Homotopy Type Theory Class

LMFI 2019-2020

Basic cubical type theory

Hugo Herbelin

The types-as-spaces correspondence

Hofmann-Streicher, 1994: can unicity of reflexivity proofs be proved in type theory?

$$\mathsf{UIP}_{\mathit{refl}} : \Pi A\, \Pi t : A\, \Pi p : t =_A t\, p =_{t=_A t} p$$

$\hookrightarrow$ equality in Martin-Löf type theory has a non-trivial groupoid (= category with inversible morphisms) interpretation, i.e.:

A type	$=$	A is a groupoid
$t : A$	$=$	$t \in \mathsf{Ob}_A$
$t =_A u$	$=$	$\mathsf{Hom}_A(t, u)$
refl t	$=$	id_t
J $t\, u$	$=$	$\ldots$

The types-as-spaces correspondence: Awodey-Warren and Voevodsky, 2006

Hofmann and Streicher's interpretation extends to iterated identity:

$$
\begin{aligned}
& t =_A u \\
& p =_{t=_A u} q \\
& \alpha =_{p=_{t=_A u} q} \beta \\
& \ldots
\end{aligned}
$$

Reflexivity, symmetry, transitivity, congruence, substitutivity at all levels give rise to an ω-groupoid.

In turn, ω-groupoids are models of topological spaces up to homotopy.

There are different variants of ω-groupoids, strict (= definitional), weak (= propositional), or semi-strict.

The univalence axiom: equality of type is equivalence of types

The groupoid approach (Hofmann-Streicher) suggests to restate equality of types as isomorphism:

$$A \simeq_{iso} B \triangleq \left\{ \begin{array}{l} f : A \Rightarrow B \\ g : B \Rightarrow A \\ \alpha : \Pi a : A.\, g(f(a)) =_A a \\ \beta : \Pi b : B.\, f(g(b)) =_B b \end{array} \right\}$$

Hofmann-Streicher's *universe extensionality* (1995):

$$(A =_{\mathsf{type}} B) \simeq_{iso} (A \simeq_{iso} B)$$

The univalence axiom: equality of type is equivalence of types

This generalises to ω-groupoids by considering a coherent form of isomorphism, *equivalence*:

$$A \simeq_{\mathsf{equiv}} B \triangleq \left\{ \begin{array}{l} f : A \Rightarrow B \\ g : B \Rightarrow A \\ \alpha : \Pi a : A.\, g(f(a)) =_A a \\ \beta : \Pi b : B.\, f(g(b)) =_B b \\ \mathsf{coh} : \Pi a : A.\, f(\alpha(a)) =_{f(g(f(a)))=_B f(g(f(a)))} \beta(f(a)) \end{array} \right\}$$

Voevodsky's *univalence axiom*:

$$(A =_{\mathsf{type}} B) \simeq_{equiv} (A \simeq_{equiv} B)$$

Notes:

- The univalence axiom also generalises propositional extensionality
- The univalence axiom actually also implies functional extensionality

Equality as path

Cubical Type Theory (Coquand *et al*, from 2013) reinterprets equality as a path over a formal interval

- Postulate a formal interval $\mathbb{I} \triangleq [0; 1]$ and treat equality as if characterised by

$$t =_A u \triangleq \{f : \mathbb{I} \Rightarrow A \mid f0 \equiv t \wedge f1 \equiv u\}$$

- This notion of equality generalises into a (cubical) "equality over": $t =_\epsilon u$ depends on a proof $\epsilon : A = B$ (i.e. itself $\epsilon : \mathbb{I} \Rightarrow \mathsf{U}$) stating that the type A of t is equal to the type B of u

$$t =_\epsilon u \triangleq \{f : (\Pi i : \mathbb{I}.\, \epsilon\, i) \mid f0 \equiv t \wedge f1 \equiv u\}$$

An analysis of the contributions of Cubical Type Theory

- It decomposes equality as a path: abstraction/application allows to enter or conceal dimensions and reason within these dimensions.

