Directed homotopy type theory

Paige Randall North

Utrecht University

1 December 2022

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
>   - Rewriting
>   - Neural networks
>   - ...

Goal

Goal

To develop a directed type theory.

To formalize theorems about:

- Higher category theory
- Directed homotopy theory
  - Concurrent processes
  - Rewriting
  - Neural networks
  - ...

The Id-type

Id-type

In MLTT, the *identity type* internalizes the notion of equality.

- There is a type $\mathrm{Id}_T(a, b)$ for any type T and $a, b : T$
- Inductively generated by $\mathrm{refl}_a : \mathrm{Id}_T(a, a)$

The Id-type

Id-type

In MLTT, the *identity type* internalizes the notion of equality.

- There is a type $\text{Id}_T(a, b)$ for any type T and $a, b : T$
- Inductively generated by $\text{refl}_a : \text{Id}_T(a, a)$

We can show:

- The relation $\text{Id}_T(a, b)$ is (reflexive), transitive, and symmetric
- The identity type can be iterated:
  $p, q : \text{Id}_T(a, b), \quad \alpha, \beta : \text{Id}_{\text{Id}_T(a,b)}(p, q), \quad ...$
- It is possible for $\text{Id}_{\text{Id}_T(a,b)}(p, q)$ to be empty.

Thus the Id-type constructor endows each type with the structure of an ∞-groupoid, or *space*.

The Id-type

Id-type

In MLTT, the *identity type* internalizes the notion of equality.

- There is a type $\mathrm{Id}_T(a, b)$ for any type T and $a, b : T$
- Inductively generated by $\mathrm{refl}_a : \mathrm{Id}_T(a, a)$

We can show:

- The relation $\mathrm{Id}_T(a, b)$ is (reflexive), transitive, and symmetric
- The identity type can be iterated:
  $p, q : \mathrm{Id}_T(a, b), \quad \alpha, \beta : \mathrm{Id}_{\mathrm{Id}_T(a,b)}(p, q), \quad ...$
- It is possible for $\mathrm{Id}_{\mathrm{Id}_T(a,b)}(p, q)$ to be empty.

Thus the Id-type constructor endows each type with the structure of an ∞-groupoid, or *space*.

$\rightarrow$ homotopy type theory

What does directed mean? (Syntactically)

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 $ip : \mathsf{Id}_T(b, a)$.

What does directed mean? (Syntactically)

> ### Id-type is symmetric/undirected
>
> 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 identity terms as *undirected* paths

What does directed mean? (Syntactically)

Id-type is symmetric/undirected

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 identity terms as *undirected* paths
- Can we design a type former of *directed* paths that resembles Id but without its inversion operation i?
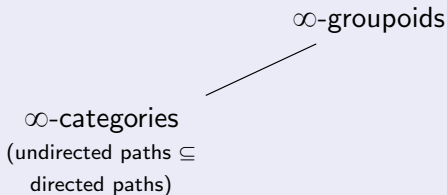
What does directed mean? (Semantically)
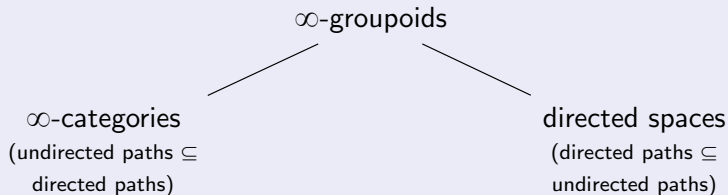
> **Categories vs. spaces**
>
> ∞-groupoids

What does directed mean? (Semantically)

Categories vs. spaces

∞-groupoids

∞-categories
(undirected paths $\subseteq$
directed paths)

What does directed mean? (Semantically)

Categories vs. spaces

∞-groupoids

∞-categories
(undirected paths $\subseteq$
directed paths)

