Type theory and directed homotopy theory

Paige Randall North

The Ohio State University

3 December 2019

Directed type theory and homotopy theory

Goal

To develop a syntax and semantics of a directed type theory.

To develop a synthetic theory for reasoning about:

- Higher category theory
- Directed homotopy theory
  - Concurrent processes
  - Rewriting

Directed type theory and homotopy theory

Goal

To develop a syntax and semantics of a directed type theory.

To develop a synthetic theory for reasoning about:

- Higher category theory
- Directed homotopy theory
    - Concurrent processes
    - Rewriting

Syntactic synthetic theories and categorical synthetic theories

- homotopy type theory ↔ weak factorization systems
- directed homotopy type theory ↔ directed weak factorization systems

Both need to be developed.

Outline

Overview of homotopy theory and type theory

Overview of directedness

Two-sided weak factorization systems

Outline

Dependent type theory

Original purpose

- A foundation of mathematics that can be computer checked (Martin-Löf 1970s).

Dependent type theory

Original purpose

- A foundation of mathematics that can be computer checked (Martin-Löf 1970s).

Basic objects of the language

Dependent types:

$$b : B \vdash E(b) \ \textsc{type}$$

Dependent type theory

Original purpose

- A foundation of mathematics that can be computer checked (Martin-Löf 1970s).

Basic objects of the language

Dependent types:

$$b : B \vdash E(b) \ \text{TYPE} \qquad \rightsquigarrow \qquad E : B \to \ \text{TYPE} \qquad \begin{array}{c} E \\ \downarrow{\scriptstyle \pi} \\ B \end{array}$$

Dependent type theory

Original purpose

- A foundation of mathematics that can be computer checked (Martin-Löf 1970s).

Basic objects of the language

Dependent types:

$$b : B \vdash E(b) \ \textsc{type} \qquad \rightsquigarrow \qquad E : B \to \ \textsc{type} \qquad \begin{array}{c} E \\ \downarrow \pi \\ B \end{array}$$

Examples

- $n : \mathbb{N} \vdash \mathsf{List}_{\mathbb{Z}}(n) \ \textsc{type}$
- $n : \mathbb{N} \vdash \mathsf{isEven}(n) \ \textsc{type}$

Dependent type theory

Basic objects of the language

Dependent terms:

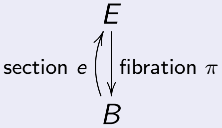
$$b : B \vdash e(b) : E(b)$$

Dependent type theory

Basic objects of the language

Dependent terms:

$$b : B \vdash e(b) : E(b) \qquad \rightsquigarrow$$



Dependent type theory

Basic objects of the language

Dependent terms:

$$b : B \vdash e(b) : E(b) \quad \rightsquigarrow \quad \begin{array}{c} E \\ \text{section } e \uparrow\!\!\downarrow \text{fibration } \pi \\ B \end{array}$$

Examples

- $n : \mathbb{N} \vdash \overline{0} : \mathsf{List}_{\mathbb{Z}}(n)$
- $n : \mathbb{N} \vdash \mathsf{isEven}(n)$ $\textsc{type}$ has no terms
- $n : \mathbb{N} \vdash [n] : \mathsf{isEven}(2n)$

Type formers

One postulates the existence of certain type formers, usually defined in the style of *inductive data types*.

Type formers

One postulates the existence of certain type formers, usually defined in the style of *inductive data types*.

Example: the natural numbers

First stipulate the type $\mathbb{N}$ exists:

$$\frac{}{\mathbb{N}\ \textsc{type}}$$

Then stipulate the 'canonical' terms in $\mathbb{N}$:

$$\frac{}{0 : \mathbb{N}} \qquad \frac{n : \mathbb{N}}{sn : \mathbb{N}}$$

Then stipulate how to get a term of a type depending on $\mathbb{N}$.

$$\frac{n : \mathbb{N} \vdash D(n)\ \textsc{type} \qquad \vdash z : D(0) \qquad n : \mathbb{N}, x : D(n) \vdash fx : D(sn)}{\begin{array}{c} n : \mathbb{N} \vdash d(n) : D(n) \qquad \vdash d(0) \equiv z : D(0) \\ n : \mathbb{N} \vdash fd(n) \equiv d(sn) : D(sn) \end{array}}$$

Type formers

One postulates the existence of certain type formers, usually defined in the style of *inductive data types*.

Example: the natural numbers

First stipulate the type $\mathbb{N}$ exists:

$$\frac{}{\mathbb{N}\ \textsc{type}}$$

Then stipulate the 'canonical' terms in $\mathbb{N}$:

$$\frac{}{0 : \mathbb{N}} \qquad \frac{n : \mathbb{N}}{sn : \mathbb{N}}$$

Then stipulate how to get a term of a type depending on $\mathbb{N}$.

$$\frac{n : \mathbb{N} \vdash D(n)\ \textsc{type} \qquad \vdash z : D(0) \qquad n : \mathbb{N}, x : D(n) \vdash fx : D(sn)}{\begin{array}{c} n : \mathbb{N} \vdash d(n) : D(n) \qquad \vdash d(0) \equiv z : D(0) \\ n : \mathbb{N} \vdash fd(n) \equiv d(sn) : D(sn) \end{array}}$$

Also define product, coproduct, functions, etc.

