Directed weak factorization systems for type theory

Paige Randall North

The Ohio State University

2 October 2019

Directed type theory

Goal

To develop a directed type theory.

To develop a synthetic theory for reasoning about:

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

Directed type theory

Goal

To develop 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 $\leftrightarrow$ weak factorization systems
- directed homotopy type theory $\leftrightarrow$ directed weak factorization systems

Both need to be developed.

Outline

Overview of directedness

Directed homotopy theory

Syntax for a directed homotopy type theory

Semantics in *Cat*

Two-sided weak factorization systems

Outline

Overview of directedness

Directed homotopy theory

Syntax for a directed homotopy type theory

Semantics in *Cat*

Two-sided weak factorization systems

What does directed mean?

Syntactically

Martin-Löf's identity type is symmetric/undirected since for any type T, and terms $a, b : T$, there is a function

$$i : \mathrm{Id}_T(a, b) \to \mathrm{Id}_T(b, a)$$

so that any *path* $p : \mathrm{Id}_T(a, b)$ can be *inverted* to obtain a path $ip : \mathrm{Id}_T(b, a)$.

What does directed mean?

Syntactically

Martin-Löf's identity type is symmetric/undirected since for any type T, and terms $a, b : T$, there is a function

$$i : \mathrm{Id}_T(a, b) \to \mathrm{Id}_T(b, a)$$

so that any *path* $p : \mathrm{Id}_T(a, b)$ can be *inverted* to obtain a path $ip : \mathrm{Id}_T(b, a)$.

- Can think of these terms as *undirected* paths

What does directed mean?

Syntactically

Martin-Löf's identity type is symmetric/undirected since for any type T, and terms $a, b : T$, there is a function

$$i : \mathrm{Id}_T(a, b) \to \mathrm{Id}_T(b, a)$$

so that any *path* $p : \mathrm{Id}_T(a, b)$ can be *inverted* to obtain a path $ip : \mathrm{Id}_T(b, a)$.

- Can think of these terms as *undirected* paths
- Can we design a type former of *directed* paths that resembles Id but without its inversion operation i?

What does directed mean?

Semantically: Theorem

Consider a cartesian closed category $\mathcal{C}$, and a reflexive graph **Id** in $[\mathcal{C}, \mathcal{C}]$:

$$1_{\mathcal{C}} \xrightarrow{r} Id \xrightarrow{\epsilon_0 \times \epsilon_1} 1_{\mathcal{C}} \times 1_{\mathcal{C}}$$

The following are equivalent.

- **Id** models identity types.
- The mapping path space factorization

$$X \xrightarrow{f} Y \quad \rightsquigarrow \quad X \xrightarrow{1 \times rf} X \times_Y Id(Y) \xrightarrow{\epsilon_1} Y$$

  underlies a weak factorization system on $\mathcal{C}$ where all red (resp. blue) maps are in the left (resp. right) class.
- **Id** is
  1. transitive,
  2. connected,
  3. symmetric.

What does directed mean?

Semantically

higher groupoids

What does directed mean?

Semantically

higher groupoids

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

What does directed mean?

Semantically

higher groupoids

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

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

Outline

Overview of directedness

Directed homotopy theory

Syntax for a directed homotopy type theory

Semantics in *Cat*

Two-sided weak factorization systems

Directed spaces

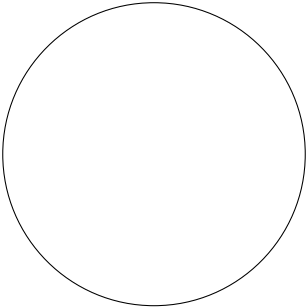
Rough definition

A space together with a subset of its paths that are marked as 'directed'

Directed spaces

Rough definition

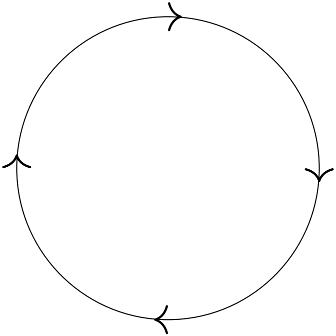
A space together with a subset of its paths that are marked as ‘directed’



