Cartesian Cubical Computational Type Theory

Carlo Angiuli
Evan Cavallo
(*) Favonia
Robert Harper
Jonathan Sterling
Todd Wilson
Cubical
features of homotopy type theory
univalence, higher inductive types
+
Computational
features of Nuprl and PVS
strict equality, strict quotients, predicative subtypes...
Cartesian  Cubical
features of homotopy type theory
univalence, higher inductive types

+  

Computational
features of Nuprl and PVS
strict equality, strict quotients,
predicative subtypes...
Computational Types

programs/realizers

computation
Computational Types

programs/realizers ← computational type theory

computation theory of computation
Computational Types

- Programs/Realizers
- Computational Type Theory
- Computation
- Theory of Computation
- Meaning/Explanation
- Pre-mathematical in M-L's work
- Martin-Löf Type Theory
A Minimum Example

\[ M := a \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M,M,M) \]
A Minimum Example

\[ M := \text{a} \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M, M, M) \]

<table>
<thead>
<tr>
<th>bool val</th>
<th>if(M, Mt, Mf) ⇔ if(M', Mt, Mf)</th>
</tr>
</thead>
<tbody>
<tr>
<td>true val</td>
<td>if(true, M, _) ⇔ M</td>
</tr>
<tr>
<td>false val</td>
<td>if(false, _, M) ⇔ M</td>
</tr>
</tbody>
</table>
A Minimum Example

\[
M := a | \text{bool} | \text{true} | \text{false} | \text{if}(M, M, M)
\]

\[
\begin{align*}
\text{bool val} & \quad \text{if}(M, M_t, M_f) \Rightarrow \text{if}(M', M_t, M_f) \\
\text{true val} & \quad \text{if}(\text{true}, M, \_ ) \Rightarrow M \\
\text{false val} & \quad \text{if}(\text{false}, \_, M) \Rightarrow M
\end{align*}
\]

The Language
## A Minimum Example

\[
M := a \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M,M,M)
\]

<table>
<thead>
<tr>
<th>Type</th>
<th>Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>bool val</td>
<td>\text{if}(M,\text{Mt},\text{Mf}) \Rightarrow \text{if}(M',\text{Mt},\text{Mf})</td>
</tr>
<tr>
<td>true val</td>
<td>\text{if}(\text{true},M,_ _ ) \Rightarrow M</td>
</tr>
<tr>
<td>false val</td>
<td>\text{if}(\text{false},_ _ ,M) \Rightarrow M</td>
</tr>
</tbody>
</table>

What are the types in **canonical forms?** \{\text{bool}\}
A Minimum Example

\[ M := a \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M,M,M) \]

**bool val**

\[ \text{if}(M,M_t,M_f) \Rightarrow \text{if}(M',M_t,M_f) \]

**true val**

\[ \text{if}(\text{true},M,\_ ) \Rightarrow M \]

**false val**

\[ \text{if}(\text{false},\_,M) \Rightarrow M \]

The Language

What are the types in **canonical forms**? \{bool\}

What are the **canonical forms** of the types?

**bool**: \{true, false\}
A Minimum Example

\[ M := a \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M, M, M) \]

\begin{align*}
\text{bool val} & \quad \text{if}(M, M_t, M_f) \leadsto \text{if}(M', M_t, M_f) \\
\text{true val} & \quad \text{if}(\text{true}, M, _) \leadsto M \\
\text{false val} & \quad \text{if}(\text{false}, _, M) \leadsto M
\end{align*}

The Language

What are the types in canonical forms? \{bool\}

What are the canonical forms of the types?

\textbf{bool}: \{true, false\}

How they are equal? \textit{syntactic equality}
A Minimum Example

\[ M := a \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M,M,M,M) \]

\text{bool val} \quad \text{if}(M,M_{t},M_{f}) \Rightarrow \text{if}(M',M_{t},M_{f})

\text{true val} \quad \text{if}(\text{true},M,\_ ) \Rightarrow M

\text{false val} \quad \text{if}(\text{false},\_,M) \Rightarrow M

The Language

What are the types in \textit{canonical forms}? \{\text{bool}\}

What are the \textit{canonical forms} of the types?