The surprising type former

Example: the identity type

First stipulate the type $\mathsf{Id}_A(a, b)$ exists:

$$\frac{A \text{ TYPE} \qquad a : A \qquad b : A}{\mathsf{Id}_A(a, b)}$$

Then stipulate the 'canonical' term:

$$\frac{a : A}{r_a : \mathsf{Id}_A(a, a)}$$

Then stipulate how to get a term of a type depending on $\mathsf{Id}_A(a, b)$.

$$\frac{A \text{ TYPE} \qquad a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash D(a, b, p) \text{ TYPE} \qquad a : A \vdash f(a) : D(a, a, r_a)}{a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash d(a, b, p) : D(a, b, p) \qquad a : A \vdash d(a) \equiv f(a) : D(a, a, r_a)}$$

The surprising type former

Example: the identity type

First stipulate the type $\mathrm{Id}_A(a,b)$ exists:

$$\frac{A \text{ TYPE} \qquad a : A \qquad b : A}{\mathrm{Id}_A(a,b)}$$

Then stipulate the 'canonical' term:

$$\frac{a : A}{r_a : \mathrm{Id}_A(a,a)}$$

Then stipulate how to get a term of a type depending on $\mathrm{Id}_A(a,b)$.

$$\frac{\begin{array}{c} A \text{ TYPE} \qquad a : A, b : A, p : \mathrm{Id}_A(a,b) \vdash D(a,b,p) \text{ TYPE} \\ a : A \vdash f(a) : D(a,a,r_a) \end{array}}{\begin{array}{c} a : A, b : A, p : \mathrm{Id}_A(a,b) \vdash d(a,b,p) : D(a,b,p) \\ a : A \vdash d(a) \equiv f(a) : D(a,a,r_a) \end{array}}$$

This makes equality *weaker* than usual set-theoretic equality.

The surprising type former

Example: the identity type

First stipulate the type $\mathsf{Id}_A(a,b)$ exists:

$$\frac{A \text{ TYPE} \qquad a : A \qquad b : A}{\mathsf{Id}_A(a,b)}$$

Then stipulate the 'canonical' term:

$$\frac{a : A}{r_a : \mathsf{Id}_A(a,a)}$$

Then stipulate how to get a term of a type depending on $\mathsf{Id}_A(a,b)$.

$$\frac{A \text{ TYPE} \qquad a : A, b : A, p : \mathsf{Id}_A(a,b) \vdash D(a,b,p) \text{ TYPE} \qquad a : A \vdash f(a) : D(a,a,r_a)}{a : A, b : A, p : \mathsf{Id}_A(a,b) \vdash d(a,b,p) : D(a,b,p) \qquad a : A \vdash d(a) \equiv f(a) : D(a,a,r_a)}$$

This makes equality *weaker* than usual set-theoretic equality.

- easier for a computer to handle

The surprising type former

Example: the identity type

First stipulate the type $\mathrm{Id}_A(a, b)$ exists:

$$\frac{A \text{ TYPE} \qquad a : A \qquad b : A}{\mathrm{Id}_A(a, b)}$$

Then stipulate the 'canonical' term:

$$\frac{a : A}{r_a : \mathrm{Id}_A(a, a)}$$

Then stipulate how to get a term of a type depending on $\mathrm{Id}_A(a, b)$.

$$\frac{A \text{ TYPE} \qquad a : A, b : A, p : \mathrm{Id}_A(a, b) \vdash D(a, b, p) \text{ TYPE} \qquad a : A \vdash f(a) : D(a, a, r_a)}{a : A, b : A, p : \mathrm{Id}_A(a, b) \vdash d(a, b, p) : D(a, b, p) \qquad a : A \vdash d(a) \equiv f(a) : D(a, a, r_a)}$$

This makes equality *weaker* than usual set-theoretic equality.

- easier for a computer to handle
- begins to look like homotopy theory

Categorical models of type theory

The identity type

A category $\mathcal{C}$ *has identity types* if for every object X of $\mathcal{C}$ there is a path object

$$X \xrightarrow{\eta} Id(X) \xrightarrow{\epsilon} X \times X$$

such that (roughly) the *mapping path space factorization* forms a wfs on $\mathcal{C}$.

Categorical models of type theory

The identity type

A category $\mathcal{C}$ *has identity types* if for every object X of $\mathcal{C}$ there is a path object

$$X \xrightarrow{\eta} Id(X) \xrightarrow{\epsilon} X \times X$$

such that (roughly) the *mapping path space factorization* forms a wfs on $\mathcal{C}$.