Directed spaces

Rough definition

A space together with a subset of its paths that are marked as ‘directed’

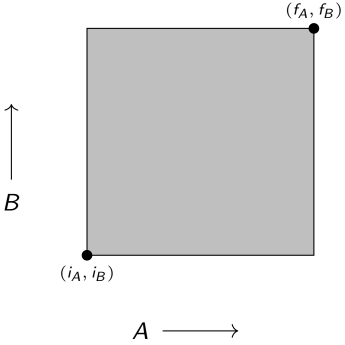


Application: concurrency

Concurrent processes can be represented by directed spaces.

Application: concurrency

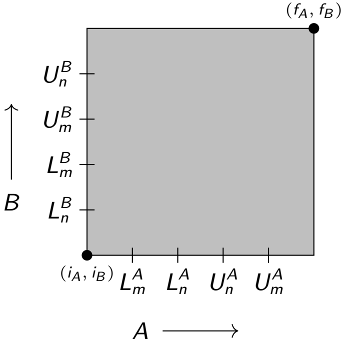
Concurrent processes can be represented by directed spaces.



- A, B are two processes

Application: concurrency

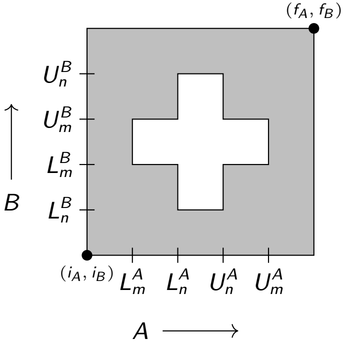
Concurrent processes can be represented by directed spaces.



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

Application: concurrency

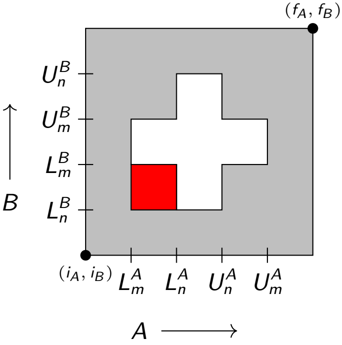
Concurrent processes can be represented by directed spaces.



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

Application: concurrency

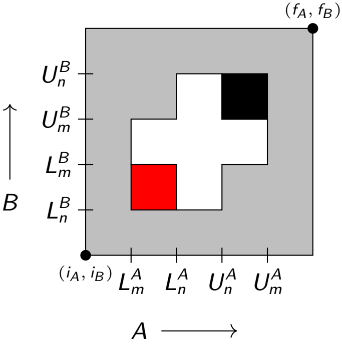
Concurrent processes can be represented by directed spaces.



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

Application: concurrency

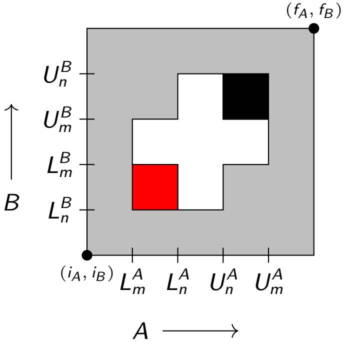
Concurrent processes can be represented by directed spaces.



- 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.)

Directedness

Semantically

higher groupoids

Directedness

Semantically

higher groupoids

higher categories
(undirected paths ⊆ directed paths)

Directedness

Semantically

higher groupoids

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

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

Outline

Criteria

- Directed paths are introduced as terms of a type former, hom, to be added to Martin-Löf type theory
- Transport along terms of hom
- Independence of hom and Id

Rules for hom: core and op

$$\frac{T \ \text{TYPE}}{T^{\text{core}} \ \text{TYPE}}$$

$$\frac{T \ \text{TYPE}}{T^{\text{op}} \ \text{TYPE}}$$

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

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

Rules for hom: formation

Id formation

$$\frac{T \;\; \text{TYPE} \qquad s : T \qquad t : T}{\mathsf{Id}_T(s,t) \;\; \text{TYPE}}$$

hom formation

$$\frac{T \;\; \text{TYPE} \qquad s : T^{\mathsf{op}} \qquad t : T}{\mathsf{hom}_T(s,t) \;\; \text{TYPE}}$$