  This provides functoriality (at all dimensions) and function extensionality which otherwise would have to be expressed by proper combinators.

  This can (a posteriori) be seen as *iterated parametricity* in *direct style*.

- It introduces *equality over* as a "consistent" heterogeneous equality (compare to Observational Type Theory which uses John Major equality).

  This allows to internalise a *cubical* geometrical shape in type theory (which otherwise is globular)[1].

- It provides an open box composition/filling structure which extends transport/ substitutivity (together with specific definitional rules).

- An extra "gluing" operation provides *univalence*.

[1]This cubical structure can natively be equipped with algebraic structure echoing to logical structural rules: contraction (cartesian structure with diagonals), exchange (symmetric group of permutation), as well as symmetry (providing inverses called reversals), connections (for oblique commutative diagrams); this structure can be given either by term combinators or by interval combinators (e.g. one gets inverse either by adding a term operation p^{-1} or by adding an interval operation $-i$). There is also a room of manoeuvre about which properties of this structure is definitional (for instance, one would like $(p^{-1})^{-1} \equiv p$, resp. $--i = i$).

Our own approach of Cubical Type Theory

- Equality on types is *defined* to be equivalence.
- Equivalence is enough to provide the substitutivity/transport/composition/filling structure.

  This structure is "minimalistic" and we believe it is definitionally compatible with the rule $J_P\,\mathsf{refl}\,t \equiv t$.
- It is aimed to be iterated univalent parametricity in direct style and we inherit definitional rules from it.
- In particular, abstraction/application over a variable in the formal interval are seen as operations.

Core Cubical Type Theory

Core equality structure in Cubical Type Theory

Syntax

$$\begin{array}{lll} r & ::= & i \mid 0 \mid 1 \\ \Gamma & ::= & \ldots \mid \Gamma, i \\ t, A, p, \epsilon & ::= & \ldots \mid t =_\epsilon u \mid \lambda i.t \mid pr \end{array}$$

Typing rules

$$\frac{\Gamma \vdash \epsilon : A =_{\lambda i.\mathsf{U}_n} B \qquad \Gamma \vdash t : A \qquad \Gamma \vdash u : B}{\Gamma \vdash t =_\epsilon u : \mathsf{U}_n}$$

$$\frac{\Gamma, i \vdash t : A}{\Gamma \vdash \lambda i.t : t[0/i] =_{\lambda i.A} t[1/i]} \qquad \frac{\Gamma \vdash v : t =_\epsilon u \qquad FV(r) \in \Gamma}{\Gamma \vdash vr : \epsilon r}$$

Conversion rules

$$\frac{\Gamma \vdash p : t =_\epsilon u \qquad i \text{ fresh}}{\Gamma \vdash \lambda i.(pi) \equiv p : t =_\epsilon u} \qquad \frac{\Gamma, i \vdash t : A \qquad FV(r) \in \Gamma}{\Gamma \vdash (\lambda i.t)\, r \equiv t[r/i] : A[r/i]}$$

$$\frac{\Gamma \vdash p : t =_\epsilon u}{\Gamma \vdash p\, 0 \equiv t : \epsilon\, 0} \qquad \frac{\Gamma \vdash p : t =_\epsilon u}{\Gamma \vdash p\, 1 \equiv u : \epsilon\, 1}$$

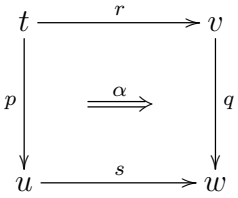
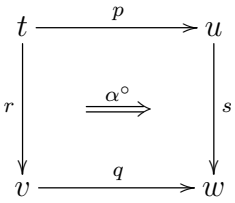
This is considered on top of an ambiant type theory with sorts U_n and types $\Sigma a : A.B$, $\Pi a : A.B$, ...

