Center for
Quantum&
Topological
Systems

# Quantum Data Types via Linear HoTT

presentation at:

## Workshop on Quantum Software @ QTML 2022

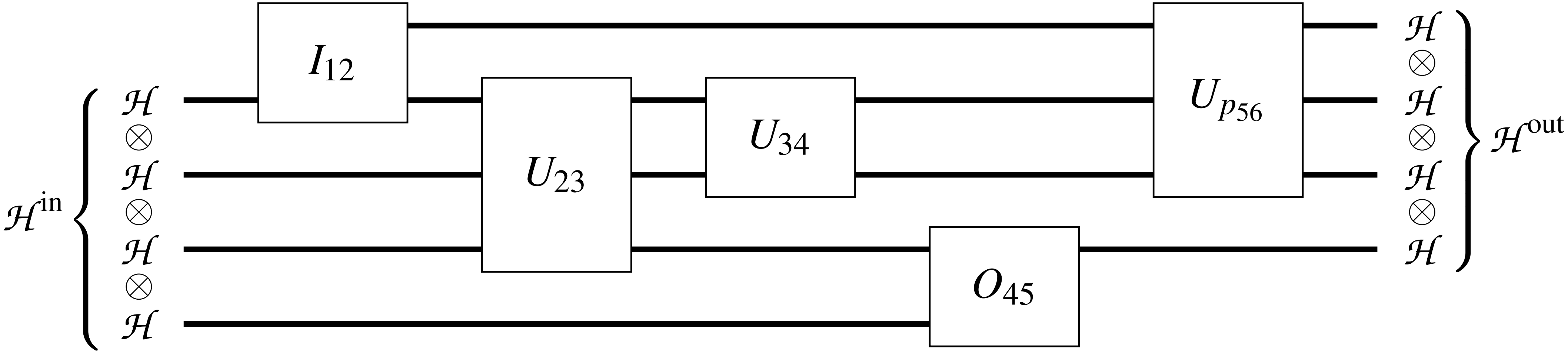
Urs Schreiber (NYU Abu Dhabi)
on joint work at CQTS with
D. J. Myers, M. Riley,
and Hisham Sati

slides and further pointers at: ncatlab.org/schreiber/show/QDataInLHoTT#QTML2022

# The Problem

# Pure quantum circuits are easy...

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

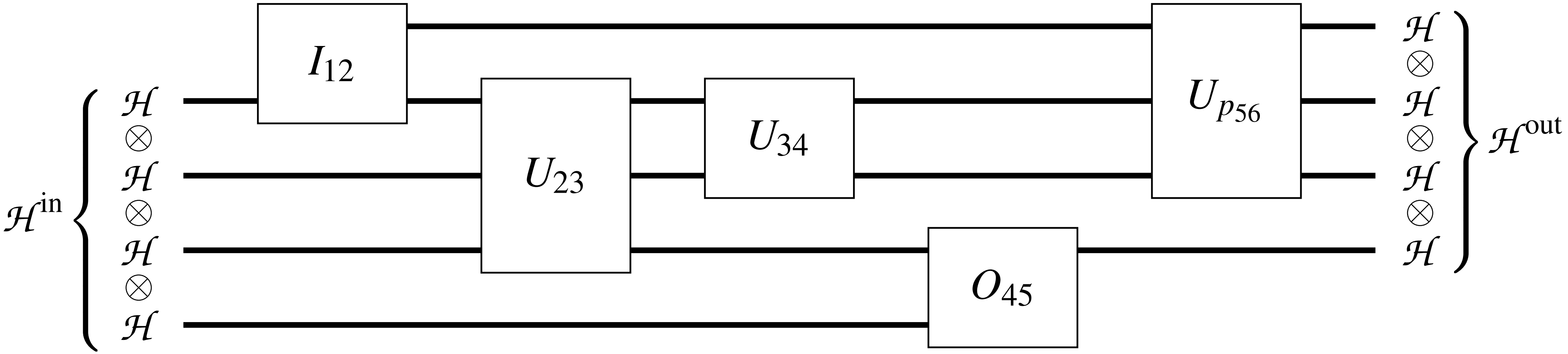


**Hilbert space of possible input quantum states** → **Hilbert space of possible output quantum states**

**linear transformation**
upon execution

# Pure quantum circuits are easy...

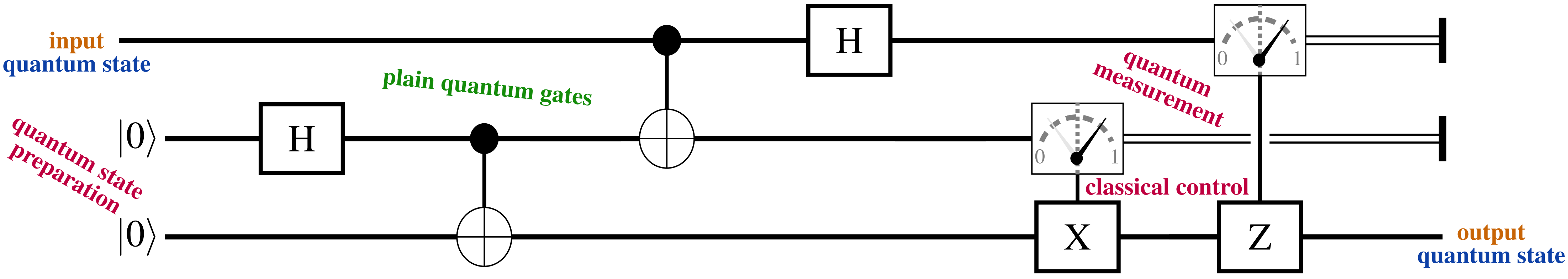
Linear operator composed & tensored from given *quantum logic gates*



**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**

**plain quantum gates**

**quantum measurement**

**quantum state preparation**

**classical control**

**output quantum state**

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

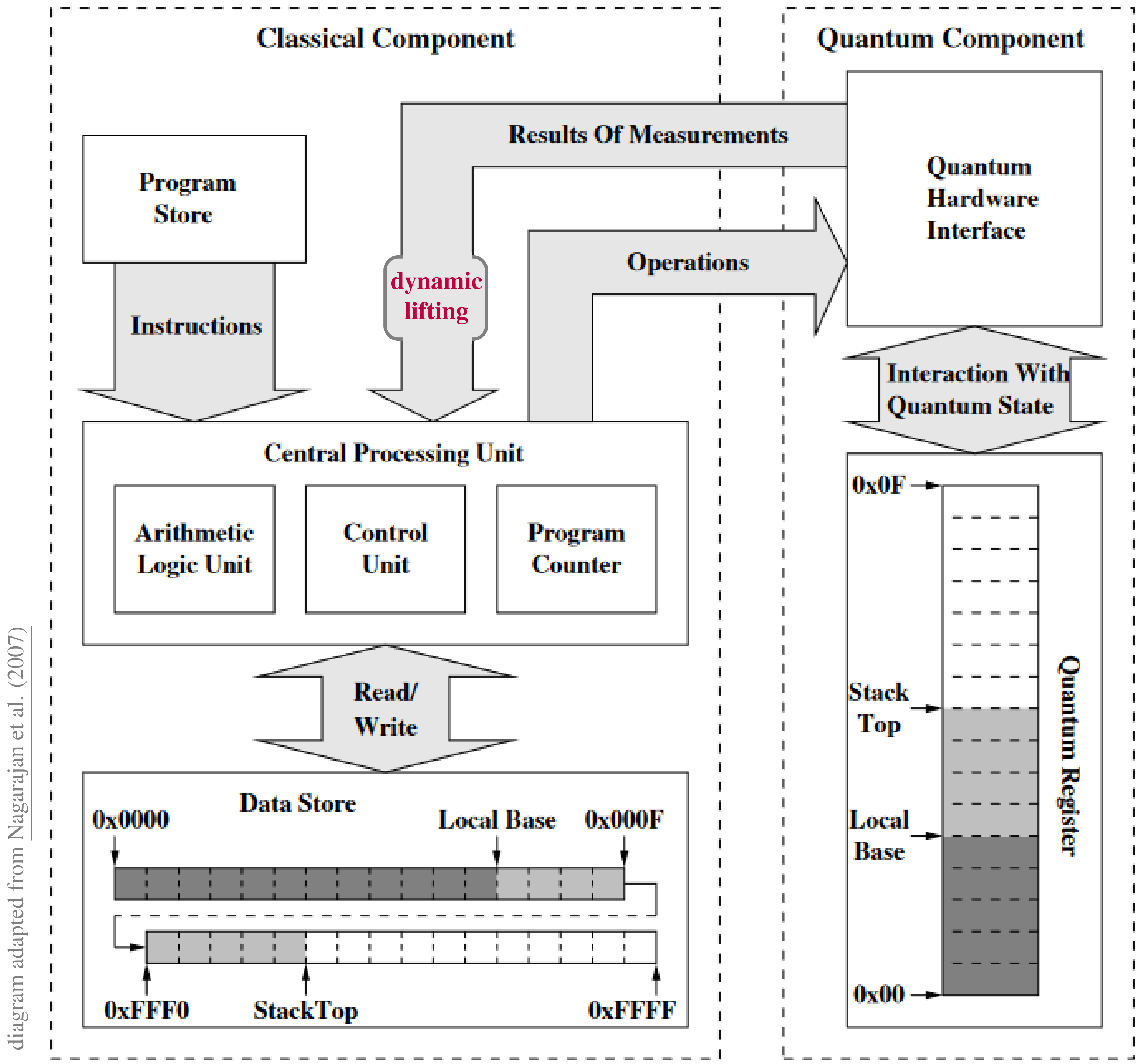
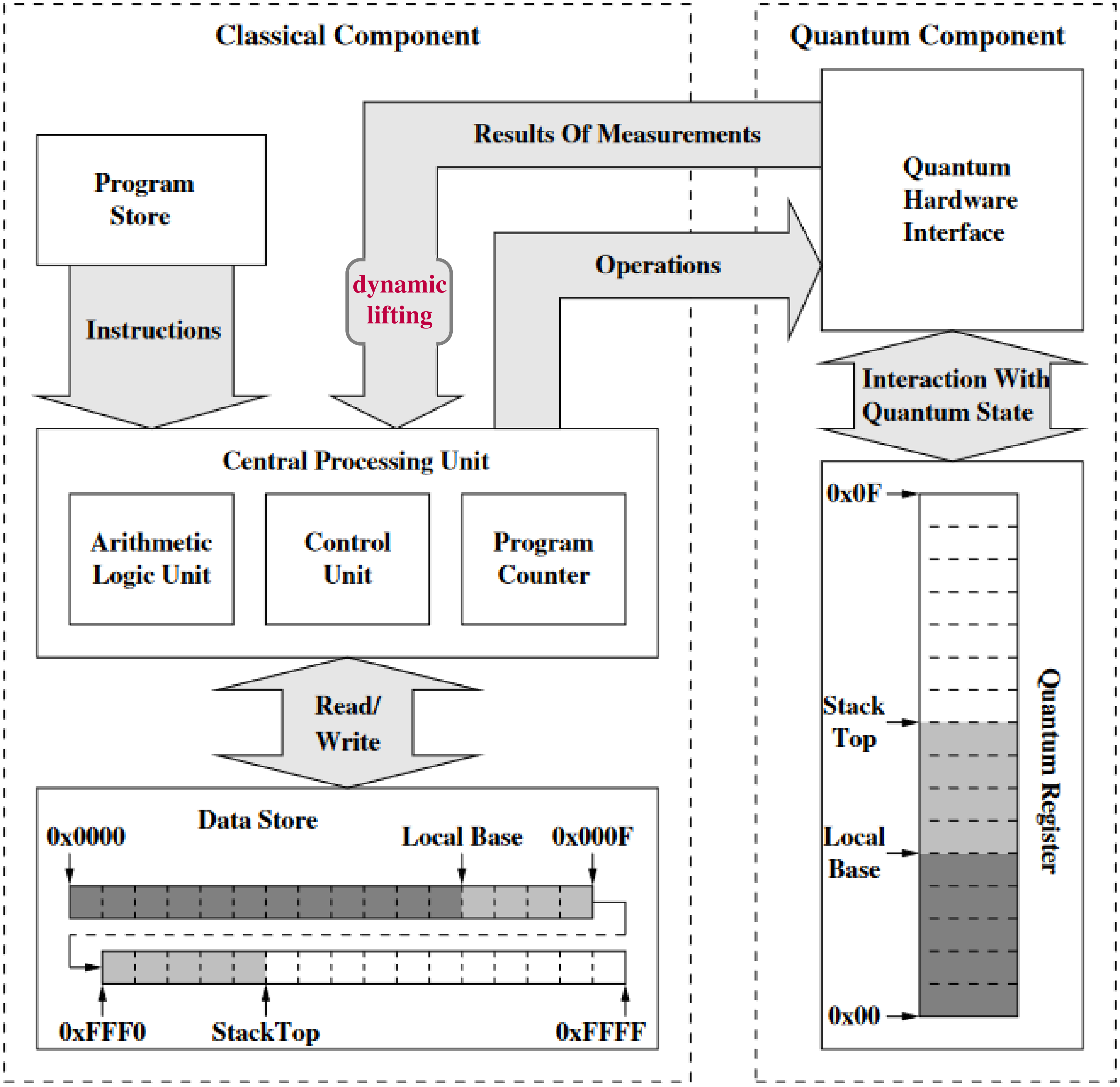


diagram adapted from Nagarajan et al. (2007)

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



full loop needed e.g. for quantum error correction

diagram adapted from Nagarajan et al. (2007)

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

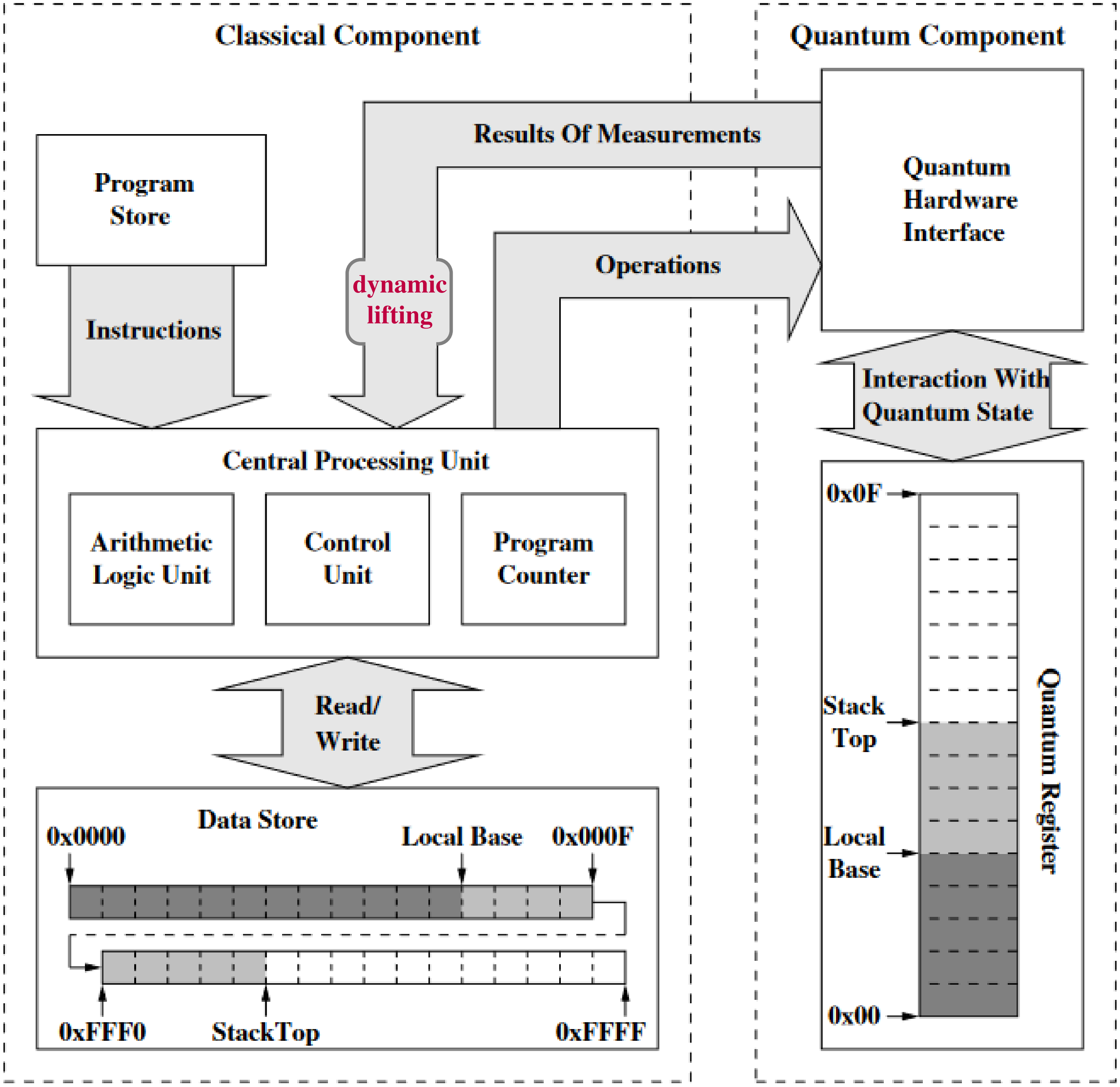


diagram adapted from Nagarajan et al. (2007)

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

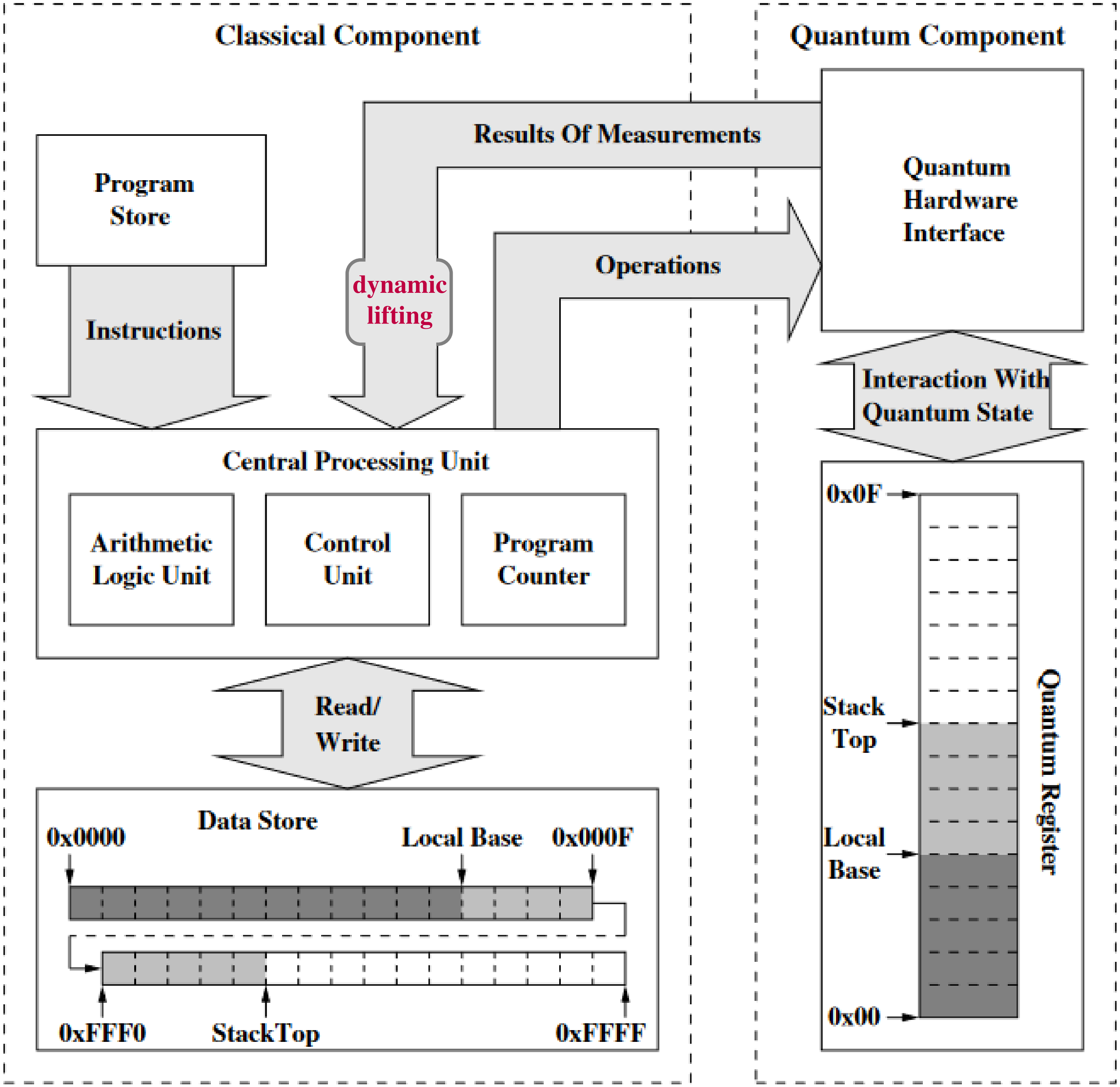


diagram adapted from Nagarajan et al. (2007)

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:

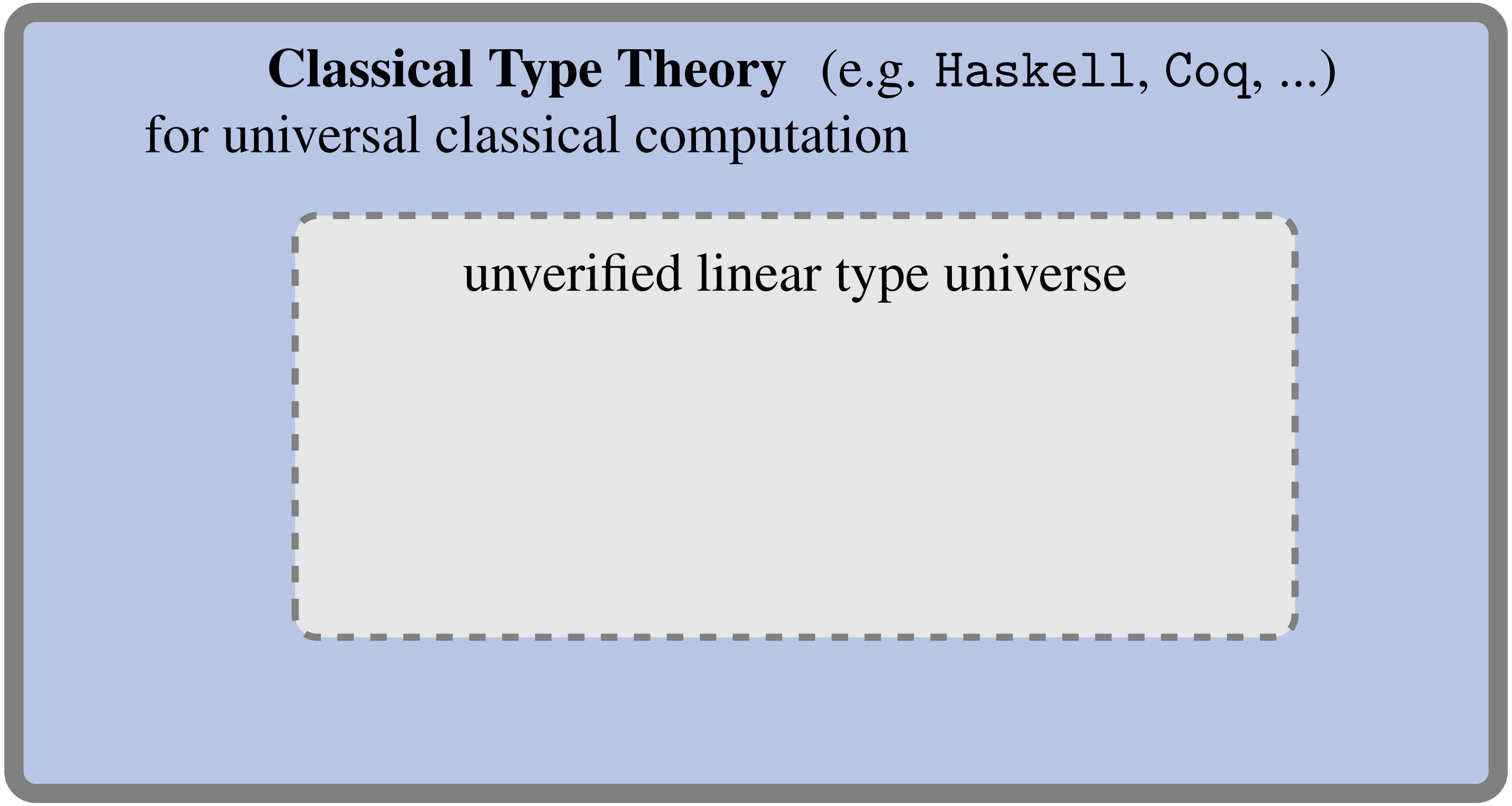
# Existing quantum typed circuit languages

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:

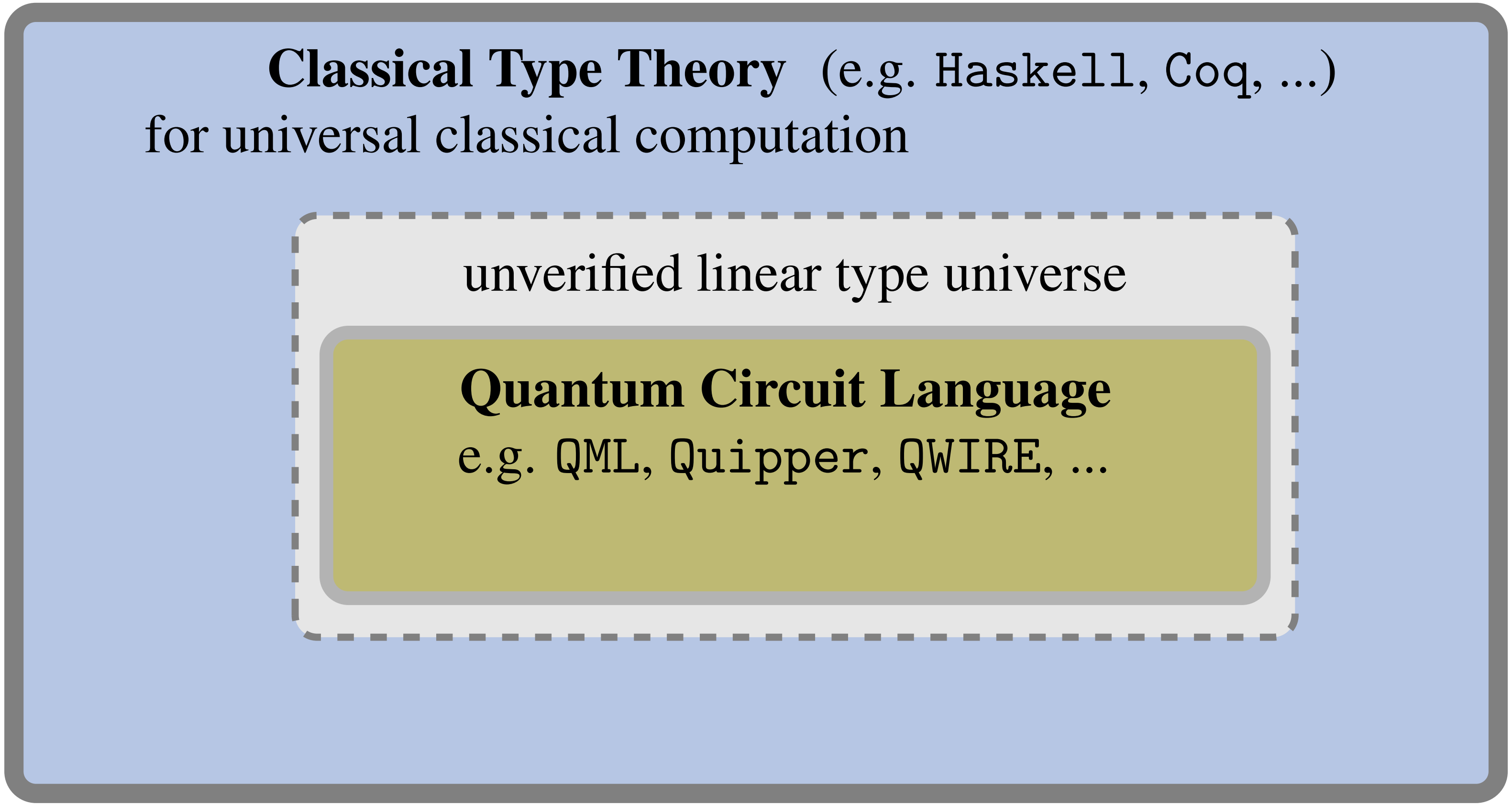


**Classical Type Theory** (e.g. `Haskell`, `Coq`, ...)
for universal classical computation

unverified linear type universe

# Existing quantum typed circuit languages

are embedded inside *classical* type theories:



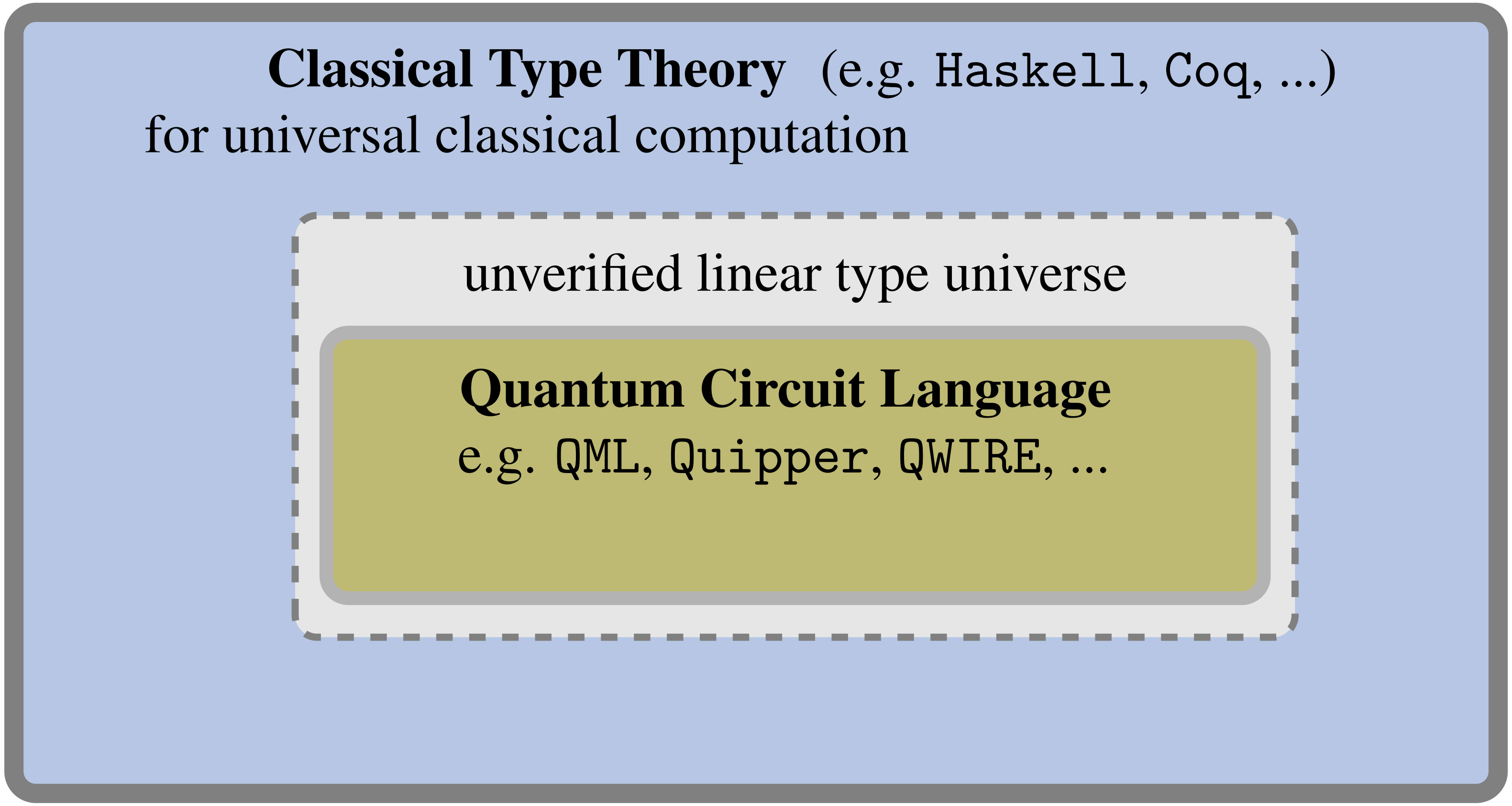
**Classical Type Theory** (e.g. `Haskell`, `Coq`, ...)
for universal classical computation

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

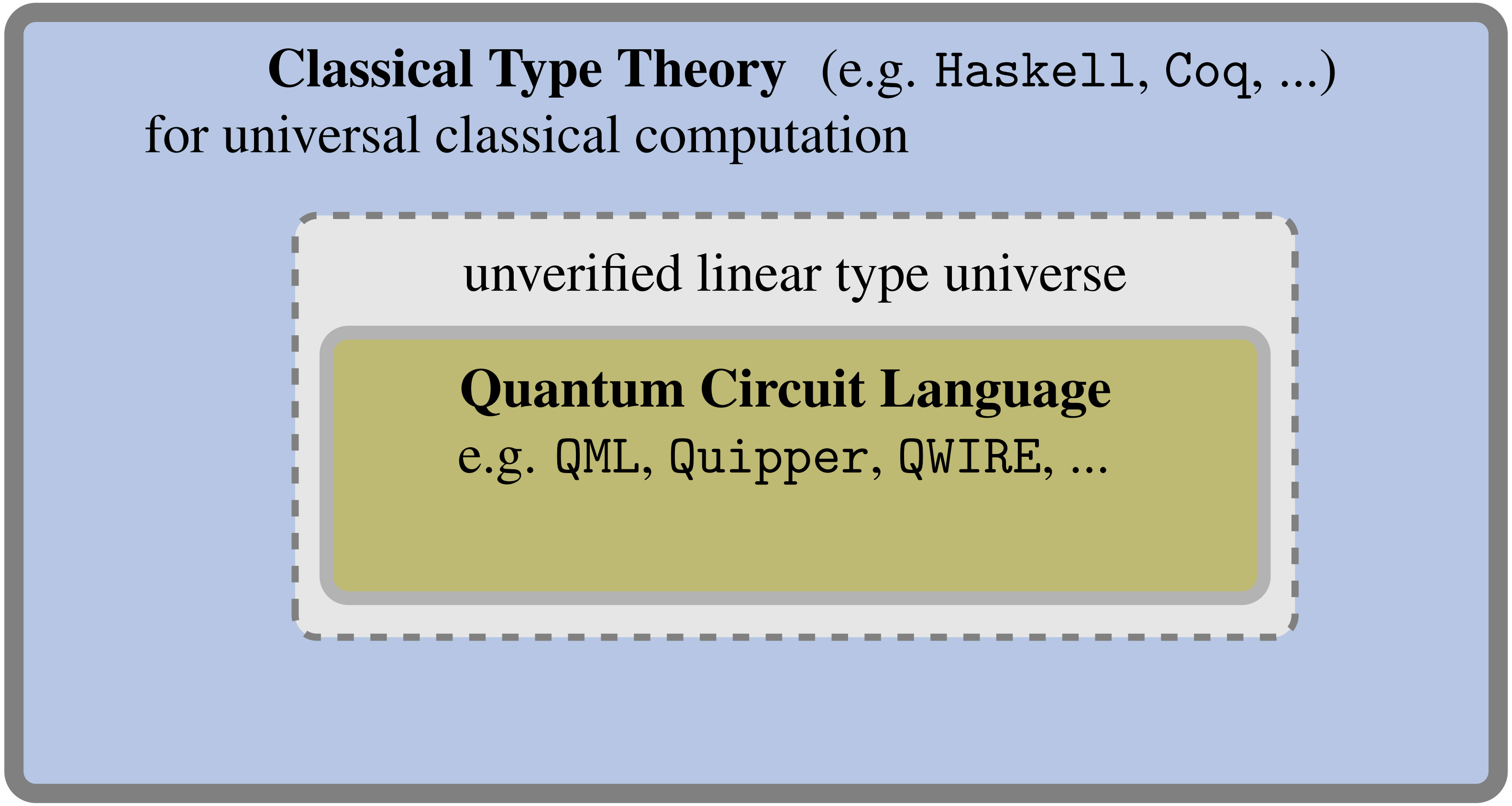
unverified linear type universe

**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.

Until now...

# Our Solution

# Dependent Linear Homotopy Type Theory (`dLHoTT`)

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

# Dependent Linear Homotopy Type Theory (`dLHoTT`)

---

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

$\exists$ *universal* quantum+classical specification language

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

# Dependent Linear Homotopy Type Theory (`dLHoTT`)

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

∃ *universal* quantum+classical specification language

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)

## **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

---

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

$\exists$ *universal* quantum+classical specification language

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)

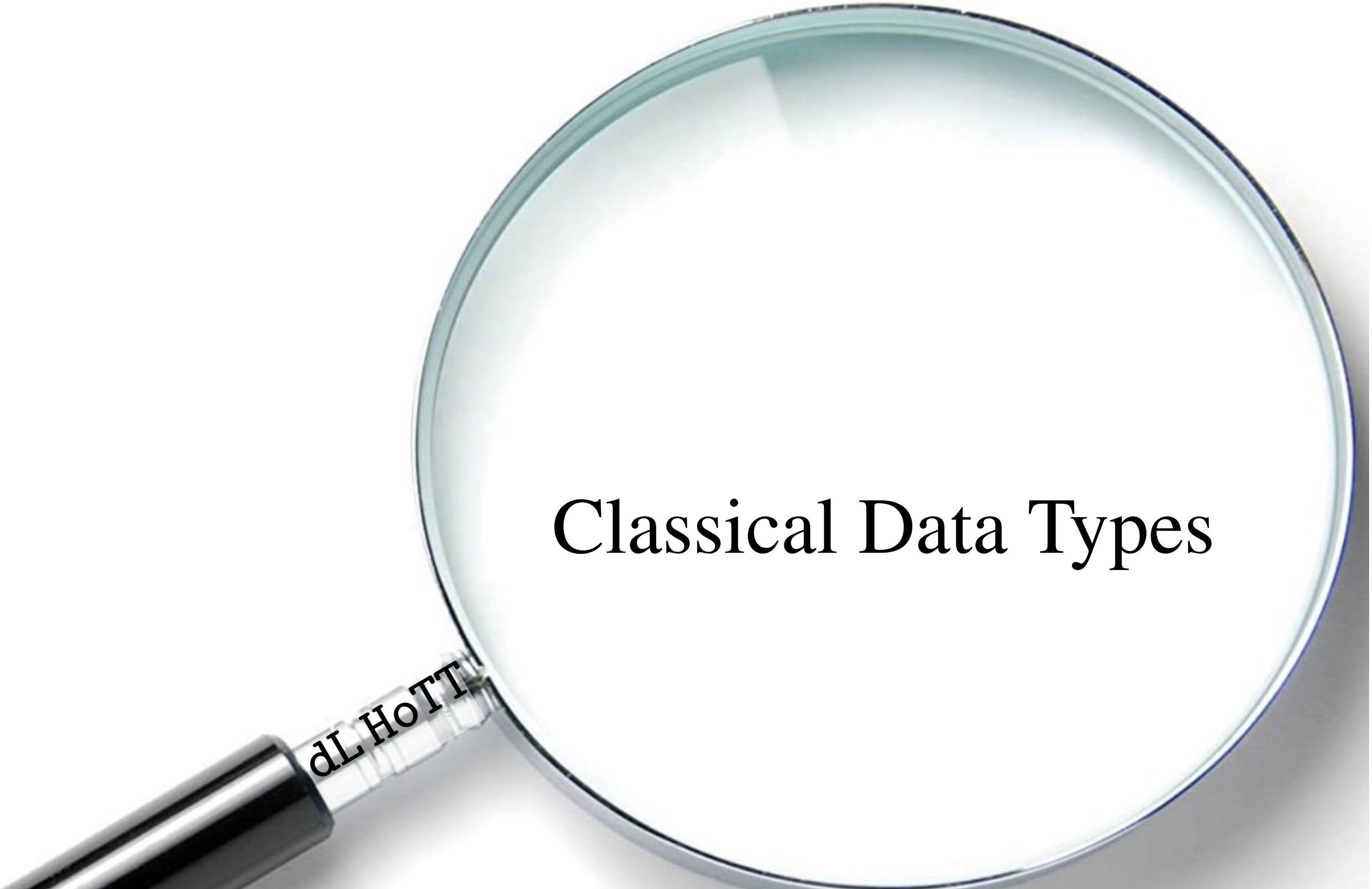
## Dependent Linear Homotopy Type Theory (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
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)

`dLHoTT` is like a quantum microscope for Classical Data Types

# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

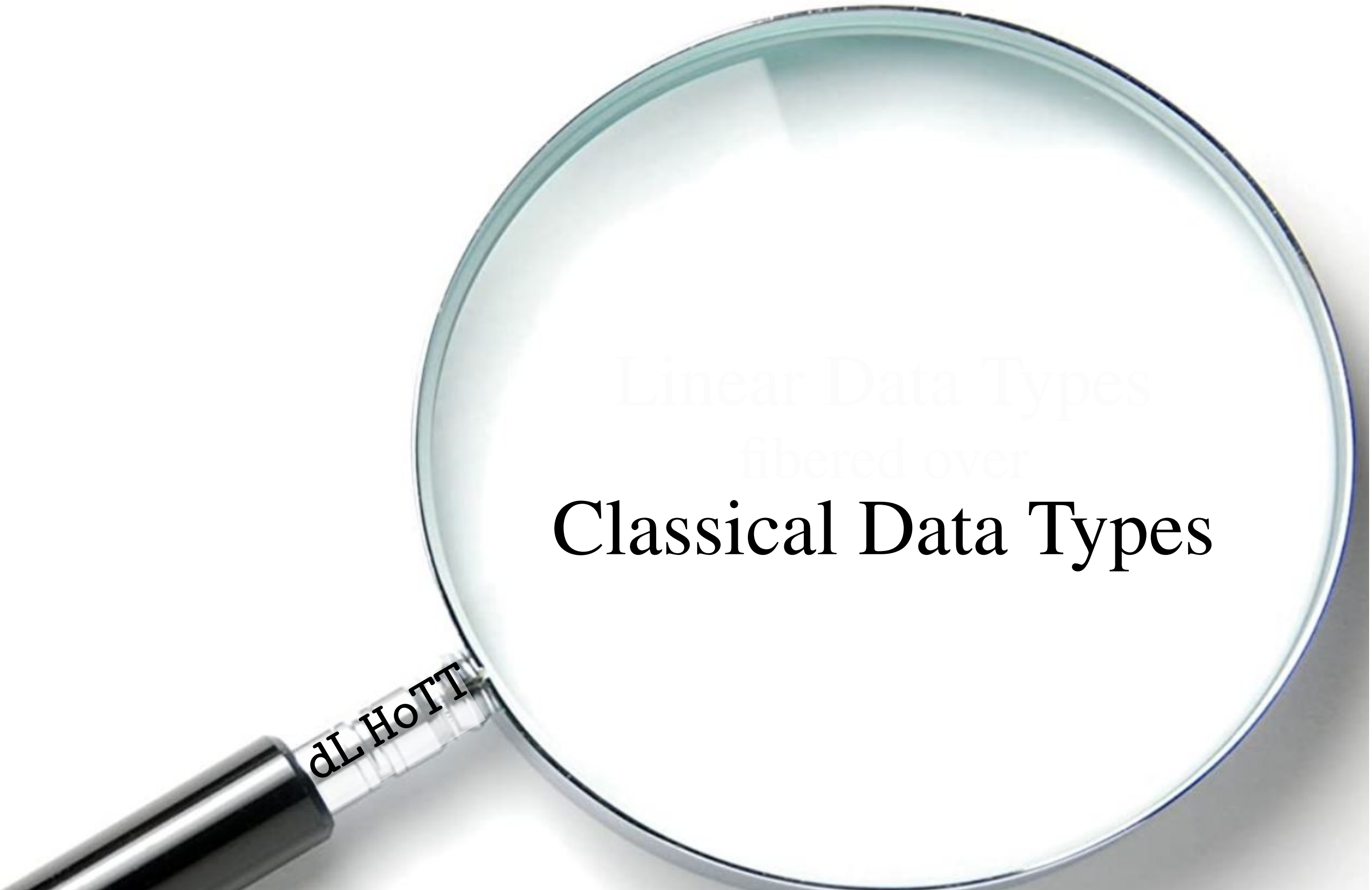
**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à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



## Dependent Linear Homotopy Type Theory (`dLHoTT`)

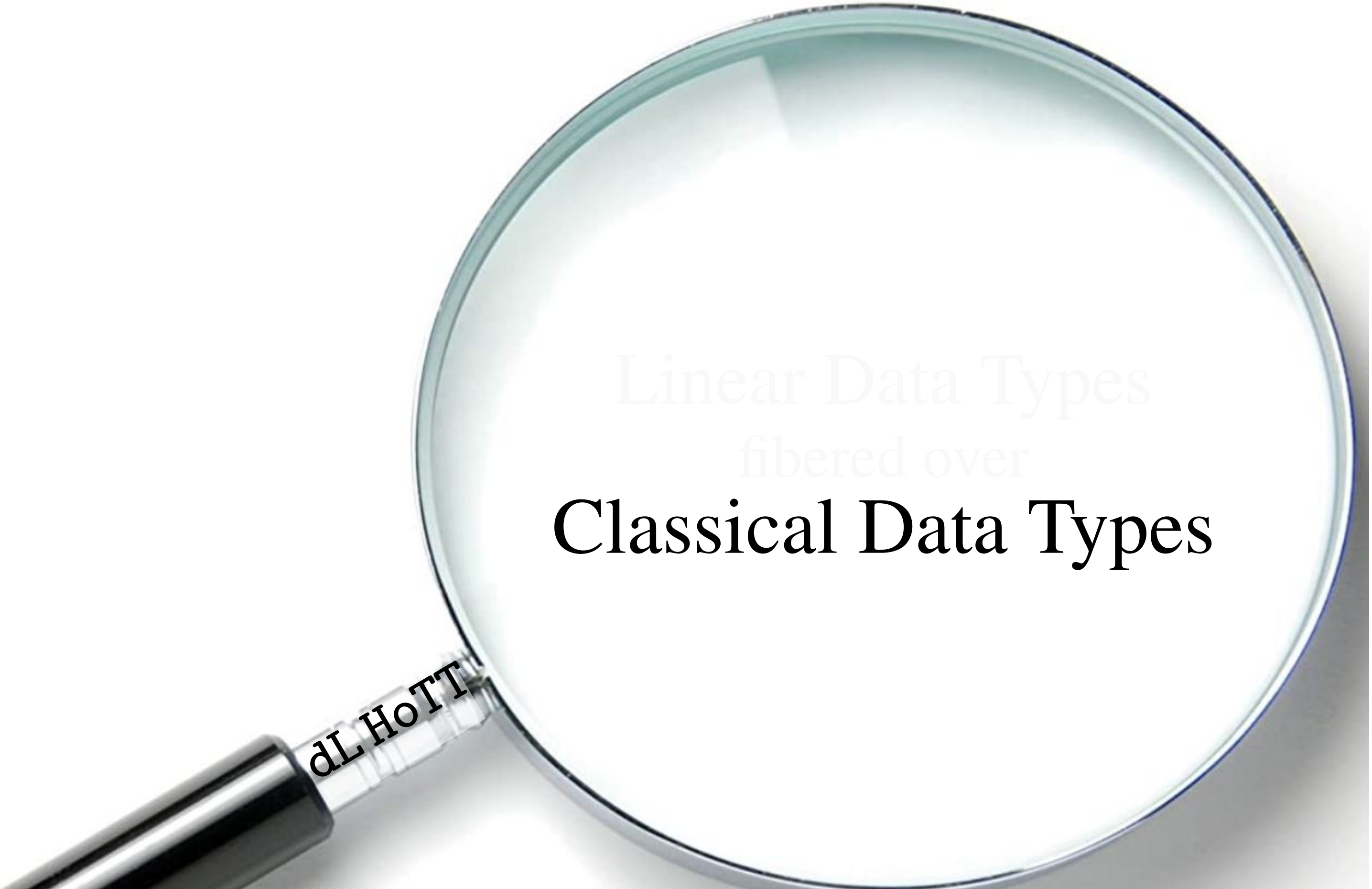
**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

$\exists$ *universal* quantum+classical specification language
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)



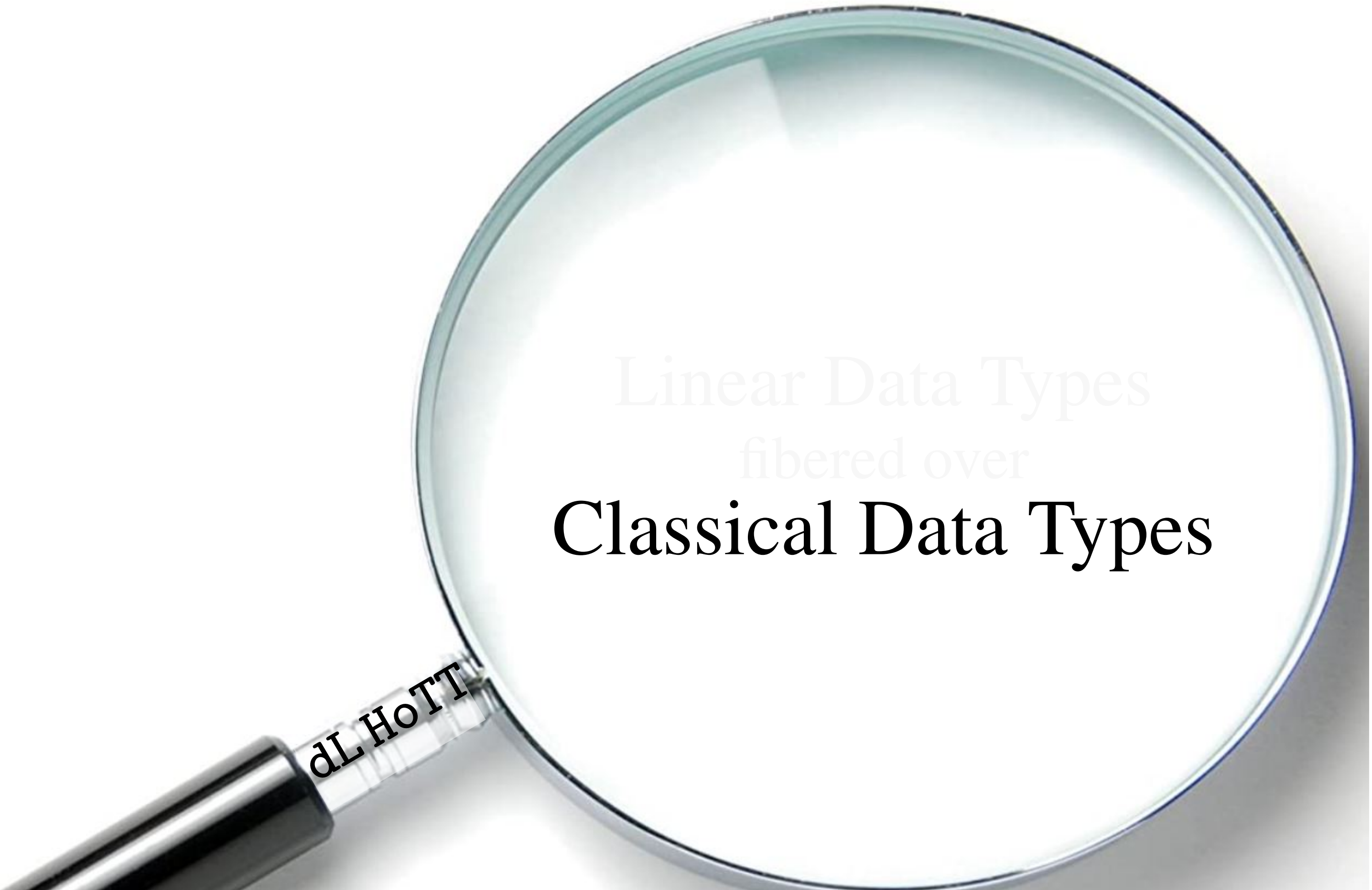
# Dependent Linear Homotopy Type Theory (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
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)



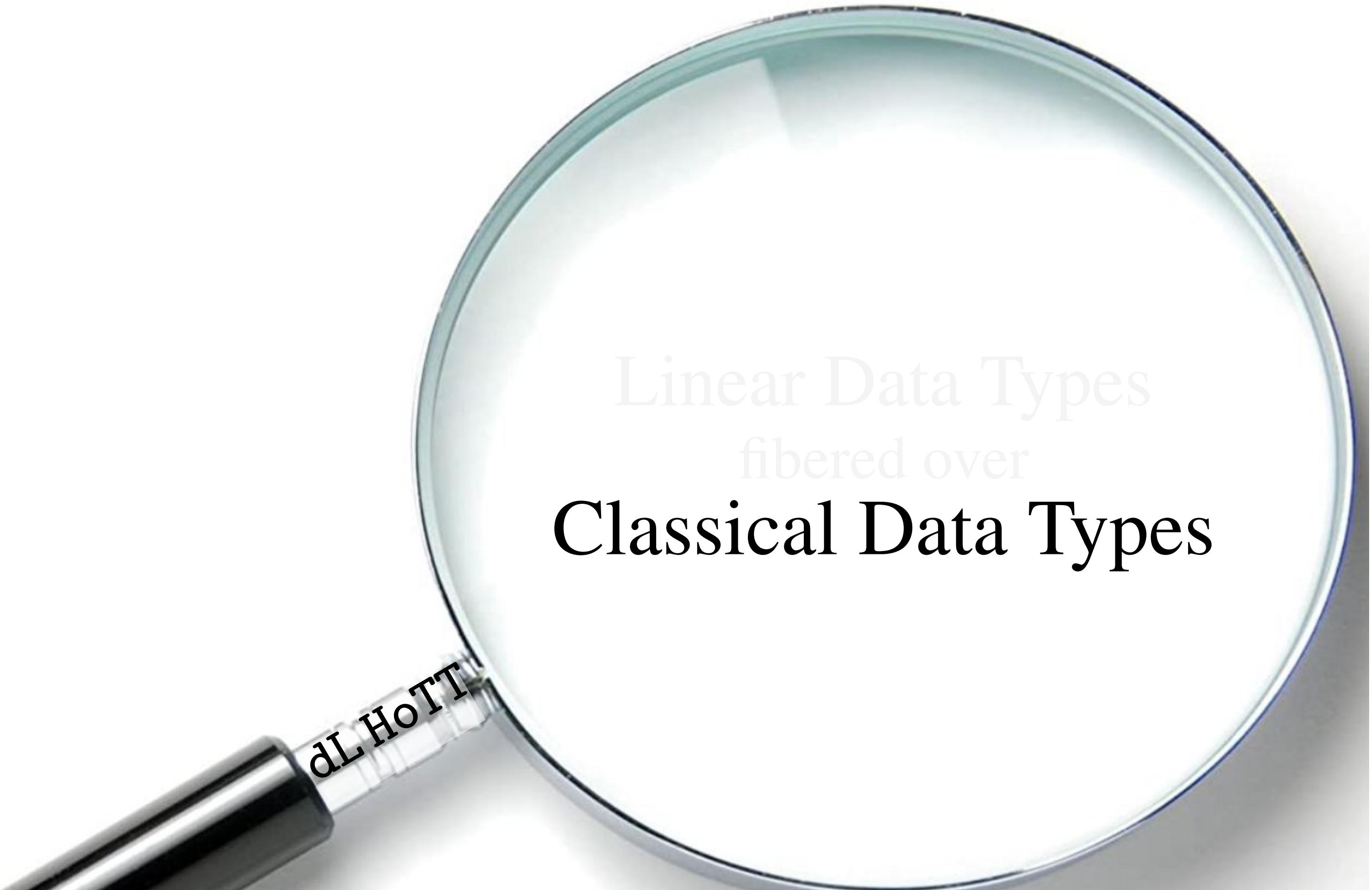
# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
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)