\textbf{bool}: \{\text{true, false}\}

How they are equal? \textit{syntactic equality}
A Minimum Example

\[ M := \text{a} \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M,M,M) \]

**types:** \{bool\} with syntactic equality \( \approx \)

**bool:** \{true, false\} with syntactic equality \( \approx_{\text{bool}} \)
A Minimum Example

M := a | bool | true | false | if(M,M,M)

types: {bool} with syntactic equality ≈
bool: {true, false} with syntactic equality ≈_{bool}

A ⊳ B type
A⇓A' B⇓B' and A'≈B'
A Minimum Example

\[
M := a \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M,M,M)
\]

types: \{\text{bool}\} with syntactic equality \(\approx\)
bool: \{true, false\} with syntactic equality \(\approx_{\text{bool}}\)

\[
A \Downarrow B \text{ type}
A \Downarrow A' \; B \Downarrow B' \text{ and } A' \approx B'
\]

\[
\text{bool} \Downarrow \text{bool type}
\]
A Minimum Example

\[
M := a \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M,M,M)
\]

\[
\text{types: \{\text{bool}\}} \text{ with syntactic equality } \approx
\]

\[
\text{bool: \{true, false\}} \text{ with syntactic equality } \approx_{\text{bool}}
\]

\[
A \triangleq B \text{ type}
\]

\[
A \downarrow A' \quad B \downarrow B' \text{ and } A' \approx B'
\]

\[
\text{bool } \triangleq \text{bool type}
\]

\[
\text{if(true, bool, bool) } \triangleq \text{bool type}
\]

\[
\downarrow \text{bool}
\]
A Minimum Example

\[ M := a \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M,M,M) \]

Types: \{\text{bool}\} with syntactic equality \( \approx \)

\text{bool}: \{\text{true}, \text{false}\} with syntactic equality \( \approx_{\text{bool}} \)

\[
\begin{align*}
\text{A} & \equiv \text{B type} \\
\text{A} \downarrow \text{A' } \text{B} \downarrow \text{B' } \text{and } \text{A'} \approx \text{B'}
\end{align*}
\]

\[
\begin{align*}
\text{bool} & \equiv \text{bool type} \\
\text{if}(\text{true}, \text{bool}, \text{bool}) & \equiv \text{bool type} \\
& \downarrow \text{bool} \\
\text{if}(\text{true}, \text{bool}, \text{any closed term}) & \equiv \text{bool type}
\end{align*}
\]
A Minimum Example

\[ M := a \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M,M,M) \]

types: \{\text{bool}\} with syntactic equality \( \approx \)

bool: \{\text{true}, \text{false}\} with syntactic equality \( \approx_{\text{bool}} \)

\[ M \equiv N \in A \]

\[ A \equiv A \text{ type, } M \downarrow M', N \downarrow N', A \downarrow A' \text{ and } M' \approx_{A'} N' \]
A Minimum Example

\[ M := a \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M,M,M) \]

\text{types: } \{\text{bool}\} \text{ with syntactic equality } \approx

\text{bool: } \{\text{true, false}\} \text{ with syntactic equality } \approx_{\text{bool}}

\[ M \doteq N \in A \]

\[ A \doteq A \text{ type, } M \downarrow M', N \downarrow N', A \downarrow A' \text{ and } M' \approx_{A'} N' \]

\[ \text{false } \doteq \text{false } \in \text{bool} \]
A Minimum Example

M := a | bool | true | false | if(M,M,M)

types: {bool} with syntactic equality \(\approx\)
bool: \{true, false\} with syntactic equality \(\approx_{\text{bool}}\)

\[ M = N \in A \]
A\(=\)A type, M\(\downarrow\)M', N\(\downarrow\)N', A\(\downarrow\)A' and M'\(\approx_{A'}\)N'

false \(\not=\) false \(\in\) bool

if(true,true,bool) \(\not=\) true \(\in\) if(true,bool,bool)
\[ \downarrow\text{true} \quad \downarrow\text{bool} \]
A Minimum Example

\[ M := a \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M, M, M) \]

\[ \text{types: \{bool\} with syntactic equality } \approx \]
\[ \text{bool: \{true, false\} with syntactic equality } \approx_{\text{bool}} \]

\[ a : A \gg M \models N \in B \]

\[ P \models Q \in A \text{ implies } M[P/a] \models N[Q/a] \in B[P/a] \]
A Minimum Example

\[ M := a \mid \text{bool} \mid \text{true} \mid \text{false} \mid \text{if}(M,M,M) \]

types: \{\text{bool}\} with syntactic equality \(\approx\)

\text{bool}: \{\text{true}, \text{false}\} with syntactic equality \(\approx_{\text{bool}}\)

\[ a:A \Rightarrow M \triangleq N \in B \]

