Directed weak factorization systems for type theory

Paige Randall North

The Ohio State University

14 June 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

Syntax for a directed homotopy type theory

Semantics in *Cat*

Two-sided weak factorization systems

Outline

Syntax for a directed homotopy type theory

Semantics in *Cat*

Two-sided weak factorization systems

Rules for hom: core and op

$$\frac{T \ \textsc{type}}{T^{\mathrm{core}} \ \textsc{type}}$$

$$\frac{T \ \textsc{type}}{T^{\mathrm{op}} \ \textsc{type}}$$

$$\frac{T \ \textsc{type} \qquad t : T^{\mathrm{core}}}{it : T}$$

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

Rules for hom: formation

Id formation

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

hom formation

$$\frac{T \ \ \textsc{type} \qquad s : T^{\mathsf{op}} \qquad t : T}{\mathsf{hom}_T(s,t) \ \ \textsc{type}}$$

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

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

hom right elimination and computation

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

Rules for hom: left elimination and computation

Id elimination and computation

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

hom left elimination and computation

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

Outline

Semantics in *Cat*

There are two functorial reflexive relations on *Cat*:

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

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

- The first models the identity type $(\Sigma_{c,c':\mathcal{C}} \mathsf{Id}(c, c')$ interpreted by $\mathcal{C}^{\cong})$
- **Goal:** to see how the second one models the homomorphism type

Semantics in *Cat*

There are two functorial reflexive relations on *Cat*:

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

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

- The first models the identity type $(\Sigma_{c,c':\mathcal{C}} \mathsf{Id}(c, c')$ interpreted by $\mathcal{C}^{\cong})$
- **Goal:** to see how the second one models the homomorphism type
- Hurdles:
  - The second factorization generates a weak factorization system, but $\mathcal{C}^{\rightarrow} \to \mathcal{C} \times \mathcal{C}$ is not a right-hand map there.

Semantics in *Cat*

There are two functorial reflexive relations on *Cat*:

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

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

- The first models the identity type ($\Sigma_{c,c':\mathcal{C}}\mathrm{Id}(c, c')$ interpreted by $\mathcal{C}^{\cong}$)
- **Goal:** to see how the second one models the homomorphism type
- Hurdles:
  - 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 $Tw(\mathcal{C}) \to \mathcal{C}^{\mathrm{op}} \times \mathcal{C}$

Semantics in *Cat*

There are two functorial reflexive relations on *Cat*:

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

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

- The first models the identity type ($\Sigma_{c,c':\mathcal{C}} \mathrm{Id}(c, c')$ interpreted by $\mathcal{C}^{\cong}$)
- **Goal:** to see how the second one models the homomorphism type
- Hurdles:
  - 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 $Tw(\mathcal{C}) \to \mathcal{C}^{\mathrm{op}} \times \mathcal{C}$
  - If we model dependent types by right maps $\mathcal{C} \to \Gamma$, there's no good way to model the operation $(\Gamma \vdash \mathcal{C}) \mapsto (\Gamma \vdash \mathcal{C}^{\mathrm{op}})$.

Semantics in *Cat*

There are two functorial reflexive relations on *Cat*:

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

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

- The first models the identity type $(\Sigma_{c,c':\mathcal{C}} \mathsf{Id}(c, c')$ interpreted by $\mathcal{C}^{\cong})$
- **Goal:** to see how the second one models the homomorphism type
- Hurdles:
  - The second factorization generates a weak factorization system, but $\mathcal{C}^{\rightarrow} \to \mathcal{C} \times \mathcal{C}$ is not a right-hand map there. Old solution: consider the twisted arrow category $Tw(\mathcal{C}) \to \mathcal{C}^{\mathrm{op}} \times \mathcal{C}$
  - If we model dependent types by right maps $\mathcal{C} \to \Gamma$, there's no good way to model the operation $(\Gamma \vdash \mathcal{C}) \mapsto (\Gamma \vdash \mathcal{C}^{\mathrm{op}})$. Old solution: we model dependent types by functors $\Gamma \to Cat$.