Examples

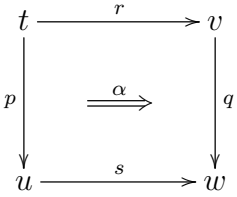
- *reflexivity*: $\widehat{t} \triangleq \lambda i.t$, for i fresh and t of type A, shall represent a proof of $t =_{\widehat{A}} t$
- *functoriality*: if $f : A \Rightarrow B$ and $p : t =_{\widehat{A}} u$ then $\lambda i.f(pi)$ is a proof of $ft =_{\widehat{B}} fu$
- *dependent functoriality*: if $f : \Pi a : A.B$ and $p : t =_{\widehat{A}} u$ then $\lambda i.f(pi)$ is a proof of $ft =_{\lambda i.B[pi/a]} fu$
- *functional extensionality* trivially provable (swap term variable with direction variable): if $p : \Pi a\!:\!A.\,(f_0\,a =_{\lambda i.B} f_1\,a)$ then $\lambda ia.pa\,i : f_0 =_{\lambda i.\Pi a:A.\,B} f_1$

Further examples

- commutation of sum with equality: if $p : t =_{\lambda i.\Sigma a{:}A.B} u$ then $\lambda i.\mathsf{snd}(pi)$ proves $\mathsf{snd}(t) =_{\lambda i.B[\mathsf{fst}(pi)/a]} \mathsf{snd}(u)$.
- nestings of equality have a *cubical* structure, stable by *permutation*

e.g. if $\alpha : p \underset{r \widetilde{=}_E s}{=} q$ (geometrically ), then $\alpha^\circ \triangleq \lambda ij.\alpha\, ji : r \underset{p \widetilde{=}_{E^\circ} q}{=} s$ ()

where we used the abbreviation $v \widetilde{=}_\xi w \triangleq \lambda i.(v\, i =_{\xi\, i} w\, i)$.

- *diagonals*: if $\alpha : p \underset{r \widetilde{=}_E s}{=} q$ (geometrically ) then $\Delta\alpha \triangleq \lambda i.\alpha ii$ proves $t =_{\Delta E} w$
- supports reasoning with equality *over* an equality without breaking the symmetry

$$v_1 =_{\lambda i.vect\ (p\ i)} v_2 \quad \text{whenever} \quad p : n_1 =_{\widehat{\mathbb{N}}} n_2$$

- appropriate to compute with *Higher Inductive Types* (HITs), and especially with quotients

Cubical equality encourages to reason by pointwise transport

Let $f : A \Rightarrow A$ and $p : \Pi a\, f(a) =_{\widehat{A}} a$. For $a : A$, let us prove that $f(pa) = p(fa)$ where $f(pa)$ is functorial application of f, i.e. $\lambda i.f(pai)$.

We need to find a "continuous" term q that evaluates into $p(fa)$ in 0 and in $f(p(a)$ in 1. To connect these terms, it is convenient to rephrase them into

$$\lambda i.f(p(\mathsf{id}\, a)i)$$

and

$$\lambda i.\mathsf{id}(p(fa)i)$$

(using η and β-expansions) so as to expose the similarity of structure. Then, for any t, the equation $ft \stackrel{?}{=} \mathsf{id}\, t$ unifies along the interval if we can find a term $?q'$ such that $ft \equiv ?q'0$ and $\mathsf{id}\, t \equiv ?q'1$. The solution is $?q' \triangleq pt$. Similarly, $\mathsf{id}\, t \stackrel{?}{=} ft$ unifies along the interval by setting $?q'' \triangleq \overline{pt}$ where $\overline{e}$ denotes a proof of $v = w$ whenever e proves $w = v$. It finally suffices to combine this into a unifier of the original problem:

$$\begin{array}{rcl}
?q\,0 & = & \lambda i.\ f\ (p\ (\mathsf{id}\ a\ \ )\ i) \\
?q\,1 & = & \lambda i.\ \mathsf{id}\ (p\ (f\ a\ \ )\ i) \\
\hline
?q\,j & = & \lambda i.\ p\ (p\ (\overline{p\ a}\ \ j)\ i)\ j
\end{array}$$

