Dependent types

Elim-rules for positive types are induction principles

$$\frac{z:\mathbb{N} \vdash P : \mathcal{U} \qquad b : [0/z]P \qquad x:\mathbb{N},\ y:[x/z]P \vdash c : [\mathrm{succ}(x)/z]P}{z:\mathbb{N} \vdash \underbrace{\mathrm{rec}[b, x, y.c](z)}_{\text{witness}} : P(z)}$$

Take advantage of deptypes, with QUANTIFIERS Π, Σ negative

Dependent function space
Dependent product
Generalized product

$$\frac{A : \mathcal{U} \qquad x : A \vdash B : \mathcal{U}}{\Pi x{:}A.\ B : \mathcal{U}}\ (\Pi\text{-F})$$

kind of technical

sometimes B is a family $B = \{B(x)\}_{x:A}$

A-indexed product

intuitively if $A = \{a_1, a_2 \dots\}$
then $\prod_{x:A} B : \mathcal{U} = \langle b_1, b_2 \dots \rangle$
w/ $b_1 : B(a_1)$, $b_2 : B(a_2) \dots$

if B is constant,
then $\prod_{x:A} B =: A \to B$

alt. note: $x{:}A \to B(x)$
$A \to B$

dependence!

$$\frac{x:A \vdash b(x) : B(x)}{\lambda x.b : \Pi x{:}A.B}\ (\Pi\text{-I})$$

$(\lambda x.b)(a) \equiv [a/x]b$

$$\frac{b : \Pi x{:}A.B \qquad a : A}{b(a) : [a/x]B}\ (\Pi\text{-E})$$

unicity (η)
$\lambda x.\,b(x) \equiv b$ ⚠

$\Pi x{:}A.B \Leftrightarrow \forall x{:}A.\ B$

domain of quantification — prop

(usually! But this could be a prop, making proof relevant mathematics)

more informative

Dependent sum
Dependent product *
Generalized sum

$$\frac{A : \mathcal{U} \qquad x : A \vdash B : \mathcal{U}}{\Sigma x : A . B : \mathcal{U}} \; (\Sigma\text{-F})$$

$$\frac{a : A \qquad b : [a/x]B}{\langle a, b \rangle : \Sigma x : A . B} \; (\Sigma\text{-I})$$

$$A \times B := \sum_{- : A} B \qquad x : A \times B$$

$$\frac{c : \Sigma x : A . B}{\mathrm{fst}(c) : A} \; (\Sigma\text{-E-L})$$

ex) derive $A \times B$ from Π

$$\frac{c : \Sigma x : A . B}{\mathrm{snd}(c) : [\mathrm{fst}(c)/x]B} \; (\Sigma\text{-E-R})$$

$$\langle \mathrm{fst}(c), \mathrm{snd}(c) \rangle \overset{?}{\equiv} c$$

Σ = constructive existence
(as opposed to mere existence)

exercise: check (nontrivial)

$$\frac{a : A \qquad B(a) \text{ true}}{\exists x : A . B(x) \text{ true}} \; (\exists I) \qquad \frac{\exists x : A . B(x) \text{ true} \qquad x : A, B(x) \text{ true} \vdash C \text{ true}}{C \text{ true}} \; (\exists E)$$

notice: this rule is quite different from Σ

$\boxed{\text{failure of logic: inability to express full meaning of } \exists\text{-qfier}}$

ex) try to go from L-$\exists$ to the other end, can't

<u>constructive existence</u>

not an axiom!

Axiom of Choice
↳ KLUDGE

THEOREM OF CHOICE

Every total binary relation contains a (choice) function.

$$(\Pi x:A.\ \Sigma y.B.\ R(x,y)) \supset \Sigma f:A \to B.\ \Pi x:A.\ R(x,f(x))$$

there could be many of these

↳ or maybe $f : \prod_{x:A} B(x)$

after ¬¬ inserted, can you see a choice fn?

math object

produce data

$$\lambda T:\prod_{x:A}\sum_{y:B} R(x,y).\ \langle \lambda a:A.\ fst(T(a)),\ \lambda a:A.\ snd(T(a))\rangle$$

axiom of replacement → depend on

"Tait's view"

When I was young, I could not believe that proving something exists by showing that it could not not exist was a cheap trick. When I talked to math professors, they thought I was a Loony Tune. Then I got to Constable and he said it was a trick, and I wanted to hug him!

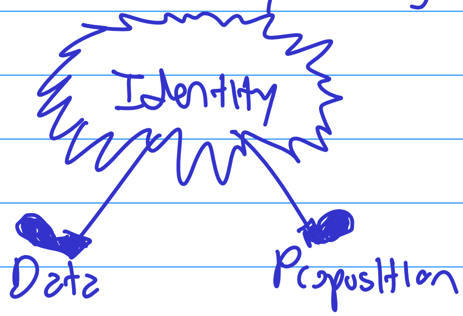
Some people think, the real axiom of choice is the one where there is a double negation, and you postulate it exists. This is true, but it's outrageous, because the whole point of type theory is that things you can run. So pulling a function out of thin air ruins the whole thing!

If I am a firm constructivist, I can still make sense of classical mathematics. I just have to reinterpret what you're saying which is weaker. I just have to hear "there cannot fail to exist a group with that property." So you could believe that all classical mathematics is done by contradiction. If I do this systematically I always have a constructively valid argument. But you're giving up information that you didn't need to give up. Constructive view is sharper, because I can draw more distinctions, whereas if I obliterate everything then I lose a lot. (Double-negation is related to GOTO in programming languages.)

SUBTLE!!

Identity relation

— trivial classically
— really interesting constructively



$$\frac{A : \mathcal{U} \qquad a : A \qquad b : B}{\mathrm{Id}_A(a,b) : \mathcal{U}} \ (\mathrm{Id}\text{-}F / \mathcal{U}\text{-}I\text{-}\mathrm{Id})$$

→ type of identifications
or proofs of identity of a & b in type A

same thing!

congruence

$$\frac{A : \mathcal{U} \qquad a : A}{\mathrm{refl}_A(a) : \mathrm{Id}_A(a,a)} \ (\mathrm{Id}\text{-}I) \qquad \frac{a \equiv b : A}{\mathrm{refl}_A(a) : \mathrm{Id}_A(a,b)}$$

$$\frac{c : \mathrm{Id}_A(a,b)}{?} \ (\mathrm{Id}\text{-}E)$$

two ways to do this

NuPRL

$a \equiv b : A$
(EXTENSIONAL IDENTITY)

X rejected

"ETT"

But thinks this is dept for sets

next time

why is this rule meaningful?

this'd be nice

Shouldn't A and B already be definitionally equal? Well, now you can show things are equal without calculation. Judgmental equality is no longer definitional.

on here

eg) $0 + x \equiv x : \mathbb{N}$