Rules for hom: introduction

Id introduction

$$\frac{T \ \ \textsc{type} \qquad t : T}{r_t : \mathrm{Id}_T(t,t) \ \ \textsc{type}}$$

hom formation

$$\frac{T \ \ \textsc{type} \qquad t : T^{\mathrm{core}}}{1_t : \mathrm{hom}_T(i^{\mathrm{op}} t, it) \ \ \textsc{type}}$$

Rules for hom: right elimination and computation

Id elimination and computation

$$\frac{T \quad \text{TYPE} \qquad s : T, t : T, f : \mathsf{Id}_T(s,t) \vdash D(f) \quad \text{TYPE} \qquad s : T \vdash d(s) : D(r_s)}{\begin{array}{c} s : T, t : T, f : \mathsf{Id}_T(s,t) \vdash j(d,f) : D(f) \\ s : T \vdash j(d, r_s) \equiv d(s) : D(r_s) \end{array}}$$

hom right elimination and computation

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

Rules for hom: left elimination and computation

Id elimination and computation

$$\frac{T \ \text{TYPE} \qquad s : T, t : T, f : \mathrm{Id}_T(s,t) \vdash D(f) \ \text{TYPE} \qquad s : T \vdash d(s) : D(r_s)}{\begin{array}{c} s : T, t : T, f : \mathrm{Id}_T(s,t) \vdash j(d,f) : D(f) \\ s : T \vdash j(d, r_s) \equiv d(s) : D(r_s) \end{array}}$$

hom left elimination and computation

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

Outline

The interpretation

- Using the framework of comprehension categories
- Dependent types are represented by functors $T : \Gamma \to Cat$.
- Dependent terms are represented by natural transformations

$$\Gamma \underset{T}{\overset{*}{\rightrightarrows}} Cat \quad (\Downarrow t)$$

where $* : \Gamma \to Cat$ is the functor which takes everything to the one-object category.

- Context extension is represented by the Grothendieck construction taking each functor $T : \Gamma \to Cat$ to the Grothendieck opfibration

$$\pi_\Gamma : \int_\Gamma T \to \Gamma.$$

Interpreting core and op in the empty context

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

For any category T,

- $T^{\text{core}} := \text{ob}(T)$
- $T^{\text{op}} := T^{\text{op}}$
- $i : T^{\text{core}} \to T$ and $i^{\text{op}} : T^{\text{core}} \to T^{\text{op}}$ are the identity on objects.

Interpreting hom formation and introduction

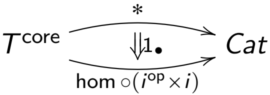
$$\frac{T \ \ \textsc{type} \qquad s : T^{\mathrm{op}} \qquad t : T}{\hom_T(s,t) \ \ \textsc{type}} \qquad\qquad \frac{T \ \ \textsc{type} \qquad t : T^{\mathrm{core}}}{1_t : \hom_T(i^{\mathrm{op}}t, it) \ \ \textsc{type}}$$

For any category T,

- Take the functor

$$\hom : T^{\mathrm{op}} \times T \to \mathit{Set} \hookrightarrow \mathit{Cat}.$$

- Take the natural transformation



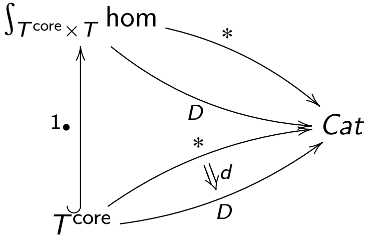
where each component $1_t : * \to \hom(t,t)$ picks out the identity morphism of t.