Hence $q \triangleq \lambda j.\lambda i.p(p(\overline{p\, a}\, j)\, i)\, j$

Focus: quotients in cubical type theory

An extension of Coq with cubical equality would support the following definition of quotients:

```
Inductive quo A R :=
| quo_in : A -> quo A R
| quo_eq : forall a b : A, R a b -> quo_in A R a = quo_in A R b.

Arguments quo_in {A} {R} _.
Arguments quo_eq {A} {R} _.

Definition quo_rect A R
  (P : quo A R -> Type)
  (f: forall a:A, P (quo_in a))
  (resp: forall a b (e:R a b), f a =_{fun i => P (quo_eq a b H i)} f b)
  (x:quo A R) : P x :=
  match x with
  | quo_in a => f a
  | quo_eq a b e i => resp a b e i
  end.
```

Typing rules for 1-sphere

introduction rules

$$\frac{}{\Gamma \vdash \mathsf{base} : \mathbb{S}^1}$$

$$\frac{}{\Gamma \vdash \mathsf{loop} : \mathsf{base} =_{\mathbb{S}^1} \mathsf{base}}$$

elimination rule

$$\frac{\Gamma \vdash p : \mathbb{S}^1 \qquad \Gamma, a : \mathbb{S}^1 \vdash A : \mathsf{U}_n \qquad \begin{array}{c}\Gamma \vdash t : A[\mathsf{base}/a] \\ \Gamma \vdash e : t =_{\lambda i.A[\mathsf{loop}\, i/a]} t\end{array}}{\Gamma \vdash \mathsf{case}\ p\ \mathsf{of}\ \mathsf{base}\ \Rightarrow\ t \mid \mathsf{loop}\, i\ \Rightarrow\ e\, i\ \mathsf{end} : A[p/a]}$$

reduction and observational rules

$$\frac{\Gamma \vdash t : A[\mathsf{base}/a] \qquad \Gamma \vdash e : t =_{\lambda i.A[\mathsf{loop}\, i/a]} t}{\Gamma \vdash \mathsf{case}\ \mathsf{base}\ \mathsf{of}\ \mathsf{base}\ \Rightarrow\ t \mid \mathsf{loop}\, i\ \Rightarrow\ e\, i\ \mathsf{end} \equiv t : A[\mathsf{base}/a]}$$

$$\frac{\Gamma \vdash t : A[\mathsf{base}/a] \qquad \Gamma \vdash e : t =_{\lambda i.A[\mathsf{loop}\, i/a]} t}{\Gamma \vdash \mathsf{case}\ \mathsf{loop}\, r\ \mathsf{of}\ \mathsf{base}\ \Rightarrow\ t \mid \mathsf{loop}\, i\ \Rightarrow\ e\, i\ \mathsf{end} \equiv e\, r : A[\mathsf{loop}\, r/a]}$$

$$\frac{\Gamma \vdash p : \mathbb{S}^1}{\Gamma \vdash \mathsf{case}\ p\ \mathsf{of}\ \mathsf{base}\ \Rightarrow\ \mathsf{base} \mid \mathsf{loop}\, i\ \Rightarrow\ \mathsf{loop}\, i\ \mathsf{end} \equiv p : \mathbb{S}^1}$$

Extending Core Cubical Type Theory

Adding symmetry of equality

There are three ways to provide symmetry of equality in Cubical Type Theory:

- Directly add an operator of symmetry:

$$\frac{\Gamma \vdash p : t =_{\epsilon} u}{\Gamma \vdash p^{-1} : u =_{\epsilon^{-1}} t}$$

  + rules of commutation between symmetry and other constructions

- Add a symmetry operator $-r$ in the structure of the interval:

$$r \ ::= \ \ldots \mid -r$$

  together with rules:

$$\begin{aligned} -0 &\equiv 1 \\ -1 &\equiv 0 \\ --r &\equiv r \end{aligned}$$

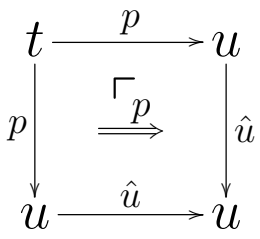
  Then, $p^{-1} \triangleq \lambda i.p\,(-i)$.

