Directed homotopy type theory

Paige Randall North

University of Pennsylvania

1 June 2021

Outline

Introduction

Directed homotopy theory

A first attempt

A second attempt (work in progress)

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
    - Rewriting

Goal

Goal

To develop a directed type theory.

To formalize theorems about:

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

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
    - Rewriting

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
  - Rewriting

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

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

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 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

higher groupoids

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

What does directed mean?

Semantically

higher groupoids

higher categories
(undirected paths ⊆ directed paths)

directed spaces
(directed paths ⊆ undirected paths)

Outline

Directed spaces

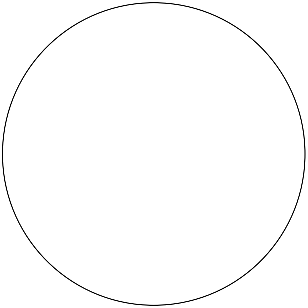
Rough definition

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

Directed spaces

Rough definition

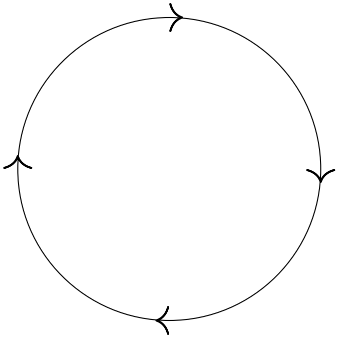
A space together with a subset of its paths that are marked as ‘directed’



Directed spaces

Rough definition

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

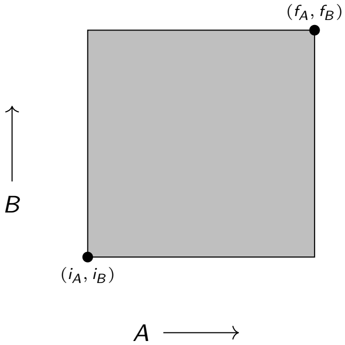


Application: concurrency

Concurrent processes can be represented by directed spaces.

Application: concurrency

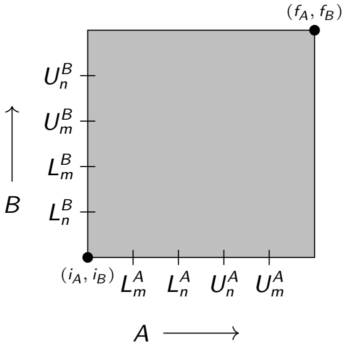
Concurrent processes can be represented by directed spaces.



- A, B are two processes

Application: concurrency

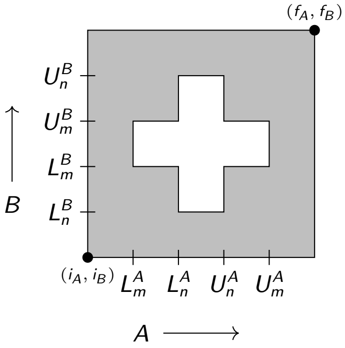
Concurrent processes can be represented by directed spaces.



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

Application: concurrency

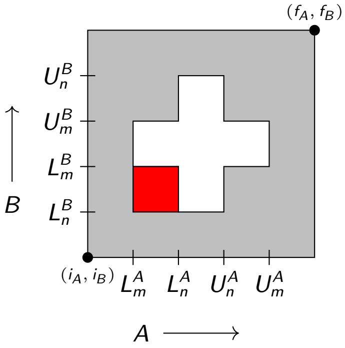
Concurrent processes can be represented by directed spaces.



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

Application: concurrency

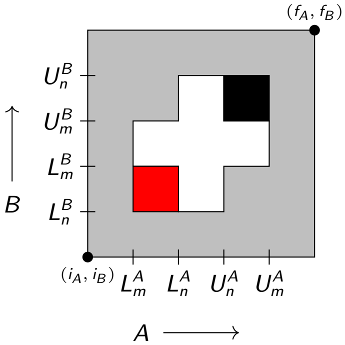
Concurrent processes can be represented by directed spaces.