Interpreting right hom elimination and computation

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

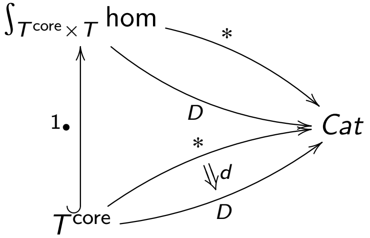
Interpreting right hom elimination and computation

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



Interpreting right hom elimination and computation

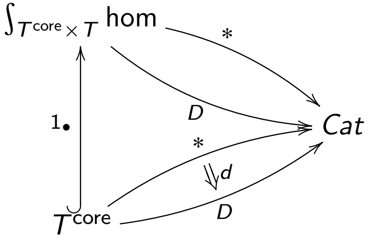
$$\frac{T \ \text{TYPE} \qquad s : T^{\text{core}}, t : T, f : \hom_T(i^{\text{op}} s, t) \vdash D(f) \ \text{TYPE} \qquad s : T^{\text{core}} \vdash d(s) : D(1_s)}{\begin{array}{c} s : T^{\text{core}}, t : T, f : \hom_T(i^{\text{op}} s, t) \vdash e_R(d, f) : D(f) \\ s : T^{\text{core}} \vdash e_R(d,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



- Use the fact that the subcategory T^{core} is coreflective:

Interpreting right hom elimination and computation

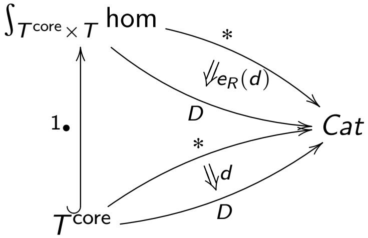
$$\frac{T \;\; \text{TYPE} \qquad s : T^{\text{core}}, t : T, f : \hom_T(i^{\text{op}} s, t) \vdash D(f) \;\; \text{TYPE} \qquad s : T^{\text{core}} \vdash d(s) : D(1_s)}{s : T^{\text{core}}, t : T, f : \hom_T(i^{\text{op}} s, t) \vdash e_R(d, f) : D(f) \qquad s : T^{\text{core}} \vdash e_R(d, 1_s) \equiv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



- Use the fact that the subcategory T^{core} is coreflective:
  - for every $(s, t, f) \in \int_{T^{\text{core}} \times T} \hom$ there is a unique morphism $(1_s, f) : (s, s, 1_s) \to (s, t, f)$ with domain in T^{core}

Interpreting right hom elimination and computation

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



- Use the fact that the subcategory T^{core} is coreflective:
  - for every $(s, t, f) \in \int_{T^{\text{core}} \times T} \hom$ there is a unique morphism $(1_s, f) : (s, s, 1_s) \to (s, t, f)$ with domain in T^{core}
- Set $e_R(d)_{(s,t,f)} := D(1_s, f) d_{(s,s,1_s)}$

Interpreting left hom elimination and computation

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

- Replace T by T^{op} and apply right hom elimination and computation.

A homotopical perspective

While the homotopy theory of isomorphisms in categories

$$\mathcal{C} \to \mathcal{C}^{(\cong)} \to \mathcal{C} \times \mathcal{C}$$

provides an interpretation of Martin-Löf's identity type, the homotopy theory of morphisms in categories

$$\mathcal{C} \to \mathcal{C}^{(\to)} \to \mathcal{C} \times \mathcal{C}$$

provides an interpretation of this hom former.

The weak factorization system

- Let $(\cong)$ denote the category with two objects and one isomorphism between them.
- Let $(\rightarrow)$ denote the category with two objects and one morphism between them.
- Then factorize the codiagonal of the one-point category in two ways

$$* + * \;\rightarrow\; (\cong) \;\rightarrow\; * \qquad * + * \;\rightarrow\; (\rightarrow) \;\rightarrow\; *$$

- which produces a factorization of any diagonal (i.e. reflexive graphs in $[Cat, Cat]$) in two ways which each generate weak factorization systems.

$$\mathcal{C} \rightarrow \mathcal{C}^{(\cong)} \rightarrow \mathcal{C} \times \mathcal{C} \qquad \mathcal{C} \rightarrow \mathcal{C}^{(\rightarrow)} \rightarrow \mathcal{C} \times \mathcal{C}$$

- The first gives an interpretation of the Id type in Cat.
- The second underlies this interpretation of the hom type in Cat.