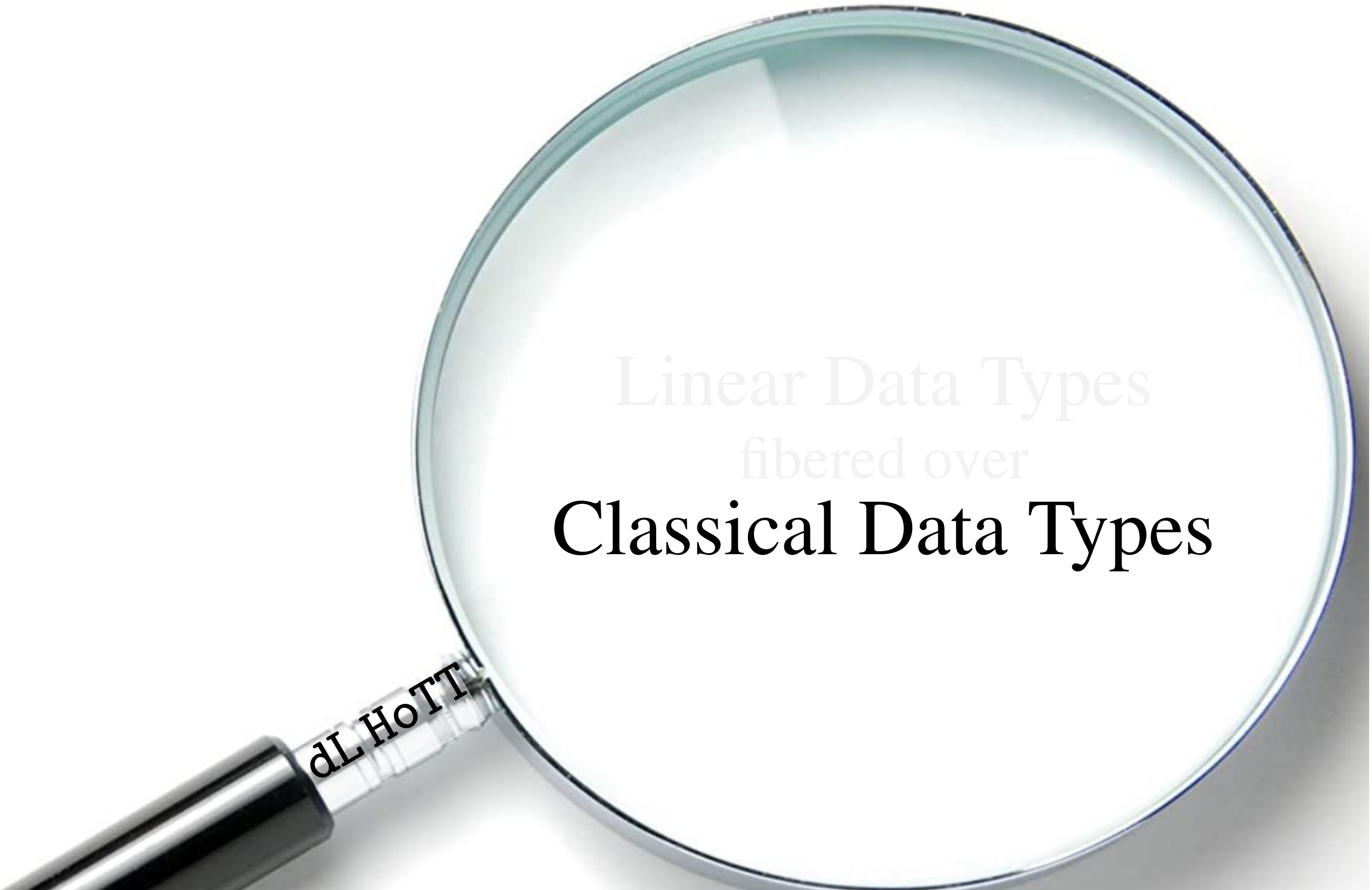
# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
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)



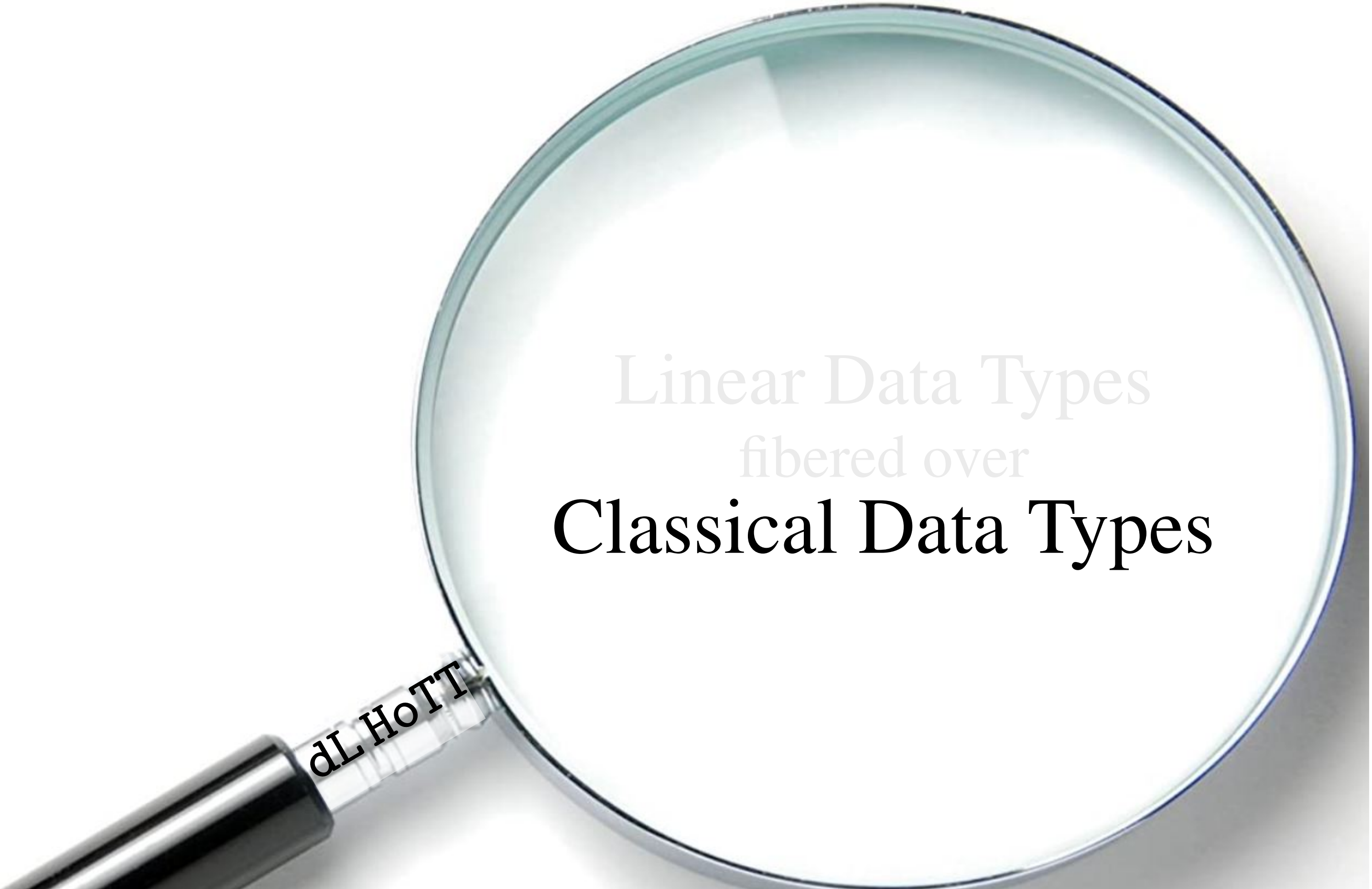
# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ *universal* quantum+classical specification language
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)



## Dependent Linear Homotopy Type Theory (`dLHoTT`)

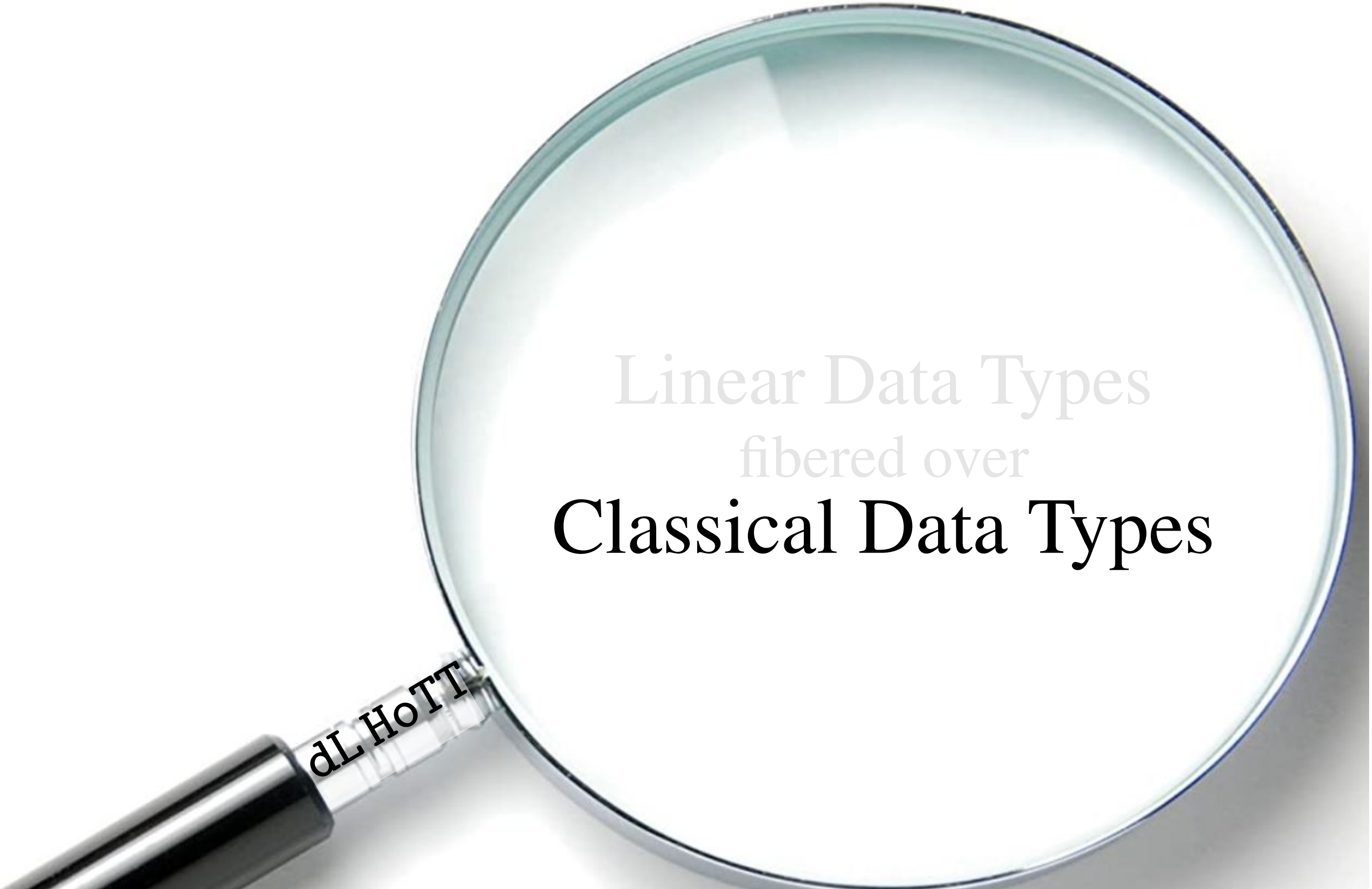
**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à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



# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

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

$\exists$ *universal* quantum+classical specification language
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)



## Dependent Linear Homotopy Type Theory (`dLHoTT`)

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

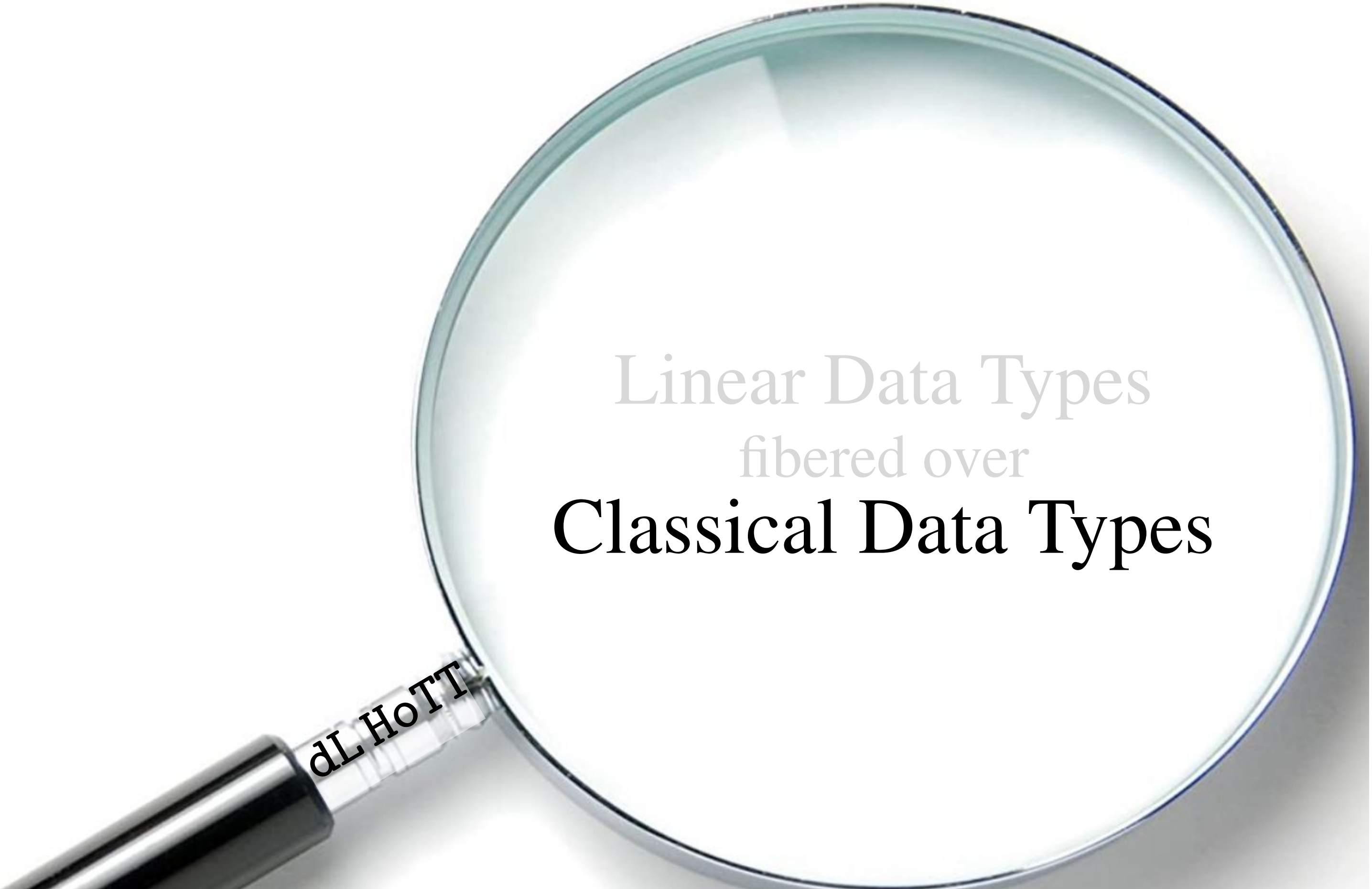
$\exists$ *universal* quantum+classical specification language

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)



# Dependent Linear Homotopy Type Theory (`dLHoTT`)

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

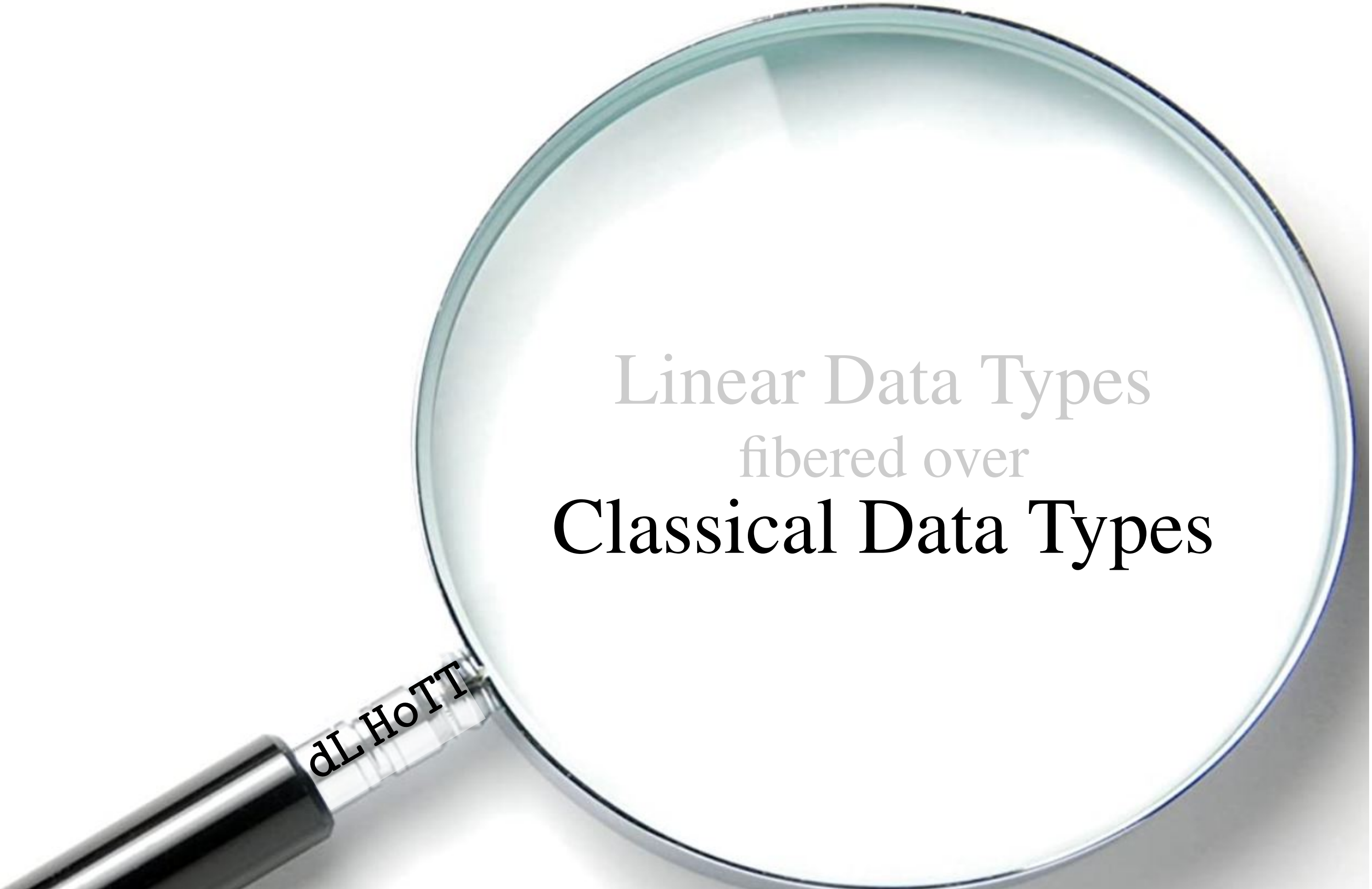
∃ *universal* quantum+classical specification language

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

and

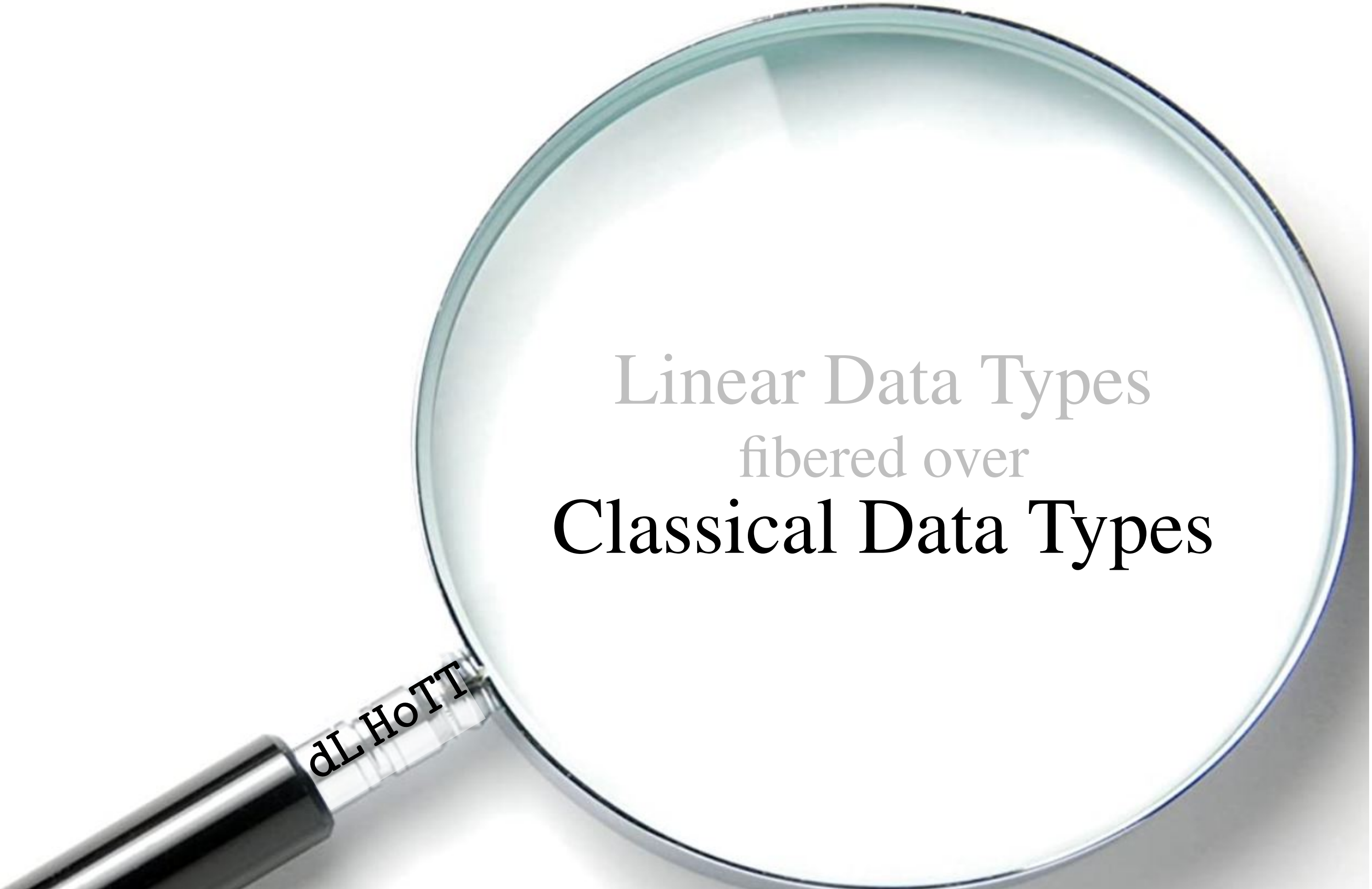
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]

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



# Dependent Linear Homotopy Type Theory (`dLHoTT`)

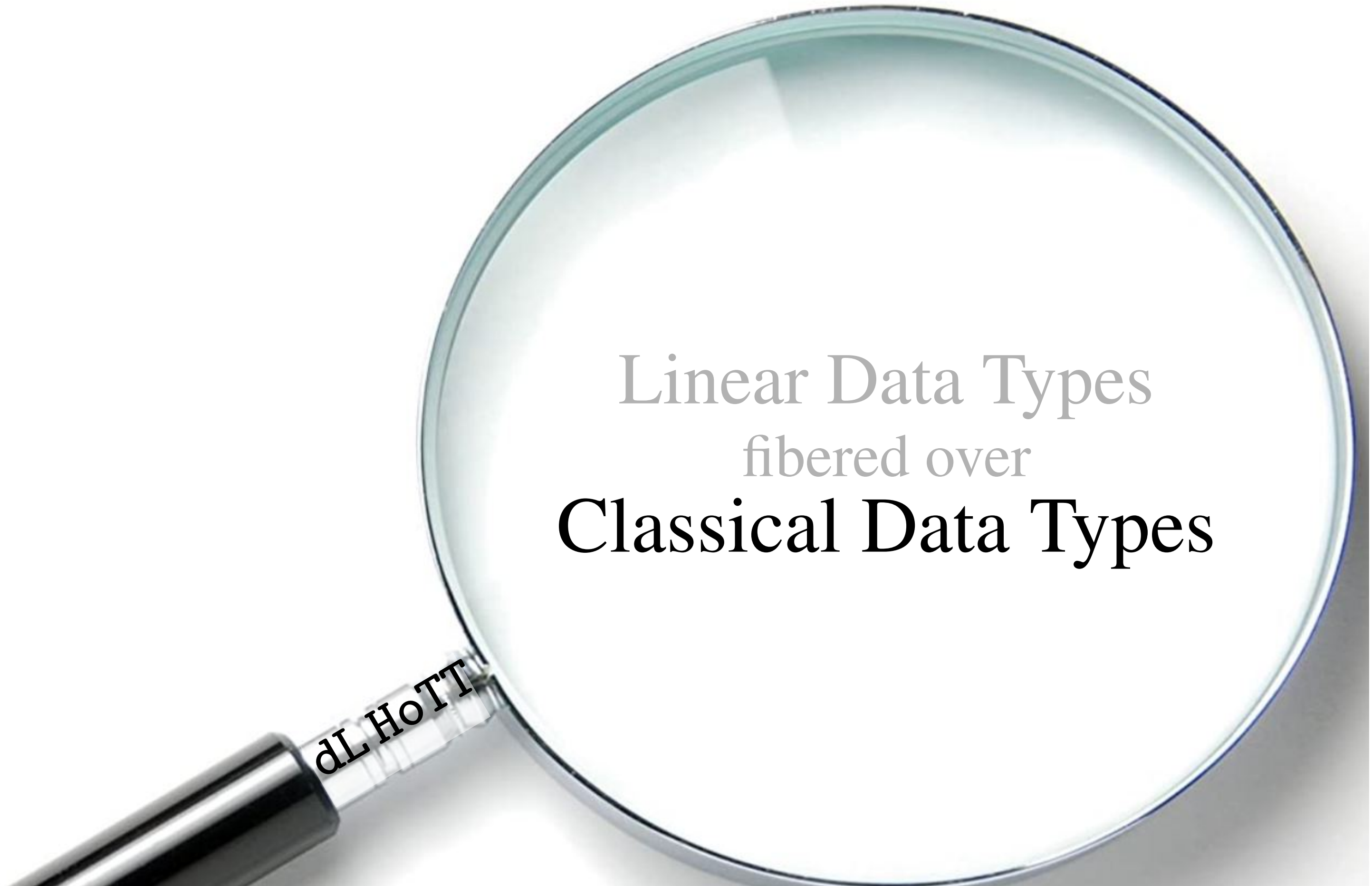
**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à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



Linear Data Types
fibered over
Classical Data Types

# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

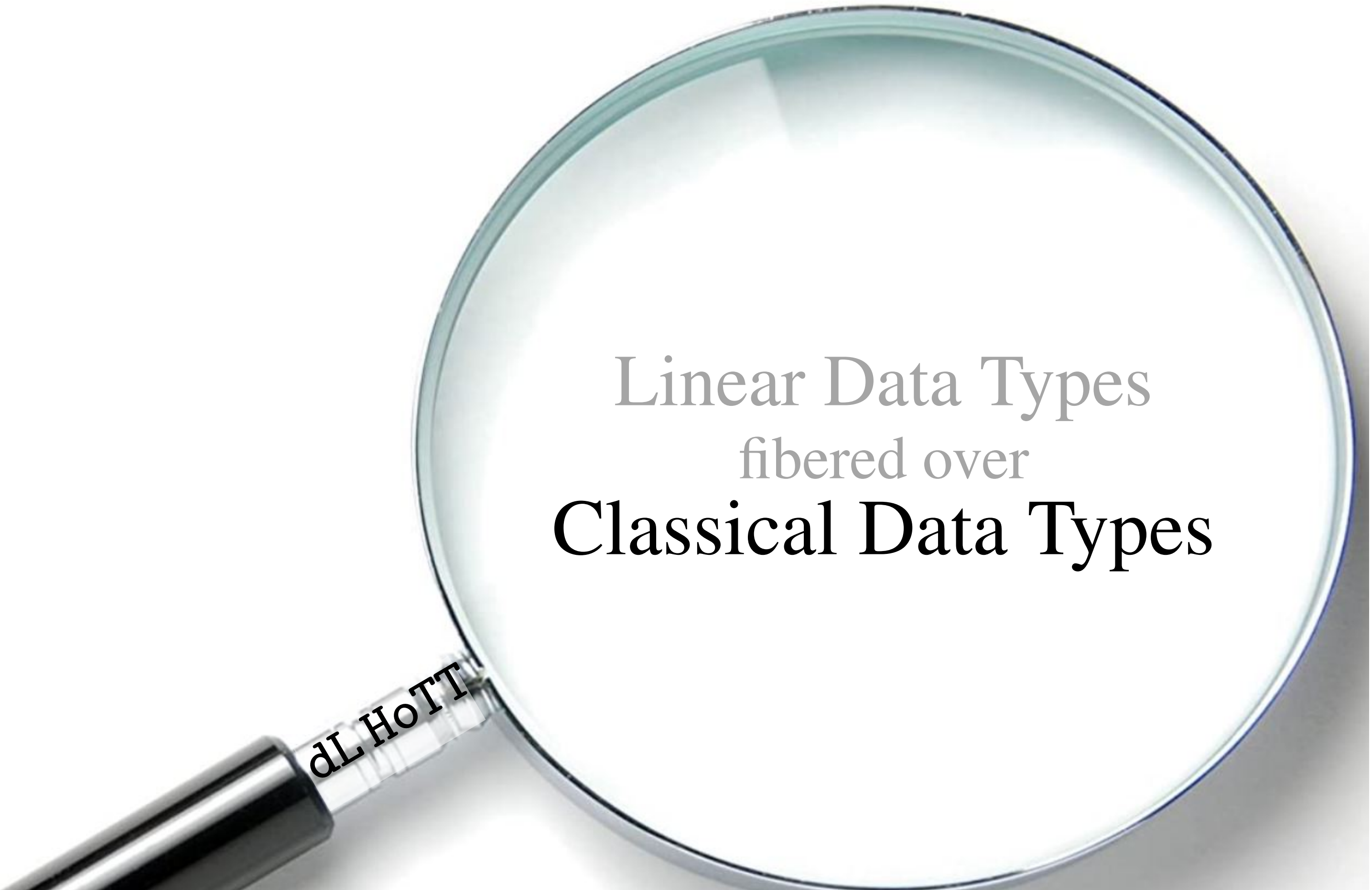
**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme “Motivic Yoga” [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck’s six operations *à la* Wirthmüller — more on all this below)