\(P \triangleq Q \in A\) implies \(M[P/a] \triangleq N[Q/a] \in B[P/a]\)

\[ b:\text{bool} \Rightarrow b \triangleq \text{if}(b, \text{true}, \text{false}) \in \text{bool}? \]
A Functional Example

\[ M := a \mid M_1 \rightarrow M_2 \mid \lambda a. M \mid M_1 M_2 \mid ... \]

\[ (M_1 \rightarrow M_2) \text{ val } \lambda a. M \text{ val } (\lambda a. M_1) M_2 \Rightarrow M_1[M_2/a] \]

Another Language
A Functional Example

M := a | M1→M2 | \a.M | M1 M2 | ...

(M1→M2) val \a.M val (\a.M1)M2 ⇒ M1[M2/a]

Another Language

What are the types in canonical forms?

the least fixed point of
S ⇒ \{M→N | M⇓, N⇓ in S\} union ...

What are the canonical forms of the types?
A→B: {\a.M}

How they are equal?
A1→B1 ≈ A2→B2 if A1 ≐ A2 and B1 ≐ B2
\a.M1 ≈_{A→B} \a.M2 if a:A >> M1 ≐ M2 ∈ B
# Variables

<table>
<thead>
<tr>
<th>Nuprl/...</th>
<th>Coq/Agda/...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vars range over closed terms</td>
<td>Vars are indet.</td>
</tr>
<tr>
<td>Defined by transition b/w closed terms</td>
<td>Defined by conversion b/w open terms</td>
</tr>
</tbody>
</table>
Open-endedness

Proof theory/tactics/editors

\downarrow

Computational type theory

\downarrow

Programming language
Open-endedness

Proof theory/tactics/editors
\downarrow
Computational type theory
\downarrow
Programming language

Canonicity always holds
Homotopy Type Theory

github.com/HoTT/book
Homotopy Type Theory

\[ a \quad b \quad \text{points} \]
Homotopy Type Theory

$p : a = b$

$a$

$q : a = b$

$b$

paths

points
Homotopy Type Theory

\[ a \xrightarrow{p: a=b} b \xrightarrow{q: a=b} \]

paths between points

paths
Homotopy Type Theory

\[ p : a = b \]
\[ h : p = q \]
\[ q : a = b \]

paths between paths

points
Equality and Paths

Equality ($\equiv$)

Silent in theory

$2 + 3 \equiv 5$

$\text{fst } \langle M, N \rangle \equiv M$
Equality and Paths

Equality (≡)

Silent in theory

\[ 2 + 3 \equiv 5 \]

\[ \text{fst} \langle M, N \rangle \equiv M \]

If \( A \equiv B \) and \( M : A \) then \( M : B \)
Equality and Paths

Equality (≡)

Silent in theory

\[ 2 + 3 \equiv 5 \]
\[ \text{fst } \langle M, N \rangle \equiv M \]
If \( A \equiv B \) and \( M : A \) then \( M : B \)

Paths (=)

Visible in theory
If \( P : A=B \) and \( M : A \) then \( \text{transport}(M,P) : B \)
Homotopy Type Theory

[Awodey and Warren] [Voevodsky et al] [van den Berg and Garner]

A : Type
a : A : Element
f : A → B : Function
C : A → Type : Dependent Type
a =_A b : Identification

Space
Point
Continuous Mapping
Fibration
Path
Features of HoTT

Univalence

If $E$ is an equivalence between types $A$ and $B$, then $\text{ua}(E):A=B$

Higher Inductive Types

circle  sphere  torus
Canonicity?

Canonicity broken by new features stated as axioms!
Canonicity?

Canonicity broken by new features stated as axioms!

Canonicity

For any $M : \text{bool}$, either $M \equiv \text{true} : \text{bool}$ or $M \equiv \text{false} : \text{bool}$
Canonicity?

Canonicity broken by new features stated as axioms!

