# Harper 5

Development of cubical type theories

computational
cartesian

point of view of PLs
PL as foundation of mathematics

~~Recall~~ Define $Id_A(M_1, M_2)$ identity identification

intro: $refl_A(M)$

elim: $J(a,b,c.C)(a.P)(Q)$ $[M_1, M_2, Q/a,b,c]$

$\uparrow Id_A(M_1, M_2)$

ML '72 if $Id_A(M,N)$ then $M \equiv N : A$

(problem for $\pi$)

2 ideas

1. add extensionality for fns
2. univalence axiom

$UA(E) : Id_{\mathcal{U}}(A,B)$ when $E : Equiv(A,B)$

non-trivial evidence & it matters

Define transport op    A indexed fam. of types

$\mathrm{tr}[a:F](P:\mathrm{Id}_A(M,N)) : F[M/a] \to F[N/a]$

e.g. rec[17]    (can define using)

$R(\_\_\_)$    $\triangleq J(a,b,\_.\underbrace{F(a) \to F(b)}_{C(a,b)})$

$(a.\mathrm{id}_{F[a]})(P)$

should be reverse, concat, etc.

HoTT    Homotopy Type Theory
(formalism)

gives funext

3) option    inductive types w/ identification
(think quotiented in certain way)

Problem    ~~What is~~

consider the "identity family"    $a.a$

$\mathrm{tr}[a.a](P:\mathrm{Id}_U(A,B)) : A \to B$    (invertible)

Tempting to call "coercion function"

of    $\dfrac{M:A \quad A \equiv B}{M:B}$    $\dfrac{M \in A \quad A \doteq B}{M \in B}$

almost $\dfrac{P : \mathrm{Id}_{\mathcal{U}}(A,B) \qquad M : A}{\mathrm{tr}(M) : B}$

looks a lot like id equivalence
proof - relevant

Idea: identification as "proof relevant equality"

$\mathrm{tr}\,[a.a](ua(E)) \equiv\ ???$
what does that compute to?

- refl ✓
- ua ?
- funext ?
- loop ?

"stuck" from computation

( Simplicial sets )

"coercion"

everyone PL eventually comes up w/ own coercion

idea ought to be to apply the equivalence E

Equiv(A, B)

invertible
preimage that's a ~~set~~ singleton

more-or-less bijection

if $\epsilon : A \to B$ acts as witness
why not 'apply' it

$H_A$ type defined in type-independent way
generic

(independent of $A$)

w/ coercion, have to say when the type is $U$ (universe)

it's actually type-specific?

cannot be fixed as stated

HoTT formalism as stated

change to different arena of discussion

beware of axioms added to TT

can make mathematical sense
not comput.

Idea: adding axioms is suspicious
↳ elements of a type

1. Gentzen (vs. Hilbert)
((implications as axioms)

entailment is $\boxed{\text{prior}}$ to implication

hyp. judgment

$$\frac{\Gamma \times A \vdash B}{\vdash A \supset B}$$ logical consequence

"ninja" move

2. Eilenberg MacLane

maps are prior to functions

$$\frac{f : C \times A \to B}{\ulcorner f \urcorner : C \to B^A}$$

Here, bring out the judgmental structure of identifications

then you can internalize them in the type

(both 1, 2 are internalizations)

generic in
A

paths

Id will emerge as
inductive type

$Path_A$ internalize identification
paths

depends on A!

in the end, these are equivalent

but... equivalence does not preserve computation

must separate if care about computation

equivalence alone unable to recover
computation

need judgments prior to inhabit a type

build rich enough judgmental apparatus,

all concepts are there

apply the Martin-Löf ninja move

Paths — between types
and within types

Where do we get paths between types?