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

Outline

Directed spaces

Rough definition

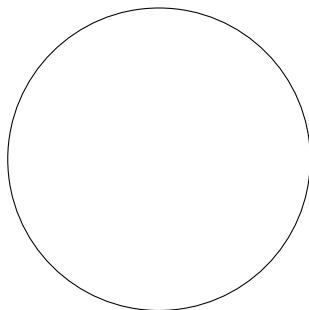A space together with a subset of its paths that are marked as 'directed'
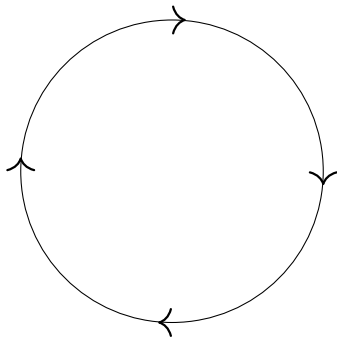
Directed spaces

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

Directed spaces

Rough definition

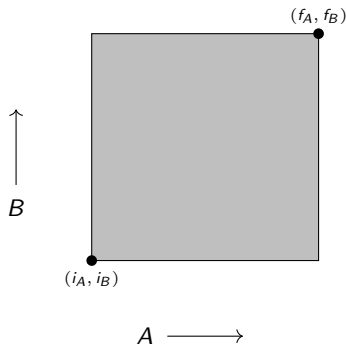A space together with a subset of its paths that are marked as 'directed'

Application: concurrency

Concurrent processes can be represented by directed spaces.

Application: concurrency

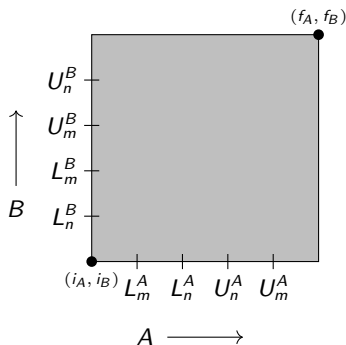Concurrent processes can be represented by directed spaces.
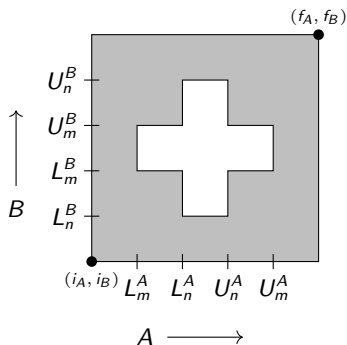

- A, B are two processes

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
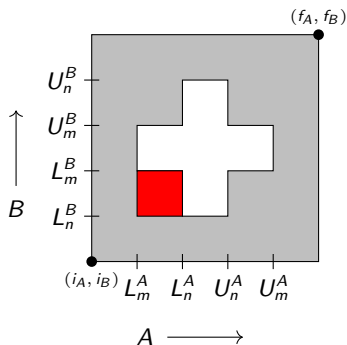- which can be locked (L) or unlocked (U) by each process

Application: concurrency

Concurrent processes can be represented by directed spaces.

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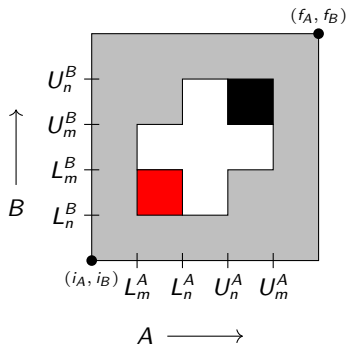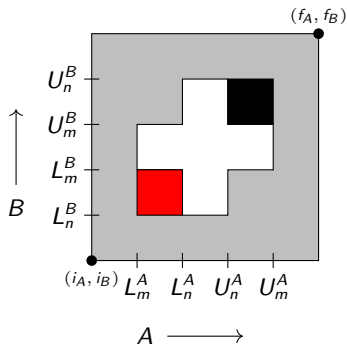- which can be locked (L) or unlocked (U) by each process