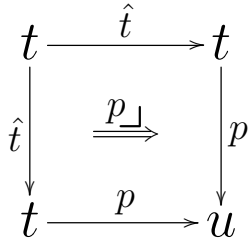
- Derive symmetry from composition (see later)

The need for "connections"

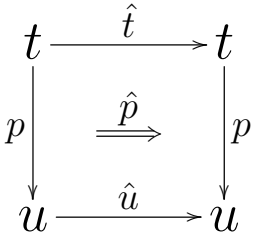
To support dependent transport, the cubical structure shall also need twisted variants of reflexivity applicable to paths:

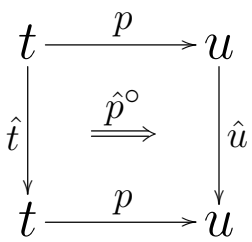
If p is a path $t \xrightarrow{p} u$, then, the North-West and South-East reflexivities of p are the squares





To be compared to the regular (West-East) and swapped regular (North-South) reflexivities of p:





$$\hat{p} \triangleq \lambda i.p : p =_{\lambda j.(t=u)} p \qquad \hat{p}^{\circ} \triangleq \lambda j.\lambda i.p\, j : \hat{t} =_{\lambda j.(pj=pj)} \hat{u}$$

The need for "connections"

These twisted reflexivities can be obtained by adding to the structure of the interval what is called *connections*:

$$r, r' \ ::= \ ... \mid r \wedge r' \mid l \vee r'$$

together with the conversion rules

$$\begin{array}{rcl@{\qquad}rcl} r \wedge 0 & \equiv & 0 & r \vee 1 & \equiv & 1 \\ 0 \wedge r & \equiv & 0 & 1 \vee r & \equiv & 1 \\ 1 \wedge 1 & \equiv & 1 & 0 \vee 0 & \equiv & 0 \\ -(r \wedge r') & \equiv & (-r) \vee (-r') & -(r \vee r') & \equiv & (-r) \wedge (-r') \end{array}$$

Then, we can define:

$$\begin{array}{rcl} \ulcorner p & \triangleq & \lambda i.\lambda j.p\,(i \vee j) \\ p_{\lrcorner} & \triangleq & \lambda i.\lambda j.p\,(i \wedge j) \end{array}$$

which have the right typing:

$$\begin{array}{rcl@{\qquad}rcl} \ulcorner p\,0\,0 & \triangleq & t & p_{\lrcorner}\,0\,0 & \triangleq & t \\ \ulcorner p\,0\,1 & \triangleq & u & p_{\lrcorner}\,0\,1 & \triangleq & t \\ \ulcorner p\,1\,0 & \triangleq & u & p_{\lrcorner}\,1\,0 & \triangleq & t \\ \ulcorner p\,1\,1 & \triangleq & u & p_{\lrcorner}\,1\,1 & \triangleq & u \end{array}$$

and similarly for the sides of the square.

Adding composition and transport

We want to add transport, i.e. that, from $t =_{\hat{A}} u$, and Pt, we get Pu.

We shall make a detour and get it as a particular case of functoriality and univalence:

- 1. from $t =_{\hat{A}} u$ and $P : A \Rightarrow U$, get $Pt =_{\hat{U}} Pu$
- 2. from $Pt =_{\hat{U}} Pu$, get $Pt \simeq Pu$
- 3. from $Pt \simeq Pu$, directly get a morphism from Pt to Pu (and actually also one from Pu to Pt)

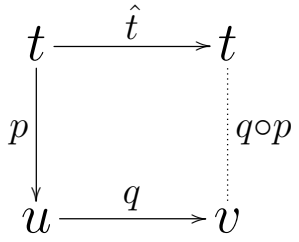
Where in step 2, we identified equality on types as equivalence, as univalence would do.