For any $M : \text{bool}$, either $M \equiv \text{true} : \text{bool}$ or $M \equiv \text{false} : \text{bool}$

$\text{ua(not)} : \text{bool} = \text{bool}$

$\text{transport(ua(not),true)} \not\equiv \text{false}$
Canonicity for All

Canonicity for bool means canonicity for everyone
Canonicity for All

Canonicity for bool means

canonicity for everyone

\[ M : \text{bool} \times A \]

\[ \text{fst}(M) \equiv ??? : \text{bool} \]
Canonicity for All

Canonicity for bool means canonicity for everyone

\[ M : \text{bool} \times A \]
\[ \text{fst}(M) \equiv ??? : \text{bool} \]

Wants \( M \equiv \langle P, Q \rangle \) and then
\[ \text{fst}(M) \equiv \text{fst}(P, Q) \equiv P \equiv \text{true or false} \]
Canonicity for Paths?

\[
\begin{align*}
  M : A \\
  \Rightarrow refl(M) : M =_A M
\end{align*}
\]
Canonicity for Paths?

\[
\begin{align*}
M & : A \\
\text{refl}(M) & : M =_A M \\
\end{align*}
\]

\[
a : A \vdash R : C(a,a,\text{refl}(a)) \quad P : M = N
\]

\[
\text{path-ind}[C](a.R,P) : C(M,N,P)
\]
Canonicity for Paths?

\[
\begin{align*}
M & : A \\
\text{refl}(M) & : M =_A M \\
\text{path-ind}[C](a.R, \text{refl}(M)) & \equiv R[M/a] \\
\end{align*}
\]
Canonicity for Paths?

\[
\begin{align*}
M &: A \\
\text{refl}(M) &: M =_A M
\end{align*}
\]

\[
a : A \vdash R : C(a,a,\text{refl}(a)) \quad P &: M = N
\]

\[
\text{path-ind}[C](a.R,P) : C(M,N,P)
\]

\[
\begin{align*}
a &: A \vdash R &: C(a,a,\text{refl}(a)) \quad M &: A \\
\text{path-ind}[C](a.R,\text{refl}(M)) &\equiv R[M/a] \\
&: C(M,M,\text{refl}(M))
\end{align*}
\]

\[
\text{path-ind}[C](a.R,ua(E)) \equiv ???
\]
Can we have a new TT with canonicity + univalence?

Yes with De Morgan cubes [CCHM 2016]
Yes with Cartesian cubes [AFH 2017]

... and higher inductive types?

Examples with De Morgan cubes [CHM 2018]
Yes with Cartesian cubes [CH 2018]
Restore Canonicity

Idea: each type manages its own paths
Restore Canonicity

Idea: each type manages its own paths

base : S1
Restore Canonicity

Idea: each type manages its own paths

base : S1
loop : base = base
Restore Canonicity

Idea: each type manages its own paths

base : S1
loop : base base
Restore Canonicity

Idea: each type manages its own paths

base \land loop

base : S1
loop : base = base
x : \mathbb{I} \vdash \text{loop}\{x\} : S1
\text{loop}\{0\} \equiv \text{base} : S1
\text{loop}\{1\} \equiv \text{base} : S1
Restore Canonicity

Idea: each type manages its own paths

Kan structure:
sufficient to implement path-ind

Kan types: types with Kan structure
Cartesian Cubes

Introducing \( I \) the formal interval
Cartesian Cubes

Introducing \( \mathbb{I} \) the formal interval

\[
\Gamma \vdash 0: \mathbb{I} \quad \Gamma \vdash 1: \mathbb{I}
\]

\[
\Gamma, \ x: \mathbb{I}, \ \Gamma' \vdash x: \mathbb{I}
\]
Cartesian Cubes

Introducing $\mathbb{I}$ the formal interval

$$\Gamma \vdash 0: \mathbb{I} \quad \Gamma \vdash 1: \mathbb{I}$$

$$\Gamma, x: \mathbb{I}, \Gamma' \vdash x: \mathbb{I}$$

$x_1: \mathbb{I}, x_2: \mathbb{I}, \ldots, x_n: \mathbb{I} \vdash M : A$

$\Leftrightarrow M$ is an n-cube in $A$
Cartesian Cubes

Introducing \( \mathbb{I} \) the formal interval

\[
\Gamma \vdash 0: \mathbb{I} \quad \Gamma \vdash 1: \mathbb{I}
\]

\( \Gamma, x: \mathbb{I}, \Gamma' \vdash x: \mathbb{I} \)