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 $N = (\mathbb{N}, 0, +)$.
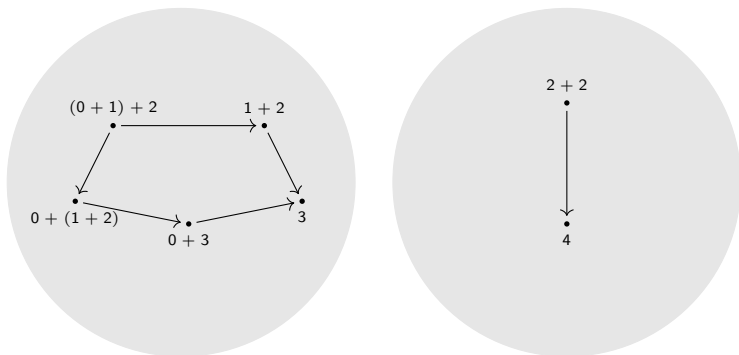
- Interested in families $D(n)$ indexed by $n \in N$ for which rewrite rules $n \to m$ induce rewrites $D(n) \to D(m)$

Outline

Rules for hom: core and op

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

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

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

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

Rules for hom: formation

Id formation

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

hom formation

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

Rules for hom: introduction

Id introduction

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

hom introduction

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

Rules for hom: right elimination and computation

Id elimination and computation

$$\frac{T \;\; \text{type} \qquad s : T, t : T, f : \mathsf{Id}_T(s, t) \vdash D(f) \;\; \text{type} \qquad s : T \vdash d(s) : D(r_s)}{s : T, t : T, f : \mathsf{Id}_T(s, t) \vdash j(d, f) : D(f)}$$
$$s : T \vdash j(d, r_s) \equiv d(s) : D(r_s)$$

hom right elimination and computation

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

Rules for hom: left elimination and computation

Id elimination and computation

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

hom left elimination and computation

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

Syntactic results

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

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

Syntactic results

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

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

- Composition: for a type T:

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

The interpretation

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

The interpretation

- Dependent types are represented by functors $T : \Gamma \to Cat$.
- T^{core} is represented by the objects of T
- T^{op} 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

Point

In this model, terms of hom-types are not always invertible, so they are not always invertible in the type theory.

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** op : $Cat \to Cat$. This does not exist for 2-categories and up.

Modal directed homotopy type theory

Solution

The solution is to properly account for core, op, etc.

- **syntactically**: modal type theory
- **semantically**: multisided weak factorization systems
  (jww van den Berg, McCloskey)

(The theory of multisided weak factorization systems accounts for
multiple fibrations – e.g. Grothendieck fibrations, opfibrations,
isofibrations – in one category and how they interact, inspired by
the two-sided fibrations of Street.)

Modal directed type theory

The idea:

- Forget about having a type constructors $T \mapsto T^{\mathrm{op}}, T^{\mathrm{core}}$

$$x : R^{\mathrm{op}}, y : S^{\mathrm{core}} \vdash T$$

- Instead op and core should be descriptions of how variables can be used.
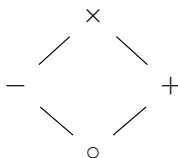
$$x \overset{-}{:} R, y \overset{\circ}{:} S \vdash T$$

- Compare with linear / n-use type theory (Reed, McBride, Licata-Shulman-Riley, Abel...)

$$x \overset{0}{:} R, y \overset{3}{:} S \vdash T$$

A modal approach

Following Licata-Shulman-Riley's (2017) modal framework for the sequent calculus.

- ▶ four modes that form a lattice:

- ▶ with an multiplication
    - ▶ $+$ is the identity
    - ▶ $\circ$ is almost an absorbing element: $\circ a = a\circ = \circ$ except $\circ \times = \times$
    - ▶ $\times$ is almost an absorbing element: $\times a = a\times = \times$ except $\times\circ = \circ$
    - ▶ $-$ is a root of unity: $-- = +$.
- ▶ This is a sub-monoidal category of $[\mathsf{Cat}, \mathsf{Cat}]$