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

Application: concurrency

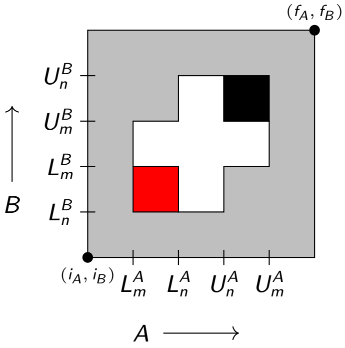
Concurrent processes can be represented by directed spaces.



- 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.



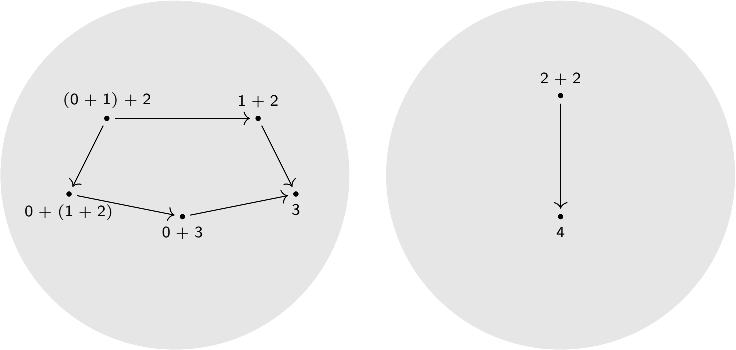
- A, B are two processes
- m, n are two memory locations
- which can be locked (L) or unlocked (U) by each process

Fundamental questions:

- Which states are safe?
- Which states are reachable?

Application: Term rewriting systems

Consider expressions in the monoid $N = (\mathbb{N}, 0, +)$.



- Interested in families $D(n)$ indexed by $n \in N$ for which rewrite rules $n \to m$ induce rewrites $D(n) \to D(m)$

Outline

Rules for hom: core and op

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

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

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

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

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

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

hom introduction

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

Rules for hom: right 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 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: left elimination and computation

Id elimination and computation

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

hom left elimination and computation

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

Syntactic results

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

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

Syntactic results

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

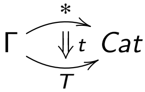
$$t : T^{\mathrm{core}}, t' : T, f : \mathrm{hom}_T(i^{\mathrm{op}} t, t'), s : S(it) \\ \vdash \mathrm{transport}_{\mathrm{R}}(s, f) : S(t')$$

- Composition: for a type T:

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

The interpretation

- Dependent types are represented by functors $T : \Gamma \to Cat$.
- Dependent terms are represented by natural transformations



- T^{core} is represented by the objects of T
- T^{core} is represented by the opposite of T
- hom is represented by hom : $T^{\text{op}} \to T \to Set$
- $1_\bullet$ is represented by the identity morphisms
- There are two computation rules since in $\Sigma_{x,y\,\text{hom}(x,y)}$, given a $f : \text{hom}(x, y)$, there are two arrows from an identity to f: postcomposing 1_x with f and precomposing 1_y with f

Outline

Problems with the first attempt

The functions op, core are problematic.

- There are no introduction rules for T^{core} or T^{op}
- Including the identity type causes the hom type to collapse to the identity type on elements of T^{core}.
- We need a op function on the universe; e.g. the **1-functor** $\text{op} : \mathit{Cat} \to \mathit{Cat}$. This does not exist for 2-categories and up.

A homotopical viewpoint

In the first attempt, we represent hom as a functor $C^{\mathrm{op}} \times C \to \mathit{Set}$ or, equivalenty under the Grothendieck construction, as a Grothendieck fibration:

$$\begin{array}{c} \Sigma_{x \in C^{\mathrm{op}}, y \in C} \hom(x, y) \\ \big\downarrow \\ C^{\mathrm{op}} \times C \end{array}$$

With van den berg and McCloskey, we are developing a notion of directed (algebraic) weak factorization system based on this.