The mapping path space factorization

$$X \xrightarrow{\qquad\qquad f \qquad\qquad} Y$$

Categorical models of type theory

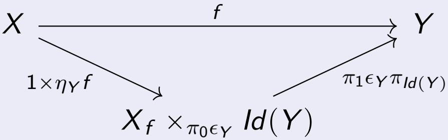
The identity type

A category $\mathcal{C}$ *has identity types* if for every object X of $\mathcal{C}$ there is a path object

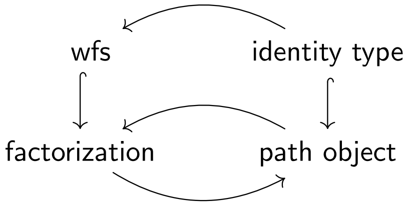
$$X \xrightarrow{\eta} Id(X) \xrightarrow{\epsilon} X \times X$$

such that (roughly) the *mapping path space factorization* forms a wfs on $\mathcal{C}$.

The mapping path space factorization

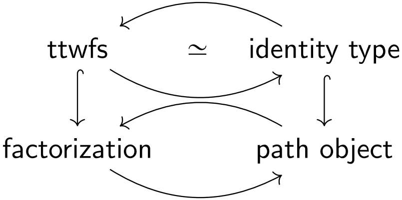


Type theoretic weak factorization systems



[1] N., Type theoretic weak factorization systems, PhD Dissertation, 2017

Type theoretic weak factorization systems



Theorem[1]

There is an equivalence between the category of identity types in $\mathcal{C}$ and the category of *type theoretic weak factorization systems* on $\mathcal{C}$.

Definition

A wfs on $\mathcal{C}$ always generates two classes of morphisms $(\mathcal{L}, \mathcal{R})$ of $\mathcal{C}$. A *type theoretic wfs* is a wfs such that

1. every morphism to the terminal object is in $\mathcal{R}$
2. $\mathcal{L}$ is stable under pullback along $\mathcal{R}$ (the *Frobenius condition*)

[1] N., Type theoretic weak factorization systems, PhD Dissertation, 2017

Outline

Overview of homotopy theory and type theory

Overview of directedness

Two-sided weak factorization systems

What does directed mean?

In topological spaces

There is an inversion

$$\begin{array}{ccccc} X & \xrightarrow{\eta} & X^{[0,1]} & \xrightarrow{\epsilon} & X \times X \\ \| & & \downarrow{\scriptstyle \iota} & & \downarrow{\scriptstyle \tau} \\ X & \xrightarrow{\eta} & X^{[0,1]} & \xrightarrow{\epsilon} & X \times X \end{array}$$

What does directed mean?

In topological spaces

There is an inversion

$$\begin{array}{ccccc} X & \xrightarrow{\eta} & X^{[0,1]} & \xrightarrow{\epsilon} & X \times X \\ \| & & \downarrow{\scriptstyle \iota} & & \downarrow{\scriptstyle \tau} \\ X & \xrightarrow{\eta} & X^{[0,1]} & \xrightarrow{\epsilon} & X \times X \end{array}$$

With identity types (or any path object in a wfs)

There is always an inversion

$$\begin{array}{ccccc} X & \xrightarrow{\eta} & Id(X) & \xrightarrow{\epsilon} & X \times X \\ \| & & \downarrow{\scriptstyle \iota} & & \downarrow{\scriptstyle \tau} \\ X & \xrightarrow{\eta} & Id(X) & \xrightarrow{\epsilon} & X \times X \end{array}$$

What does directed mean?

In topological spaces

There is an inversion

$$\begin{array}{ccccc} X & \xrightarrow{\eta} & X^{[0,1]} & \xrightarrow{\epsilon} & X \times X \\ \| & & \downarrow{\iota} & & \downarrow{\tau} \\ X & \xrightarrow{\eta} & X^{[0,1]} & \xrightarrow{\epsilon} & X \times X \end{array}$$

With identity types (or any path object in a wfs)

There is always an inversion

$$\begin{array}{ccccc} X & \xrightarrow{\eta} & Id(X) & \xrightarrow{\epsilon} & X \times X \\ \| & & \downarrow{\iota} & & \downarrow{\tau} \\ X & \xrightarrow{\eta} & Id(X) & \xrightarrow{\epsilon} & X \times X \end{array}$$

- Can think of these as *undirected* path objects

What does directed mean?

In topological spaces

There is an inversion

$$\begin{array}{ccccc} X & \xrightarrow{\eta} & X^{[0,1]} & \xrightarrow{\epsilon} & X \times X \\ \| & & \downarrow \iota & & \downarrow \tau \\ X & \xrightarrow{\eta} & X^{[0,1]} & \xrightarrow{\epsilon} & X \times X \end{array}$$

With identity types (or any path object in a wfs)

There is always an inversion

$$\begin{array}{ccccc} X & \xrightarrow{\eta} & Id(X) & \xrightarrow{\epsilon} & X \times X \\ \| & & \downarrow \iota & & \downarrow \tau \\ X & \xrightarrow{\eta} & Id(X) & \xrightarrow{\epsilon} & X \times X \end{array}$$

- Can think of these as *undirected* path objects
- Can we design a type former of *directed* paths that resembles identity types but without its inversion?