We are thus left to relate our previous definition of equality with equivalence.

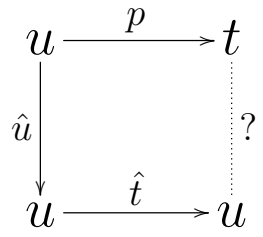
Deriving transitivity

In passing, transport shall give us transitivity (and symmetry) back:

Transitivity: From $q : u =_{\hat{A}} v$, we get a proof $(\lambda i.t = qi) : (t =_{\hat{A}} u) =_{\hat{U}} (t =_{\hat{A}} v)$, thus, by transport of p:



Symmetry: From $p : t =_{\hat{A}} u$, we get a proof of $(\lambda i.pi = u) : (t =_{\hat{A}} t) =_{\hat{U}} (u =_{\hat{A}} t)$, thus, by transport of $\hat{t}$:



A symmetric definition of equivalence

(exercise 4.2 of the HoTT Book)

We extend the theory with a record type $A \simeq_s B$ (equivalence in sort s) defined as follows. If $\epsilon : A \simeq_s B$, the following projections are available:

$$
\begin{array}{ll}
=_\epsilon & : A \Rightarrow B \Rightarrow \mathsf{U}_n; \\
\overrightarrow{\epsilon} & : A \Rightarrow B; \\
\overRightarrow{\epsilon} & : \Pi a : A.\, a =_\epsilon \overrightarrow{\epsilon} a; \\
\overrightarrow{\mathsf{coe}_\epsilon} & : \Pi a : A . \Pi b : B.\, a =_\epsilon b \Rightarrow \overrightarrow{\epsilon} a =_B b \\
\overRightarrow{\mathsf{coe}_\epsilon} & : \Pi a : A . \Pi b : B . \Pi p : a =_\epsilon b.\, \overRightarrow{\epsilon} a =_{\lambda i.(a =_\epsilon \overrightarrow{\mathsf{coe}_\epsilon}(p) i)} p \\
\overleftarrow{\epsilon} & : B \Rightarrow A; \\
\overLeftarrow{\epsilon} & : \Pi b : B.\, \overleftarrow{\epsilon}(b) =_\epsilon b; \\
\overleftarrow{\mathsf{coe}_\epsilon} & : \Pi a : A . \Pi b : B.\, a =_\epsilon b \Rightarrow a =_A \overleftarrow{\epsilon} b \\
\overLeftarrow{\mathsf{coe}_\epsilon} & : \Pi a : A . \Pi b : B . \Pi p : a =_\epsilon b.\, p =_{\lambda i.(\overleftarrow{\mathsf{coe}_\epsilon}(p) i =_\epsilon b)} \overLeftarrow{\epsilon}(b)
\end{array}
$$

In particular, setting $(A =_{\widehat{s}} B) \triangleq (A \simeq_s B)$, substitutivity shall become a consequence of $t =_\xi u \Rightarrow P(t) =_{\widehat{s}} P(u)$

Excerpt of rules defining $\lambda i.A$ as a proof of equivalence (ad hoc polymorphism)

Excerpt of the semantics of $\widehat{\mathsf{U}_n}$:

$$(A =_{\widehat{\mathsf{U}_n}} B) \equiv A \simeq_{\mathsf{U}_n} B$$

$$\overrightarrow{\widehat{\mathsf{U}_n}}\, A \equiv A \qquad \overleftarrow{\widehat{\mathsf{U}_n}}\, B \equiv B$$

$$\overrightarrow{\widehat{\mathsf{U}_n}}\, A \equiv \widehat{A} \qquad \overleftarrow{\widehat{\mathsf{U}_n}}\, A \equiv \widehat{A}$$