Linear Data Types
fibered over
Classical Data Types

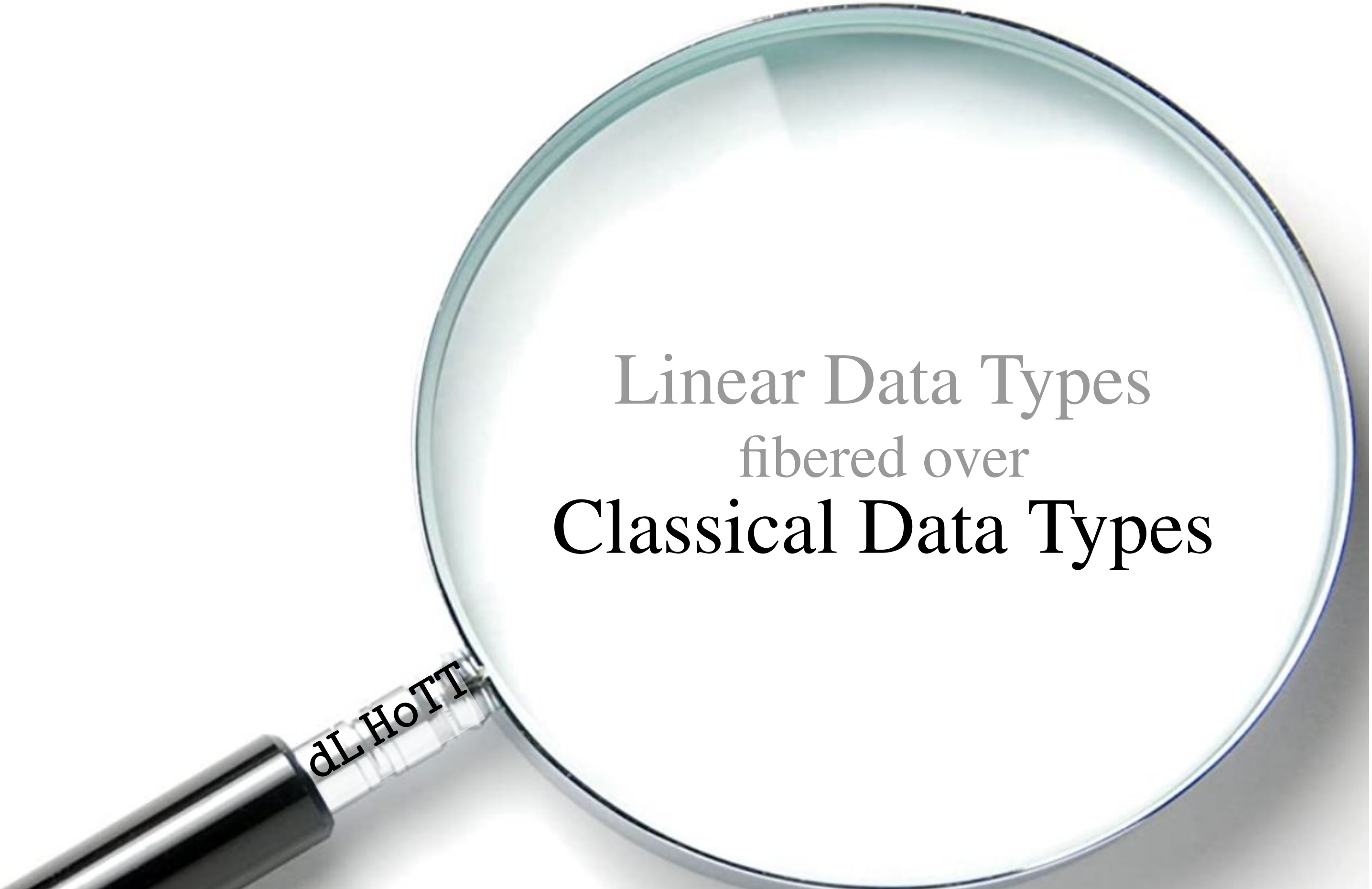
# Dependent Linear Homotopy Type Theory (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ *universal* quantum+classical specification language
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)



## Dependent Linear Homotopy Type Theory (`dLHoTT`)

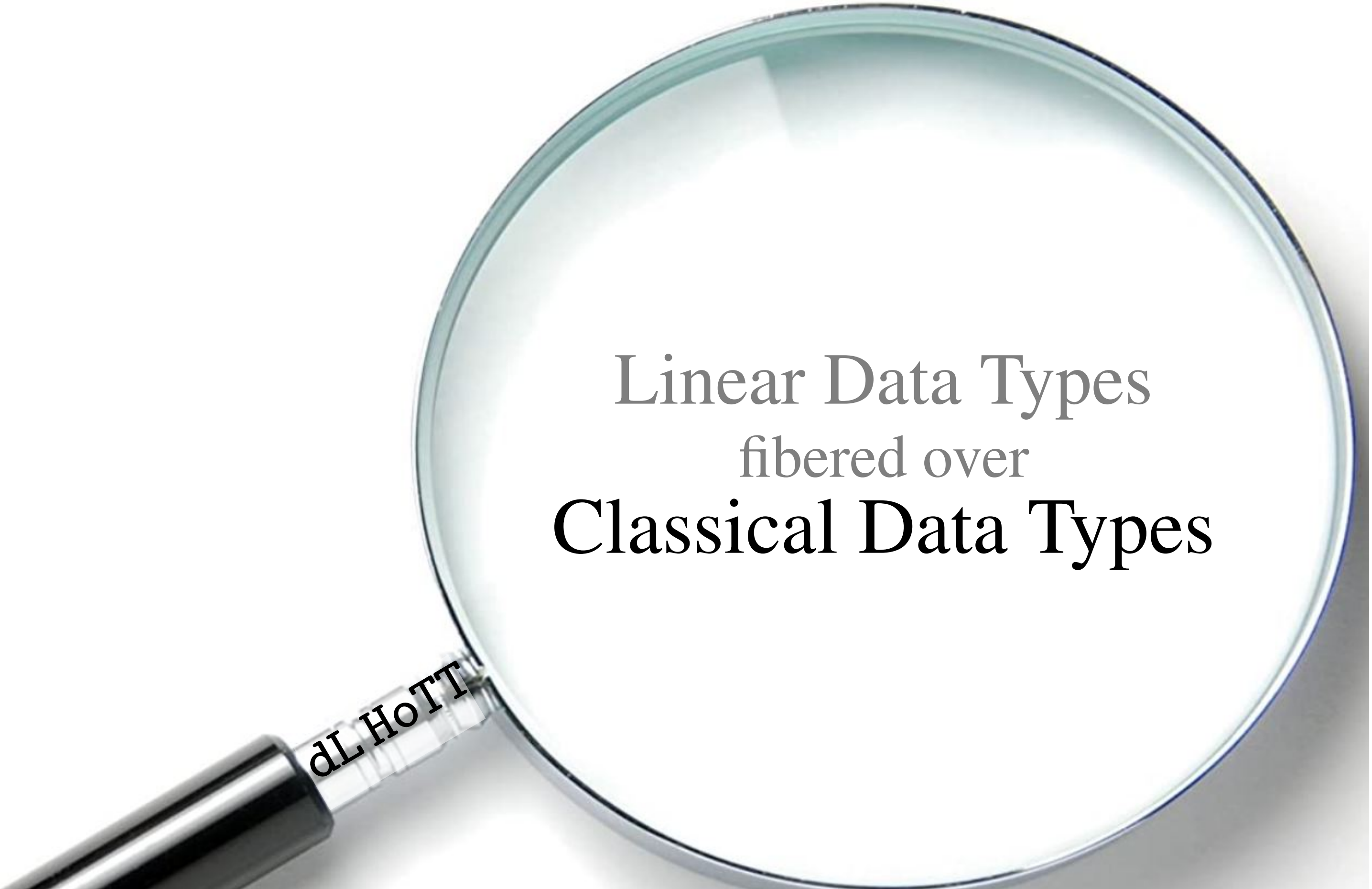
**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations *à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



Linear Data Types
fibered over
Classical Data Types

# Dependent Linear Homotopy Type Theory (`dLHoTT`)

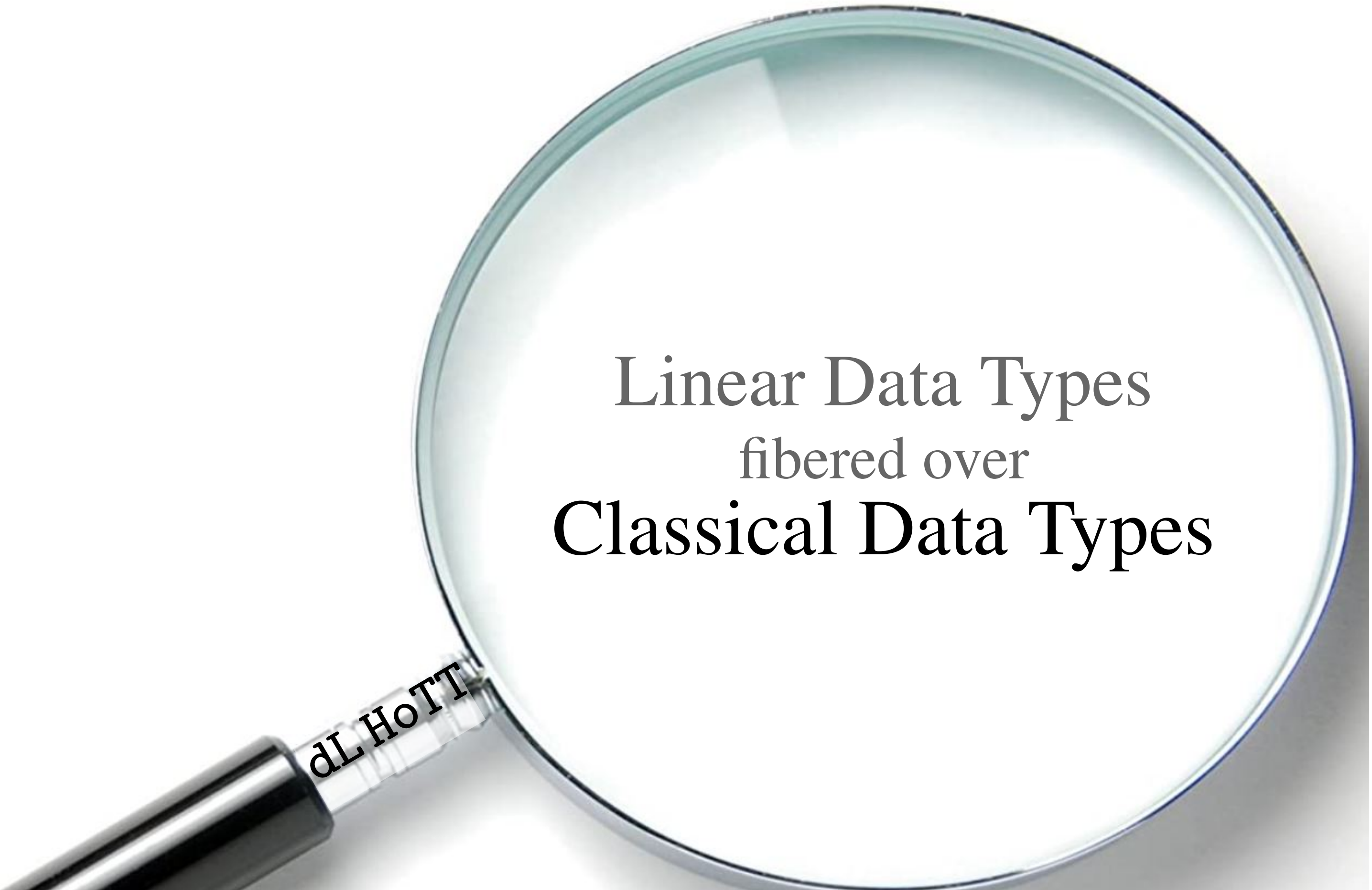
**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ *universal* quantum+classical specification language
conservative over classical *Homotopy Type Theory* (`HoTT`)
and
verifying axiom scheme "Motivic Yoga" [Riley, §2.4, anticipated in S. (2014), §3.2]
(i.e. Grothendieck's six operations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



Linear Data Types
fibered over
Classical Data Types

# Dependent Linear Homotopy Type Theory (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
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)

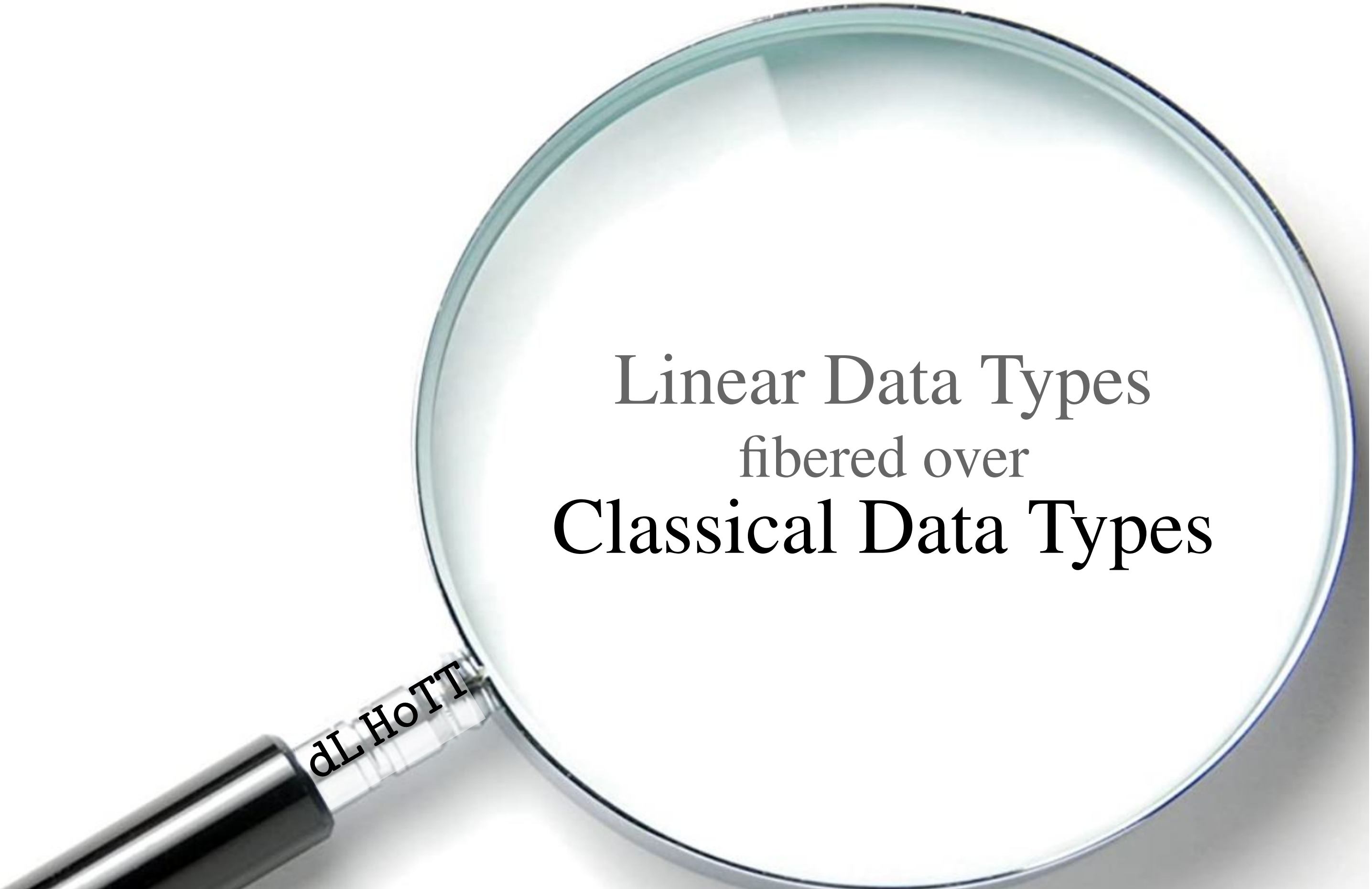


Linear Data Types
fibered over
Classical Data Types

# Dependent Linear Homotopy Type Theory (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ *universal* quantum+classical specification language
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

# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

---

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

$\exists$ *universal* quantum+classical specification language

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

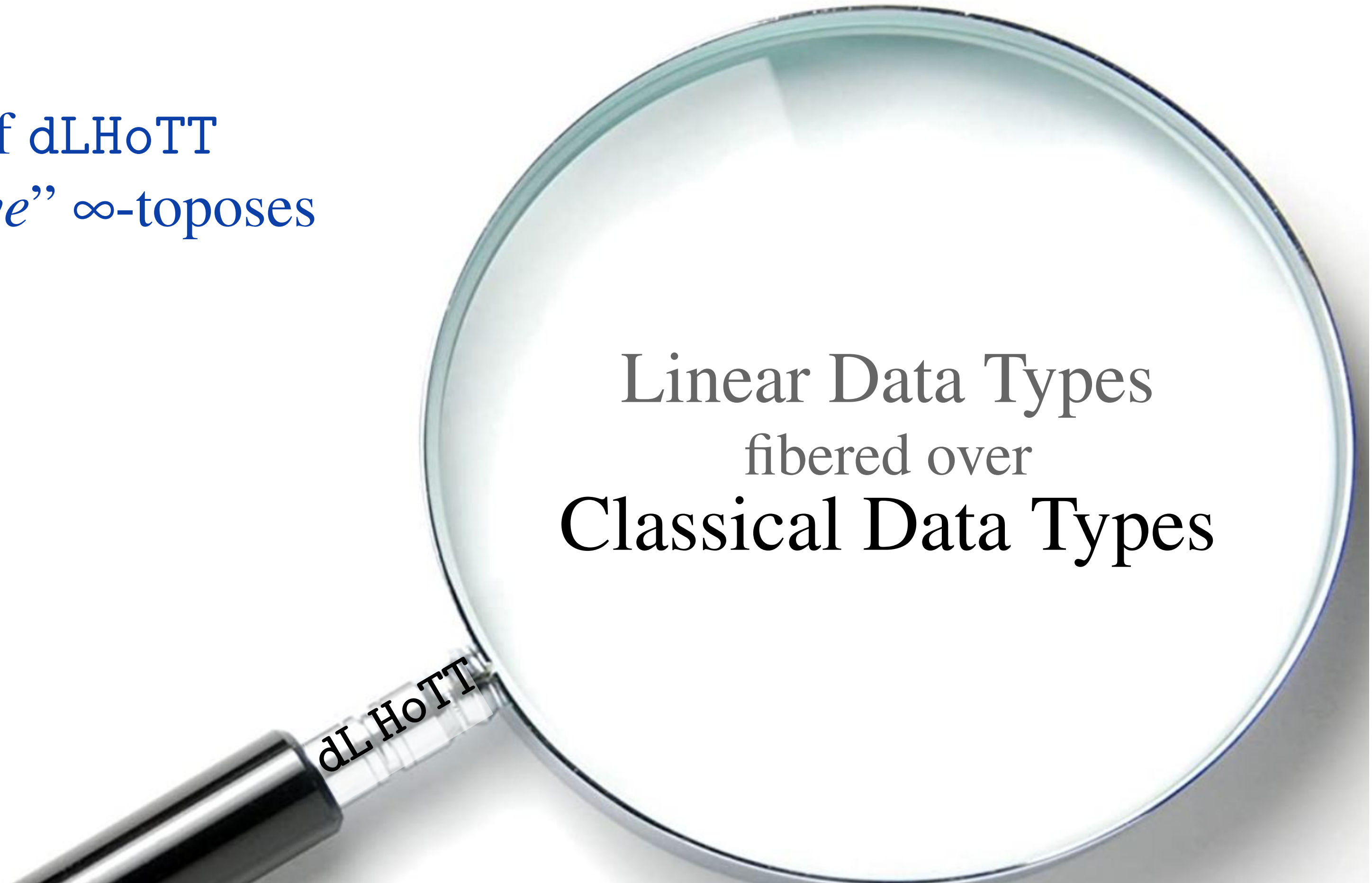
and

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 `dLHoTT`
is in "*infinitesimally cohesive*" $\infty$-toposes



Linear Data Types
fibered over
Classical Data Types

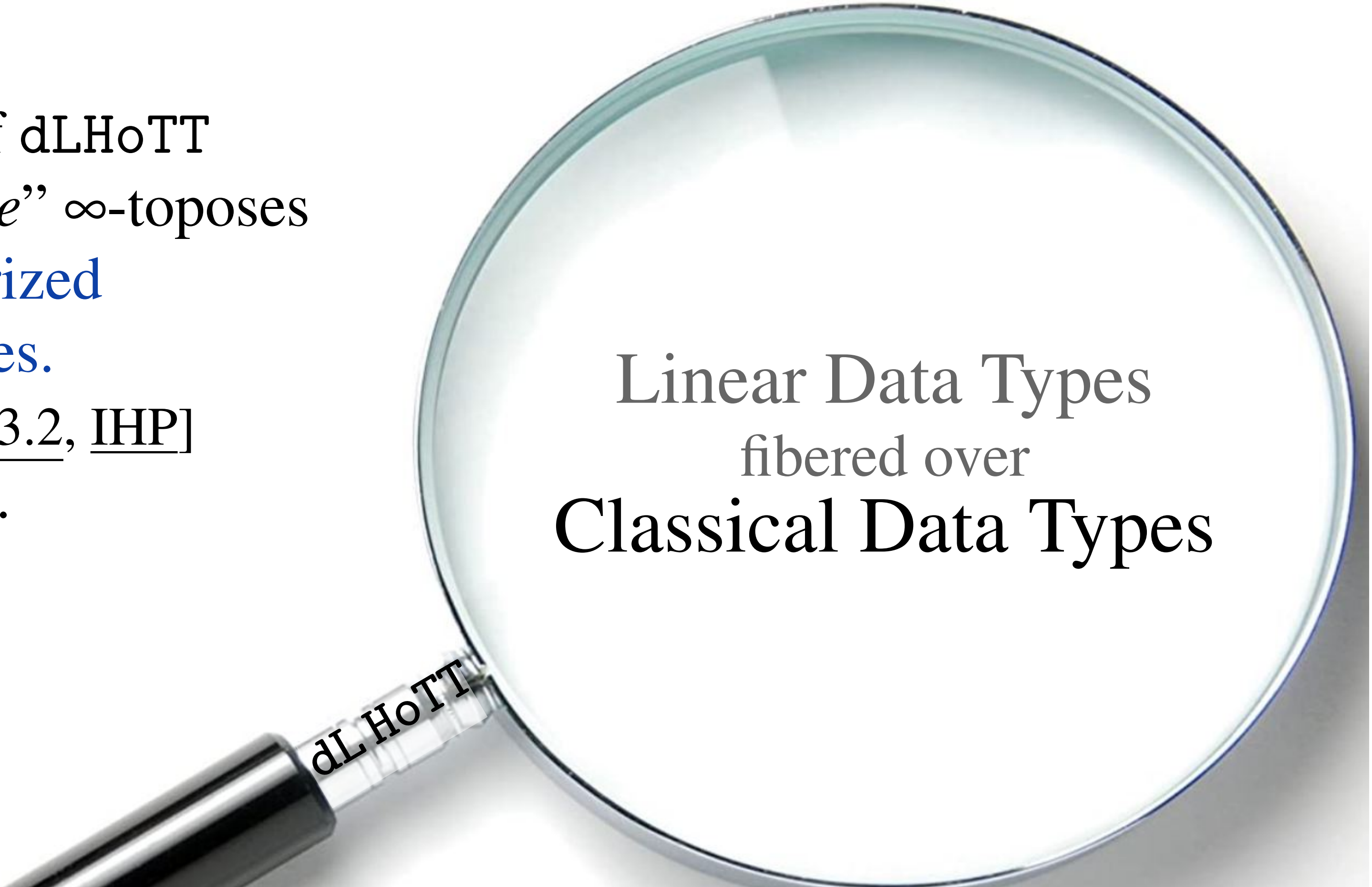
# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

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

$\exists$ *universal* quantum+classical specification language
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 `dLHoTT`
is in "*infinitesimally cohesive*" $\infty$-toposes
of module spectra parameterized
over classical homotopy types.
[S. (2013), §4.1.2] [S. (2014), §3.2, IHP]
[Riley, Finster & Licata (2021)].



# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

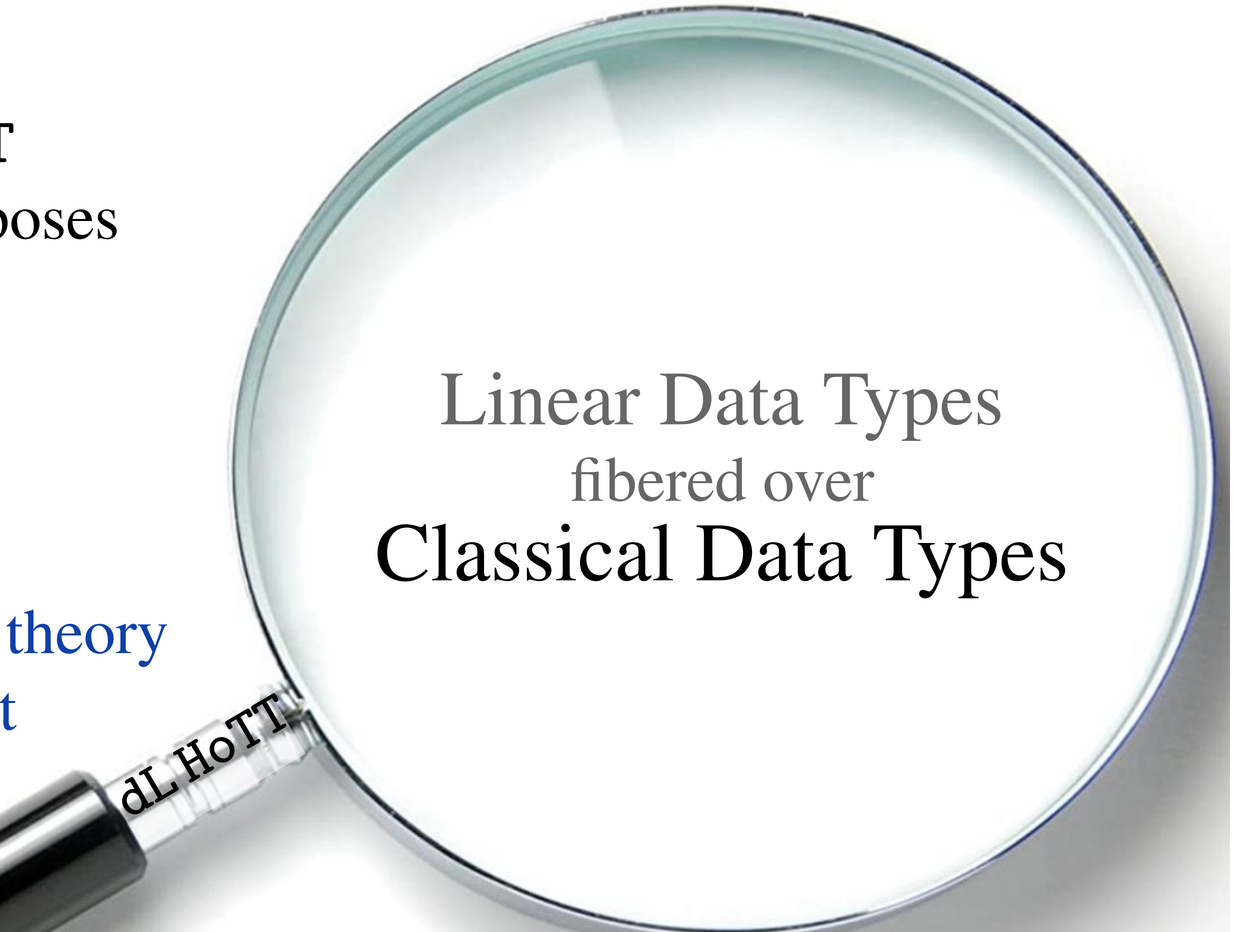
**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:

$\exists$ *universal* quantum+classical specification language
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 `dLHoTT`
is in "*infinitesimally cohesive*" $\infty$-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



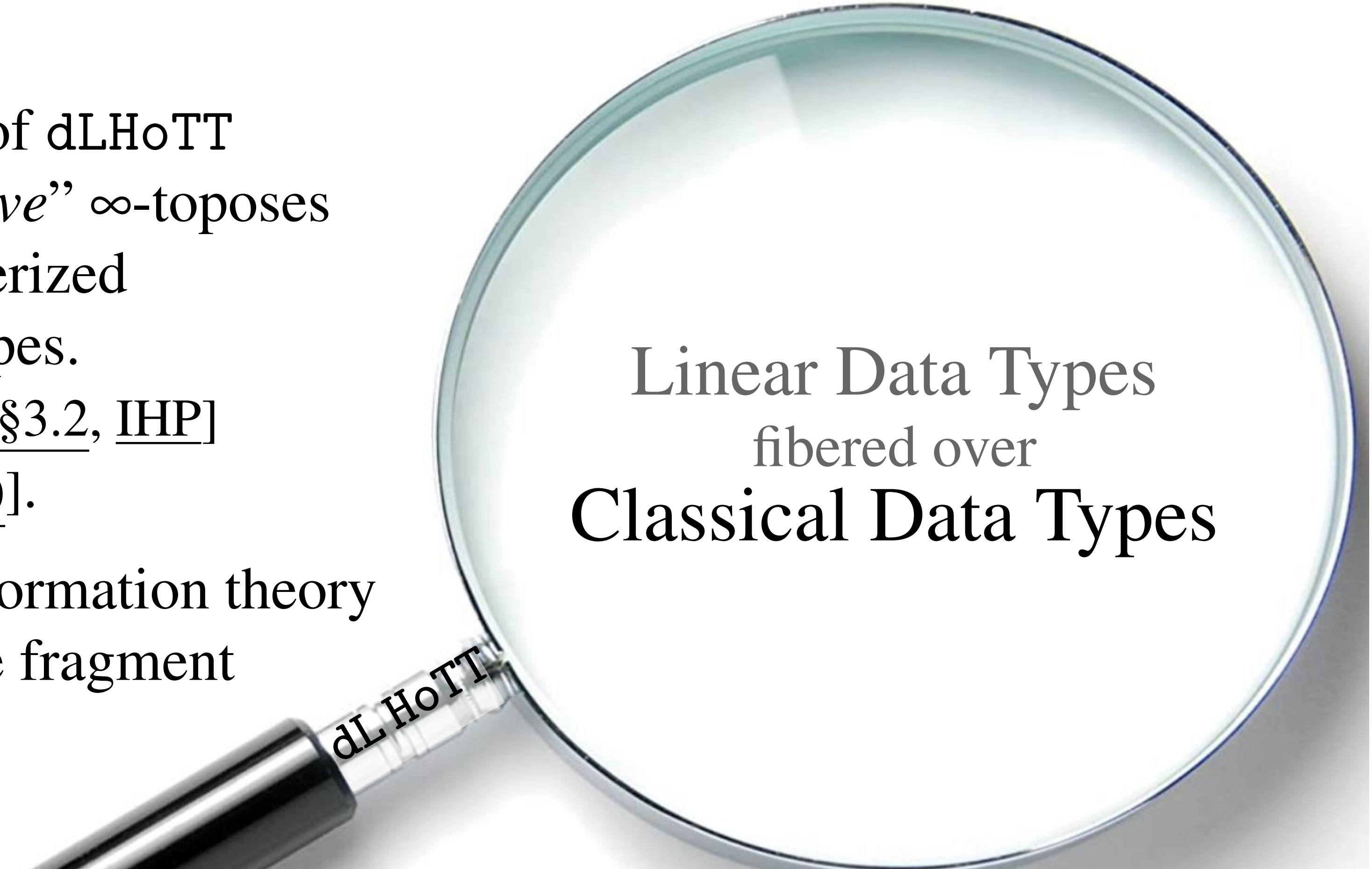
# Dependent Linear Homotopy Type Theory (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ *universal* quantum+classical specification language
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 `dLHoTT` is in "*infinitesimally cohesive*" $\infty$-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

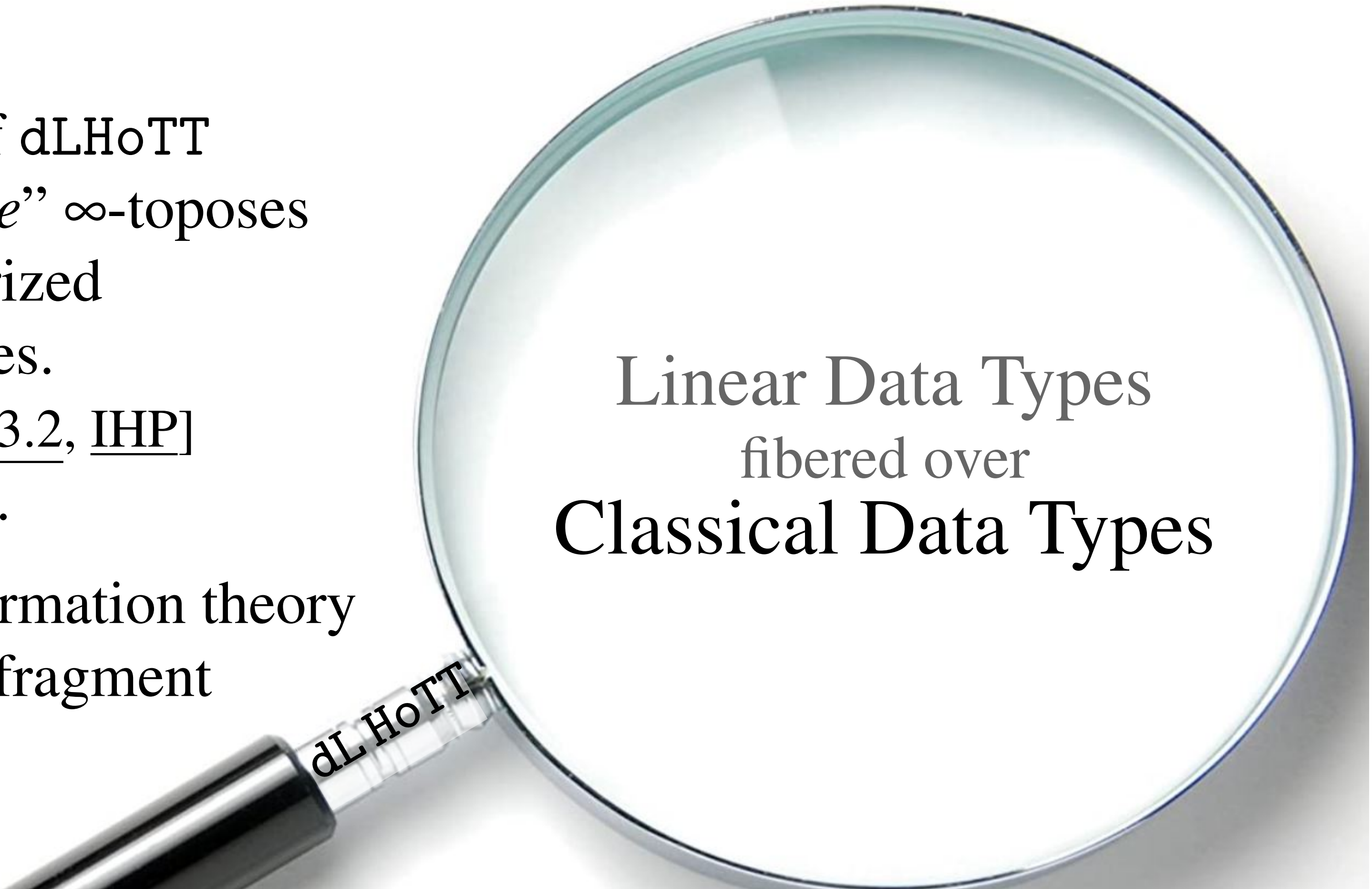
# Dependent Linear Homotopy Type Theory (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
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 `dLHoTT`
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.



# Dependent Linear Homotopy Type Theory (`dLHoTT`)

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

$\exists$ *universal* quantum+classical specification language
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)

## **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
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)]:

# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ *universal* quantum+classical specification language
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

## **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

---

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ *universal* quantum+classical specification language
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**:

## **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ *universal* quantum+classical specification language
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

# Dependent Linear Homotopy Type Theory (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ *universal* quantum+classical specification language
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 |

# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
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 |

# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
$\exists$ *universal* quantum+classical specification language
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 |

# **Dependent Linear Homotopy Type Theory** (`dLHoTT`)

**Theorem** [M. Riley (2022), `doi:10.14418/wes01.3.139`]:
∃ *universal* quantum+classical specification language
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

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

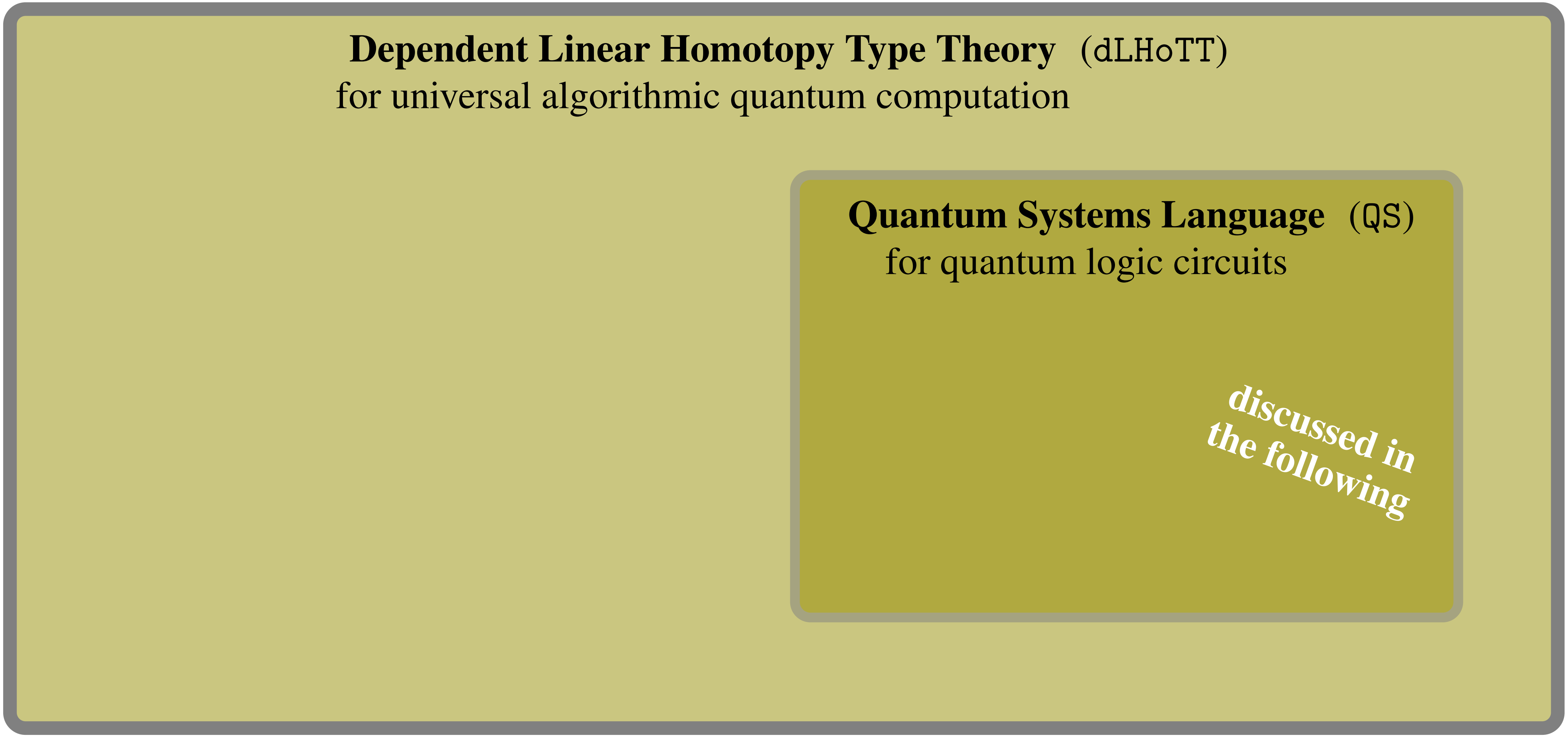
# `QS` – Quantum Systems language @ CQTS

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

**Dependent Linear Homotopy Type Theory** (`dLHoTT`)
for universal algorithmic quantum computation

# `QS` – Quantum Systems language @ CQTS

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



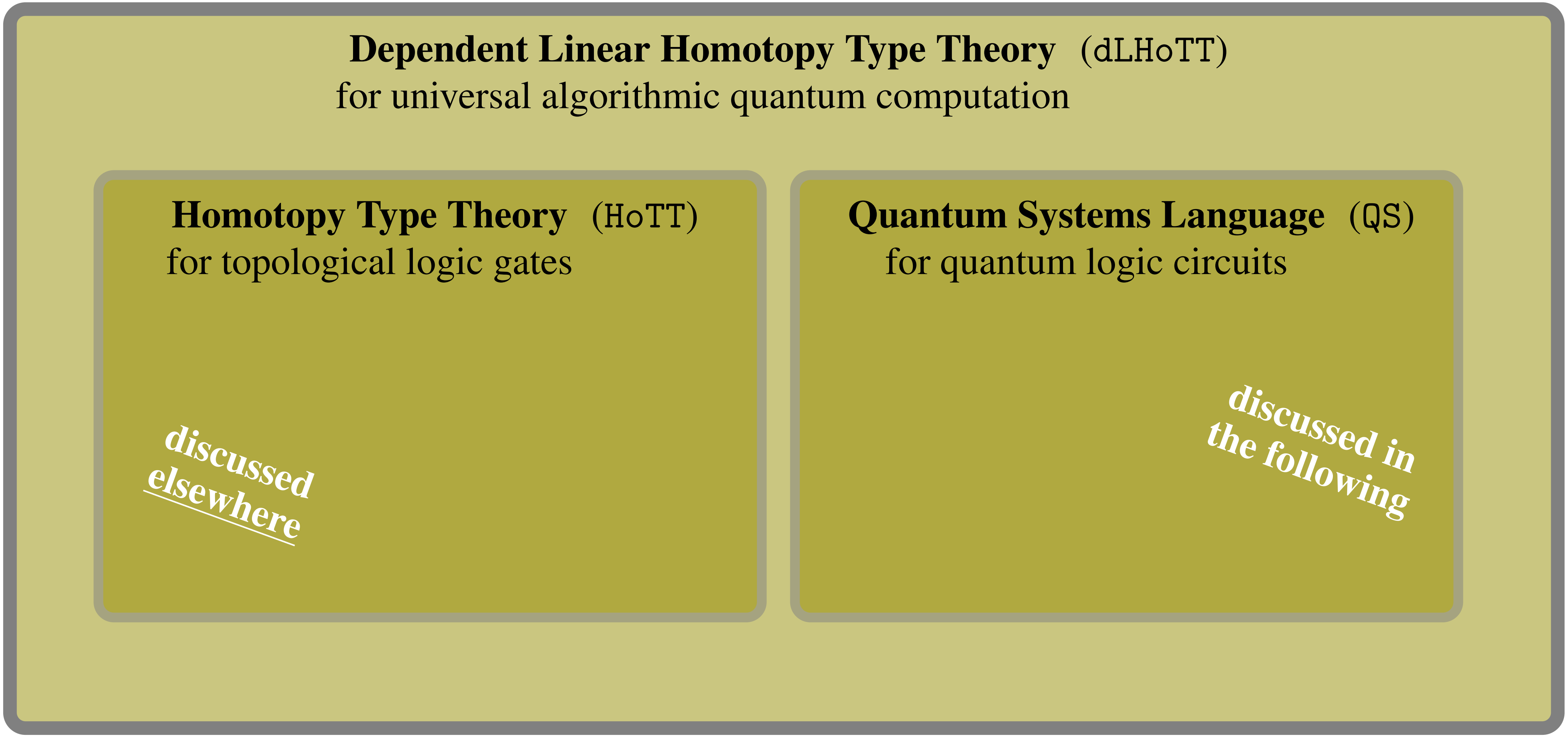
**Dependent Linear Homotopy Type Theory** (`dLHoTT`)
for universal algorithmic quantum computation

**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 `dLHoTT`



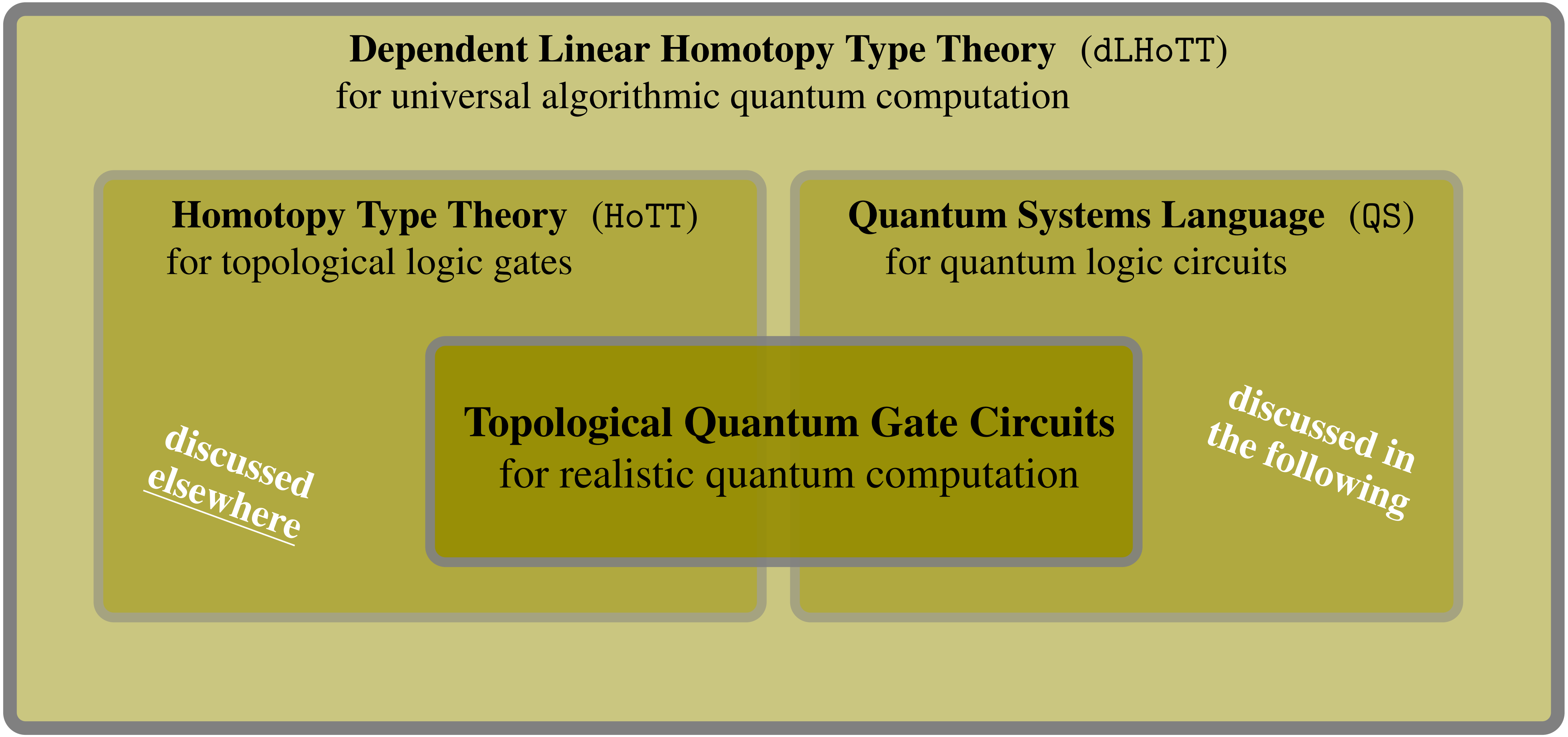
**Dependent Linear Homotopy Type Theory** (`dLHoTT`)
for universal algorithmic quantum computation

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

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

# `QS` – Quantum Systems language @ CQTS

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



**Dependent Linear Homotopy Type Theory** (`dLHoTT`)
for universal algorithmic quantum computation

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

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

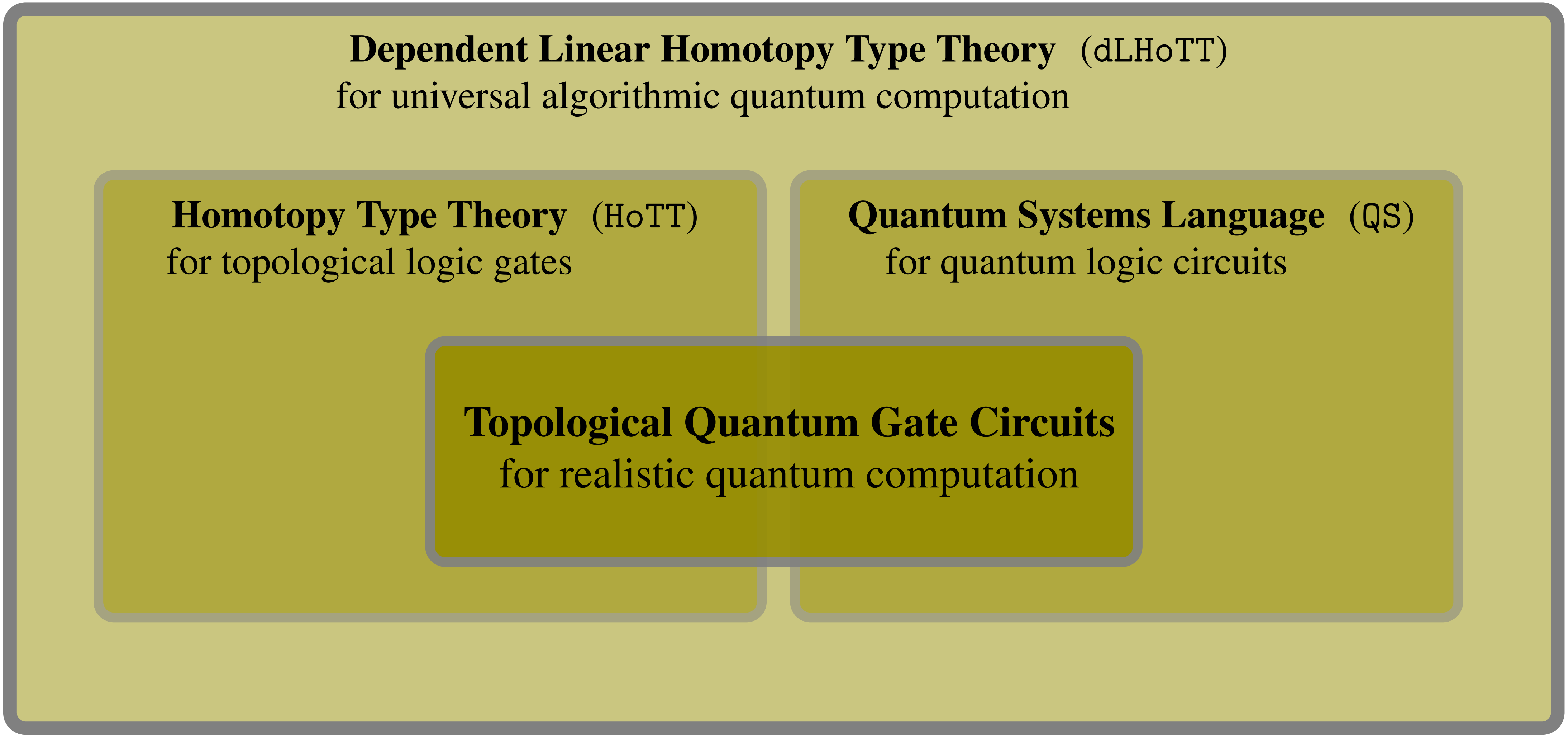
**Topological Quantum Gate Circuits**
for realistic quantum computation

**discussed elsewhere**

**discussed in the following**

# QS – Quantum Systems language @ CQTS

⇝ full-blown Quantum Systems language emerges embedded in dLHoTT



**Dependent Linear Homotopy Type Theory** (dLHoTT)
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 dLHoTT | 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

| | | | |
|---|---|---|---|
| **Characteristic Property** | | | |
| **Symbol** | | | |
| **Formula**<br>(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** | | | |
| **Formula**<br>(for $B$ : FinType) | | | |
| **AlgTop Jargon** | | | |
| | | | |
| **Linear Logic** | | | |
| **Physics Meaning** | | | |

# Linear/Quantum Data Types

<table>
<tr><td rowspan="1"><b>Characteristic Property</b></td><td><b>1.</b> their cartesian product blends into the co-product:</td><td></td><td></td></tr>
<tr><td><b>Symbol</b></td><td>$\oplus$ direct sum</td><td></td><td></td></tr>
<tr><td><b>Formula</b><br>(for $B$ : FinType)</td><td></td><td></td><td></td></tr>
<tr><td rowspan="2"><b>AlgTop Jargon</b></td><td rowspan="2"></td><td></td><td></td></tr>
<tr><td colspan="2"></td></tr>
<tr><td><b>Linear Logic</b></td><td></td><td></td><td></td></tr>
<tr><td><b>Physics Meaning</b></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 | |
|---|---|---|---|
| **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

| **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 | |
| **Formula** (for $B$ : FinType) | | | |
| **AlgTop Jargon** | | | |
| | | | |
| **Linear Logic** | | | |
| **Physics Meaning** | | | |

# Linear/Quantum Data Types

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

# Linear/Quantum Data Types

<table>
<tr><td><b>Characteristic Property</b></td><td><b>1.</b> their cartesian product blends into the co-product:</td><td><b>2.</b> a tensor product appears & distributes over direct sum</td><td><b>3.</b> a linear function type appears adjoint to tensor</td></tr>
<tr><td><b>Symbol</b></td><td>$\bigoplus$ direct sum</td><td>$\bigotimes$ tensor product</td><td>$\multimap$ linear function type</td></tr>
<tr><td><b>Formula</b><br>(for $B$ : FinType)</td><td>$\overset{\textbf{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\textbf{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\textbf{co-product}}{\coprod_B \mathcal{H}_b}$</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>$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$</td></tr>
<tr><td rowspan="2"><b>AlgTop Jargon</b></td><td rowspan="2">biproduct,<br>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><b>Linear Logic</b></td><td></td><td></td><td></td></tr>
<tr><td><b>Physics Meaning</b></td><td></td><td></td><td></td></tr>
</table>

# Linear/Quantum Data Types

<table>
<tr><td><b>Characteristic Property</b></td><td><b>1.</b> their cartesian product blends into the co-product:</td><td><b>2.</b> a tensor product appears & distributes over direct sum</td><td><b>3.</b> a linear function type appears adjoint to tensor</td></tr>
<tr><td><b>Symbol</b></td><td>$\oplus$ direct sum</td><td>$\otimes$ tensor product</td><td>$\multimap$ linear function type</td></tr>
<tr><td><b>Formula</b><br>(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>$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$</td></tr>
<tr><td rowspan="2"><b>AlgTop Jargon</b></td><td rowspan="2">biproduct,<br>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><b>Linear Logic</b></td><td>additive disjunction</td><td>multiplicative conjunction</td><td>linear implication</td></tr>
<tr><td><b>Physics Meaning</b></td><td></td><td></td><td></td></tr>
</table>