What does directed mean?

Theorem[2]

A functorial choice of path object

$$X \xrightarrow{\eta} Id(X) \xrightarrow{\epsilon} X \times X$$

for every object X constitutes an identity type in $\mathcal{C}$ if these path objects are

1. (left or right) transitive,
2. (left or right) connected,
3. symmetric.

[2] N., Type theoretic weak factorization systems, PhD Dissertation, 2017

What does directed mean?

Semantically

higher groupoids

What does directed mean?

Semantically

higher groupoids

higher categories
(undirected paths ⊆ directed paths)

What does directed mean?

Semantically

higher groupoids

higher categories
(undirected paths $\subseteq$ directed paths)

directed spaces
(directed paths $\subseteq$ undirected paths)

Directed spaces

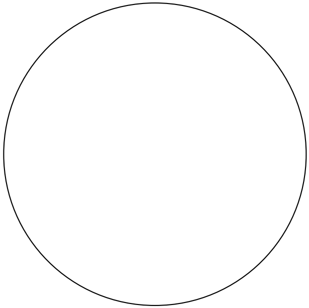
Rough definition

A directed space is a space together with a subset of its paths which are marked as *directed*.

Directed spaces

Rough definition

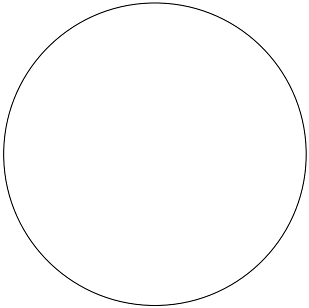
A directed space is a space together with a subset of its paths which are marked as *directed*.

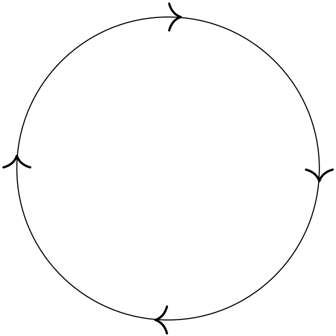


Directed spaces

Rough definition

A directed space is a space together with a subset of its paths which are marked as *directed*.



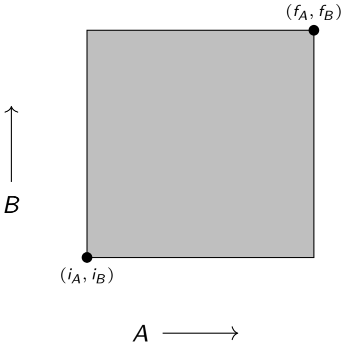


Application: concurrency

Concurrent processes can be represented by directed spaces.

Application: concurrency

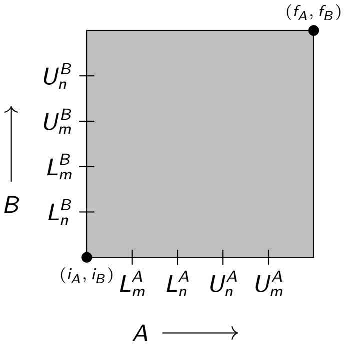
Concurrent processes can be represented by directed spaces.



- A, B are two processes

Application: concurrency

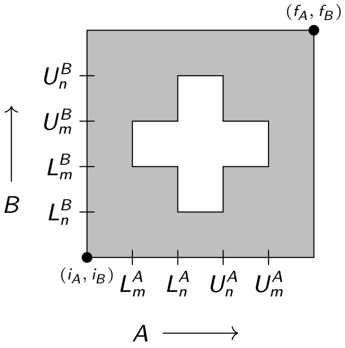
Concurrent processes can be represented by directed spaces.



- A, B are two processes
- m, n are two memory locations
- which can be locked (L) or unlocked (U) by each process

Application: concurrency

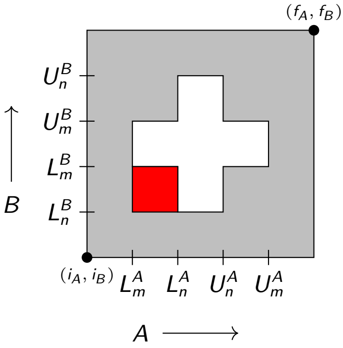
Concurrent processes can be represented by directed spaces.



- A, B are two processes
- m, n are two memory locations
- which can be locked (L) or unlocked (U) by each process

Application: concurrency

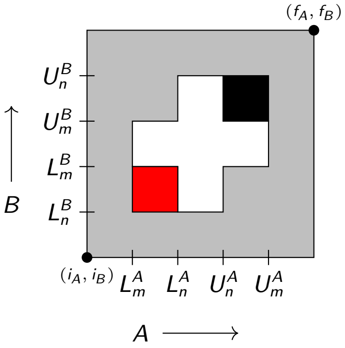
Concurrent processes can be represented by directed spaces.