Excerpt of the semantics of $\lambda i.\Sigma a : A.B$:

$$\begin{aligned}
(t =_{\lambda i.\Sigma a:A.B} u) &\equiv \Sigma a : (\mathsf{fst}\, t =_{\lambda i.A} \mathsf{fst}\, u).(\mathsf{snd}\, t =_{\lambda i.B[\mathsf{fst}\,(ai)/a]} \mathsf{snd}\, u) \\
\overrightarrow{\lambda i.\Sigma a : A.B}\, t &\equiv (\overrightarrow{\lambda i.A}(\mathsf{fst}\, t), \overrightarrow{\lambda i.B[\overrightarrow{\lambda i.A}\, a\, i/a]}(\mathsf{snd}\, t)) \\
\overleftarrow{\lambda i.\Sigma a : A.B}\, t &\equiv (\overleftarrow{\lambda i.A}(\mathsf{fst}\, t), \overleftarrow{\lambda i.B[\overleftarrow{\lambda i.A}\, a\, i/a]}(\mathsf{snd}\, t)) \\
\overrightarrow{\overrightarrow{\lambda i.\Sigma a : A.B}}\, t &\equiv (\overrightarrow{\overrightarrow{\lambda i.A}}(\mathsf{fst}\, t), \overrightarrow{\lambda i.B[\overrightarrow{\overrightarrow{\lambda i.A}}\, a\, i/a]}(\mathsf{snd}\, t)) \\
\overleftarrow{\overleftarrow{\lambda i.\Sigma a : A.B}}\, t &\equiv (\overleftarrow{\overleftarrow{\lambda i.A}}(\mathsf{fst}\, t), \overleftarrow{\lambda i.B[\overleftarrow{\overleftarrow{\lambda i.A}}\, a\, i/a]}(\mathsf{snd}\, t))
\end{aligned}$$

And similar other rules, including for $A \simeq_s B$ and $\Pi a : A.B$ (though the design for the latter is not yet stabilised)

Getting univalence for free

We conjecture that we can justify by parametricity the following rule giving univalence by definition:

$$\frac{\Gamma \vdash A \simeq_{\mathsf{U}_n} B}{\Gamma \vdash A =_{\hat{\mathsf{U}}_n} B}$$

Works on "parametric"-like interpretations of type theory

Takeuti (1953), Gandy (1956): setoid interpretation in Church's simple type theory

Hofmann (1995), Altenkirch (1999): setoid interpretation in type theory

Altenkirch-McBride-Swiestra (2007): setoid interpretation in direct style (Observational Type Theory)

Licata-Harper (2012): two-dimensional type theory

Barras-Coquand-Huber (2015): semi-simplicial interpretation

Bernardy-Coquand-Moulin (2015): iterated parametricity in direct style

Altenkirch-Kaposi (2015): towards univalent parametricity

Tabareau-Tanter-Sozeau (2018): univalent parametricity at dimension 1

More generally, a motto is that we should eventually have a "polysemy" between some type theory in direct style, a corresponding indirect interpretation type theory by translation, a corresponding higher-dimensional presheaf interpretation.

In particular, we generalise Bernardy-Coquand-Moulin into an iterated *univalent* parametricity translation (in progress).

A bit of terminology

Let $\Gamma \vdash t : A$ be a sequent in a given object language.

Semantic realisability interprets Γ and A as sets of realisers in some language and proves:

$$\Gamma \vdash t : A \text{ implies that for all } \gamma \in |\Gamma|, [\![t]\!]_\gamma \in |A|_\gamma$$

Syntactic realisability does the same, but within another (or the same) object language:

$$\Gamma \vdash t : A \text{ implies } \gamma \,\mathsf{r}\, \Gamma \vdash [\![t]\!]_\gamma : t \,\mathsf{r}\, A$$

for some internal definition of $t \,\mathsf{r}\, A$ and some translation of proofs $[\![t]\!]$.

