Outline

Lectures 1–3: Rules of type theory
w/ $\Pi, \Sigma, \emptyset, \mathbb{1}, \mathbb{N}, +, =$
(§1–5 of Rijke)

Today:
- Universes $\mathcal{U}$ (§6)
- Propositions as types, aka the Curry–Howard interpretation (§7)

Why do we need universes?

1) To prove $0 \neq 1$ in $\mathbb{N}$, i.e., $(0 =_{\mathbb{N}} 1) \to \emptyset$

 – more generally, define type families by induction.

2) To write polymorphic terms, e.g., instead of def.

$id_A : A \to A$, for each type A, with a universe $\mathcal{U}$

we have $id^{\mathcal{U}} : \prod_{X:\mathcal{U}} \mathcal{T}(X) \to \mathcal{T}(X)$

3) To do category theory in a streamlined way,

$\rightsquigarrow$ Grothendieck universes.

What is a universe?

$\{\Pi, \Sigma, \emptyset, \mathbb{1}, \mathbb{N}, +, =\}$

<u>Slogan</u>: Whatever we can do with types, we can do with a universe.

(Just as $=$ is an internalization of $\doteq$, a universe $\mathcal{U}$ is an internalization of (A type))

<u>Def</u> (at meta level) A universe $(\mathcal{U}, \mathcal{T})$ is a type family

$$\mathcal{U} \text{ type}, \quad X : \mathcal{U} \vdash \mathcal{T}(X) \text{ type}$$

with:

- $\check{\Pi} : \Pi_{X:\mathcal{U}}((\mathcal{T}(X) \to \mathcal{U}) \to \mathcal{U})$ s.t. $\mathcal{T}(\check{\Pi}(X,Y)) \doteq \prod_{x:\mathcal{T}(X)} \mathcal{T}(Y_x)$ for every $X : \mathcal{U}$, $Y : \mathcal{T}(X) \to \mathcal{U}$
- $\check{\Sigma} : \Pi_{X:\mathcal{U}}((\mathcal{T}(X) \to \mathcal{U}) \to \mathcal{U})$ s.t. $\mathcal{T}(\check{\Sigma}(X,Y)) \doteq \sum_{x:\mathcal{T}(X)} \mathcal{T}(Y_x)$
- $\check{\emptyset}, \check{\mathbb{1}}, \check{\mathbb{N}} : \mathcal{U}$ s.t. $\mathcal{T}(\check{\emptyset}) \doteq \emptyset$, $\mathcal{T}(\check{\mathbb{1}}) \doteq \mathbb{1}$, $\mathcal{T}(\check{\mathbb{N}}) \doteq \mathbb{N}$
- $\_\check{+}\_ : \mathcal{U} \to \mathcal{U} \to \mathcal{U}$ s.t. $\mathcal{T}(X \check{+} Y) \doteq \mathcal{T}(X) + \mathcal{T}(Y)$

& $\_ \overset{\vee}{=} \_ : \prod_{X : \mathcal{U}} (\mathcal{T}(X) \to \mathcal{T}(X) \to \mathcal{U})$

s.t. $\mathcal{T}(x \overset{\vee}{=} y) \doteq (x =_{\mathcal{T}(X)} y)$ for every $X : \mathcal{U}$, $x, y : \mathcal{T}(X)$ $\square$

Assume enough universes

Postulate: Whenever we have finitely many type families:

$$\Gamma_1 \vdash A_1 \text{ type} \quad , \ldots , \quad \Gamma_n \vdash A_n \text{ type}$$

there is a universe $(\mathcal{U}, \mathcal{T})$ in the empty context that contains these:

w/ terms $\Gamma_i \vdash \check{A}_i : \mathcal{U}$ s.t. $\Gamma_i \vdash \mathcal{T}(\check{A}_i) \doteq A_i$ type for all i.

Examples

- $n=0$ (no type families): we get a base universe $(\mathcal{U}_0, T_0)$
- If $(\mathcal{U}, \tau)$ is a universe, then there is a successor universe $(\mathcal{U}^+, \tau^+)$: $\vdash \mathcal{U}$ type $\rightsquigarrow$ $\vdash \check{\mathcal{U}} : \mathcal{U}^+$ & $\vdash \tau^+(\check{\mathcal{U}}) \doteq \mathcal{U}$

$X : \mathcal{U} \vdash \tau(X)$ type $\rightsquigarrow$ $X : \mathcal{U} \vdash \check{\tau}(X) : \mathcal{U}^+$ &

$X : \mathcal{U} \vdash \tau^+(\check{\tau}(X)) \doteq \tau(X)$

- If $(\mathcal{U}, \tau_{\mathcal{U}}), (\mathcal{V}, \tau_{\mathcal{V}})$ are universes, then we have a universe $(\mathcal{U} \sqcup \mathcal{V}, \tau_{\mathcal{U} \sqcup \mathcal{V}})$

$\mathcal{U}$ type

$X : \mathcal{U} \vdash \tau_{\mathcal{U}}(X)$ type

$\mathcal{V}$ type

$X : \mathcal{V} \vdash \tau_{\mathcal{V}}(X)$ type

$\check{\mathcal{U}} : \mathcal{U} \sqcup \mathcal{V}$

$X : \mathcal{U} \vdash \check{\tau}_{\mathcal{U}}(X) : \mathcal{U} \sqcup \mathcal{V}$

$\check{\mathcal{V}} : \mathcal{U} \sqcup \mathcal{V}$

$X : \mathcal{V} \vdash \check{\tau}_{\mathcal{V}}(X) : \mathcal{U} \sqcup \mathcal{V}$

s.t. $\tau_{\mathcal{U} \sqcup \mathcal{V}}(\check{\mathcal{U}}) \doteq \mathcal{U}$ etc.

