A type theory for directed homotopy theory

Paige Randall North

Ohio State University

18 June 2018

Outline

Outline

Goal

Goal

To develop a directed type theory.

Goal

Goal

To develop a directed type theory.

To formalize theorems about:

Goal

Goal

To develop a directed type theory.

To formalize theorems about:

- Higher category theory

Goal

Goal

To develop a directed type theory.

To formalize theorems about:

- Higher category theory
- Directed homotopy theory

Goal

Goal

To develop a directed type theory.

To formalize theorems about:

- Higher category theory
- Directed homotopy theory
    - Concurrent processes

Goal

Goal

To develop a directed type theory.

To formalize theorems about:

- Higher category theory
- Directed homotopy theory
  - Concurrent processes

Criteria

- Directed paths are introduced as terms of a type former, hom, to be added to Martin-Löf type theory

Goal

Goal

To develop a directed type theory.

To formalize theorems about:

- Higher category theory
- Directed homotopy theory
  - Concurrent processes

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

Goal

Goal

To develop a directed type theory.

To formalize theorems about:

- Higher category theory
- Directed homotopy theory
  - Concurrent processes

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

What does directed mean?

Syntactically

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

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

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

- Can think of these terms as *undirected* paths

What does directed mean?

Syntactically

Martin-Löf's Id 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?

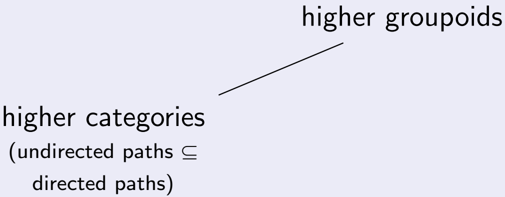
What does directed mean?

Semantically

higher groupoids

What does directed mean?

Semantically



What does directed mean?

Semantically

higher groupoids

higher categories
(undirected paths ⊆ directed paths)

directed spaces
(directed paths ⊆ undirected paths)

Outline

Directed spaces

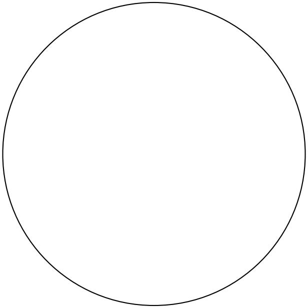
Rough definition

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

Directed spaces

Rough definition

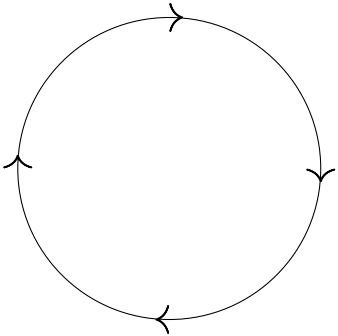
A (topological) space together with a subset of its paths that are marked as ‘directed’



Directed spaces

Rough definition

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

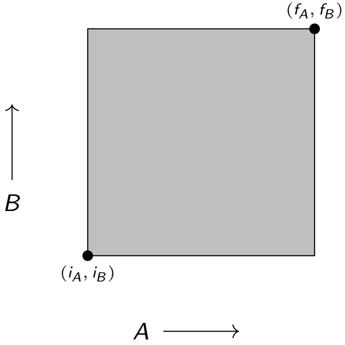


Directed spaces and concurrency

Concurrent processes can be represented by directed spaces.

Directed spaces and concurrency

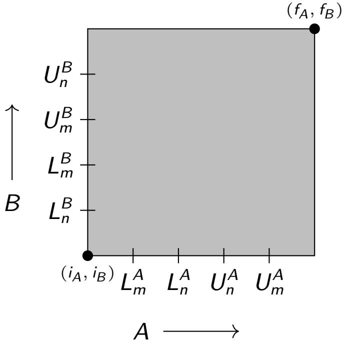
Concurrent processes can be represented by directed spaces.



- A, B are two processes

Directed spaces and concurrency

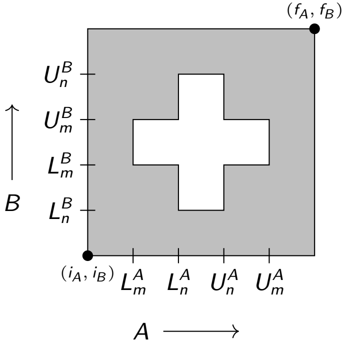
Concurrent processes can be represented by directed spaces.