# Linear/Quantum Data Types

<table>
<tr><td><b>Characteristic Property</b></td><td><b>1.</b> their cartesian product blends into the co-product:</td><td><b>2.</b> a tensor product appears & distributes over direct sum</td><td><b>3.</b> a linear function type appears adjoint to tensor</td></tr>
<tr><td><b>Symbol</b></td><td>$\oplus$ direct sum</td><td>$\otimes$ tensor product</td><td>$\multimap$ linear function type</td></tr>
<tr><td><b>Formula</b><br>(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>$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$</td></tr>
<tr><td rowspan="2"><b>AlgTop Jargon</b></td><td rowspan="2">biproduct,<br>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><b>Linear Logic</b></td><td>additive disjunction</td><td>multiplicative conjunction</td><td>linear implication</td></tr>
<tr><td><b>Physics Meaning</b></td><td>superselection sectors<br>/ quantum parallelism</td><td>compound quantum systems<br>/ quantum entanglement</td><td>QRAM systems</td></tr>
</table>

# Linear/Quantum Data Types

<table>
<tr><td><b>Characteristic Property</b></td><td><b>1.</b> their cartesian product blends into the co-product:</td><td><b>2.</b> a tensor product appears & distributes over direct sum</td><td><b>3.</b> a linear function type appears adjoint to tensor</td></tr>
<tr><td><b>Symbol</b></td><td>$\oplus$ direct sum</td><td>$\otimes$ tensor product</td><td>$\multimap$ linear function type</td></tr>
<tr><td><b>Formula</b><br>(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>$(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$</td></tr>
<tr><td rowspan="2"><b>AlgTop Jargon</b></td><td rowspan="2">biproduct,<br>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><b>Linear Logic</b></td><td>additive disjunction</td><td>multiplicative conjunction</td><td>linear implication</td></tr>
<tr><td><b>Physics Meaning</b></td><td>superselection sectors<br>/ quantum parallelism</td><td>compound quantum systems<br>/ 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**<br>(for $B$ : FinType) | $\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$ | $\mathcal{V} \otimes \Big( \bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$ | $(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$ |

**Dependent linear Type Formers**

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

**reference context**

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

**classical type system dependent on context**

$\mathrm{BType}_B$

$\mathrm{BType}$

**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**<br>(for $B$ : FinType) | **cart. product** **co-product**<br>$\prod_B \mathcal{H}_b \simeq \bigoplus_B \mathcal{H}_b \simeq \coprod_B \mathcal{H}_b$<br>**direct sum** | $\mathcal{V} \otimes \Big( \bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$ | $(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$ |

**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**

$$\begin{tikzcd} \mathrm{BType}_B & & \mathrm{BType} \arrow[ll, "*_B \times" description] \end{tikzcd}$$

**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) | $\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$ | $\mathcal{V} \otimes \Big( \bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$ | $(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$ |

**Dependent linear Type Formers**

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

**reference context**

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

co-product

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

**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**<br>(for $B$ : FinType) | $\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$ | $\mathcal{V} \otimes \Big( \bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$ | $(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$ |

**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**

**classical type system**

$$\begin{tikzcd}
\mathrm{BType}_B \arrow[rr, shift left=6, "\overset{\text{co-product}}{\coprod_{b:B}}"] \arrow[rr, shift right=6, "\underset{\text{product}}{\prod_{b:B}}"'] & & \mathrm{BType} \arrow[ll, "\overset{\perp}{\underset{\perp}{\ast_B \times}}" description]
\end{tikzcd}$$

# 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 | $\bigotimes$ tensor product | $\multimap$ linear function type |
| **Formula**<br>(for $B$ : FinType) | $\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$ | $\mathcal{V} \otimes \Big( \bigoplus\limits_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus\limits_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$ | $(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$ |

**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**

$$\begin{tikzcd}[column sep=huge] \mathrm{BType}_B \arrow[r, shift left=3ex, "\overset{\text{co-product}}{\coprod_{b:B}}"] \arrow[r, phantom, shift left=1.5ex, "\perp"] \arrow[r, phantom, shift right=1.5ex, "\perp"] \arrow[r, shift right=3ex, "\underset{\text{product}}{\prod_{b:B}}"'] & \mathrm{BType} \arrow[l, "*_B \times" description] \end{tikzcd}$$

**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** | $\oplus$ direct sum | $\otimes$ tensor product | $\multimap$ linear function type |
| **Formula**<br>(for $B$ : FinType) | $\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$ | $\mathcal{V} \otimes \Big( \bigoplus\limits_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus\limits_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$ | $(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$ |

**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**

co-product

$$\begin{tikzcd}[column sep=huge]
\mathrm{BType}_B \arrow[r, shift left=8, "\coprod_{b:B}"] \arrow[r, phantom, shift left=4, "\perp"] \arrow[r, leftarrow, "*_B \times" description] \arrow[r, phantom, shift right=4, "\perp"] \arrow[r, shift right=8, "\prod_{b:B}"'] & \mathrm{BType}
\end{tikzcd}$$

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**<br>(for $B$ : FinType) | $\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$ | $\mathcal{V} \otimes \Big( \bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$ | $(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$ |

**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**

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

**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**<br>(for $B$ : FinType) | $\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$ | $\mathcal{V} \otimes \Big( \bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$ | $(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$ |

**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**

$$\begin{tikzcd}[column sep=huge]
\text{BType}_B \arrow[r, shift left=6ex, "\coprod_{b:B}" description, "\text{co-product}"] \arrow[r, phantom, shift left=3ex, "\perp"] \arrow[r, phantom, shift right=3ex, "\perp"] \arrow[r, shift right=6ex, "\prod_{b:B}" description, "\text{product}"'] & \text{BType} \arrow[l, "*_B \times" description]
\end{tikzcd}$$

**classical type system**

classical base change / classical quantification

**linear type system in classical context**

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

**linear 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 | $\otimes$ tensor product | $\multimap$ linear function type |
| **Formula** (for $B$ : FinType) | $\overset{\textbf{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\textbf{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\textbf{co-product}}{\coprod_B \mathcal{H}_b}$ | $\mathcal{V} \otimes \Big( \bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$ | $(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$ |

**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**

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

**classical type system**

**linear type system in classical context**

$$\begin{tikzcd}[column sep=huge]
\Big(\mathrm{LType}_B, \overset{\text{tensor}}{\otimes_B}\Big) \arrow[r, shift left=5, "\overset{\text{direct sum}}{\bigoplus_{b:B}}"] \arrow[r, phantom, shift left=2.5, "\perp"] \arrow[r, phantom, shift right=2.5, "\perp"] \arrow[r, shift right=5, "\bigoplus_{b:B}"'] & \Big(\mathrm{LType}, \overset{\text{tensor}}{\otimes}\Big) \arrow[l, "\mathbb{1}_B \otimes" description]
\end{tikzcd}$$

**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** | $\bigoplus$ direct sum | $\bigotimes$ tensor product | $\multimap$ linear function type |
| **Formula**<br>(for $B$ : FinType) | $\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$ | $\mathcal{V} \otimes \Big( \bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} \big(\mathcal{V} \otimes \mathcal{H}_b\big)$ | $(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$ |

**Dependent linear Type Formers**

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

**reference context**

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

**classical type system dependent on context**

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

**classical type system**

classical base change / classical quantification

**linear type system in classical context**

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

**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 | $\bigotimes$ tensor product | $\multimap$ linear function type |
| **Formula** (for $B$ : FinType) | $\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$ | $\mathcal{V} \otimes \Big( \bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} (\mathcal{V} \otimes \mathcal{H}_b)$ | $(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$ |

**Dependent linear Type Formers**

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

**reference context**

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

**classical type system dependent on context**

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

**classical type system**

classical base change / classical quantification

**linear type system in classical context**

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

**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** | $\bigoplus$ direct sum | $\bigotimes$ tensor product | $\multimap$ linear function type |
| **Formula**<br>(for $B$ : FinType) | $\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$ | $\mathcal{V} \otimes \Big( \bigoplus_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus_{b:B} \big( \mathcal{V} \otimes \mathcal{H}_b \big)$ | $(\mathcal{V} \otimes \mathcal{H}) \multimap \mathcal{K}$<br>$\simeq \; \mathcal{V} \multimap (\mathcal{H} \multimap \mathcal{K})$ |

all available in dLHoTT

**Dependent linear Type Formers**

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

**reference context**

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

**classical type system dependent on context**

co-product

$$\begin{tikzcd}
\mathrm{BType}_B \arrow[r, shift left=6, "\coprod_{b:B}"] \arrow[r, phantom, shift left=3, "\perp"] \arrow[r, phantom, shift right=3, "\perp"] \arrow[r, shift right=6, "\prod_{b:B}"'] & \mathrm{BType} \arrow[l, "*_B \times"']
\end{tikzcd}$$

product

**classical type system**

classical base change / classical quantification

**linear type system in classical context**

direct sum

$$\begin{tikzcd}
\Big(\mathrm{LType}_B, \overset{\text{tensor}}{\otimes_B}\Big) \arrow[r, shift left=6, "\bigoplus_{b:B}"] \arrow[r, phantom, shift left=3, "\perp"] \arrow[r, phantom, shift right=3, "\perp"] \arrow[r, shift right=6, "\bigoplus_{b:B}"'] & \Big(\mathrm{LType}, \overset{\text{tensor}}{\otimes}\Big) \arrow[l, "\mathbb{1}_B \otimes"']
\end{tikzcd}$$

**linear type system**

quantum base change / Motivic Yoga

# Quantum Effects

# Recall: **Monadic computational effects.**

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

**effectful program**

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

**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**

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

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

**second program**

$$\begin{tikzcd} D_2 \arrow[r, "\mathrm{prog}_{23}"] & \mathscr{E}(D_3) \end{tikzcd}$$

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

$\mapsto$ **bind previous effects**
**into second program**

$$\begin{tikzcd} D_1 \arrow[r, "\mathrm{prog}_{12}"] & \mathscr{E}(D_2) & \mathscr{E}(D_2) \arrow[r, "\mathrm{bind}^{\mathscr{E}}\mathrm{prog}_{23}"] & \mathscr{E}(D_3) \end{tikzcd}$$

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

$\mapsto$ **compose**

**$\mathscr{E}$-composite program**

$$\begin{tikzcd} D_1 \arrow[r, "\mathrm{bind}^{\mathscr{E}}\mathrm{prog}_{23} \circ \mathrm{prog}_{12}"] & \mathscr{E}(D_3) \end{tikzcd}$$

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

# Recall: **Monadic effect handlers.**

$$\begin{tikzcd} D_1 \arrow[r, "\mathrm{prog}_{12}", "\text{in-effectful program}"'] & D_2 \end{tikzcd}$$ **data type to absorb $\mathscr{E}$-effects**

# Recall: **Monadic effect handlers.**

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

# Recall: **Monadic effect handlers.**

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

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

**consistency conditions**

# Recall: **Monadic effect handlers.**

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

**consistency conditions**

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

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

**handle effects...** **consecutively**

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

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

**“$\mathscr{E}$-modal data type”**

**Monadicity**:

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

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

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

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

**in-effectful program**

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

**“$\mathcal{E}$-modal data type”**

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

**Monadicity**:

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

**monad**

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

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

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

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

**“$\mathscr{E}$-modal data type”**

## **Monadicity**:

$$\begin{tikzcd}
& \mathrm{Type} \arrow[loop right, "\mathscr{E}"] \arrow[dl, shift right, "F^{\mathscr{E}}"'] \\
\mathrm{Type}^{\mathscr{E}} \arrow[ur, shift right, "U^{\mathscr{E}}"'] \arrow[ur, phantom, "\perp"] &
\end{tikzcd}$$

**induced monad**

**monadic adjunction**

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

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

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

**“$\mathcal{E}$-modal data type”**

**Monadicity**:

$$\begin{tikzcd}
\mathrm{Type}' \arrow[rr, shift right=2, "U"'] \arrow[d, "K^{UF}"] & & \mathrm{Type} \arrow[ll, shift right=2, "F"'] \arrow[ll, phantom, "\perp"] \arrow[dll, shift right=2, "F^{\mathcal{E}}"'] \arrow[loop right, "\mathcal{E}"] \\
\mathrm{Type}^{\mathcal{E}} \arrow[urr, shift right=2, "U^{\mathcal{E}}"'] \arrow[urr, phantom, "\perp"] & &
\end{tikzcd}$$

**any adjunction**
on type system

**induced monad**

**terminal comparison functor**

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

**monadic adjunction**

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

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

## **Monadicity**:

$$\begin{tikzcd}
\text{free } \mathscr{E}\text{-modales in Type (“Kleisli category”)} & \mathrm{Type}_{\mathscr{E}} \arrow[d, hook, "K_{UF}", "\text{initial comparison functor}"'] \arrow[drr, bend left=10, "U_{\mathscr{E}}"] & & \\
\text{any adjunction on type system} & \mathrm{Type}' \arrow[d, "K^{UF}", "\text{terminal comparison functor}"'] \arrow[rr, bend right=10, "U"'] \arrow[rr, phantom, "\perp"] & & \mathrm{Type} \arrow[ull, two heads, bend right=10, "F_{\mathscr{E}}"'] \arrow[ll, bend right=10, "F"'] \arrow[dll, bend left=10, "F^{\mathscr{E}}"] \arrow[loop right, "\mathscr{E}"] & \text{induced monad} \\
\mathscr{E}\text{-modales in Type (“EM-category”)} & \mathrm{Type}^{\mathscr{E}} \arrow[urr, bend right=10, "U^{\mathscr{E}}"', "\text{monadic adjunction}"] & &
\end{tikzcd}$$

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

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

**in-effectful program**

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

**“$\mathscr{E}$-modal data type”**

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

**Monadicity**:

$$\begin{tikzcd}
\mathrm{Type}_{\mathscr{E}} \arrow[d, hook, "K_{UF}"] \arrow[dr, shift right, "U_{\mathscr{E}}"'] \arrow[dr, phantom, "\perp"] & \\
\mathrm{Type}' \arrow[r, shift right, "U"'] \arrow[r, phantom, "\perp"] \arrow[d, "K^{UF}"] & \mathrm{Type} \arrow[l, shift right, "F"'] \arrow[ul, two heads, shift right, "F_{\mathscr{E}}"'] \arrow[dl, shift right, "F^{\mathscr{E}}"'] \arrow[dl, phantom, "\perp"] \arrow[loop right, "\mathscr{E}"] \\
\mathrm{Type}^{\mathscr{E}} \arrow[ur, shift right, "U^{\mathscr{E}}"'] &
\end{tikzcd}$$

**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”)

**induced monad**

**monadic adjunction**

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

Given $B : \mathrm{BType}$ 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**

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

**actual data** $\perp$ (between $\Diamond_B$ and $\Box_B$); $\coprod_B \perp \times B \perp \prod_B$; $\perp$ **potential data** (between $\star_B$ and $\bigcirc_B$)

**necessarily** **indefinitely**

Given $B : \mathrm{BType}$ 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** $\perp$ $\perp$ **potential data**

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

$$\coprod_B \dashv \times B \dashv \prod_B$$

**necessarily** **indefinitely**

---

**necessarily $P_\bullet$**

$$\Box_B P_\bullet$$

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

Given $B$ : BType 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**

$$\begin{tikzcd} \mathrm{Type}_B \arrow[loop above, "\lozenge_B"] \arrow[loop below, "\Box_B"] \arrow[r, shift left=4, "\coprod_B"] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] \arrow[r, shift right=4, "\prod_B"'] & \mathrm{Type} \arrow[l, "\times B" description] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"] \end{tikzcd}$$

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

**necessarily** **indefinitely**

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

$$\Box_B P_\bullet \xrightarrow{\quad \epsilon_{P_\bullet}^{\Box_B} \quad} P_\bullet$$

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

Given $B : \mathrm{BType}$ 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** $\perp$

$$\begin{tikzcd}
\mathrm{Type}_B \arrow[loop above, "\lozenge_B"] \arrow[loop below, "\Box_B"] \arrow[r, shift left=4, "\coprod_B"] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] \arrow[r, shift right=4, "\prod_B"'] & \mathrm{Type} \arrow[l, "\times B"'] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"]
\end{tikzcd}$$

**necessarily** **indefinitely**

---

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

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

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

Given $B$ : BType 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**

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

$\perp \quad \perp \quad \perp \quad \perp$

**necessarily** **indefinitely**

---

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

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

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

**randomly** $P$

$$\star_B P$$

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

Given $B$ : BType 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

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

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

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

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

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

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

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

Given $B$ : BType 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** $\perp$ $\perp$ **potential data**

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

**necessarily** **indefinitely**

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

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

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

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

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

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

Given $B$ : BType 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** $\perp$

**necessarily** **indefinitely**

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

Given $B$ : BType 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

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

**necessarily $\mathcal{H}_\bullet$**

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

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

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

**measurement result**

Given $B$ : BType 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**

$$\begin{tikzcd}
& \text{possibly} & & & & \text{randomly} & \\
& \mathrm{LType}_B \simeq \mathrm{LType}^{\bigcirc_B} \arrow[loop above, "\Diamond_B"] \arrow[loop below, "\Box_B"'] \arrow[r, two heads, bend left, "\oplus_B"] \arrow[r, two heads, bend right, "\oplus_B"'] & (\mathrm{LType}_B)_{\Diamond_B} \arrow[r, hook, "\sim"] \arrow[d, phantom, "\perp"] & (\mathrm{LType}_B)^{\Diamond_B} \arrow[rr, phantom, "\simeq"] & & \mathrm{LType} \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"'] & \\
\text{actual data}\ \perp & & \mathrm{LType}_{\bigcirc_B} \arrow[ul, hook] \arrow[d, phantom, "\perp"] & & & & \perp\ \text{potential data} \\
& & (\mathrm{LType}_B)_{\Box_B} \arrow[r, hook, "\sim"'] & (\mathrm{LType}_B)^{\Box_B} \arrow[rr, phantom, "\simeq"] & & \\
& \text{necessarily} & & & & \text{indefinitely} &
\end{tikzcd}$$

$$\mathrm{LType}_B \simeq \mathrm{LType}^{\bigcirc_B} \xleftarrow{\quad} \mathrm{LType}_{\bigcirc_B} \twoheadleftarrow \mathbb{1}_B\otimes \text{———} \mathrm{LType}$$

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

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

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

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

**measurement result**

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

Given $B$ : BType 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

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

---

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

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

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

**measurement result**

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

Given $B$ : BType 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** $\perp$ $\quad$ $\perp$ **potential data**

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

**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...**

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

**measurement result**

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

Given $B : \mathrm{BType}$ 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**

$$\begin{tikzcd}
 & & (\mathrm{LType}_B)_{\Diamond_B} \arrow[r, hook, "\sim"] & (\mathrm{LType}_B)^{\Diamond_B} & \\
\mathrm{LType}_B \simeq \mathrm{LType}^{\bigcirc_B} \arrow[urr, two heads, "\bigoplus_B"] \arrow[drr, two heads, "\bigoplus_B"'] \arrow[loop above, "\Diamond_B"] \arrow[loop below, "\Box_B"'] & & \mathrm{LType}_{\bigcirc_B} \arrow[ll, hook'] \arrow[u, phantom, "\perp"] \arrow[d, phantom, "\perp"] & & \mathrm{LType} \arrow[ll, two heads, "\mathbb{1}_B \otimes"'] \arrow[ul, phantom, "\simeq"] \arrow[dl, phantom, "\simeq"] \arrow[loop above, "\star_B"] \arrow[loop below, "\bigcirc_B"'] \\
 & & (\mathrm{LType}_B)_{\Box_B} \arrow[r, hook, "\sim"'] & (\mathrm{LType}_B)^{\Box_B} &
\end{tikzcd}$$

**necessarily** **indefinitely**

---

**principle of quantum compulsion:**

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

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

**ambidexterity**

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

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

**measurement result**

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

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

$\star_B \mathcal{H}$

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

Given $B$ : BType 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**

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

**necessarily** **indefinitely**

**principle of quantum compulsion:**

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

$$\begin{tikzcd}
\square_B \mathcal{H}_\bullet \arrow[r, "\epsilon^{\square_B}_{\mathcal{H}_\bullet}"] & \mathcal{H}_\bullet \arrow[r, "\eta^{\lozenge_B}_{\mathcal{H}_\bullet}"] & \lozenge_B \mathcal{H}_\bullet \arrow[r, phantom, "\simeq"] \arrow[r, phantom, "\text{ambidexterity}"', shift right=3] & \square_B \mathcal{H}_\bullet
\end{tikzcd}$$

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

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

**measurement result**

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

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

$$\begin{tikzcd}
\star_B \mathcal{H} \arrow[r, "\epsilon^{\star_B}_{\mathcal{H}}"] & \mathcal{H}
\end{tikzcd}$$

$$\begin{tikzcd}
\bigoplus_{b:B} \mathcal{H} \arrow[r, "\oplus_b |\psi_b\rangle \mapsto \Sigma_b |\psi_b\rangle", "\text{quantum superposition}"'] & \mathcal{H}
\end{tikzcd}$$

Given $B$ : BType 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

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

**principle of quantum compulsion:**

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

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

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

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

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

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

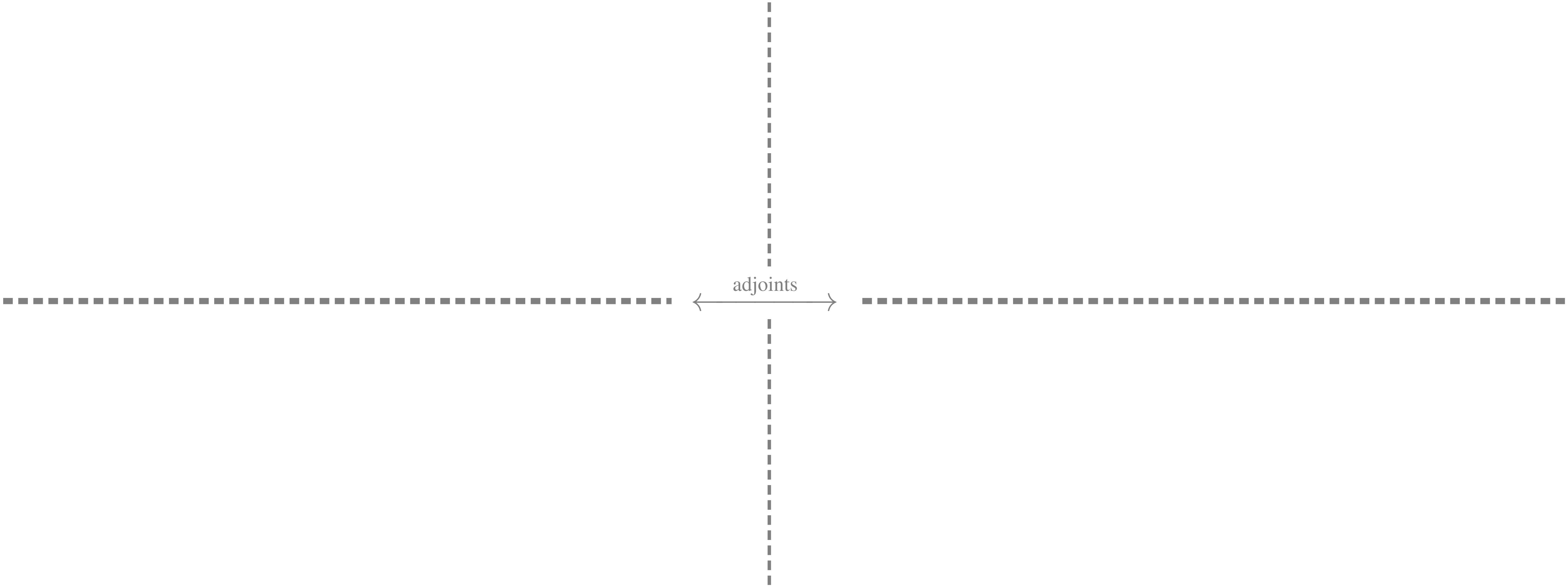
$$\begin{tikzcd}
\star_B\,\mathcal{H} \arrow[r, "\varepsilon^{\star_B}_{\mathcal{H}}"] & \mathcal{H} \arrow[r, "\eta^{\bigcirc_B}_{\mathcal{H}}"] & \bigcirc_B\,\mathcal{H}
\end{tikzcd}$$

$$\begin{tikzcd}
\bigoplus\limits_{b:B}\mathcal{H} \arrow[r, "\oplus_b\,|\psi_b\rangle \mapsto \Sigma_b\,|\psi_b\rangle", "\text{quantum superposition}"'] & \mathcal{H} \arrow[r, "|\psi\rangle \mapsto \oplus_b |\psi\rangle_b", "\text{quantum parallelization}"'] & \bigoplus\limits_{b:B}\mathcal{H}
\end{tikzcd}$$

# 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

# 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 ↔

$$\mathcal{H}_\bullet \xrightarrow[\textbf{possibility unit}]{\eta^{\Diamond_B}_{\mathcal{H}_\bullet}} \overbrace{(p_B)^*(p_B)_!}^{\Diamond_B}\mathcal{H}_\bullet$$

$$b : B \vdash \mathcal{H}_b \xrightarrow[\text{quantum }\textbf{state preparation}]{|\psi_b\rangle \;\mapsto\; \oplus_{b'}\begin{cases}|\psi_b\rangle & \text{if } b'=b\\ 0 & \text{else}\end{cases}} \bigoplus_{b':B}\mathcal{H}_{b'}$$

“ the actual is possible ”

## 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 ”

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:

$$\overbrace{(p_B)^*(p_B)_*}^{\Box_B}\mathcal{H}_\bullet \xrightarrow[\textbf{necessity counit}]{\epsilon^{\Box_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet$$

$$b : B \vdash \bigoplus_{b':B}\mathcal{H}_{b'} \xrightarrow[\text{quantum }\textbf{measurement}]{\oplus_{b'}|\psi_{b'}\rangle \;\mapsto\; |\psi_b\rangle} \mathcal{H}_b$$

“ 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 potential is inderminate ”

$$\mathcal{H} \xrightarrow[\textbf{indeterminacy unit}]{\eta^{\bigcirc_B}_{\mathcal{H}}} \overbrace{(p_B)_*(p_B)^*}^{\bigcirc_B}\mathcal{H}$$

$$\mathcal{H} \xrightarrow[\text{quantum }\textbf{parallelism}]{|\psi\rangle \;\mapsto\; \oplus_b|\psi\rangle_b} \bigoplus_{b:B}\mathcal{H}$$

# Q-bits are the free linear indeterminacy-effect handlers over $\mathrm{Bit} = \{0, 1\}$

**Coherent q-bits:**

$$\begin{array}{l} \mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}\otimes} \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_B} \\ \quad \overset{\cdot\cdot}{\|} \\ \bigcirc_{\mathrm{Bit}}\mathbb{1} \end{array}$$

**De-cohered (measured) q-bits:**

**Quantum gate with q-bit output:**

# Q-bits are the free linear indeterminacy-effect handlers over $\mathrm{Bit} = \{0, 1\}$

## Coherent q-bits:

▬ $$\mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}\otimes} \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\oplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_B}$$

$$\overset{\cdot\cdot}{\|}$$

$$\bigcirc_{\mathrm{Bit}}\mathbb{1} \;=\; \bigoplus_{\{0,1\}}\mathbb{C} \;=\; \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle$$

## Quantum gate with q-bit output:

## De-cohered (measured) q-bits:

# Q-bits are the free linear indeterminacy-effect handlers over $\mathrm{Bit} = \{0, 1\}$

## Coherent q-bits:

$$\begin{tikzcd} \mathrm{QBit} \arrow[d, phantom, "\rotatebox{90}{$=:$}"] & : & \mathrm{LType} \arrow[r, hook, "\mathbb{1}_{\mathrm{Bit}}\otimes"] & \mathrm{LType}_{\mathrm{Bit}} \arrow[r, "\bigoplus_{\mathrm{Bit}}", "\sim"'] & \mathrm{LType}^{\bigcirc_B} \\ \bigcirc_{\mathrm{Bit}}\mathbb{1} & = & \bigoplus_{\{0,1\}}\mathbb{C} & = & \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle \end{tikzcd}$$

## De-cohered (measured) q-bits:

## Quantum gate with q-bit output:

# **Q-bits are the free linear indeterminacy-effect handlers over** $\mathrm{Bit} = \{0,1\}$

## **Coherent q-bits:**

$$\mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}\otimes} \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_B}$$

$$\ddot{\|}$$

$$\bigcirc_{\mathrm{Bit}}\mathbb{1} \;=\; \bigoplus_{\{0,1\}}\mathbb{C} \;=\; \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle$$

$$\mathrm{QBit}$$

$$\otimes$$

$$\mathcal{H}$$

$$\|$$

$$\bigcirc_{\mathrm{Bit}}\mathcal{H} \;=\; \bigoplus_{\{0,1\}}\mathcal{H} \;=\; \mathcal{H}\otimes|0\rangle \;\oplus\; \mathcal{H}\otimes|0\rangle$$