The weak factorization system continued

- The right class of this weak factorization system are those functors $p : E \to B$ which have the enriched right lifting property

$$\begin{array}{ccc} * & \longrightarrow & E \\ {\scriptstyle \mathrm{DOM}}\big\downarrow & \nearrow & \big\downarrow{\scriptstyle p} \\ (\rightarrow) & \longrightarrow & B \end{array}$$

- so all Grothendieck opfibrations (dependent projections) are in the right class.
- The functor $1_\bullet : T^{\mathrm{core}} \hookrightarrow \int_{T^{\mathrm{core}} \times T} \mathrm{hom}$ is the left part of the factorization of
$$i : T^{\mathrm{core}} \hookrightarrow T.$$
- Then the right hom elimination and computation rules arise from the weak factorization system.

$$\begin{array}{ccc} T^{\mathrm{core}} & \xrightarrow{\;d\;} & \int_{\int_{T^{\mathrm{core}} \times T} \mathrm{hom}} D \\ {\scriptstyle 1_\bullet}\big\downarrow & \overset{e_R(d)}{\nearrow} & \big\downarrow{\scriptstyle \pi} \\ \int_{T^{\mathrm{core}} \times T} \mathrm{hom} & = & \int_{T^{\mathrm{core}} \times T} \mathrm{hom} \end{array}$$

Outline

Semantics in *Cat*

Goal

To generalize the construction of this model of the homomorphism type to other categories $\mathcal{C}$ with reflexive graphs in $[\mathcal{C}, \mathcal{C}]$.

$$1_{\mathcal{C}} \to hom \to 1_{\mathcal{C}} \times 1_{\mathcal{C}}$$

Semantics in *Cat*

Goal

To generalize the construction of this model of the homomorphism type to other categories $\mathcal{C}$ with reflexive graphs in $[\mathcal{C}, \mathcal{C}]$.

$$1_{\mathcal{C}} \to \mathit{hom} \to 1_{\mathcal{C}} \times 1_{\mathcal{C}}$$

Hurdles

- If we model dependent types by right-hand maps $\mathcal{C} \to \Gamma$, there's no good way to model the operation $(\Gamma \vdash \mathcal{C}) \mapsto (\Gamma \vdash \mathcal{C}^{\mathsf{op}})$.
  - Old solution: we model dependent types by functors $\Gamma \to \mathit{Cat}$.
- The second factorization generates a weak factorization system, but $\mathcal{C}^{\to} \to \mathcal{C} \times \mathcal{C}$ is not a right-hand map there.
  - Old solution: consider the twisted arrow category $\int_{\mathcal{C}^{\mathsf{op}} \times \mathcal{C}} \mathrm{hom} \to \mathcal{C}^{\mathsf{op}} \times \mathcal{C}$
- Generally, we rely too much on properties of *Cat*. A synthetic categorical theory of direction should be simpler.

WFS from graph

How do we get weak factorization systems from a functorial reflexive graph (Id-type) on a category?

$$X \xrightarrow{\eta} \Gamma(X) \xrightarrow{\epsilon_0 \times \epsilon_1} X \times X$$

WFS from graph

How do we get weak factorization systems from a functorial reflexive graph (Id-type) on a category?

$$X \xrightarrow{\eta} \Gamma(X) \xrightarrow{\epsilon_0 \times \epsilon_1} X \times X$$

First, we need to factor any map $f : X \to Y$. We do this using the mapping path space:

$$X \xrightarrow{\eta} X_f \times_{\epsilon_0} \Gamma(Y) \xrightarrow{\epsilon_1} Y$$

But this introduces an asymmetry.

In models of identity types, this is resolved because a 'symmetry' involution on $\Gamma(X)$ is required that preserves η and switches ϵ_0 and ϵ_1.