1. Induced by paths between element
   of the indexed type of a family

   $a : A \vdash F$ type

   path within A will be
   preserved by F

   why? set up meaning of hyp
   judgm to preserve paths

generalizes points
to lines bet points, square
then lines bet lines (cubes)
lines bet square
lines bet cubes (hyper cube)
⋮

need algebra!

$$\underbrace{M \xrightarrow{P} N}_{A} \quad \text{induce} \quad \underbrace{F[M] \xrightarrow{F[P]} F[N]}_{\text{multiverse of types}}$$

universes are not universal
(don't contain self)

need a word for everything

multiverse of all types

2. Univalence

$\text{Bool} \times \text{Nat} \xrightarrow{\text{swap}} \text{Nat} \times \text{Bool}$ instance of univalence

swap - identify between types

UA applied to swap-equiv

(show swap is equiv)

ua (swap-equiv)
= swap

$\text{Bool} \times \text{Nat} \text{ ——— } \text{Nat} \times \text{Bool}$

Paths induce $\boxed{\text{coercions}}$ !

coe (swap) : $\text{Bool} \times \text{Nat} \to \text{Nat} \times \text{Bool}$

other kinds of identification

$$(Bool \times Nat) \to C \xrightarrow{\text{"swap} \to C\text{"}} (Nat + Bool) \to C$$

swap will be a type

elements of those types path left endpoints

what is

coe (swap → C)

More specific about what is a <u>Path</u>

Diagram

$$Bool \times Nat \xrightarrow{\text{"swap"}} Nat + Bool$$

"swap" is a type

left endpoint Bool × Nat
right endpoint Nat + Bool

Paths are traced out by dimension

variables that "range over unit interval [0,1]

$$\begin{array}{l} [0,1] \longrightarrow \text{ez} \\ \;\subseteq \\ \;\mathbb{R} \end{array}$$

continuous & uniformly

$\boxed{swap_x \ \ Type \ \ [x]}$ ($x$ coord variable)

obtain endpts by subst.

$swap_x \langle 0/x \rangle = Bool \times Nat$

$swap_x \langle 1/x \rangle = Nat + Bool$

swap mediates between them

don't know anything about $x$ other than $0, 1$

can't be "half-way"

$Bool \times Nat \xrightarrow{swap_x} Nat + Bool$

$swap_0 \mapsto$ $\qquad\qquad$ $swap_1 \mapsto$

$swap_0 \mapsto Bool \times Nat$

$swap_1 \mapsto Nat \times Bool$

"a $_x$ line of types"

↳ line of types traced out by $x$

If its a type,
what are its <u>elements</u> of <u>$swap_x$</u>?

in general:
heterogeneous lines

canonical element

$swapel_x(N) \in swap_x \; [x]$

if $\underline{N \in A_2 \times A_1 \; [x]}$

$swapel_0(N)$

$(swapel_x(N))\langle 0/x \rangle = swap_0(N\langle 0/x \rangle).$

same for: 1

$swapel_0(N) \longmapsto ap(swapfn, N)$

$swapel_1(N) \longmapsto N$

Idea

$ap(swapfn(N_0))$

$\|$

$N_0$ ———————— $N_1$

Why is that the definition?
(may not get to why)

0 - side $A_0 \times A_1$ o/k
1 - side
"heterogeneous"

associated w/ type lines
flavor of:
lines of elements
lines of types

<u>General setup</u>

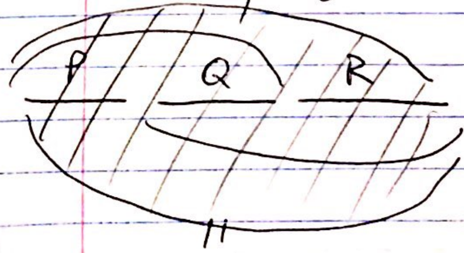
1. type lines induce coercions

2. in general must be able to <u>compose lines</u>

must be the case, has to be reversible

$0 \xrightarrow{P} 1 \qquad 0 \xleftarrow{P^{-1}} 1$

concat lines




', '' should be same

ie. must be lines between lines
double line, or <u>shade</u> "filled in"

associativity should be same
miraculous thing

## Kan condition

program for performing compositions that gets us all of this structure !

One version (most basic)

$\frac{x}{l_y}$

$$\begin{tikzcd} \bullet \arrow[r, "Q"] \arrow[d, "P"'] & \bullet \arrow[d, "R"] \\ \bullet \arrow[r, dash, dashed, "\text{``}P^{-1}\cdot Q\cdot R\text{''}"'] & \bullet \end{tikzcd}$$

reversal for free

shows we have lines bet lines (identification)