- A, B are two processes
- m, n are two memory locations
- which can be locked (L) or unlocked (U) by each process

Application: concurrency

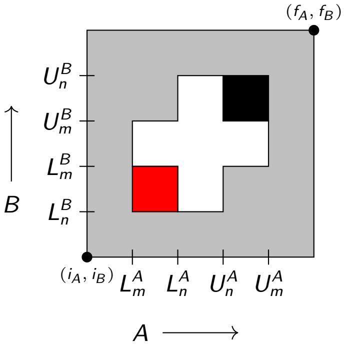
Concurrent processes can be represented by directed spaces.



- A, B are two processes
- m, n are two memory locations
- which can be locked (L) or unlocked (U) by each process

Application: concurrency

Concurrent processes can be represented by directed spaces.



- A, B are two processes
- m, n are two memory locations
- which can be locked (L) or unlocked (U) by each process

Fundamental questions:

- Which states are safe? (Predicate $S(x)$ on X^{op}.)
- Which states are reachable? (Predicate $R(x)$ on X.)
- How many traces are there up to homotopy? ($\hom(\bar{i}, \bar{f}) = 2$).

The homomorphism type[3]

$$\frac{T \ \text{TYPE}}{T^{\text{core}} \ \text{TYPE}} \qquad \frac{T \ \text{TYPE}}{T^{\text{op}} \ \text{TYPE}} \qquad \frac{T \ \text{TYPE} \qquad t : T^{\text{core}}}{it : T} \qquad \frac{T \ \text{TYPE} \qquad t : T^{\text{core}}}{i^{\text{op}} t : T^{\text{op}}}$$

$$\frac{T \ \text{TYPE} \qquad s : T^{\text{op}} \qquad t : T}{\hom_T(s, t) \ \text{TYPE}} \qquad \frac{T \ \text{TYPE} \qquad t : T^{\text{core}}}{1_t : \hom_T(i^{\text{op}} t, it) \ \text{TYPE}}$$

$$\frac{\begin{array}{c} T \ \text{TYPE} \qquad s : T^{\text{core}}, t : T, f : \hom_T(i^{\text{op}} s, t) \vdash D(f) \ \text{TYPE} \\ s : T^{\text{core}} \vdash d(s) : D(1_s) \end{array}}{\begin{array}{c} s : T^{\text{core}}, t : T, f : \hom_T(i^{\text{op}} s, t) \vdash e_R(d, f) : D(f) \\ s : T^{\text{core}} \vdash e_R(d, 1_s) \equiv d(s) : D(1_s) \end{array}}$$

$$\frac{\begin{array}{c} T \ \text{TYPE} \qquad s : T^{\text{op}}, t : T^{\text{core}}, f : \hom_T(s, it) \vdash D(f) \ \text{TYPE} \\ s : T^{\text{core}} \vdash d(s) : D(1_s) \end{array}}{\begin{array}{c} s : T^{\text{op}}, t : T^{\text{core}}, f : \hom_T(s, it) \vdash e_L(d, f) : D(f) \\ s : T^{\text{core}} \vdash e_L(d, 1_s) \equiv d(s) : D(1_s) \end{array}}$$

[3] N., Towards a directed homotopy type theory, MFPS 2019

Outline

Path objects in *Cat*

- Let $\cong$ denote the category consisting of one isomorphism.
- Let $\rightarrow$ denote the category consisting of one morphism.
- These produce two path objects of any category $\mathcal{C}$ which each generate wfs.

$$\mathcal{C} \to \mathcal{C}^{\cong} \to \mathcal{C} \times \mathcal{C} \qquad \mathcal{C} \to \mathcal{C}^{\rightarrow} \to \mathcal{C} \times \mathcal{C}$$

- The first forms an identity type in *Cat*.
- The second does not because it is not symmetric.
- There is a 'model' of the homomorphism type in *Cat* that captures the behavior of $\mathcal{C}^{\rightarrow}$, but the construction is very specific to *Cat*.

Path objects in *Cat*

- Let $\cong$ denote the category consisting of one isomorphism.
- Let $\rightarrow$ denote the category consisting of one morphism.
- These produce two path objects of any category $\mathcal{C}$ which each generate wfs.

$$\mathcal{C} \to \mathcal{C}^{\cong} \to \mathcal{C} \times \mathcal{C} \qquad \mathcal{C} \to \mathcal{C}^{\rightarrow} \to \mathcal{C} \times \mathcal{C}$$

- The first forms an identity type in *Cat*.
- The second does not because it is not symmetric.
- There is a 'model' of the homomorphism type in *Cat* that captures the behavior of $\mathcal{C}^{\rightarrow}$, but the construction is very specific to *Cat*.

Goal

To develop a general, categorical theory of such homomorphism types just like the one for identity types.

This will allow us to improve the syntax and use it to reason about directed phenomena.