In the directed case (e.g. $\mathcal{C}^{\rightarrow}$), this isn't resolved and we get two factorizations underlying two weak factorization systems.

$$X \xrightarrow{\eta} X_f \times_{\epsilon_0} \Gamma(Y) \xrightarrow{\epsilon_1} Y \qquad X \xrightarrow{\eta} \Gamma(Y)_{\epsilon_1} \times_f X \xrightarrow{\epsilon_0} Y$$

We want to see these two wfs's as part of the same structure.

Graph from WFS

How do we get a functorial reflexive graph (Id-type) back from a wfs on a category?

We factor the diagonal of every object.

$$X \xrightarrow{\lambda(\Delta_X)} M(\Delta_X) \xrightarrow{\rho(\Delta_X)} X \times X$$

Graph from WFS

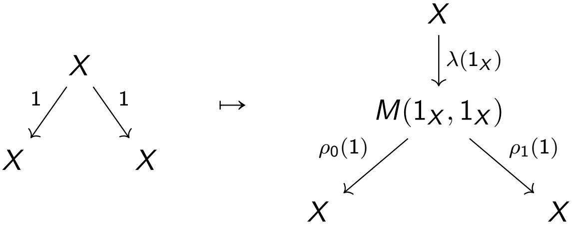
How do we get a functorial reflexive graph (Id-type) back from a wfs on a category?

We factor the diagonal of every object.

$$X \xrightarrow{\lambda(\Delta_X)} M(\Delta_X) \xrightarrow{\rho(\Delta_X)} X \times X$$

In our new notion of directed weak factorization, we need to preserve this ability.

We can think of this as the following operation.



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

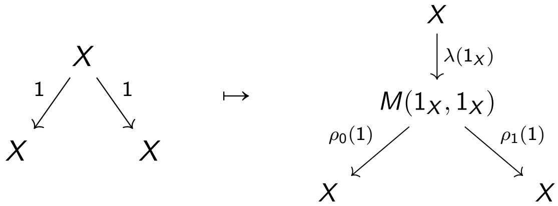
- a factorization of every span into a **sprout**

$$Y \xleftarrow{f} X \xrightarrow{g} Z \qquad \mapsto \qquad X \xrightarrow{\lambda(f,g)} M(f,g), \quad Y \xleftarrow{\rho_0(f,g)} M(f,g) \xrightarrow{\rho_1(f,g)} Z$$

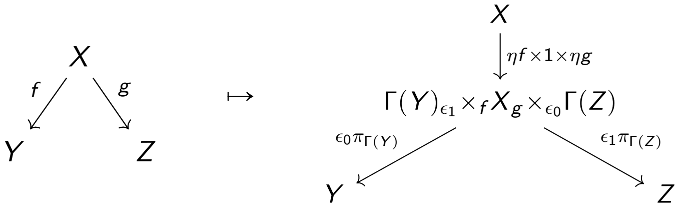
- that extends to morphisms of spans

Graphs

From any two-sided factorization, we obtain a reflexive graph on every object



Conversely, from a reflexive graph $X \xrightarrow{\eta} \Gamma(X) \xrightarrow{\epsilon} X, X$ on each object, we obtain a two-sided factorization (Street 1974)



Comma category

Notation

Write a span

$$\begin{array}{ccc} & X & \\ {\scriptstyle f}\swarrow & & \searrow{\scriptstyle g} \\ Y & & Z \end{array}$$

as $f, g : X \to Y, Z$.

Then a factorization maps

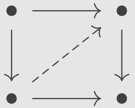
$$X \xrightarrow{f,g} Y, Z \qquad \mapsto \qquad X \xrightarrow{\lambda(f,g)} M(f,g) \xrightarrow{\rho(f,g)} Y, Z$$

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

Lifting

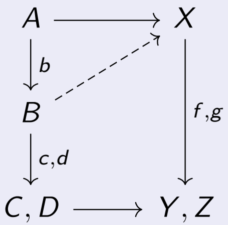
Lifting

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



Two-sided lifting

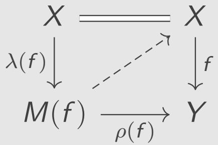
A sprout $A \xrightarrow{b} B \xrightarrow{c,d} C, D$ **lifts** against a span $X \xrightarrow{f,g} Y, Z$ if for any commutative diagram of solid arrows, there is a dashed arrow making the whole diagram commute.