Orientations

Contexts are annotated by *orientations*. We write:

$$w \overset{\times}{:} A, x \overset{+}{:} B, y \overset{-}{:} C, z \overset{\circ}{:} D \vdash T$$

or

$$w : A, x : B, y : C, z : D \vdash_{x, y^-, z^\circ} T$$

Orientations

Contexts are annotated by *orientations*. We write:

$$w \overset{\times}{:} A, x \overset{+}{:} B, y \overset{-}{:} C, z \overset{\circ}{:} D \vdash T$$

or

$$w : A, x : B, y : C, z : D \vdash_{x, y^-, z^\circ} T$$

Orientations on contexts inherit an order from the lattice, so we use the following rule.

$$\frac{\Gamma \vdash_\ell \mathcal{J} \qquad m \leqslant \ell \; \Gamma\text{-ort}}{\Gamma \vdash_m \mathcal{J}} \; \text{Ort-Subst}$$

Structural rules

$$\frac{\Gamma, x : \sigma, \Delta \text{ ctx}}{\Gamma, x : \sigma, \Delta \vdash_x x : \sigma} \text{ Var}$$

$$\frac{\Gamma, \Delta \vdash_{\ell,m} \mathcal{J} \qquad \Gamma \vdash_\ell \rho \text{ type}}{\Gamma, x : \rho, \Delta \vdash_{\ell,m} \mathcal{J}} \text{ Weak}$$

$$\frac{\begin{array}{c} \Gamma, x : \rho, \Delta \vdash_{\ell,\omega,m} T \\ \Gamma \vdash_\ell U : \rho \qquad n \leqslant \ell \ \Gamma\text{-ort} \qquad n \leqslant \omega \cdot \ell \ \Gamma\text{-ort} \end{array}}{\Gamma, \Delta[U/x] \vdash_{n,m} T[U/x]} \text{ Subst}$$

The new hom-type

hom formation

$$\frac{\Gamma \vdash_\ell A \text{ type}}{\Gamma, x : A, y : A \vdash_{\ell, x^-, y} \text{hom}_A(x, y) \text{ type}} \text{ hom-}\textsc{form}$$

hom introduction

$$\frac{\Gamma \vdash_\ell A \text{ type}}{\Gamma, x : A \vdash_{\ell, x^\circ} 1_x : \text{hom}_A(x, x) \text{ type}} \text{ hom-}\textsc{intro}$$

The new Id-type

Id° formation

$$\frac{\Gamma \vdash_\ell A \text{ type}}{\Gamma, x : A, y : A \vdash_{\ell, x^\circ, y^\circ} \mathsf{Id}_A^\circ(x, y) \text{ type}} \ \mathsf{Id}^\circ\text{-}\mathrm{FORM}$$

Id° introduction

$$\frac{\Gamma \vdash_\ell A \text{ type}}{\Gamma, x : A \vdash_{\ell, x^\circ} \mathsf{refl}_x : \mathsf{Id}_A^\circ(x, x) \text{ type}} \ \mathsf{Id}^\circ\text{-}\mathrm{INTRO}$$

The new Id-type

$\mathsf{Id}^\times$ formation

$$\frac{\Gamma \vdash_\ell A \text{ type}}{\Gamma, x : A, y : A \vdash_{\ell, x^\times, y^\times} \mathsf{Id}^\times(x, y) \text{ type}} \; \mathsf{Id}^\times\text{-FORM}$$

$\mathsf{Id}^\times$ introduction

$$\frac{\Gamma \vdash_\ell A \text{ type}}{\Gamma, x : A \vdash_{\ell, x^\times} \mathsf{refl}_x : \mathsf{Id}^\times(x, x) \text{ type}} \; \mathsf{Id}^\times\text{-INTRO}$$

Inside the type theory

What can we do?

- Find inclusions $\mathsf{Id}^\circ(a, b) \to \hom(a, b) \to \mathsf{Id}^\times(a, b)$, but not $\hom(a, b) \to \mathsf{Id}^\circ(a, b)$.
- Transport and compose.

What can't we do?

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

Future work

- Connect this formally with the intended semantics (jww van den Berg-McCloskey and Ahrens-van der Weide)
- Understand which Σ types exist.
- Π-types, directed univalence, higher inductive types, etc...

Thank you!