Semantics in *Cat*

There are two functorial reflexive relations on *Cat*:

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

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

- The first models the identity type $(\Sigma_{c,c':\mathcal{C}} \mathsf{Id}(c, c')$ interpreted by $\mathcal{C}^{\cong})$
- **Goal:** to see how the second one models the homomorphism type
- Hurdles:
  - 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 $Tw(\mathcal{C}) \to \mathcal{C}^{\mathsf{op}} \times \mathcal{C}$
  - If we model dependent types by right 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 $\Gamma \to Cat$.
  - Problem: we rely too much on properties of *Cat*. A synthetic categorical theory of direction should be simpler.

WFS from relation

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

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

WFS from relation

How do we get weak factorization systems from a functorial reflexive relation (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.

Relation from WFS

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

We factor the diagonal of every object.

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

Relation from WFS

How do we get a functorial reflexive relation (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.

$$\begin{tikzcd} & X \arrow[dl, "1"'] \arrow[dr, "1"] & \\ X & & X \end{tikzcd} \quad \mapsto \quad \begin{tikzcd} & X \arrow[d, "\lambda(1_X)"] & \\ & M(1_X, 1_X) \arrow[dl, "\rho_0(1)"'] \arrow[dr, "\rho_1(1)"] & \\ X & & X \end{tikzcd}$$

Outline

Syntax for a directed homotopy type theory

Semantics in *Cat*

Two-sided weak factorization systems

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**

$$\begin{tikzcd} & X \arrow[dl, "f"'] \arrow[dr, "g"] & \\ Y & & Z \end{tikzcd} \qquad \mapsto \qquad \begin{tikzcd} & X \arrow[d, "\lambda(f,g)"] & \\ & M(f,g) \arrow[dl, "\rho_0(f,g)"'] \arrow[dr, "\rho_1(f,g)"] & \\ Y & & Z \end{tikzcd}$$

- that extends to morphisms of spans

Relations

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

$$\begin{tikzcd} & X \arrow[dl, "1"'] \arrow[dr, "1"] & \\ X & & X \end{tikzcd} \quad \mapsto \quad \begin{tikzcd} & X \arrow[d, "\lambda(1_X)"] & \\ & M(1_X, 1_X) \arrow[dl, "\rho_0(1)"'] \arrow[dr, "\rho_1(1)"] & \\ X & & X \end{tikzcd}$$

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

$$\begin{tikzcd} & X \arrow[dl, "f"'] \arrow[dr, "g"] & \\ Y & & Z \end{tikzcd} \quad \mapsto \quad \begin{tikzcd} & X \arrow[d, "\eta f \times 1 \times \eta g"] & \\ & \Gamma(Y)_{\epsilon_1} \times_f X_g \times_{\epsilon_0} \Gamma(Z) \arrow[dl, "\epsilon_0 \pi_{\Gamma(Y)}"'] \arrow[dr, "\epsilon_1 \pi_{\Gamma(Z)}"] & \\ Y & & Z \end{tikzcd}$$

Comma category

Notation

Write a span $\begin{tikzcd} & X \arrow[dl, "f"'] \arrow[dr, "g"] & \\ Y & & Z \end{tikzcd}$ as $f, g : X \to Y, Z$.

Then a factorization maps

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

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

Lifting

Lifting

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

$$\begin{tikzcd} \bullet \arrow[r] \arrow[d] & \bullet \arrow[d] \\ \bullet \arrow[r] \arrow[ur, dashed] & \bullet \end{tikzcd}$$

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.

$$\begin{tikzcd} A \arrow[r] \arrow[d, "b"] & X \arrow[dd, "f,g"] \\ B \arrow[d, "c,d"] \arrow[ur, dashed] & \\ C, D \arrow[r] & Y, Z \end{tikzcd}$$

Two-sided fibrations

Fibrations.