Two-sided fibrations

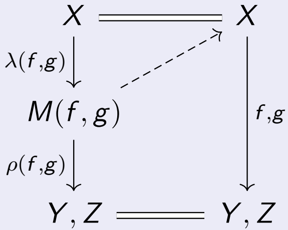
Fibrations.

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



Two-sided fibrations

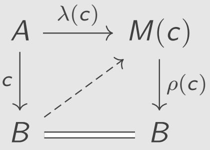
Given a two-sided factorization, a **two-sided fibration** is a span $f, g : X \to Y, Z$ for which there is a lift in



Rooted cofibrations

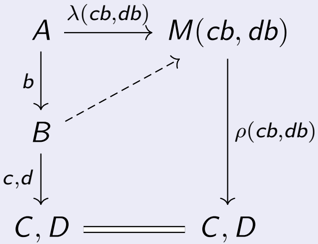
Cofibrations

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



First results

For a factorization...

- every isomorphism is both a cofibration and fibration
- cofibrations and fibrations are closed under retracts
- cofibrations and fibrations are closed under composition
- fibrations are stable under pullback
- cofibrations lift against fibrations

For a two-sided factorization...

- every sprout whose top morphism is an isomorphism is a rooted cofibration
- every product projection $X \times Y \to X, Y$ is a two-sided fibration
- the span-composition of two two-sided fibrations is a two-sided fibration
- two-sided fibrations are stable under pullback
- rooted cofibrations lift against two-sided fibrations

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 (Rosický-Tholen 2002)

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.

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

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

Two weak factorization systems

Proposition

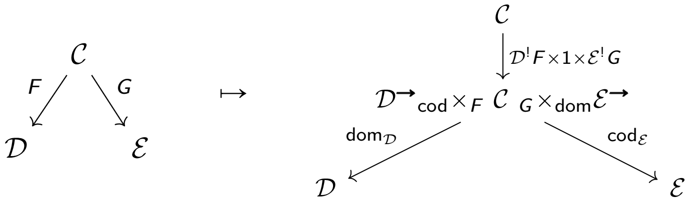
Consider a two-sided fibration $f, g : X \to Y, Z$ in a 2swfs. Then f is a past fibration and g is a future fibration.

Proposition

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

The example in *Cat*

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



- The past fibrations contain the Grothendieck fibrations
- The future fibrations contain the Grothendieck opfibrations
- The two-sided fibrations contain the (Grothendieck) two-sided fibrations (Street 1974)

2SWFSs from graphs

We want to understand which 2swfs's arise from functorial reflexive graphs, since this is how we will model the homomorphism type.

First, we characterize those functorial reflexive graphs which give rise to 2swfs.

Theorem

Consider a functorial reflexive graph $X \to \Gamma(X) \to X, X$. 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 weakly left transitive and weakly left connected.

Theorem

Consider a functorial reflexive graph $X \to \Gamma(X) \to X, X$. 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 weakly left transitive, weakly right transitive, weakly left connected, and weakly right connected.

Type-theoretic 2SWFSs

Theorem

The following are equivalent for a wfs:

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

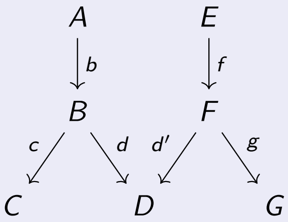
Fibrant object in a 2swfs

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

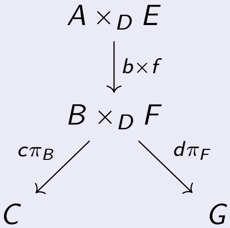
Type-theoretic 2SWFSs

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 (North 2017)

The following are equivalent for a wfs:

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

Examples

- In Cat, $\mathcal{C}^{\to}$
- In simplicial sets, free internal category on $X^{y(1)}$
- In cubical sets with connections, free internal category on $X^{y(1)}$
- 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!