conditions guarantee lines bet lines

and so forth

# Cartesian cubes

A type $[x_1 \dots x_n]$

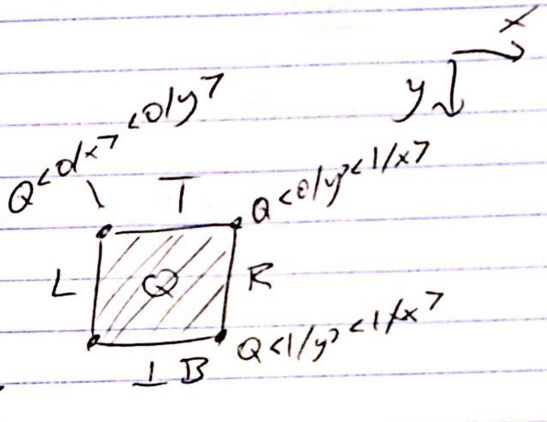
$M \in A$ $[x_1 \dots x_n]$

$n \geq 0$

points

lines

2 diff lines




$Q\langle 0/x\rangle = L$

$Q\langle 1/x\rangle = R$

$Q\langle 0/y\rangle = T$

$Q\langle 1/y\rangle = B$

worry $(Q\langle 0/x\rangle)\langle 1/y\rangle$

value then compute

<u>runs</u>

same as $(Q\langle 0/x\rangle\langle 1/y\rangle)$

$\boxed{\text{coherence}}$

(subject to coherence requirement)

can take faces

can think of vacuously extent in ~~z~~ realm

- ~~degeneracy~~

- diagonals ↳ make it cartesian

(transfer over to mathematical setting)

taking faces causes computation to run

what are those coercions?

have to have enough of them

has to be well-behaved (like a groupoid, group laws)

weak ∞-groupoid to infin dimensions

How we lift / deal w/ swap

$$\mathrm{coe}^{0 \rightsquigarrow 1}_{x.(A_1 \to A_2)}\left(\begin{array}{l} M \\ \in A_1\langle 0/x\rangle \\ \to A_2\langle 0/x\rangle \end{array}\right) \in \begin{array}{l} A_1\langle 1/x\rangle \\ \to A_2\langle 1/x\rangle \end{array}$$

what should it be / turn into?

$\mapsto \lambda a_1 \in A_1\langle 1/x\rangle$

every type coercion & composition

"Kan type"

$$\mathrm{coe}^{0 \rightsquigarrow 1}_{x.A_2}\left(M\left(\mathrm{coe}^{1 \rightsquigarrow 0}_{x.A_1}(a_1)\right)\right)$$

also need coercion from 1 to 0

$\mathrm{coe}^{0 \rightsquigarrow 1}_{x.\,\mathrm{swap}_x}$

$\in \mathrm{swap}_x\langle 0/x\rangle$
Bool $\times$ Nat

$\to \mathrm{swap}_x\langle 1/x\rangle$
Nat $\times$ Bool

coercion swaps!

all depends on defn of swap-equiv

(could also swap bool values etc.)

Coercions $0 \rightsquigarrow 1$, $1 \rightsquigarrow 0$

coercion $0 \rightsquigarrow x$, $x \rightsquigarrow 0$, $1 \rightsquigarrow x$, $x \rightsquigarrow 1$, $x \rightsquigarrow y$

all has to be possible

$$\mathrm{coe}^{0\rightsquigarrow 0}_{y.B}(M) \xrightarrow{\mathrm{coe}^{0\rightsquigarrow x}_{y.B}(M)} \mathrm{coe}^{0\rightsquigarrow 1}_{y.B}(M)$$

$\equiv M.$

think of $B$ as $\mathrm{swap}_x$

elements of $M$ are lines w/ swapped endpoints.

evals to something derived from <u>$M$ itself</u>!

That's why $\mathrm{swap}_x$ is defined as it is

what if dep type

$$\mathrm{coe}^{0\rightsquigarrow 1}_{x.(a:A_1 \to A_2(a))}$$

for dependency, need $\mathrm{coe}^{1\rightsquigarrow x}_{A_1}(-)$

Wrap up

Id type J type

throw in UA, etc.

ninja move

enrich judgmental structure

add line of types

coercion - resolve tension
of Id dep on type

J in Id case

once you have lines
then lines bet. lines
on forever

have to def lot of notions of
coercion that all cohere together

compositions U-diagrams
by constraints
adjacent requirement

requirement: have enough cubes

comp of lines/cubes of types
also have to be ~~type~~ lines/cubes of types
gives comput. meaning to HoTT