A homotopical viewpoint

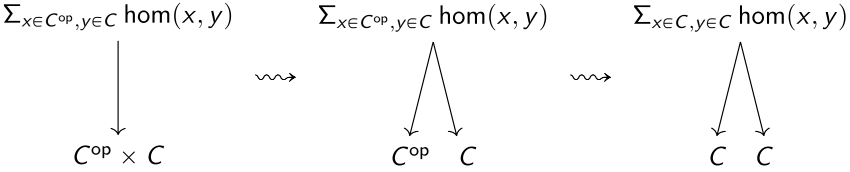
In the first attempt, we represent hom as a functor $C^{\mathrm{op}} \times C \to \mathit{Set}$ or, equivalenty under the Grothendieck construction, as a Grothendieck fibration:

$$
\begin{array}{c}
\Sigma_{x \in C^{\mathrm{op}}, y \in C} \hom(x, y) \\
\downarrow \\
C^{\mathrm{op}} \times C
\end{array}
\quad \rightsquigarrow \quad
\begin{array}{c}
\Sigma_{x \in C^{\mathrm{op}}, y \in C} \hom(x, y) \\
\swarrow \quad \searrow \\
C^{\mathrm{op}} \qquad C
\end{array}
$$

With van den berg and McCloskey, we are developing a notion of directed (algebraic) weak factorization system based on this.

A homotopical viewpoint

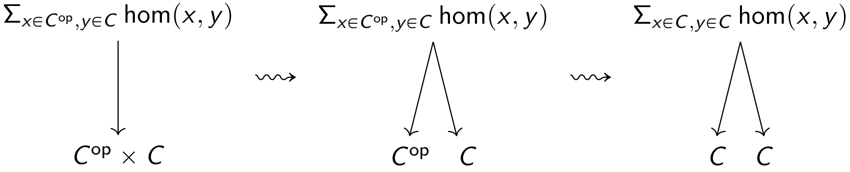
In the first attempt, we represent hom as a functor $C^{\mathrm{op}} \times C \to Set$ or, equivalenty under the Grothendieck construction, as a Grothendieck fibration:



With van den berg and McCloskey, we are developing a notion of directed (algebraic) weak factorization system based on this.

A homotopical viewpoint

In the first attempt, we represent hom as a functor $C^{\mathrm{op}} \times C \to \mathit{Set}$ or, equivalenty under the Grothendieck construction, as a Grothendieck fibration:



The latter is a two-sided fibration of Street (1973).

- It is a pair of a Grothendieck fibration and a Grothendieck opfibration such that when one 'lifts'/'transports' along one morphism of one base, the result is in the same fiber wrt the other base.
- This is what should happen in type theory: Along $f : a \to a'$ one transports $T(a, b) \to T(a', b)$.

With van den berg and McCloskey, we are developing a notion of directed (algebraic) weak factorization system based on this.

A modal approach

Following Licata-Shulman-Riley's (2017) modal framework for the sequent calculus.
We start with a 2-category of modes:

- Objects: $\bullet$
- 1-Morphisms: $\mathsf{op}, \mathsf{core} : \bullet \to \bullet$
- 2-Morphisms: $i : \mathsf{core} \Rightarrow 1_\bullet$, $i^{\mathsf{op}} : \mathsf{core} \Rightarrow \mathsf{op}$
- Equalities: $\mathsf{op} \circ \mathsf{op} = 1_\bullet$, $x \circ \mathsf{core} = \mathsf{core} \circ x = \mathsf{core}$ for all x, ...

Context descriptors

Every judgment is annotated by a *context descriptor*. These are inductively generated by:

$$\frac{}{x_1, ..., x_n \vdash x_i} \qquad \frac{}{\overline{x} \vdash} \qquad \frac{\overline{x} \vdash y}{\overline{x} \vdash \mathsf{op}(y)} \qquad \frac{\overline{x} \vdash y}{\overline{x} \vdash \mathsf{core}(y)} \qquad \frac{\overline{x} \vdash y \qquad \overline{x} \vdash z}{\overline{x} \vdash y, z}$$

The variables $x_1, ..., x_n$ coincide with the variables $x_1 : A_1, ..., x_n : A_n \vdash ...$ of a context.