For instance, Kreisel's syntactic realisability defines

$$t \,\mathsf{r}\, (A \Rightarrow B) \triangleq \Pi a : |A| \,(a \,\mathsf{r}\, A \Rightarrow t\, a \,\mathsf{r}\, B)$$

Type variables are interpreted in realisability as sets of realisers. Parametricity differs in using (possibly heteregenous) instead relations. However, realisability is generally enough to get theorems for free.

Different variants of parametricity can be defined (e.g. Tabareau-Tanter-Sozeau *univalent parametricity*).

Expressiveness of a translation

Syntactic interpretations extend the strength of a formalism.

For instance:

- Kleene's realisability for Heyting Arithmetic exactly captures the strength of Heyting Arithmetic with Markov's principle and the extended Church's Thesis.

- Kreisel's realisability for Heyting Arithmetic in finite types exactly captures the intensional axiom of choice and the independence of premisses.

- Gödel's Dialectica realisability for Heyting Arithmetic in finite types exactly captures the intensional axiom of choice, the independence of premisses and Markov's principle.

One step of the parametricity translation, example

The translation of a sequent

$$A : \mathsf{U}_n, a : A \vdash t : C$$

has the form

$$\begin{array}{l} A_0 : \mathsf{U}_n, \quad A_1 : \mathsf{U}_n, \quad A_\star : A_0 \times A_1 \Rightarrow \mathsf{U}_n, \\ a_0 : A_0, \quad a_1 : A_1, \quad a_\star : A_\star(a_0, a_1) \\ \vdash \\ (t_0, t_1, t_\star) : \Sigma c_0 : C_0 . \Sigma c_1 : C_1 . C_\star(c_0, c_1) \end{array}$$

Iterated parametricity, example

The declaration of a variable of axis increases the level of nesting of the parametricity translation. The translation of a sequent

$$A : \mathsf{U}_n, a : A, i, B : \mathsf{U}_n, b : A, j \vdash t : C$$

has the form

$$\begin{array}{l}
A : \mathsf{U}_n \\
a : A \\
B_0 : \mathsf{U}_n, \quad B_1 : \mathsf{U}_n, \quad B_\star : B_0 \times B_1 \Rightarrow \mathsf{U}_n, \\
b_0 : A_0, \quad b_1 : A_1, \quad b_\star : B_\star(b_0, b_1) \\
\vdash \\
((t_{00}, t_{01}, t_{0\star}), (t_{10}, t_{11}, t_{1\star}), (t_{\star 0}, t_{\star 1}, t_{\star\star})) \\
: \Sigma c_0 : (\Sigma c_{00} : C_{00}.\Sigma c_{01} : C_{01}.C_{0\star}(c_{00}, c_{01})) \\
\ \Sigma c_1 : (\Sigma c_{10} : C_{10}.\Sigma c_{11} : C_{11}.C_{1\star}(c_{10}, c_{11})) \\
\ \Sigma c_{\star 0} : C_{\star 0}(c_{00}, c_{01}).\Sigma c_{\star 1} : C_{\star 1}(c_{01}, c_{11}).C_{\star\star}(c_{0\star}, c_{1\star}, c_{\star 0}, c_{\star 1})
\end{array}$$

Note: up to some currification...

The iterated parametricity translation

Excerpt of ingredients:

- To traverse the axis variables and implement the axiom rule, (e.g. $a : A, i \vdash a : A$), reflexivities are used (with interpretation $a : A \vdash (a, a, \mathsf{refl}\, a) : \Sigma a : A.\Sigma a : A.a = a$ here).

- To anticipate an arbitrary increase in dimension by reduction, types have to be interpreted by an infinite stream $(A, =_A, =_{=_A}, ...)$ of relations in all dimensions, so that, e.g., in $A : \mathsf{U}_n, a : A \vdash a =_{\lambda i.A} a$, the equal sign is interpreted as a component of the interpretation of A.

Outcome:

- The "effect" of the translation is to equip types with a higher-dimensional structure of equalities

- Extensionality of pairs holds by definition

- We expect univalence to hold by definition

- We expect function extensionality to hold by definition (in the 0-dimensional context)