Factorization from path object

How do we get factorizations from a path objects?

$$\mathcal{C} \xrightarrow{\eta} \mathcal{C}^{\rightarrow} \xrightarrow{\epsilon_0 \times \epsilon_1} \mathcal{C} \times \mathcal{C}$$

We factor through using the mapping path space:

$$\mathcal{C} \xrightarrow{\eta} \mathcal{C}_F \times_{\epsilon_0} \mathcal{D}^{\rightarrow} \xrightarrow{\epsilon_1} \mathcal{D}$$

Factorization from path object

How do we get factorizations from a path objects?

$$\mathcal{C} \xrightarrow{\eta} \mathcal{C}^{\rightarrow} \xrightarrow{\epsilon_0 \times \epsilon_1} \mathcal{C} \times \mathcal{C}$$

We factor through using the mapping path space:

$$\mathcal{C} \xrightarrow{\eta} \mathcal{C}{}_F \times_{\epsilon_0} \mathcal{D}^{\rightarrow} \xrightarrow{\epsilon_1} \mathcal{D}$$

But we could have factored through the other one:

$$\mathcal{C} \xrightarrow{\eta} \mathcal{D}^{\rightarrow}{}_{\epsilon_1} \times_F \mathcal{C} \xrightarrow{\epsilon_0} \mathcal{D}$$

In the case of identity types, this is resolved because the symmetry makes them equivalent.

In the directed case, we have a tale of two factorizations (wfs) which we want to see as part of the same structure.

Path object from factorization

We get a path object back from a factorization by factoring the diagonal of every object.

$$X \to Path(X) \to X \times X$$

Path object from factorization

We get a path object back from a factorization by factoring the diagonal of every object.

$$X \to Path(X) \to X \times X$$

In a weak factorization system, a commutative square whose left edge is a left factor and whose right edge is a right factor always has a lift.

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{\eta} & \mathcal{C}^{\to} \\ \downarrow{\scriptstyle \eta} & & \downarrow{\scriptstyle \epsilon} \\ \mathcal{C}^{\to} & \xrightarrow{\tau\epsilon} & \mathcal{C} \times \mathcal{C} \end{array}$$

Path object from factorization

We get a path object back from a factorization by factoring the diagonal of every object.

$$X \to Path(X) \to X \times X$$

In a weak factorization system, a commutative square whose left edge is a left factor and whose right edge is a right factor always has a lift.

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{\eta} & \mathcal{C}^{\to} \\ \downarrow{\scriptstyle \eta} & \nearrow & \downarrow{\scriptstyle \epsilon} \\ \mathcal{C}^{\to} & \xrightarrow{\tau\epsilon} & \mathcal{C} \times \mathcal{C} \end{array}$$

Path object from factorization

We get a path object back from a factorization by factoring the diagonal of every object.

$$X \to Path(X) \to X \times X$$

In a weak factorization system, a commutative square whose left edge is a left factor and whose right edge is a right factor always has a lift.

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{\eta} & \mathcal{C}^{\to} \\ \Big\downarrow{\scriptstyle \eta} & \nearrow & \Big\downarrow{\scriptstyle \epsilon} \\ \mathcal{C}^{\to} & \xrightarrow{\tau\epsilon} & \mathcal{C} \times \mathcal{C} \end{array}$$

We need to separate the two endpoints, so we think of factoring the diagonal as:

$$\begin{array}{ccc} & X & \\ {\scriptstyle 1}\swarrow & & \searrow{\scriptstyle 1} \\ X & & X \end{array} \quad \mapsto \quad \begin{array}{ccc} & X & \\ & \downarrow & \\ & Path(X) & \\ \swarrow & & \searrow \\ X & & X \end{array}$$

Two-sided factorization

Factorization on a category

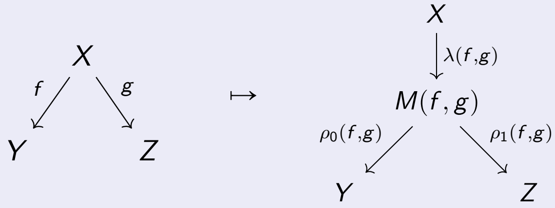
- a factorization of every morphism

$$X \xrightarrow{f} Y \qquad \mapsto \qquad X \xrightarrow{\lambda(f)} Mf \xrightarrow{\rho(f)} Y$$

- that extends to morphisms of morphisms

Two-sided factorization on a category

- a factorization of every span into a **sprout**



- that extends to morphisms of spans

Comma category

Notation

Write a span $\begin{array}{ccc} & X & \\ {}^{f}\swarrow & & \searrow^{g} \\ Y & & Z \end{array}$ as $f, g : X \to Y, Z$.