A modal approach continued

The variable rule:

$$\frac{}{\Gamma, x : \sigma, \Delta \vdash_{\overline{\gamma}, x, \overline{\delta} \vdash x} x : \sigma}$$

The weakening rule:

$$\frac{\Gamma, \Delta \vdash_{\overline{\gamma}, \overline{\delta} \vdash g, d} T \qquad \Gamma \vdash_{\overline{\gamma} \vdash g} \rho}{\Gamma, \Delta \vdash_{\overline{\gamma}, x, \overline{\delta} \vdash g, d} T}$$

The substitution rule:

$$\frac{\Gamma, x : \rho, \Delta \vdash_{\overline{\gamma}, x, \overline{\delta} \vdash g, \chi, d} T \qquad \Gamma \vdash_{\overline{\gamma} \vdash h} U : \rho}{\Gamma, \Delta[U/x] \vdash_{\overline{\gamma}, \overline{\delta} \vdash g, \chi[h/x], d} T[U/x]}$$

The modal substitution rule:

$$\frac{a \Rightarrow b \qquad \Gamma \vdash_{\overline{\gamma} \vdash \ldots, b(x), \ldots} T}{\Gamma \vdash_{\overline{\gamma} \vdash \ldots, a(x), \ldots} T}$$

The new hom type

hom formation

$$\frac{\Gamma \vdash_{\overline{\gamma} \vdash g} A \qquad \Gamma, \Delta \vdash_{\overline{\gamma}, \overline{\delta} \vdash g, \mathrm{op}(d)} a : A \qquad \Gamma, \Delta \vdash_{\overline{\gamma}, \overline{\delta} \vdash g, d} b : A}{\Gamma, \Delta \vdash_{\overline{\gamma}, \overline{\delta} \vdash g, d} \mathrm{hom}_A(a, b)}$$

hom introduction

$$\frac{\Gamma \vdash_{\overline{\gamma} \vdash g} A \qquad \Gamma, \Delta \vdash_{\overline{\gamma}, \overline{\delta} \vdash g, \mathrm{core}(d)} a : A}{\Gamma, \Delta \vdash_{\overline{\gamma}, \overline{\delta} \vdash g, \mathrm{core}(d)} 1_a : \mathrm{hom}_A(a, a)}$$

The new hom type

right hom computation and elimination

$$\frac{\begin{array}{c}\Gamma, a : A, b : A, f : \hom_A(a,b) \vdash_{\overline{\gamma}, a, b, f \vdash g, \mathrm{core}(a), b, f} D(f) \\ \Gamma, a : A \vdash_{\overline{\gamma}, a \vdash g, \mathrm{core}(a)} d(a) : D(1_a)\end{array}}{\begin{array}{c}\Gamma, a : A, b : A, f : \hom_A(a,b) \vdash_{\overline{\gamma}, a, b, f \vdash \underline{g}, \mathrm{core}(a), b, f} \overline{d}(f) : D(f) \\ \Gamma, a : A \vdash_{\overline{\gamma}, a \vdash g, \mathrm{core}(a)} d(a) \equiv \overline{d}(1_a) : D(1_a)\end{array}}$$

Inside the type theory

What can we do?

- Define the Id type analogously.
- Find an inclusion $\mathrm{Id}(a, b) \to \hom(a, b)$ in the context of the cores of a and b, but not $\hom(a, b) \to \mathrm{Id}(a, b)$.
- Transport and compose.

What can't we do?

- Form all Σ types (F types in LSR). For example, the one you should get from $a : A \vdash_{a \vdash \mathrm{op}(a)} 1$ is A^{op}.

Future work

- Connect this formally with the intended semantics.
- Understand which Σ types exist.
- Π-types, directed univalence, higher inductive types, etc...

Thank you!