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

Directed spaces and concurrency

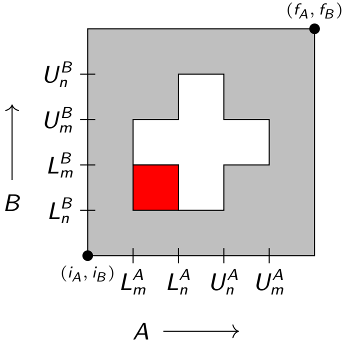
Concurrent processes can be represented by directed spaces.



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

Directed spaces and concurrency

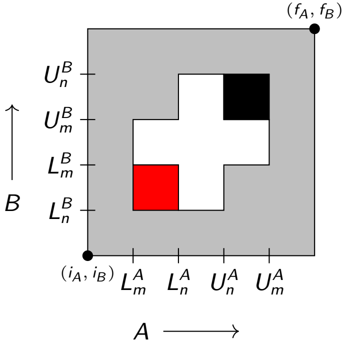
Concurrent processes can be represented by directed spaces.



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

Directed spaces and concurrency

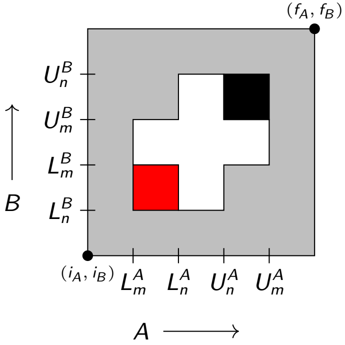
Concurrent processes can be represented by directed spaces.



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

Directed spaces and 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?
- Which states are reachable?

Outline

Rules for hom: core and op

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

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

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

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

Rules for hom: formation

$$\frac{T \ \ \textsc{type} \qquad s : T^{\mathrm{op}} \qquad t : T}{\hom_T(s,t) \ \ \textsc{type}}$$

Rules for hom: formation

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

Id formation

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

Rules for hom: introduction

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

Rules for hom: introduction

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

Id introduction

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

Rules for hom: right elimination and computation

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

Rules for hom: right elimination and computation

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

Id elimination and computation

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

Rules for hom: left elimination and computation

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

Rules for hom: left elimination and computation

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

Id elimination and computation

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

Syntactic results

- Transport: for a dependent type $t : T \vdash S(t)$:

$$t : T^{\text{core}}, t' : T, f : \hom_T(i^{\text{op}} t, t'), s : S(it) \\ \vdash \text{transport}_{\text{R}}(s, f) : S(t')$$

Syntactic results

- Transport: for a dependent type $t : T \vdash S(t)$:

$$t : T^{\mathrm{core}}, t' : T, f : \mathrm{hom}_T(i^{\mathrm{op}}t, t'), s : S(it) \\ \vdash \mathrm{transport}_{\mathrm{R}}(s, f) : S(t')$$

- Composition: for a type T:

$$r : T^{\mathrm{op}}, s : T^{\mathrm{core}}, t : T, f : \mathrm{hom}_T(r, is), g : \mathrm{hom}_T(i^{\mathrm{op}}s, t) \\ \vdash \mathrm{comp}_{\mathrm{R}}(f, g) : \mathrm{hom}_T(r, t)$$

Outline

The interpretation

- Use the framework of comprehension categories

The interpretation

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

The interpretation

- Use the framework of comprehension categories
- Dependent types are represented by functors $T : \Gamma \to Cat$.
- In the empty context, $\hom_T(s, t)$ is the usual homset,

The interpretation

- Use the framework of comprehension categories
- Dependent types are represented by functors $T : \Gamma \to Cat$.
- In the empty context, $\hom_T(s, t)$ is the usual homset,
  and the introduction rule just gives $1_t : hom_T(t, t)$.

The interpretation

- Use the framework of comprehension categories
- Dependent types are represented by functors $T : \Gamma \to Cat$.
- In the empty context, $\hom_T(s, t)$ is the usual homset,
  and the introduction rule just gives $1_t : hom_T(t, t)$.
- Plugs into the homotopy theory of morphisms, as the interpretation of the Id type plugs into the homotopy theory of isomorphisms.

Outline

Summary & future work

Summary

We have:

- a directed type theory

Summary & future work

Summary

We have:

- a directed type theory
- with a model in *Cat*.

Summary & future work

Summary

We have:

- a directed type theory
- with a model in *Cat*.

Future work

We need to:

- integrate this into traditional Martin-Löf type theory
  - integrate Id and hom in the same theory
  - specify Σ, Π, etc

Summary & future work

Summary

We have:

- a directed type theory
- with a model in *Cat*.