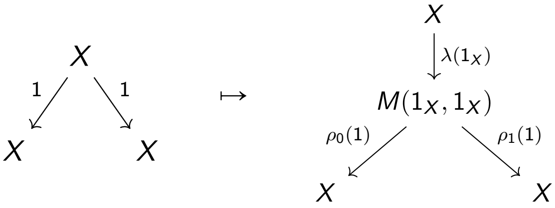
Then a factorization maps

$$X \xrightarrow{f,g} Y, Z \qquad \mapsto \qquad X \xrightarrow{\lambda(f,g)} M(f,g) \xrightarrow{\rho(f,g)} Y, Z$$

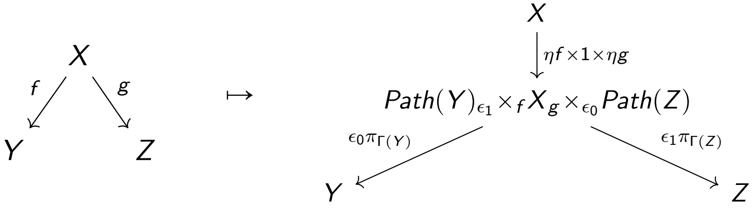
We're in the comma category $\Delta_{\mathcal{C}} \downarrow \mathcal{C} \times \mathcal{C}$.

Path objects

From any two-sided factorization, we obtain a path object for every object



Conversely, from a path object $X \xrightarrow{\eta} Path(X) \xrightarrow{\epsilon} X, X$ on each object, we obtain a two-sided factorization[4]

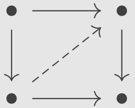


[4] Street, Fibrations and Yoneda's lemma in a 2-category, 1974

Lifting

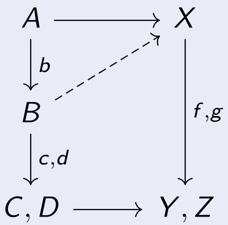
Lifting

A lifting problem is a commutative square, and a solution is a diagonal morphism making both triangles commute.



Two-sided lifting

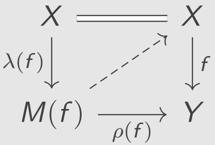
A sprout $A \xrightarrow{b} B \xrightarrow{c,d} C, D$ **lifts** against a span $X \xrightarrow{f,g} Y, Z$ if for any commutative diagram of solid arrows, there is a dashed arrow making the whole diagram commute.



Two-sided fibrations

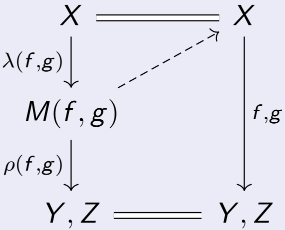
Fibrations.

Given a factorization, a **fibration** is a morphism $f : X \to Y$ for which there is a lift in



Two-sided fibrations

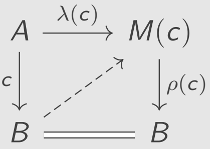
Given a two-sided factorization, a **two-sided fibration** is a span $f, g : X \to Y, Z$ for which there is a lift in



Rooted cofibrations

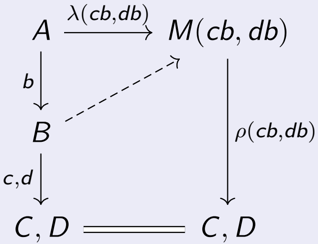
Cofibrations

Given a factorization, a **cofibration** is a morphism $c : A \to B$ for which there is a lift in



Rooted cofibrations

Given a two-sided factorization, a **rooted cofibration** is a sprout $A \xrightarrow{b} B \xrightarrow{c,d} C, D$ for which there is a lift in



Two-sided weak factorization systems

Weak factorization system

A factorization (λ, ρ) such that $\lambda(f)$ is a cofibration and $\rho(f)$ is a fibration for each morphism f

Two-sided weak factorization system

A two-sided factorization (λ, ρ) such that the span $\rho(f, g)$ is a two-sided fibration and the sprout in green is a cofibration for each span (f, g).

$$
\begin{array}{ccccc}
X & = & X & = & X \\
\downarrow{\scriptstyle \lambda(f,!)} & & \downarrow{\scriptstyle \lambda(f,g)} & & \downarrow{\scriptstyle \lambda(!,g)} \\
M(f,!) & \xleftarrow{M(1,1,!)} & M(f,g) & \xrightarrow{M(1,!,1)} & M(!,g) \\
\downarrow{\scriptstyle \rho(f,!)} & & \downarrow{\scriptstyle \rho(f,g)} & & \downarrow{\scriptstyle \rho(!,g)} \\
Y,* & \xleftarrow{1,!} & Y,Z & \xrightarrow{!,1} & *,Z
\end{array}
$$

Two-sided weak factorization systems

Theorem[5]

In a weak factorization system, the cofibrations are exactly the morphisms with the left lifting property against the fibrations and vice versa.

Theorem

In a two-sided weak factorization system, the rooted cofibrations are exactly the morphisms with the left lifting property against the two-sided fibrations and vice versa.

[5] Rosický and Tholen, Lax factorization algebras, 2002

Two weak factorization systems

Proposition

Consider a 2swfs $(\lambda, \rho_0, \rho_1)$ on a category with a terminal object. This produces two weak factorization systems: a **future** wfs whose underlying factorization is given by

$$X \xrightarrow{f} Y \qquad \mapsto \qquad X \xrightarrow{\lambda(!,f)} M(!,f) \xrightarrow{\rho_1(!,f)} Y$$

and a **past** wfs whose underlying factorization is given by

$$X \xrightarrow{f} Y \qquad \mapsto \qquad X \xrightarrow{\lambda(f,!)} M(f,!) \xrightarrow{\rho_0(f,!)} Y$$

Proposition

Consider a two-sided fibration $f, g : X \to Y, Z$ in a 2swfs. Then f is a past fibration and g is a future fibration.