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

$$\begin{tikzcd} X \arrow[r, equal] \arrow[d, "\lambda(f)"'] & X \arrow[d, "f"] \\ M(f) \arrow[r, "\rho(f)"'] \arrow[ur, dashed] & Y \end{tikzcd}$$

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

$$\begin{tikzcd} X \arrow[r, equal] \arrow[d, "\lambda(f{,}g)"'] & X \arrow[dd, "f{,}g"] \\ M(f, g) \arrow[d, "\rho(f{,}g)"'] \arrow[ur, dashed] & \\ Y, Z \arrow[r, equal] & Y, Z \end{tikzcd}$$

Rooted cofibrations

Cofibrations

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

$$
\begin{tikzcd}
A \arrow[r, "\lambda(c)"] \arrow[d, "c"'] & M(c) \arrow[d, "\rho(c)"] \\
B \arrow[r, equal] \arrow[ur, dashed] & B
\end{tikzcd}
$$

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

$$
\begin{tikzcd}
A \arrow[r, "\lambda(cb{,}db)"] \arrow[d, "b"'] & M(cb, db) \arrow[dd, "\rho(cb{,}db)"] \\
B \arrow[d, "c{,}d"'] \arrow[ur, dashed] & \\
C, D \arrow[r, equal] & C, D
\end{tikzcd}
$$

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).

$$
\begin{tikzcd}
X \arrow[r, equal] \arrow[d, "\lambda(f,!)"] & X \arrow[r, equal] \arrow[d, "\lambda(f,g)"] & X \arrow[d, "\lambda(!,g)"] \\
M(f,!) \arrow[d, "\rho(f,!)"] & M(f,g) \arrow[l, "M(1,1,!)"'] \arrow[r, "M(1,!,1)"] \arrow[d, "\rho(f,g)"] & M(!,g) \arrow[d, "\rho(!,g)"] \\
Y,* & Y,Z \arrow[l, "1,!"'] \arrow[r, "!,1"] & *,Z
\end{tikzcd}
$$

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 cofibrations 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 $h' : Z \to Z'$. 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

$$\begin{tikzcd} & \mathcal{C} \arrow[dl, "F"'] \arrow[dr, "G"] & \\ \mathcal{D} & & \mathcal{E} \end{tikzcd} \quad \mapsto \quad \begin{tikzcd} & \mathcal{C} \arrow[d, "\mathcal{D}^! F \times 1 \times \mathcal{E}^! G"] & \\ & \mathcal{D}^{\rightarrow} {}_{\mathrm{cod}}\times_{F} \mathcal{C} {}_{G}\times_{\mathrm{dom}} \mathcal{E}^{\rightarrow} \arrow[dl, "\mathrm{dom}_{\mathcal{D}}"'] \arrow[dr, "\mathrm{cod}_{\mathcal{E}}"] & \\ \mathcal{D} & & \mathcal{E} \end{tikzcd}$$

- 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 relations

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

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

Theorem (North 2017)

Consider a functorial reflexive relation $X \to \Gamma(X) \to X, X$. 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 relation $X \to \Gamma(X) \to X, X$. 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 (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 relation $X \to \Gamma(X) \to X, X$.
- 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,

$$\begin{tikzcd}
 & A \arrow[d, "b"] & & E \arrow[d, "f"] & \\
 & B \arrow[dl, "c"'] \arrow[dr, "d"] & & F \arrow[dl, "d'"'] \arrow[dr, "g"] & \\
C & & D & & G
\end{tikzcd}$$

the 'composite' is a cofibration.

$$\begin{tikzcd}
 & A \times_D E \arrow[d, "b \times f"] & \\
 & B \times_D F \arrow[dl, "c\pi_B"'] \arrow[dr, "d\pi_F"] & \\
C & & G
\end{tikzcd}$$

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 relation $X \to \Gamma(X) \to X, X$.
- 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 relation $X \to \Gamma(X) \to X, X$.
- it is type-theoretic: (1) all objects are fibrant and (2) the two-sided Frobenius condition holds.

Examples

- In Cat, $\mathcal{C}^{\rightarrow}$
- 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!