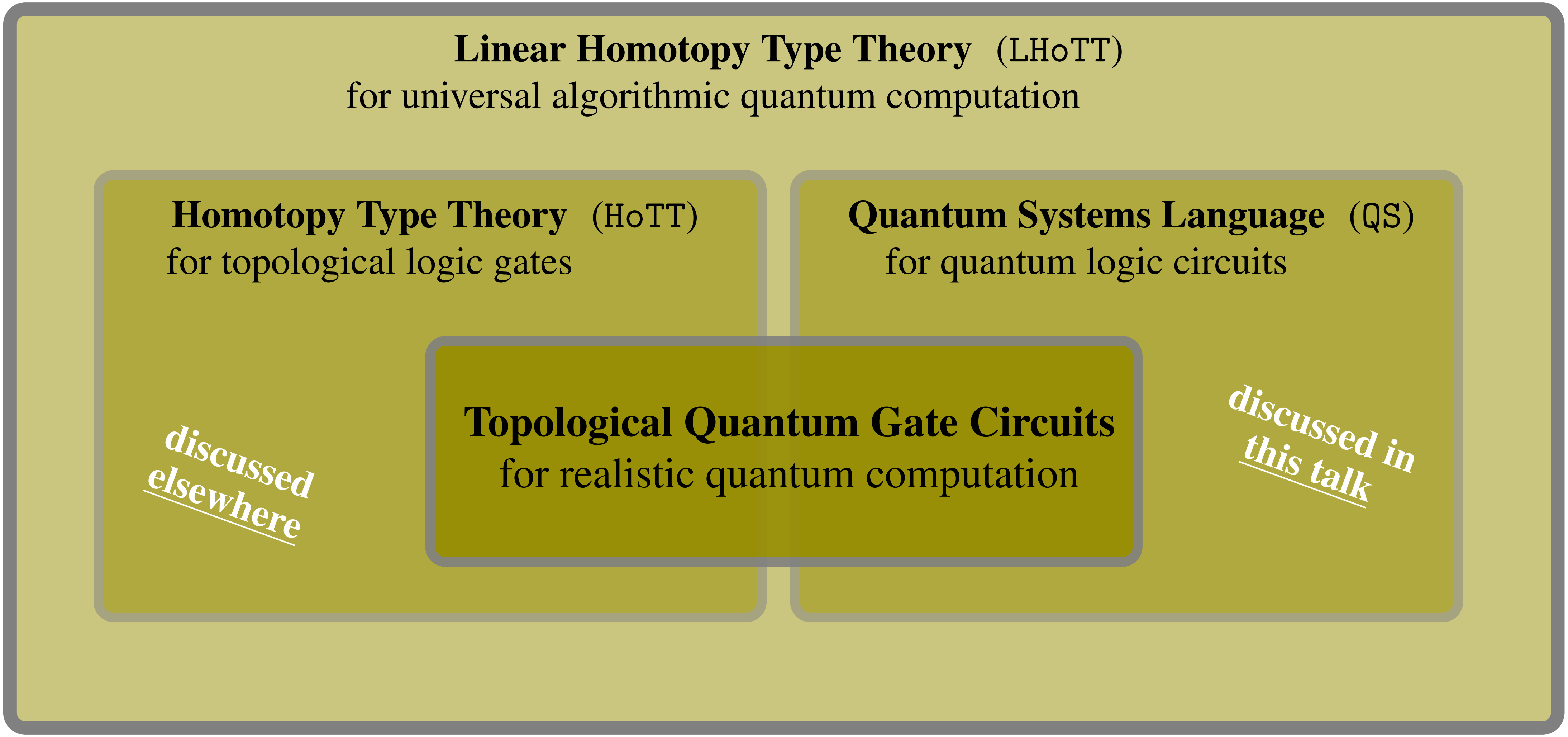


$$\square_{\mathrm{Bit}^2}\big(\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet\big) \longrightarrow \mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet$$

$$b_1, b_2 : \mathrm{Bit} \quad \vdash \quad \mathbb{C}^2\otimes\mathbb{C}^2 \xrightarrow{\;\Sigma_{b'_1 b'_2} q_{b'_1 b'_2}\cdot|b'_1\rangle\otimes|b'_2\rangle \;\mapsto\; \big(q_{0,b_2}+(-1)^{b_1}\cdot q_{1,(1-b_2)}\big)\cdot|b_1\rangle\otimes|b_2\rangle\;} \mathbb{C}$$

`QS` – Quantum Systems language @ CQTS

⇝ full-blown Quantum Systems language emerges embedded in `LHoTT`



Linear Homotopy Type Theory (`LHoTT`)
for universal algorithmic quantum computation

Homotopy Type Theory (`HoTT`)
for topological logic gates

Quantum Systems Language (`QS`)
for quantum logic circuits

Topological Quantum Gate Circuits
for realistic quantum computation

Effective Quantum Certification via Linear Homotopy Types

Urs Schreiber (NYU Abu Dhabi)
on joint work at CQTS with
D. J. Myers, M. Riley,
and Hisham Sati

Thanks!

CENTER FOR
QUANTUM &
TOPOLOGICAL
SYSTEMS

presentation at:

The Topos Institute Colloquium, 13 Apr 2023

slides and further pointers at: ncatlab.org/Quantum+Certification+via+Linear+Homotopy+Types#TI2023