The example in *Cat*

There is a 2swfs in *Cat* given by the factorization

$$\begin{array}{ccc} & \mathcal{C} & \\ {\scriptstyle F}\swarrow & & \searrow{\scriptstyle G} \\ \mathcal{D} & & \mathcal{E}\end{array} \quad\mapsto\quad \begin{array}{ccc} & \mathcal{C} & \\ & \downarrow{\scriptstyle \mathcal{D}^{!}F\times 1\times \mathcal{E}^{!}G} & \\ & \mathcal{D}^{\to}{}_{\mathrm{cod}}\times_{F}\mathcal{C}{}_{G}\times_{\mathrm{dom}}\mathcal{E}^{\to} & \\ {\scriptstyle \mathrm{dom}_{\mathcal{D}}}\swarrow & & \searrow{\scriptstyle \mathrm{cod}_{\mathcal{E}}} \\ \mathcal{D} & & \mathcal{E}\end{array}$$

- The past fibrations contain the Grothendieck fibrations
- The future fibrations contain the Grothendieck opfibrations
- The two-sided fibrations contain the (Grothendieck) two-sided fibrations [6]

[6] Street, Fibrations and Yoneda's lemma in a 2-category, 1974

2SWFSs from path objects

We want to understand which 2swfs's arise from path objects.

First, we characterize those path objects which give rise to 2swfs.

Theorem

Consider a choice of path objects $X \to \Gamma(X) \to X, X$. Then the factorization that sends $f : X \to Y$ to $X \to X \times_Y \Gamma(Y) \to Y$ underlies a weak factorization system if and only if Γ is left transitive and left connected.

Theorem[7]

Consider a choice of path objects $X \to \Gamma(X) \to X, X$. Then the factorization that sends $f, g : X \to Y, Z$ to $X \to \Gamma(Y) \times_Y X \times_Z \Gamma(Z) \to Y, Z$ is a two-sided weak factorization system if and only if Γ it is left transitive, right transitive, left connected, and right connected.

[7] N., Type theoretic weak factorization systems, PhD Dissertation, 2017

Type-theoretic 2SWFSs

Theorem[8]

The following are equivalent for a wfs:

- it is generated by a left transitive, left connected, and symmetric choice of path objects $X \to \Gamma(X) \to X, X$.
- it is type-theoretic: (1) all morphisms to the terminal object are fibrations and (2) the Frobenius condition, that cofibrations are stable under pullback along fibrations, holds

Fibrant object in a 2swfs

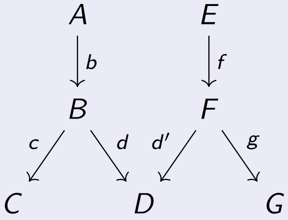
An object X such that $!, ! : X \to *, *$ is a two-sided fibration.

[8] N., Type theoretic weak factorization systems, PhD Dissertation, 2017

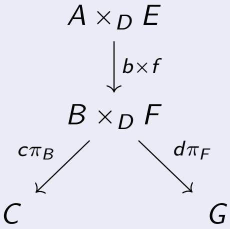
Type-theoretic 2SWFSs

Two-sided Frobenius condition.

The two-sided Frobenius condition holds when for any 'composable' two rooted cofibrations where db is a future fibration and $d'f$ is a past fibration,



the 'composite' is a cofibration.



Type-theoretic 2SWFSs

Theorem[9]

The following are equivalent for a wfs:

- it is generated by a left transitive, left connected, and symmetric choice of path objects $X \to \Gamma(X) \to X, X$.
- it is type-theoretic: (1) all objects are fibrant and (2) the Frobenius condition, that cofibrations are stable under pullback along fibrations, holds

Theorem

The following are equivalent for a 2swfs:

- it is generated by a left transitive, right transitive, left connected, right connected, choice of path objects $X \to \Gamma(X) \to X, X$.
- it is type-theoretic: (1) all objects are fibrant and (2) the two-sided Frobenius condition holds.

[9] N., Type theoretic weak factorization systems, PhD Dissertation, 2017

Examples

- In *Cat*, $\mathcal{C}^{\rightarrow}$
- In simplicial sets, free internal category on X^{Δ^1}
- In cubical sets with connections, free internal category on $X^{\Box^1}$
- In d-spaces (Grandis 2003), Moore paths $\Gamma(X)$

Summary

We now have

- a syntactic synthetic theory of direction and
- a categorical synthetic theory of direction
- which behave similarly.

We need

- to formalize the connection between the two,
- to get rid of the op and core operations on types using a modal type theory à la Licata-Riley-Shulman.

Thank you!