Discussion

- In practice, we leave out the $T(-)$s, because this can be inferred from context (no pun intended).
- Universes are open-ended: no requirement that $\mathcal{U}_0, \mathcal{U}^+, \mathcal{U} \sqcup \mathcal{V}$ are minimal! + If/when we add new type formers, we'll want the universes to be closed under these. (W-types, HITs, ...)
- $\mathcal{U}_0, \mathcal{U}_0^+, \mathcal{U}_0^{++}, \ldots$ is a universe hierarchy, but there's no requirement that all types lie in this.

  OTOH, the reflection principle doesn't give more universes than these.
- We might expect, for a $\mathcal{U}, \mathcal{U}^+$, where $\mathrm{Lift} : \mathcal{U} \to \mathcal{U}^+$, "$\lambda X : \mathcal{U}, \check{T}(X)$"

  $\mathrm{Lift}(\check{\mathbb{N}}) \doteq \check{\mathbb{N}}^+ : \mathcal{U}^+$. This is not assumed, here and in Agda.

  It's called: cumulativity

Type in type?!

Could we have a universe $\mathcal{U}$, w/ $\check{\mathcal{U}} : \mathcal{U}$, $T(\check{\mathcal{U}}) \doteq \mathcal{U}$?

(This was the case in MLTT '71)

No! Girard showed in his '72 thesis this is inconsistent.

Simpler version due Hurkens '95 ⇝ see github.

If we further assume a gen. ind. type in $\mathcal{U}$, then a version of Russell's paradox appears.

$V :$ $\quad \mathrm{sup} : \prod_{X:\mathcal{U}} ((X \to V) \to V)$

$V : \mathcal{U}^+$

Larger universes?

Reflection on the universe postulate, we can propose larger universes, e.g., Palmgren's super-universe. ⇝ cf. large cardinals in set theory.

Universe polymorphism

For every universe $\mathcal{U}$, we now have $id^{\mathcal{U}} : \prod_{X:\mathcal{U}} (X \to X)$

$id^{\mathcal{U}} = \lambda X \, \lambda x . x$.

this applies to types in $\mathcal{U}$, not to all types, proof assistants (Agda, Coq, Lean, ...) have universe polymorphism.

mechanisms of different sorts.

In Agda, $\text{Type} \doteq \mathcal{U}_0$

$\text{Type1} \doteq \mathcal{U}_1$, $\rightsquigarrow$ universe level judgments

$\rightsquigarrow$ Level type.

Examples of universes in use

- $$\left.\begin{array}{l} \text{is-true} : \text{bool} \to \mathcal{U} \\ \text{is-true false} \doteq \emptyset \\ \text{is-true true} \doteq \mathbb{1} \end{array}\right\} \ni \text{true-not-false} : \text{true} =_{\text{bool}} \text{false} \to \emptyset$$

  by $\text{transport}^{\text{is-true}}_{p} : \mathbb{1} \to \emptyset$

  $p : \text{true} = \text{false}$

- Need a universe to do this, otherwise we a model, "types as propositions" model where X is a type is interpreted as a subset of $\mathbb{1}$.

Observational equality on $\mathbb{N}$

We want $\mathrm{Eq}\text{-}\mathbb{N} : \mathbb{N} \to \mathbb{N} \to \mathcal{U}$ that repr. $=_{\mathbb{N}}$.

$$\begin{aligned}
\mathrm{Eq}\text{-}\mathbb{N}\ 0\ 0 &\doteq \mathbb{1} \\
\mathrm{Eq}\text{-}\mathbb{N}\ 0\ (\mathrm{suc}\ m) &\doteq \emptyset \\
\mathrm{Eq}\text{-}\mathbb{N}\ (\mathrm{suc}\ n)\ 0 &\doteq \emptyset \\
\mathrm{Eq}\text{-}\mathbb{N}\ (\mathrm{suc}\ n)\ (\mathrm{suc}\ m) &\doteq \mathrm{Eq}\text{-}\mathbb{N}\ n\ m
\end{aligned}$$

or, w/ $\mathbb{N}$-elim:

$$\begin{aligned}
\mathrm{Eq}\text{-}\mathbb{N}\ n\ m :=\ \mathrm{ind}\text{-}\mathbb{N}^{x.\mathbb{N}\to\mathcal{U}}\Big(&\lambda y.\,\mathrm{ind}\text{-}\mathbb{N}^{z.\mathcal{U}}(\mathbb{1}, zX.\emptyset, y), && \leftarrow n \doteq 0 \\
&xf.\,\lambda y.\,\mathrm{ind}\text{-}\mathbb{N}^{z.\mathcal{U}}(\emptyset, zX.f\,z, y), && \leftarrow n \doteq \mathrm{suc}\ x \\
&n\Big)\ m
\end{aligned}$$