## **De-cohered (measured) q-bits:**

## **Quantum gate with q-bit output:**

# Q-bits are the free linear indeterminacy-effect handlers over $\mathrm{Bit} = \{0,1\}$

**Coherent q-bits:**

$$\mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}\otimes} \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_B}$$

$$\overset{\cdot\cdot}{\shortparallel}$$

$$\bigcirc_{\mathrm{Bit}}\mathbb{1} \;=\; \bigoplus_{\{0,1\}}\mathbb{C} \;=\; \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle$$

$$\mathrm{QBit}$$

$$\otimes$$

$$\mathcal{H}$$

$$\|$$

$$\bigcirc_{\mathrm{Bit}}\mathcal{H} \;=\; \bigoplus_{\{0,1\}}\mathcal{H} \;=\; \mathcal{H}\otimes|0\rangle \;\oplus\; \mathcal{H}\otimes|0\rangle$$

**De-cohered (measured) q-bits:**

$$\mathbb{1}_{\mathrm{Bit}} \;:\; \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_{\mathrm{Bit}}}$$

$$b : \mathrm{Bit} \quad\vdash\quad \mathbb{C}\cdot|b\rangle \;:\; \mathrm{LType}$$

**Quantum gate with q-bit output:**

# Q-bits are the free linear indeterminacy-effect handlers over $\mathrm{Bit} = \{0,1\}$

## Coherent q-bits:

$$\mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}\otimes} \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_B}$$

$\overset{\cdot\cdot}{\shortparallel}$

$$\bigcirc_{\mathrm{Bit}}\mathbb{1} \;=\; \bigoplus_{\{0,1\}}\mathbb{C} \;=\; \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle$$

$\mathrm{QBit}$

$\otimes$

$\mathcal{H}$

$\shortparallel$

$$\bigcirc_{\mathrm{Bit}}\mathcal{H} \;=\; \bigoplus_{\{0,1\}}\mathcal{H} \;=\; \mathcal{H}\otimes|0\rangle \;\oplus\; \mathcal{H}\otimes|0\rangle$$

## De-cohered (measured) q-bits:

$$\mathbb{1}_{\mathrm{Bit}} \;:\; \mathrm{LType}_{\mathrm{Bit}} \xrightarrow[\sim]{\bigoplus_{\mathrm{Bit}}} \mathrm{LType}^{\bigcirc_{\mathrm{Bit}}}$$

$$b : \mathrm{Bit} \quad\vdash\quad \mathbb{C}\cdot|b\rangle \;:\; \mathrm{LType}$$

$\mathbb{1}_{\mathrm{Bit}}$

$\otimes$

$\mathcal{H}$

$$b : \mathrm{Bit} \quad\vdash\quad \mathcal{H}\otimes|b\rangle \;:\; \mathrm{LType}$$

## Quantum gate with q-bit output:

# Q-bits are the free linear indeterminacy-effect handlers over $\mathrm{Bit} = \{0,1\}$

## Coherent q-bits:

$$\mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}\otimes} \mathrm{LType}_{\mathrm{Bit}} \underset{\sim}{\xrightarrow{\oplus_{\mathrm{Bit}}}} \mathrm{LType}^{\bigcirc_B}$$

$$\rotatebox{90}{$=:$}$$

$$\bigcirc_{\mathrm{Bit}}\mathbb{1} \;=\; \bigoplus_{\{0,1\}}\mathbb{C} \;=\; \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle$$

$$\mathrm{QBit}$$

$$\otimes$$

$$\mathcal{H}$$

$$\|$$

$$\bigcirc_{\mathrm{Bit}}\mathcal{H} \;=\; \bigoplus_{\{0,1\}}\mathcal{H} \;=\; \mathcal{H}\otimes|0\rangle \;\oplus\; \mathcal{H}\otimes|0\rangle$$

## De-cohered (measured) q-bits:

$$\mathbb{1}_{\mathrm{Bit}} \;:\; \mathrm{LType}_{\mathrm{Bit}} \underset{\sim}{\xrightarrow{\oplus_{\mathrm{Bit}}}} \mathrm{LType}^{\bigcirc_{\mathrm{Bit}}}$$

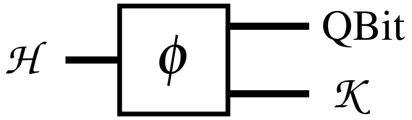
$$b : \mathrm{Bit} \quad\vdash\quad \mathbb{C}\cdot|b\rangle \;:\; \mathrm{LType}$$

$$\mathbb{1}_{\mathrm{Bit}} \otimes \mathcal{H}$$

$$b : \mathrm{Bit} \quad\vdash\quad \mathcal{H}\otimes|b\rangle \;:\; \mathrm{LType}$$

## 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 indeterminacy-effect handlers over $\mathrm{Bit} = \{0,1\}$

**Coherent q-bits:**

$$\mathrm{QBit} \;:\; \mathrm{LType} \xhookrightarrow{\mathbb{1}_{\mathrm{Bit}}\otimes} \mathrm{LType}_{\mathrm{Bit}} \underset{\sim}{\xrightarrow{\oplus_{\mathrm{Bit}}}} \mathrm{LType}^{\bigcirc_B}$$

$$\overset{\cdot\cdot}{||}$$

$$\bigcirc_{\mathrm{Bit}}\mathbb{1} \;=\; \bigoplus_{\{0,1\}}\mathbb{C} \;=\; \mathbb{C}\cdot|0\rangle \oplus \mathbb{C}\cdot|1\rangle$$

$$\begin{array}{c} \mathrm{QBit} \\ \otimes \\ \mathcal{H} \\ || \end{array}$$

$$\bigcirc_{\mathrm{Bit}}\mathcal{H} \;=\; \bigoplus_{\{0,1\}}\mathcal{H} \;=\; \mathcal{H}\otimes|0\rangle \;\oplus\; \mathcal{H}\otimes|0\rangle$$

**De-cohered (measured) q-bits:**

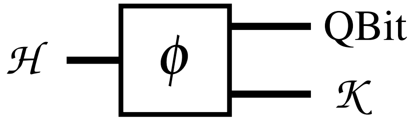
$$\mathbb{1}_{\mathrm{Bit}} \;:\; \mathrm{LType}_{\mathrm{Bit}} \underset{\sim}{\xrightarrow{\oplus_{\mathrm{Bit}}}} \mathrm{LType}^{\bigcirc_{\mathrm{Bit}}}$$

$$b : \mathrm{Bit} \quad \vdash \quad \mathbb{C}\cdot|b\rangle \;:\; \mathrm{LType}$$

$$\begin{array}{c} \mathbb{1}_{\mathrm{Bit}} \\ \otimes \\ \mathcal{H} \end{array} \qquad b : \mathrm{Bit} \quad \vdash \quad \mathcal{H}\otimes|b\rangle \;:\; \mathrm{LType}$$

**Quantum gate with q-bit output:**

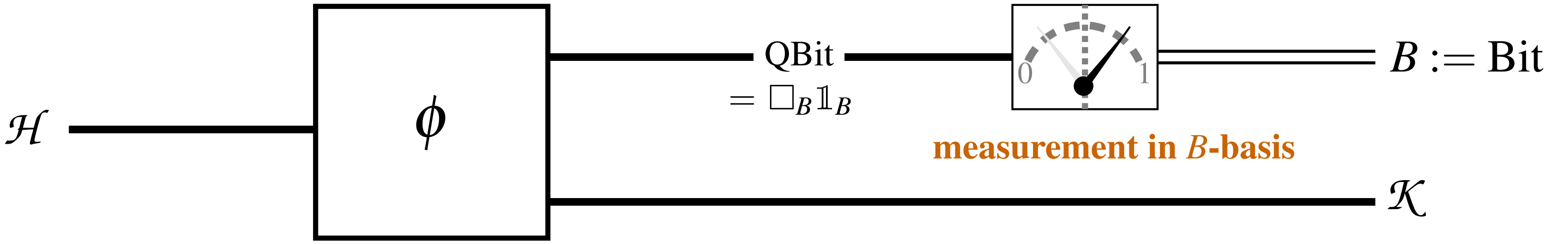
A quantum gate which may handle $\bigcirc_{\mathrm{Bit}}$-effects is one with a QBit-output:



$$\mathcal{H} \xrightarrow{\phi} \mathrm{QBit}\otimes\mathcal{K} \simeq \bigcirc_{\mathrm{Bit}}\mathcal{K}$$

# Quantum measurement is Linear indefiniteness-effect handling.

**quantum circuit**



**measurement in $B$-basis**

**quantum gate**

$$\mathcal{H} \xrightarrow{\phi} \mathrm{QBit} \otimes \mathcal{K} \simeq \bigcirc_B \mathcal{K}$$

**$\bigcirc_B$-effect handling**

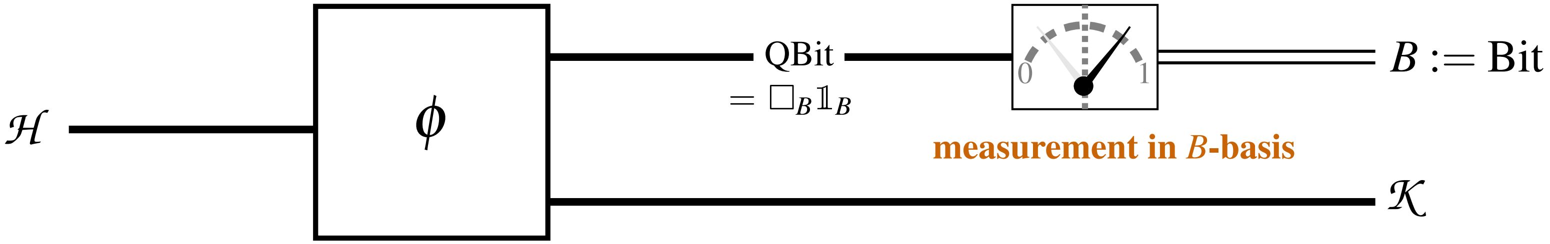
# Quantum measurement is Linear indefiniteness-effect handling.

**quantum circuit**

formalization

**$\bigcirc_B$-modal linear types**

$\mathrm{LType}_{\bigcirc_B}$



**measurement in $B$-basis**

**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.

**quantum circuit**

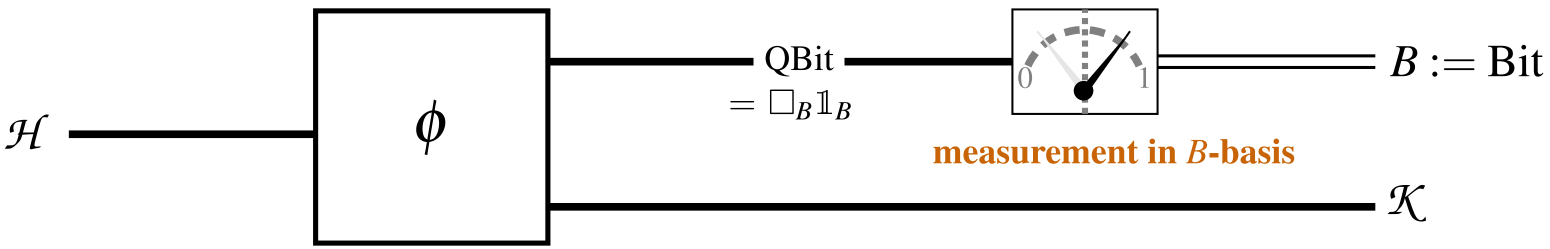
$$\begin{tikzcd}
\text{quantum circuit} \arrow[d, squiggly, "\text{formalization}"] \\
\mathrm{LType}_{\bigcirc_B} \arrow[d, "\text{comparison functor}", "K_{(p_B)_*(p_B)^*}"'] \\
\mathrm{LType}_B
\end{tikzcd}$$

**$\bigcirc_B$-modal linear types**

$\mathrm{LType}_{\bigcirc_B}$

$\mathrm{LType}_B$

***B*-dependent linear types**



**subjection to $\bigcirc_B$-effects**

**quantum gate**

$$\begin{tikzcd}[column sep=huge]
\bigcirc_B \mathcal{H} \arrow[loop above, "\bigcirc_B"] \arrow[r, "\bigcirc_B\left(\mathcal{H} \xrightarrow{\phi} \mathrm{QBit}\otimes\mathcal{K} \simeq \bigcirc_B \mathcal{K}\right)"] & \bigcirc_B\bigcirc_B\mathcal{K} \arrow[loop above, "\bigcirc_B"] \arrow[r, "\mathrm{hndl}^{\bigcirc_B}_{\bigcirc_B \mathcal{K}}", "\text{$\bigcirc_B$-effect handling}"'] & \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"]
\end{tikzcd}$$

# Quantum measurement is Linear indefiniteness-effect handling.

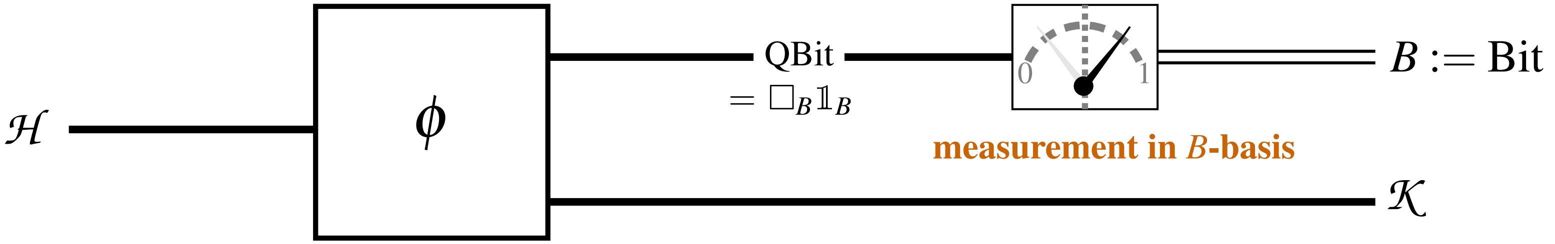
**quantum circuit**

$\rightsquigarrow$ formalization

**$\bigcirc_B$-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**



**quantum gate**

**subjection to $\bigcirc_B$-effects**

$$\begin{tikzcd}
\bigcirc_B \mathcal{H} \arrow[loop above, "\bigcirc_B"] \arrow[rr, "\bigcirc_B\left(\mathcal{H} \xrightarrow{\phi} \mathrm{QBit} \otimes \mathcal{K} \simeq \bigcirc_B \mathcal{K}\right)"] & & \bigcirc_B \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"] \arrow[rr, "\mathrm{hndl}^{\bigcirc_B}_{\bigcirc_B \mathcal{K}}", "\bigcirc_B\text{-effect handling}"'] & & \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"]
\end{tikzcd}$$

$$\begin{tikzcd}
\mathbb{1}_B \otimes \mathcal{H} \arrow[rr, "\mathbb{1}_B \otimes \phi"] & & \Box_B \mathbb{1}_B \otimes \mathcal{K} \arrow[rr, "\epsilon^{\Box_B}_{\mathbb{1}_B} \otimes \mathcal{K}", "\Box_B\text{-counit}"'] & & \mathbb{1}_B \otimes \mathcal{K}
\end{tikzcd}$$

$$\begin{tikzcd}
b : B \vdash \mathcal{H} \arrow[rr, "\phi"] & & \mathrm{QBit} \otimes \mathcal{K} \arrow[rr, "\text{quantum state collapse}"] & & \mathcal{K}
\end{tikzcd}$$

**measurement result**

$$\begin{array}{c} |0\rangle \otimes |\psi_0\rangle \\ + |1\rangle \otimes |\psi_1\rangle \end{array} \mapsto |\psi_b\rangle$$

# Quantum measurement is Linear indefiniteness-effect handling.

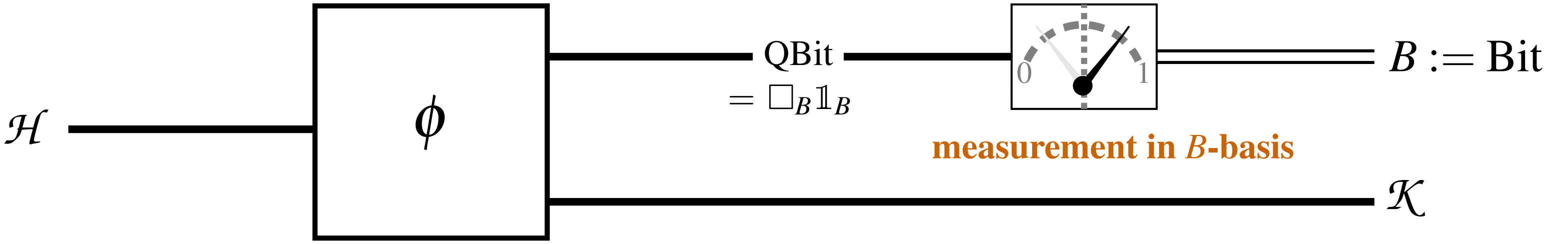
**quantum circuit**

formalization

**$\bigcirc_B$-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**

$$\begin{tikzcd}[column sep=huge] \bigcirc_B \mathcal{H} \arrow[loop above, "\bigcirc_B"] \arrow[r, "\overset{\text{quantum gate}}{\bigcirc_B\left(\mathcal{H} \xrightarrow{\phi} \mathrm{QBit} \otimes \mathcal{K} \simeq \bigcirc_B \mathcal{K}\right)}"] & \bigcirc_B \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"] \arrow[r, "\mathrm{hndl}^{\bigcirc_B}_{\bigcirc_B \mathcal{K}}", "\bigcirc_B\text{-effect handling}"'] & \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"] \end{tikzcd}$$

$$\begin{tikzcd}[column sep=huge] \mathbb{1}_B \otimes \mathcal{H} \arrow[r, "\mathbb{1}_B \otimes \phi"] & \Box_B \mathbb{1}_B \otimes \mathcal{K} \arrow[r, "\epsilon^{\Box_B}_{\mathbb{1}_B} \otimes \mathcal{K}", "\Box_B\text{-counit}"'] & \mathbb{1}_B \otimes \mathcal{K} \end{tikzcd}$$

$$\begin{tikzcd}[column sep=huge] b : B \vdash \mathcal{H} \arrow[r, "\phi"] & \mathrm{QBit} \otimes \mathcal{K} \arrow[r, "\text{quantum state collapse}"] & \mathcal{K} \end{tikzcd}$$

**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 dLHoTT

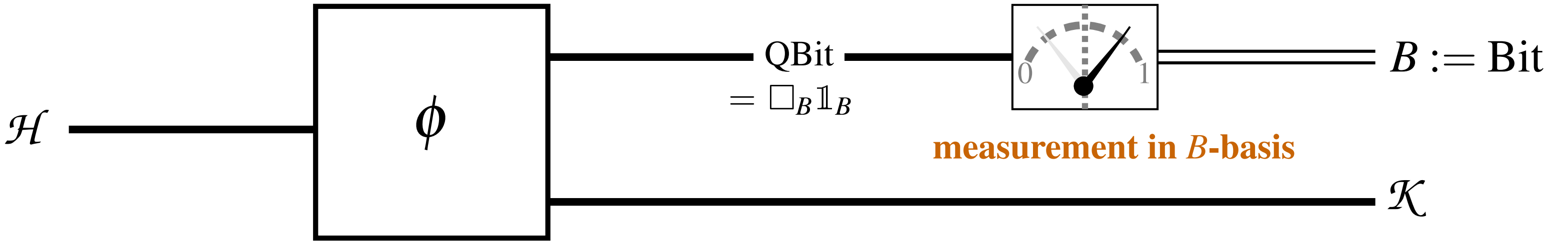
# Quantum measurement is Linear indefiniteness-effect handling.

**quantum circuit**

$$\begin{tikzcd} \text{quantum circuit} \arrow[d, squiggly, "\text{formalization}"] \\ \bigcirc_B\text{-modal linear types} \\ \mathrm{LType}_{\bigcirc_B} \arrow[d, hook, "\text{comparison functor}", "K_{(p_B)_*(p_B)^*}"'] \\ \mathrm{LType}_B \\ B\text{-dependent linear types} \end{tikzcd}$$

**$\bigcirc_B$-modal linear types**

**$B$-dependent linear types**



**measurement in $B$-basis**

$$\begin{tikzcd} \bigcirc_B \mathcal{H} \arrow[loop above, "\bigcirc_B"] \arrow[rr, "\bigcirc_B\left(\mathcal{H} \xrightarrow{\phi} \mathrm{QBit} \otimes \mathcal{K} \simeq \bigcirc_B \mathcal{K}\right)"] & & \bigcirc_B \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"] \arrow[rr, "\mathrm{hndl}^{\bigcirc_B}_{\bigcirc_B \mathcal{K}}", "\bigcirc_B\text{-effect handling}"'] & & \bigcirc_B \mathcal{K} \arrow[loop above, "\bigcirc_B"] \end{tikzcd}$$

**subjection to $\bigcirc_B$-effects**

**quantum gate**

**$\bigcirc_B$-effect handling**

$$\begin{tikzcd} \mathbb{1}_B \otimes \mathcal{H} \arrow[rr, "\mathbb{1}_B \otimes \phi"] & & \Box_B \mathbb{1}_B \otimes \mathcal{K} \arrow[rr, "\varepsilon^{\Box_B}_{\mathbb{1}_B} \otimes \mathcal{K}", "\Box_B\text{-counit}"'] & & \mathbb{1}_B \otimes \mathcal{K} \end{tikzcd}$$

**$\Box_B$-counit**

$$\begin{tikzcd} b : B \vdash \mathcal{H} \arrow[rr, "\phi"] & & \mathrm{QBit} \otimes \mathcal{K} \arrow[rr, "\text{quantum state collapse}"] & & \mathcal{K} \end{tikzcd}$$

**measurement result**

**quantum state collapse**

$$\begin{array}{l} |0\rangle \otimes |\psi_0\rangle \\ + |1\rangle \otimes |\psi_1\rangle \end{array} \mapsto |\psi_b\rangle$$

full linearly-typed detail of
quantum measurement logic
is emergent effect in `dLHoTT`

# Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

(see nLab:quantum+reader+monad)

$$\begin{array}{c} \bigcirc_B \\ \\ \\ \\ \\ \rotatebox{-90}{$\in$} \\ \mathrm{Monad}\big(\mathrm{LType}\big) \end{array}$$

# Aside: **Linear indefiniteness monad recovers Coecke’s “classical structures”.**

(see nLab:quantum+reader+monad)

$$\begin{array}{c} \bigcirc_B \\ \rotatebox{90}{$:=$} \\ B\text{-Reader} \\ \\ \\ \rotatebox{-90}{$\in$} \\ \mathrm{Monad}\big(\mathrm{LType}\big) \end{array}$$

# Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

(see nLab:quantum+reader+monad)

$$B : \mathrm{FinType} \;\vdash\; \begin{array}{c} \bigcirc_B \\ \rotatebox{90}{$=:$} \\ B\text{-Reader} \\ \wr\rotatebox{90}{$=$} \\ \mathbb{1}^B\text{-Writer} \\ \\ \rotatebox{90}{$\in$} \\ \mathrm{Monad}\big(\mathrm{LType}\big) \end{array}$$

$$\mathbb{1}^B\text{-Writer}(D) := \mathbb{1}^B \oplus D$$

$$\mathrm{bind}^{\mathbb{1}^B\mathrm{Writer}}\Big(D_1 \xrightarrow{\mathrm{prog}} \mathbb{1}^B \oplus D_2\Big) := \mathbb{1}^B \oplus D_1 \xrightarrow{\mathbb{1}^B \oplus \mathrm{prog}} \mathbb{1}^B \oplus \mathbb{1}^B \oplus D_2 \xrightarrow{\mu \oplus \mathrm{id}_{D_2}} \mathbb{1}^B \oplus D_2$$

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**

$$\mathbb{1}^B \otimes \mathbb{1}^B \xrightarrow[\;P_b \otimes P_{b'} \mapsto \left\{ \begin{array}{l} P_b \text{ if } b{=}b' \\ 0 \text{ else} \end{array} \right.\;]{\mu} \mathbb{1}^B$$

# Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

(see nLab:quantum+reader+monad)

$$\begin{tikzcd}[column sep=small]
 & & \bigcirc_B \arrow[d, phantom, "\overset{!}{=}"] \\
 & & B\text{-Reader} \arrow[d, phantom, "\wr|"] \\
 & & \mathbb{1}^B\text{-Writer} \\
B : \mathrm{FinType} \;\vdash & \mathbb{1}^B\text{-DualWriter} \arrow[ur, phantom, "\simeq"] \arrow[dr, phantom, "\in"] & \\
 & & \mathrm{Monad}\big(\mathrm{LType}\big) \\
 & \mathrm{FrobMonad}\big(\mathrm{LType}\big) \arrow[ur] &
\end{tikzcd}$$

$$\mathbb{1}^B\text{-Writer}(D) := \mathbb{1}^B \oplus D$$

$$\mathrm{bind}^{\mathbb{1}^B\mathrm{Writer}}\Big(D_1 \xrightarrow{\mathrm{prog}} \mathbb{1}^B \oplus D_2\Big) := \mathbb{1}^B \oplus D_1 \xrightarrow{\mathbb{1}^B \oplus \mathrm{prog}} \mathbb{1}^B \oplus \mathbb{1}^B \oplus D_2 \xrightarrow{\mu\oplus\mathrm{id}_{D_2}} \mathbb{1}^B \oplus D_2$$

Where $\mathbb{1}^B = \bigoplus\limits_{b:B} \mathbb{C}\cdot P_b \;\in\; \mathrm{CMon}\big(\mathrm{LType}\big)$ is Frobenius algebra of $B$-projection operators :

$$\begin{tikzcd}
\mathbb{1} \arrow[r, "\substack{\textbf{unit}\\ \eta}", "1 \mapsto \sum\limits_{b:B} P_b"'] & \mathbb{1}^B \arrow[r, "\substack{\textbf{co-product}\\ \delta}", "P_b \mapsto P_b \otimes P_b"'] & \mathbb{1}^B \otimes \mathbb{1}^B \arrow[r, "\substack{\textbf{product}\\ \mu}", "P_b \otimes P_{b'} \mapsto \begin{cases} P_b \text{ if } b=b' \\ 0 \text{ else} \end{cases}"'] & \mathbb{1}^B \arrow[r, "\substack{\textbf{co-unit}\\ \varepsilon}", "P_b \mapsto 1"'] & \mathbb{1}
\end{tikzcd}$$

# Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

(see nLab:quantum+reader+monad)

$B : \mathrm{FinType}\ \vdash$

$$\begin{tikzcd}
 & & & \bigcirc_B \arrow[d, phantom, "\rotatebox[origin=c]{90}{$:=$}"] \\
 & & & B\text{-Reader} \arrow[d, phantom, "\rotatebox[origin=c]{90}{$\simeq$}"] \\
 & \mathbb{1}^B\text{-CoWriter} \arrow[dr, phantom, "\simeq" sloped] & & \mathbb{1}^B\text{-Writer} \arrow[dl, phantom, "\simeq" sloped] \\
 & & \mathbb{1}^B\text{-DualWriter} \arrow[d, phantom, "\pitchfork"] & \\
 & \mathrm{CoMonad}\big(\mathrm{LType}\big) & \phantom{x} & \mathrm{Monad}\big(\mathrm{LType}\big) \\
 & & \mathrm{FrobMonad}\big(\mathrm{LType}\big) \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$$\mathbb{1}^B\text{-Writer}(D) := \mathbb{1}^B \oplus D$$

$$\mathrm{bind}^{\mathbb{1}^B\mathrm{Writer}}\Big(D_1 \xrightarrow{\mathrm{prog}} \mathbb{1}^B \oplus D_2\Big) := \mathbb{1}^B \oplus D_1 \xrightarrow{\mathbb{1}^B \oplus \mathrm{prog}} \mathbb{1}^B \oplus \mathbb{1}^B \oplus D_2 \xrightarrow{\mu \oplus \mathrm{id}_{D_2}} \mathbb{1}^B \oplus D_2$$

Where $\mathbb{1}^B = \bigoplus\limits_{b:B} \mathbb{C} \cdot P_b \ \in\ \mathrm{CMon}\big(\mathrm{LType}\big)$ is Frobenius algebra of $B$-projection operators :

$$\begin{tikzcd}[column sep=huge]
\mathbb{1} \arrow[r, "\overset{\textbf{unit}}{\eta}", "1 \mapsto \sum\limits_{b:B} P_b"'] & \mathbb{1}^B \arrow[r, "\overset{\textbf{co-product}}{\delta}", "P_b \mapsto P_b \otimes P_b"'] & \mathbb{1}^B \otimes \mathbb{1}^B \arrow[r, "\overset{\textbf{product}}{\mu}", "P_b \otimes P_{b'} \mapsto \begin{cases} P_b \text{ if } b=b' \\ 0 \text{ else} \end{cases}"'] & \mathbb{1}^B \arrow[r, "\overset{\textbf{co-unit}}{\varepsilon}", "P_b \mapsto 1"'] & \mathbb{1}
\end{tikzcd}$$

# Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

(see nLab:quantum+reader+monad)

$$\begin{tikzcd}
 & & & \bigcirc_B \arrow[d, phantom, "\rotatebox{-90}{$:=$}"] \\
 & \color{blue}{B\text{-CoReader}} \arrow[d, phantom, "\wr"] & & B\text{-Reader} \arrow[d, phantom, "\wr"] \\
 & \mathbb{1}^B\text{-CoWriter} \arrow[dr, phantom, "\simeq"] & & \mathbb{1}^B\text{-Writer} \arrow[dl, phantom, "\simeq"] \\
B : \mathrm{FinType} \;\vdash & & \mathbb{1}^B\text{-DualWriter} \arrow[d, phantom, "\pitchfork"] & \\
 & \mathrm{CoMonad}\big(\mathrm{LType}\big) & & \mathrm{Monad}\big(\mathrm{LType}\big) \\
 & & \mathrm{FrobMonad}\big(\mathrm{LType}\big) \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$$\mathbb{1}^B\text{-Writer}(D) := \mathbb{1}^B \oplus D$$

$$\mathrm{bind}^{\mathbb{1}^B\mathrm{Writer}}\Big(D_1 \xrightarrow{\mathrm{prog}} \mathbb{1}^B \oplus D_2\Big) := \mathbb{1}^B \oplus D_1 \xrightarrow{\mathbb{1}^B \oplus \mathrm{prog}} \mathbb{1}^B \oplus \mathbb{1}^B \oplus D_2 \xrightarrow{\mu \oplus \mathrm{id}_{D_2}} \mathbb{1}^B \oplus D_2$$

Where $\mathbb{1}^B = \bigoplus_{b:B} \mathbb{C} \cdot P_b \;\in\; \mathrm{CMon}(\mathrm{LType})$ 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)

$$\begin{tikzcd}
 & \color{blue}{\star_B} \arrow[d, phantom, "\rotatebox{-90}{$=:$}"] & & \bigcirc_B \arrow[d, phantom, "\rotatebox{-90}{$=:$}"] \\
 & B\text{-CoReader} \arrow[d, phantom, "\rotatebox{90}{$\simeq$}"] & & B\text{-Reader} \arrow[d, phantom, "\rotatebox{90}{$\simeq$}"] \\
 & \mathbb{1}^B\text{-CoWriter} \arrow[dr, phantom, "\simeq"] & & \mathbb{1}^B\text{-Writer} \arrow[dl, phantom, "\simeq"] \\
B : \mathrm{FinType} \;\; \vdash & & \mathbb{1}^B\text{-DualWriter} \arrow[d, phantom, "\Cap"] & \\
 & \mathrm{CoMonad}\big(\mathrm{LType}\big) & & \mathrm{Monad}\big(\mathrm{LType}\big) \\
 & & \mathrm{FrobMonad}\big(\mathrm{LType}\big) \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$$\mathbb{1}^B\text{-Writer}(D) := \mathbb{1}^B \otimes D$$

$$\mathrm{bind}^{\mathbb{1}^B\mathrm{Writer}}\Big(D_1 \xrightarrow{\mathrm{prog}} \mathbb{1}^B \otimes D_2\Big) := \mathbb{1}^B \otimes D_1 \xrightarrow{\mathbb{1}^B \otimes \mathrm{prog}} \mathbb{1}^B \otimes \mathbb{1}^B \otimes D_2 \xrightarrow{\mu \otimes \mathrm{id}_{D_2}} \mathbb{1}^B \otimes D_2$$

Where $\mathbb{1}^B = \bigoplus_{b:B} \mathbb{C} \cdot P_b \;\in\; \mathrm{CMon}(\mathrm{LType})$ is Frobenius algebra of $B$-projection operators :

$$\begin{tikzcd}[column sep=huge]
\mathbb{1} \arrow[r, "\substack{\textbf{unit} \\ \eta}", "1 \mapsto \sum\limits_{b:B} P_b"'] & \mathbb{1}^B \arrow[r, "\substack{\textbf{co-product} \\ \delta}", "P_b \mapsto P_b \otimes P_b"'] & \mathbb{1}^B \otimes \mathbb{1}^B \arrow[r, "\substack{\textbf{product} \\ \mu}", "P_b \otimes P_{b'} \mapsto \begin{cases} P_b \text{ if } b = b' \\ 0 \text{ else} \end{cases}"'] & \mathbb{1}^B \arrow[r, "\substack{\textbf{co-unit} \\ \varepsilon}", "P_b \mapsto 1"'] & \mathbb{1}
\end{tikzcd}$$

# Aside: **Linear indefiniteness monad recovers Coecke's "classical structures".**

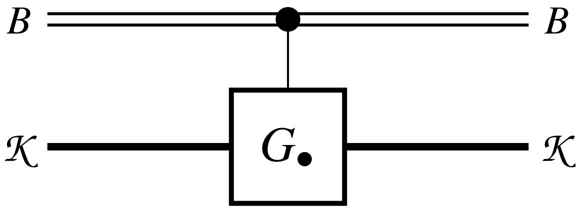
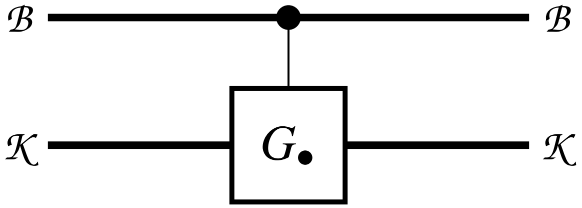
(see nLab:quantum+reader+monad)

$$\begin{tikzcd}
 & \star_B \arrow[rr, equal] \arrow[dr, equal] & & \bigcirc_B \arrow[dl, equal] \\
B : \mathrm{FinType} \;\vdash & & \mathbb{1}^B\text{-DualWriter} & & \\
 & & \pitchfork & & \\
 & \mathrm{CoMonad}(\mathrm{LType}) & & \mathrm{Monad}(\mathrm{LType}) & \\
 & & \mathrm{FrobMonad}(\mathrm{LType}) \arrow[ul] \arrow[ur] & &
\end{tikzcd}$$

Where $\mathbb{1}^B = \bigoplus_{b:B} \mathbb{C} \cdot P_b \in \mathrm{CMon}(\mathrm{LType})$ is Frobenius algebra of $B$-projection operators :

$$\begin{tikzcd}
\mathbb{1} \arrow[r, "\substack{\textbf{unit}\\ \eta}", "1 \mapsto \sum_{b:B} P_b"'] & \mathbb{1}^B \arrow[r, "\substack{\textbf{co-product}\\ \delta}", "P_b \mapsto P_b \otimes P_b"'] & \mathbb{1}^B \otimes \mathbb{1}^B \arrow[r, "\substack{\textbf{product}\\ \mu}", "P_b \otimes P_{b'} \mapsto \begin{cases} P_b \text{ if } b=b' \\ 0 \text{ else} \end{cases}"'] & \mathbb{1}^B \arrow[r, "\substack{\textbf{co-unit}\\ \varepsilon}", "P_b \mapsto 1"'] & \mathbb{1}
\end{tikzcd}$$

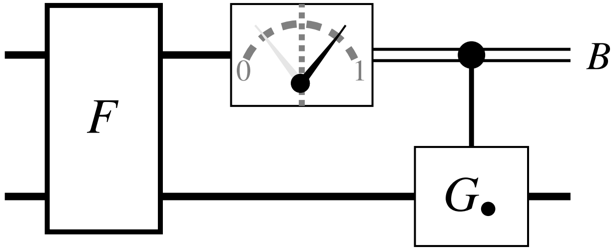
# Exmp: Deferred measurement principle – Proven by monadic effect logic.

| classically controlled gate | quantumly controlled gate |
|---|---|
| <br> <br>$\mathcal{B}_\bullet \boxtimes \mathcal{K} \xrightarrow{G_\bullet} \mathcal{B}_\bullet \boxtimes \mathcal{K}$<br>$b : B \;\vdash\; \mathcal{K} \xrightarrow{G_b} \mathcal{K}$ | <br> <br>$\Box_B \mathcal{B}_\bullet \boxtimes \mathcal{K} \xrightarrow{\Box_B G_\bullet} \Box_B \mathcal{B}_\bullet \boxtimes \mathcal{K}$<br>$b : B \;\vdash\; \bigoplus_{b'B} \mathcal{K} \xrightarrow{\bigoplus_{b':B} G_{b'}} \bigoplus_{b'B} \mathcal{K}$ |

# Exmp: Deferred measurement principle – Proven by monadic effect logic.

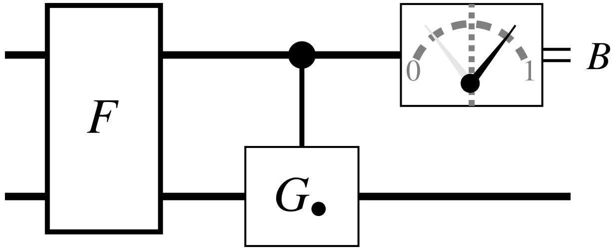
$$\Box_B\mathcal{H}_\bullet \xrightarrow{F} \Box_B\mathcal{H}_\bullet \xrightarrow{\varepsilon^{\Box_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet \xrightarrow{G_\bullet} \mathcal{H}_\bullet \quad\mapsto\quad \Box_B\mathcal{H}_\bullet \xrightarrow{\mathrm{diag}_B(G_\bullet)\circ F} \Box_B\mathcal{H}_\bullet \quad\mapsto\quad \Box_B\mathcal{H}_\bullet \xrightarrow{\mathrm{diag}_B(G_\bullet)\circ F} \Box_B\mathcal{H}_\bullet \xrightarrow{\varepsilon^{\Box_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet$$

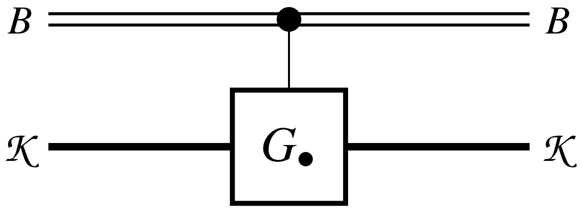
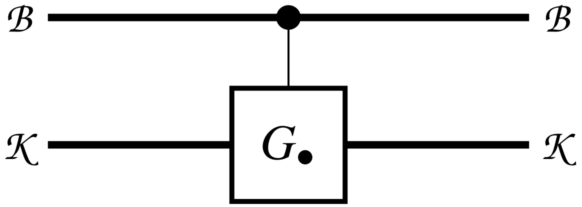
**measurement-controlled quantum gate** **quantum-controlled quantum gate...** **...followed by measurement**



**Deferred Measurement Principle**

$\longleftrightarrow$



| **classically controlled gate** | **quantumly controlled gate** |
|---|---|
|  <br>$\mathcal{B}_\bullet \boxtimes \mathcal{K} \xrightarrow{G_\bullet} \mathcal{B}_\bullet \boxtimes \mathcal{K}$<br>$b : B \;\vdash\; \mathcal{K} \xrightarrow{G_b} \mathcal{K}$ |  <br>$\Box_B\mathcal{B}_\bullet \boxtimes \mathcal{K} \xrightarrow{\Box_B G_\bullet} \Box_B\mathcal{B}_\bullet \boxtimes \mathcal{K}$<br>$b : B \;\vdash\; \bigoplus_{b'B}\mathcal{K} \xrightarrow{\bigoplus_{b':B} G_{b'}} \bigoplus_{b'B}\mathcal{K}$ |

# Exmp: Deferred measurement principle – Proven by monadic effect logic.

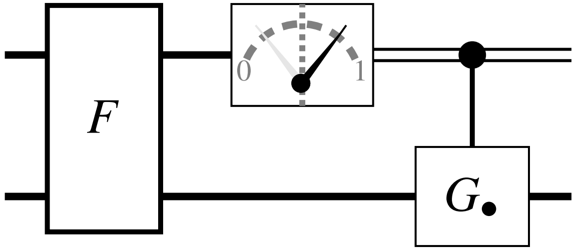
$$\begin{tikzcd}
\mathrm{Kl}\big(\square_B\big) \arrow[r, "\sim", "\delta^B \circ \square_B(-)"'] \arrow[rr, bend left, "\mathrm{id}"] & \mathrm{LType}_{B\square_B} \arrow[r, "\sim", "\varepsilon^{\square_B} \circ (-)"'] & \mathrm{Kl}\big(\square_B\big)
\end{tikzcd}$$

**Kleisli equivalence**

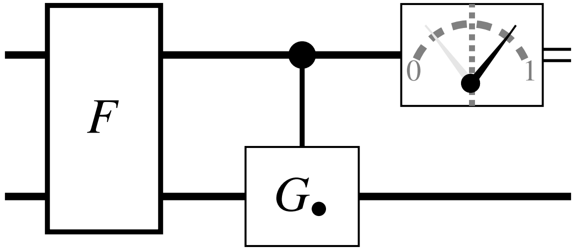
**$\square_B$-Kleisli morphisms** **$\square_B$-coalgebra homomorphisms** **$\square_B$-Kleisli morphisms**

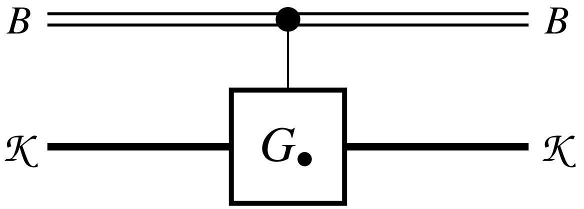
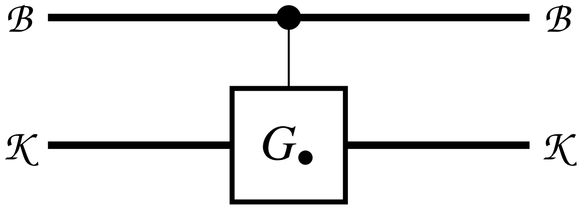
$$\square_B \mathcal{H}_\bullet \xrightarrow{F} \square_B \mathcal{H}_\bullet \xrightarrow{\varepsilon^{\square_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet \xrightarrow{G_\bullet} \mathcal{H}_\bullet \;\mapsto\; \square_B \mathcal{H}_\bullet \xrightarrow{\mathrm{diag}_B(G_\bullet) \circ F} \square_B \mathcal{H}_\bullet \;\mapsto\; \square_B \mathcal{H}_\bullet \xrightarrow{\mathrm{diag}_B(G_\bullet) \circ F} \square_B \mathcal{H}_\bullet \xrightarrow{\varepsilon^{\square_B}_{\mathcal{H}_\bullet}} \mathcal{H}_\bullet$$

**measurement-controlled quantum gate** **quantum-controlled quantum gate...** **...followed by measurement**



$\longleftrightarrow$ **Deferred Measurement Principle**



| **classically controlled gate** | **quantumly controlled gate** |
|---|---|
| <br>$\mathcal{B}_\bullet \boxtimes \mathcal{K} \xrightarrow{G_\bullet} \mathcal{B}_\bullet \boxtimes \mathcal{K}$<br>$b : B \;\vdash\; \mathcal{K} \xrightarrow{G_b} \mathcal{K}$ | <br>$\square_B \mathcal{B}_\bullet \boxtimes \mathcal{K} \xrightarrow{\square_B G_\bullet} \square_B \mathcal{B}_\bullet \boxtimes \mathcal{K}$<br>$b : B \;\vdash\; \bigoplus_{b' B} \mathcal{K} \xrightarrow{\bigoplus_{b' : B} G_{b'}} \bigoplus_{b' B} \mathcal{K}$ |

# The Quantum modality.

Also the *exponential modality* traditionally postulated in linear logic is an emergent effect in `dLHoTT`,

**linear randomization**

aka: stabilization/motivization

$$\begin{tikzcd}
\mathrm{Type} \arrow[r, shift left=2, "\Sigma^{\infty}_{+} \;:\; B \;\mapsto\; \star_{B}\mathbb{1}"] \arrow[r, phantom, "\perp"] & \mathrm{LType} \arrow[l, shift left=2, "\Omega^{\infty}"] \arrow[loop right, "!"]
\end{tikzcd}$$

**classical types**

**linear types**

**exponential modality**

## The Quantum modality.

Also the *exponential modality* traditionally postulated in linear logic
is an emergent effect in `dLHoTT`,
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 `dLHoTT`,
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 $\mathrm{QBit} = Q(\mathrm{Bit})$...

# The Quantum modality.

Also the *exponential modality* traditionally postulated in linear logic
is an emergent effect in `dLHoTT`,
as is the crucial *Quantum Modality*, not considered before:

**linear randomization**

aka: stabilization/motivization

**quantum modality**

$$\begin{tikzcd} \mathrm{Type} \arrow[loop left, "Q"] \arrow[rr, shift left=2, "\Sigma^{\infty}_{+} \,:\, B \,\mapsto\, \star_{B}\mathbb{1}"] \arrow[rr, phantom, "\perp"] && \mathrm{LType} \arrow[loop right, "!"] \arrow[ll, shift left=2, "\Omega^{\infty}"] \end{tikzcd}$$

**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:

$$\bigcirc_B\big(D \otimes D'\big) \;\simeq\; \big(\bigcirc_B D\big) \otimes D' \,, \qquad \text{☆}_B\big(D \otimes D'\big) \;\simeq\; \big(\text{☆}_B D\big) \otimes D'$$

There is a whole system of them:

$$\bigcirc_B \bigcirc_{B'} \;\simeq\; \bigcirc_{B \times B'}, \qquad \text{NB: } \bigcirc_B \bigcirc'_B \;\simeq\; \bigcirc_B \mathbb{1} \otimes \bigcirc'_B$$

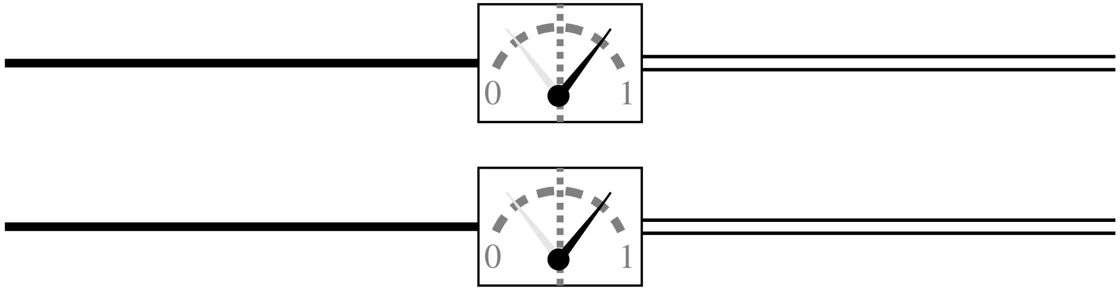
which under dynamic lifting (monadicity comparison functor)
gives the external tensor product of dependent linear types:

$$\begin{tikzcd}
\textbf{free $\bigcirc_B$-effect handlers in linear data types} & \mathrm{LType}_{\bigcirc_B} \arrow[d, hook, "K_{\oplus_B \mathbb{1}_B \otimes}", "\textbf{comparison functor}"'] & \overset{\overset{\bigcirc}{B \times B'}}{\curvearrowright} \\[-1.5em]
 & & \bigcirc_{B \times B'} \bigcirc_{B \times B'} \mathcal{H} \arrow[d, mapsto] \\
\textbf{$B$-dependent linear data types} & \mathrm{LType}_B & \big(\Box_B \mathbb{1}_B\big) \underset{\textbf{external tensor product}}{\boxtimes} \big(\Box_{B'} \mathbb{1}_{B'}\big) \otimes \mathcal{H}
\end{tikzcd}$$

# 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:



**type of a pair of coherent qbits**

**pair of measurements**

**type of collapsed qbits dependent on measured bits $b, b'$**

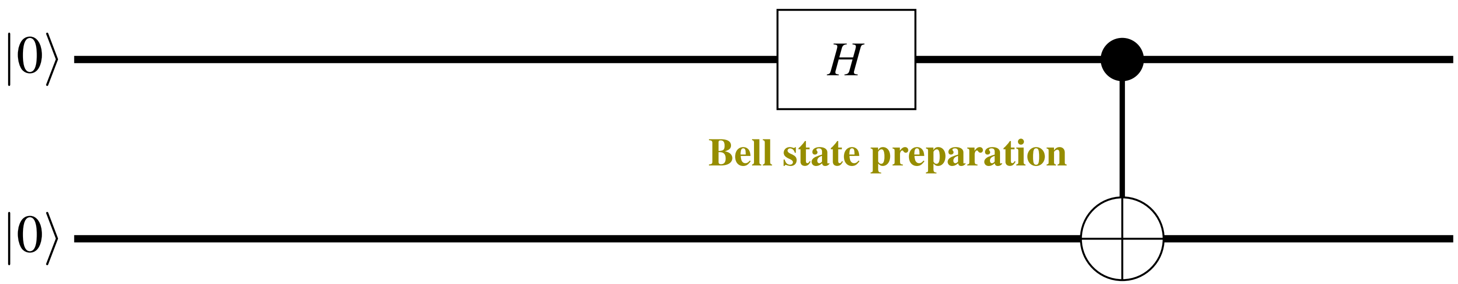
$$\Box_{\mathrm{Bit}^2}\big(\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet\big) \xrightarrow{\;\varepsilon_{\mathrm{Bit}^2}(\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet)\;} \mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet$$

**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 $\mathrm{LType}_{\mathrm{Bit}\times\mathrm{Bit}}$):

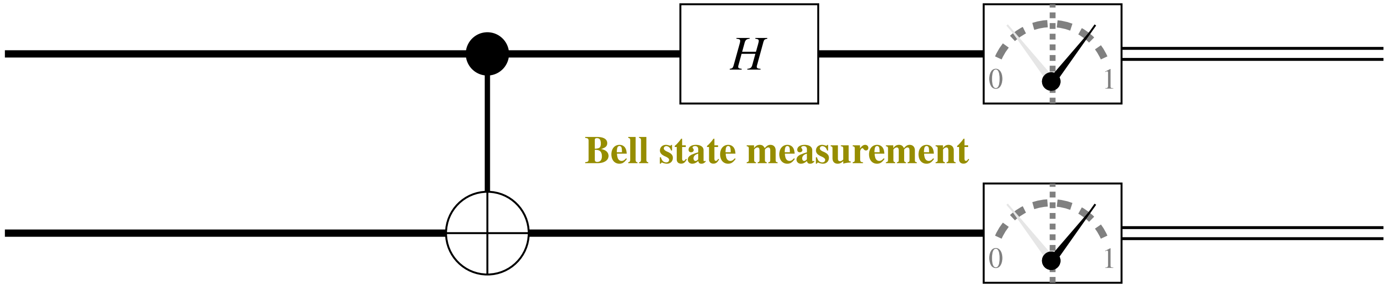
---



$$\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet \longrightarrow \big(\lozenge_{\mathrm{Bit}}\mathrm{QBit}_\bullet\big) \boxtimes \big(\lozenge_{\mathrm{Bit}}\mathrm{QBit}_\bullet\big) \simeq \square_{\mathrm{Bit}^2}\big(\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet\big) \longrightarrow \square_{\mathrm{Bit}^2}\big(\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet\big)$$

$$b, b' : \mathrm{Bit} \vdash \;\; \mathbb{C} \xrightarrow{\;1 \;\mapsto\; |0\rangle\otimes|0\rangle \;\mapsto\; \frac{1}{\sqrt{2}}\big(|0\rangle+|1\rangle\big)\otimes|0\rangle \;\mapsto\; \frac{1}{\sqrt{2}}\big(|0\rangle\otimes|0\rangle+|1\rangle\otimes|1\rangle\big)\;} \mathbb{C}^2\otimes\mathbb{C}^2$$

---

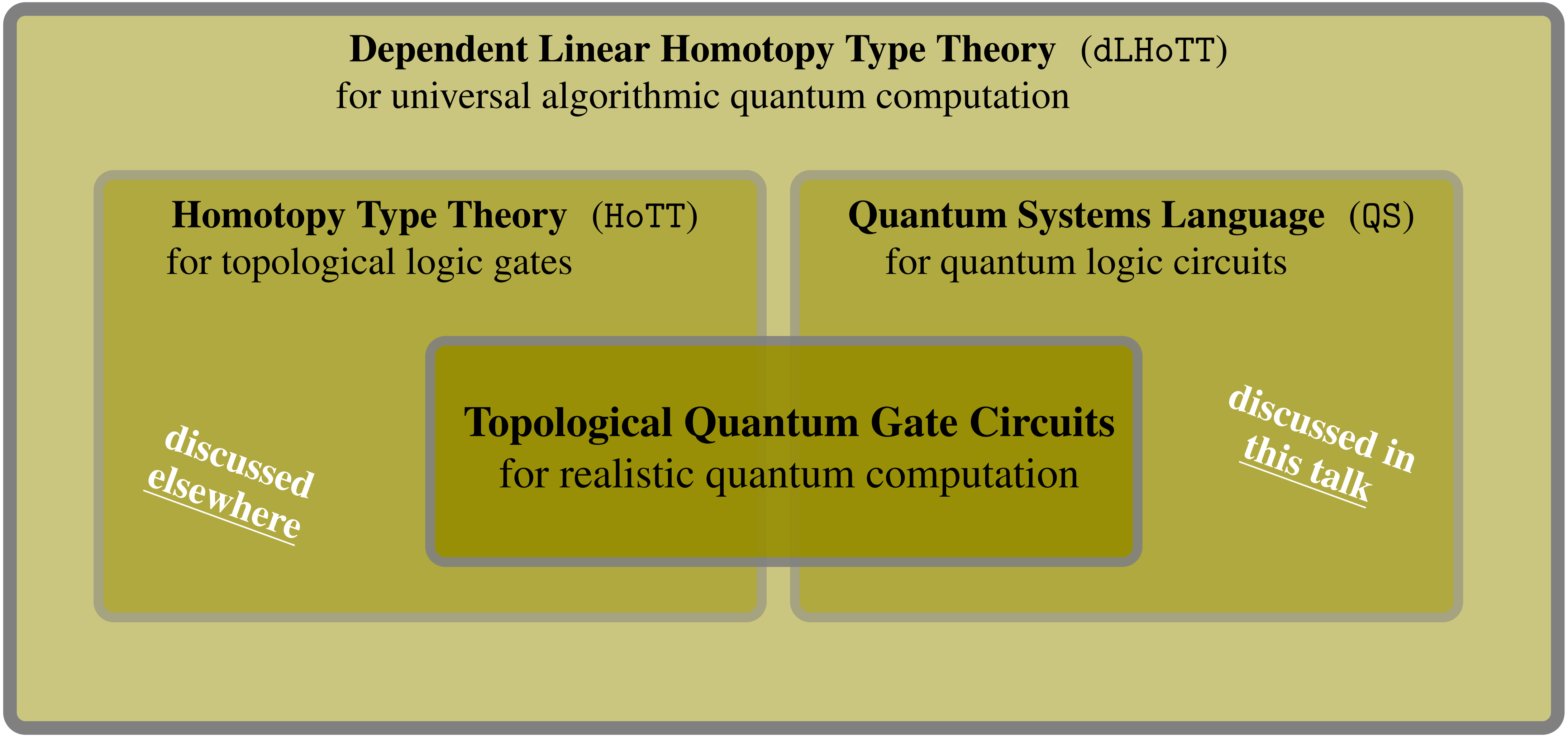


$$\square_{\mathrm{Bit}^2}\big(\mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet\big) \longrightarrow \mathrm{QBit}_\bullet \boxtimes \mathrm{QBit}_\bullet$$

$$b_1, b_2 : \mathrm{Bit} \quad \vdash \quad \mathbb{C}^2\otimes\mathbb{C}^2 \xrightarrow{\;\Sigma_{b'_1 b'_2}\, q_{b'_1 b'_2}\cdot|b'_1\rangle\otimes|b'_2\rangle \;\mapsto\; \big(q_{0,b_2}+(-1)^{b_1}\cdot q_{1,(1-b_2)}\big)\cdot|b_1\rangle\otimes|b_2\rangle\;} \mathbb{C}$$

# `QS` – Quantum Systems language @ CQTS

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



**Dependent Linear Homotopy Type Theory** (`dLHoTT`)
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 <u>elsewhere</u>***

***discussed in <u>this talk</u>***

**Thanks!**

Center for
Quantum&
Topological
Systems

# Quantum Data Types via Linear HoTT

presentation at:

**Workshop on Quantum Software @ QTML 2022**

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

slides and further pointers at: ncatlab.org/schreiber/show/QDataInLHoTT#QTML2022