Computation Type Theory

Bob Harper
Carnegie Mellon University

July 19, 2018

1 Recap

1.1 Example: Last Time

Suppose $a, b : A, c : \texttt{Eq}_A(a, b) >> C$ `type`
If $a : A >> Q \in C[a, a, */a, b, c]$ and $P \in \texttt{Eq}_A(M, N)$
then $A[M/a] \in C[M, N, P/a, b, c]$

1.2 Proof

$P = * \in \texttt{Eq}_A(M, N)$ and $M \doteq N \in A$
$C[M, N, P] \doteq C[M, N, *]$, $P[M] \in M$

1.3 Examining Formalism (”abstract implementations”) for type theory

Inductively defined by rules for deriving judgments of the forms:

$$\Gamma \vdash A \; \texttt{type}$$

$$\Gamma \vdash A \equiv A'$$

$$\Gamma \vdash M : A$$

$$\Gamma \vdash M \equiv M' : A$$

$$\frac{}{\Gamma, x : A, \Gamma' \vdash x : A}$$

$$\frac{\Gamma \vdash A_1 \; \texttt{type} \; \Gamma \vdash A_2 \; \texttt{type}}{\Gamma \vdash A_1 \times A_2 \; \texttt{type}}$$

$$\frac{\Gamma \vdash M : A \; \Gamma \vdash A \equiv A'}{\Gamma \vdash M : A'}$$

$$\frac{\Gamma \vdash M_1 : A_1 \; \Gamma \vdash M_2 : A_2}{\Gamma \vdash < M_1 ; M_2 >: A_1 \times A_2}$$

$$\frac{\Gamma \vdash M : A_1 \times A_2}{\Gamma \vdash M \cdot i : A_i}(i = 1, 2)$$

$$\frac{\Gamma \vdash M_1 : A_1 \; \Gamma \vdash M_2 : A_2}{\Gamma \vdash < M_1; M_2 > \cdot i \equiv M_i : A_i} \checkmark$$

$$\frac{\Gamma \vdash M : A_1 \times A_2}{\Gamma \vdash < M \cdot 1; M \cdot 2 = M : A_1 \times A_2} ?$$

2 Idea

Formalism is just a means of deriving truths (about computations)
Props as types / proofs as programs
extraction

1. Define erasure of formal terms types $|M|, |A|$ (more or less trivial)

2. Soundness w.r.t. erasure
   $\Gamma \vdash M : A \implies |\Gamma| >> |M| \in |A|$
   $\Gamma \vdash M_1 \equiv M_2 : A \implies |\Gamma| >> |M_1| \doteq |M_2| \in |A|$
   $\Gamma \vdash A$ `type` $\implies |\Gamma| >> |A|$ `type`
   $\Gamma \vdash A_1 \equiv A_2 \implies |\Gamma| >> |A_1| \doteq |A_2|$

2.1 Corollary

1. If M : `Bool` then $M \Downarrow$ `true` or $M \Downarrow$ `false`

2. Consistency

2.2 Soundness

Soundness tells us that proofs have a computational context.
Would like to internalize computation as definitional equivalence.

2.2.1 Cononicity

If M : `Bool`, then $M \equiv$ `true` : `Bool` or $M \equiv$ `false` : `Bool`
"Internal Completeness"

2.3 How to formalize equality?

2.3.1 Example

$a : \texttt{Nat} >> a + a \doteq 2 \times a \in \texttt{Nat}$
$\lambda a.a + a = \lambda a.2 \times a \in \texttt{Nat} \to \texttt{Nat}$

1. ETT - Equality reflection uniqueness
   expressive - not decidable

2. ITT - M-L
   Formalizes Id_A as the least reflexive relation.

$$\frac{\Gamma \vdash A \ \texttt{type} \ \Gamma \vdash M_1 : A \ \Gamma \vdash M_1 : A}{\Gamma \vdash \texttt{Id}_A(M_1, M_2)}$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \texttt{refl}_A(M) : \texttt{Id}_A(M, M)}$$

$$\frac{\Gamma, a, b : A, c : \texttt{Id}_A(a, b) \vdash C \ \texttt{type} \ \Gamma \vdash P : \texttt{Id}_A(M, N) \ \Gamma, a : A \vdash Q : C[a, a, \texttt{refl}_A(M)]}{\Gamma \vdash J(a, b, c.C)(a.Q)(P) : C[M, N, P] \ J(a, b, c.C)(a.Q)(\texttt{refl}_A(M)) \equiv Q[M/a]}$$

2.3.2 Idea of Soundness

$|\texttt{Id}_A(M, N)| \triangleq \texttt{Eq}_{|A|}(|M|, |N|)$
$|\texttt{refl}_A(M)| \triangleq *$
$|J \dots| \triangleq |Q|[|M|/a]$
But a theorem of M-L

2.3.3 What to do about this?

A common approach is to add an axiom (??)
$FunExt : a : A_1 \to \texttt{Id}_{A_2}(ap(M, a), ap(N, a)) \to \texttt{Id}_{a:A_1 \to A_2}(M, N)$
$FunExt(H) : \texttt{Id}_{a:A_1 \to A_2}(M, N)$
Okay mathematically - but it ruins cononicity!
$J(a, b, c.C)(a.Q)(FunExt(H)) \equiv ???$

2.3.4 Strength in Weakness

ITT cannot prove (internally) that there is only one identification!

- "groupoid model"

Yet it's valid under erasure: $|FE(H)| \triangleq *$

3 Univalence

Motivation: It is common to informally "identify structures up to isomorphism"

3.1 Example

$A \times B \simeq B \times A$ by swap is a bijection.
Add an axiom: $\texttt{UA("swap")} : \texttt{Id}_{\mathcal{U}}(A \times B, B \times A)$ where $\mathcal{U}$ is the "universe".
For specific choices of A and B, then there can be many isomorphisms!
I.e.: $\texttt{Bool} \simeq \texttt{Bool}$

1. Identity is a mesh for interchangability

2. Stipulate that equivalent types are identical

1. $J\_(FunExt(H)) \equiv ?$
2. $J\_(\mathtt{UA}(E)) \equiv ?$

ruins canonicity
Cases:

1. Has a computational interprection
2. ??? what would be a computational of univalence ???
   "ought to exist"