$(f : \mathbb{N} \to \mathcal{U})$

repr. $\mathrm{Eq}\text{-}\mathbb{N}(x, -)$

$\uparrow m = 0$ $\qquad \uparrow m = \mathrm{suc}\ z$

$\text{Eq-}\mathbb{N}\ 0\ 0 \doteq \mathbb{1}$
$\text{Eq-}\mathbb{N}\ 0\ (\text{suc}\ m) \doteq \emptyset$
$\text{Eq-}\mathbb{N}\ (\text{suc}\ n)\ 0 \doteq \emptyset$
$\text{Eq-}\mathbb{N}\ (\text{suc}\ n)\ (\text{suc}\ m) \doteq \text{Eq-}\mathbb{N}\ n\ m$

We prove this is reflexive, i.e.,

$$\prod_{n:\mathbb{N}} \text{Eq-}\mathbb{N}\ n\ n \quad \text{by ind on } n$$

Then by path ind. we get

$$\prod_{n,m:\mathbb{N}} \left(n = m \to \text{Eq-}\mathbb{N}\ n\ n \right).$$

By double ind. we get

$$g : \prod_{n,m:\mathbb{N}} \left(\text{Eq-}\mathbb{N}\ n\ m \longrightarrow n = m \right)$$

$$g\ 0\ 0\ c \doteq \text{refl}_0$$
$$g\ 0\ (\text{suc}\ m)\ c \doteq \emptyset\text{-ind}\ c$$
$$g\ (\text{suc}\ n)\ 0\ c \doteq \emptyset\text{-ind}\ c$$
$$g\ (\text{suc}\ n)\ (\text{suc}\ m)\ c \doteq \text{ap}\ \text{suc}\ (\overbrace{g\ n\ m\ c}^{: n = m}) : \text{suc}\ n = \text{suc}\ m$$

Curry–Howard (Propositions as types)

prop's P (math'l statements	types A
proof of P	terms/elts of A
eq. of proofs of P	judgmental eq. of terms
$\top$	$\mathbb{1}$
$\bot$	$\emptyset$
$P \Rightarrow Q$	$A \to B$
$\neg P$	$A \to \emptyset$
$P \vee Q$	$A + B$
$\forall_{x:A} P(x)$	$\Pi_{x:A} B(x)$
$\exists_{x:A} P(x)$	$\Sigma_{x:A} B(x)$
$x =_A y$	$x =_A y$

Brief history (p. 77)

1908: Brouwer, rejects LEM

20's, 30's: Heyting, Kolmogorov

1934: Curry

(1936: Turing machine)

1958: Curry–Feys

realizability (Kleene)

1969: Howard, for the case of arithmetic

type theory

Per Martin-Löf

Dana Scott

de Bruijn

etc.

Example: Divisibility on $\mathbb{N}$

Def If $k, n : \mathbb{N}$, $k \mid n$ type "there exist $d : \mathbb{N}$ s.t. $k \cdot d = n$"

$$(k \mid n) :\equiv \sum_{d:\mathbb{N}} k \cdot d = n$$

Prop For all n, $1 \mid n$ and $n \mid n$.

encoded as: $\prod_{n:\mathbb{N}} (1 \mid n) \times (n \mid n)$

w/ proof: $\lambda n. ((n, p_n), (1, q_n))$

$$p_n : 1 \cdot n = n \quad , \quad q_n : n \cdot 1 = n$$

Prop For all n, $n \mid 0$, $\rightsquigarrow$ $\prod_{n:\mathbb{N}} n \mid 0$ w/ proof/term

$$\lambda n \ (0, r_n), \ r_n : n \cdot 0 = 0$$

NB $0 \mid 0 \doteq \sum_{d:\mathbb{N}} \underbrace{0 \cdot d = 0}$

we have terms $(d, s_d) : 0 \mid 0$, $s_d : 0 \cdot d = 0$.

<u>Example</u> Type theoretic axiom of choice:

Suppose A, B types, $x : A, y : B \vdash R(x,y)$ type

(correspondence/ heterogeneous rel.)

Then $$\left(\prod_{x:A} \sum_{y:B} R(x,y)\right) \leftrightarrow \left(\sum_{f:A\to B} \prod_{x:A} R(x, f\,x)\right)$$

(C-H int. of AC, $\to$)

"$\to$" $\quad H \longmapsto ((\lambda x.\, pr_1(Hx)),\ (\lambda x.\, pr_2(Hx)))$

"$\leftarrow$" $\quad (f,h) \longmapsto \lambda x.\, (f\,x, h\,x)$

(Σ-ind)

We'll return later to the "real AC"