Future work

We need to:

- integrate this into traditional Martin-Löf type theory
  - integrate Id and hom in the same theory
  - specify Σ, Π, etc
- find interpretations in categories of directed spaces
  - build 'directed' weak factorization systems
  - build universes

Summary & future work

The future

We aim to define and reason about

$$isReachable(T) := \Sigma_{x:T} \hom_T(i, x)$$

$$isSafe(T) := \Sigma_{x:T^{op}} \hom_T(x, f)$$

for any type T with terms $i : T^{op}, f : T$.

Thank you!

The interpretation

- Use 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 which takes 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

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

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

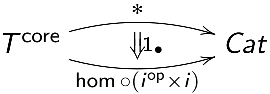
$$\frac{T \ \ \text{TYPE} \qquad s : T^{\mathrm{op}} \qquad t : T}{\hom_T(s, t) \ \ \text{TYPE}} \qquad\qquad \frac{T \ \ \text{TYPE} \qquad t : T^{\mathrm{core}}}{1_t : \hom_T(i^{\mathrm{op}} t, it) \ \ \text{TYPE}}$$

For any category T,

- Take the functor

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

- Take the natural transformation



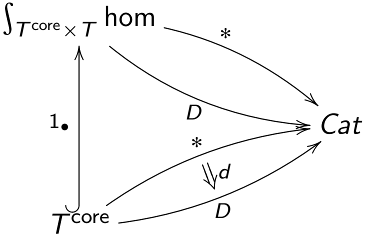
where each component $1_t : * \to \hom(t, t)$ picks out the identity morphism of t.

Interpreting right hom elimination and computation

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

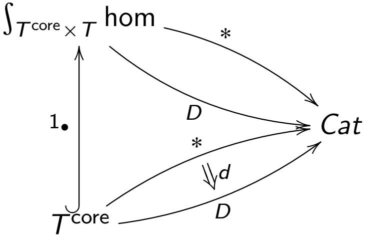
Interpreting right hom elimination and computation

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



Interpreting right hom elimination and computation

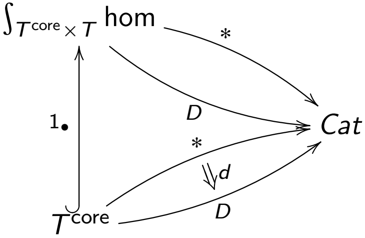
$$\frac{T \ \text{TYPE} \qquad s : T^{\text{core}}, t : T, f : \hom_T(i^{\text{op}}s, t) \vdash D(f) \ \text{TYPE} \qquad s : T^{\text{core}} \vdash d(s) : D(1_s)}{\begin{array}{c} s : T^{\text{core}}, t : T, f : \hom_T(i^{\text{op}}s, t) \vdash e_R(d, f) : D(f) \\ s : T^{\text{core}} \vdash e_R(d, 1_s)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



- Use the fact that the subcategory T^{core} is ‘initial’:

Interpreting right hom elimination and computation

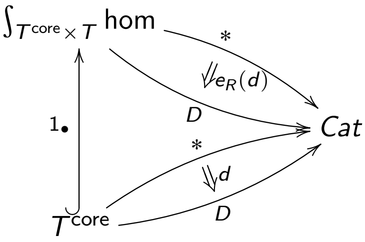
$$\frac{T \ \ \text{TYPE} \qquad s : T^{\text{core}}, t : T, f : \hom_T(i^{\text{op}}s, t) \vdash D(f) \ \ \text{TYPE} \\ s : T^{\text{core}} \vdash d(s) : D(1_s)}{s : T^{\text{core}}, t : T, f : \hom_T(i^{\text{op}}s, t) \vdash e_R(d, f) : D(f) \\ s : T^{\text{core}} \vdash e_R(d, 1_s) \equiv d(s) :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



- Use the fact that the subcategory T^{core} is 'initial':
  - 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

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



- Use the fact that the subcategory T^{core} is 'initial':
  - 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

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

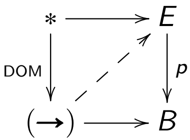
- which produces a factorization of any diagonal in two ways which each generate weak factorization systems.

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

- 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 lifting property



- so all Grothendieck opfibrations (dependent projections) are in the right class.
- The functor $1_\bullet : T^{\text{core}} \hookrightarrow \int_{T^{\text{core}} \times T} \text{hom}$ is the left part of the factorization of

$$i : T^{\text{core}} \to T.$$

- Then the right hom elimination and computation rule arises from the weak factorization system.