Cartesian: works as normal contexts

\[
M(0/x) \quad M(1/x) \quad M(y/x)
\]
Cubical Programming

\[
\text{dim expr } r := 0 \mid 1 \mid x
\]

indeterminate
Circle

\[ M := S1 \mid \text{base} \mid \text{loop}\{r\} \]
\[ \mid S1\text{elim}(a.M, M, M, x.M) \mid \ldots \]
Circle

\[ M := S1 \mid \text{base} \mid \text{loop}\{r\} \mid S1\text{elim}(a.M, M, M, x.M) \mid ... \]
Circle

\[ M := S1 | \text{base} | \text{loop\{r\}} | S1\text{elim(a.M, M, M, x.M)} | ... \]
Circle

\[ M := S1 \mid \text{base} \mid \text{loop}\{r\} \mid S1\text{elim}(a.M, M, M, x.M) \mid \ldots \]
Circle

\[ M \Rightarrow M' \]

\[ S1\text{elim}(a.A, M, B, x.L) \Rightarrow S1\text{elim}(a.A, M', B, x.L) \]
Circle

\[ \text{S1elim}(a.A, \text{base}, B, x._) \xrightarrow{\text{loop}\{x\}} B \]

\[ M \Rightarrow M' \]

\[ \text{S1elim}(a.A, M, B, x.L) \Rightarrow \text{S1elim}(a.A, M', B, x.L) \]

\[ \text{S1elim}(a.A, \text{base}, B, x._) \Rightarrow B \]
Circle

\[ \text{S1elim}(a.A, \text{base}, B, x._) \mapsto B \]

\[ \text{S1elim}(a.A, \text{loop}\{x\}, _, y.L) \mapsto L<x/y> \]

\[ M \Rightarrow M' \]

\[ \text{S1elim}(a.A, M, B, x.L) \Rightarrow \text{S1elim}(a.A, M', B, x.L) \]

\[ \text{S1elim}(a.A, \text{base}, B, x._) \Rightarrow B \]

\[ \text{S1elim}(a.A, \text{loop}\{x\}, _, y.L) \Rightarrow L<x/y> \]
Kan 1/2: Coercion

\[ \exists \ x.A \]
Kan 1/2: Coercion

coe[θ→1]

{x.A}(M)

M

\{ x.A \}(M)

x.A

x

Kan 1/2: Coercion

\[
\text{coe}[\emptyset \mapsto 1]
\{x.A\}(M) \\
\subseteq A^{<r'/x>}
\]

\[
\text{coe}[r \leadsto r']\{x.A\}(M) \in A^{<r'/x>}
\]

\[
\subseteq A^{<r/x>}
\]
Kan 1/2: Coercion

\[
\begin{align*}
\text{coe} [\theta \sim 1] \{ x \cdot A \}(M) & \in A <r'/x> \\
\text{coe} [r \sim r'] \{ x \cdot A \}(M) & \in A <r'/x> \\
\text{coe} [r \sim r'] \{ x \cdot A \}(M) & \triangleq M \in A <r/x>
\end{align*}
\]
Kan 1/2: Coercion

\[ \text{coe}[\emptyset \mapsto x] \quad \text{coe}[\emptyset \mapsto 1] \]

\[ M \quad \{x.A\}(M) \quad \{x.A\}(M) \]

\[ x.A \]

\[ \text{coe}[r \mapsto r']\{x.A\}(M) \in A\langle r'/x\rangle \]

\[ \{x.A\}(M) \simeq M \in A\langle r/x\rangle \]
Kan 2/2: Homogeneous Comp.
Kan 2/2: Homogeneous Comp.

hcom[0→1]\{A\}(M)
[x=0→y.N_0, x=1→y.N_1]
Kan 2/2: Homogeneous Comp.

\[ \text{hcom}[0 \rightarrow 1]\{A\}(M) \]
\[ [x=0 \rightarrow y.N_0, \ x=1 \rightarrow y.N_1] \]

\[ \text{hcom}[r \rightarrow r']\{A\}(M) \] \[ [\ldots, \ r_i=r'_i \rightarrow y.N_i, \ \ldots] \in A \]
Kan 2/2: Homogeneous Comp.

\[ hcom[0 \rightarrow 1] \{A\}(M) \]

\[ \left[ x = 0 \rightarrow y.N_0, \ x = 1 \rightarrow y.N_1 \right] \]

\[ hcom[r \sim r'] \{A\}(M) \[\ldots, r_i = r'_i \rightarrow y.N_i, \ldots\] \in A \]

\[ hcom[r \sim r] \{A\}(M) \triangleq M \in A \]

\[ hcom[r \sim r'] \{A\}(M) \[\ldots, r_i = r_i \rightarrow y.N_i, \ldots\] \]

\[ \triangleq N_i <r'/y> \in A \]
Kan 2/2: Homogeneous Comp.
Kan Circle

coe[r ~ r']{_.S1}(M) ↦ M
Kan Circle

\[\text{coe}[r \sim r']\{_.S1\}(M) \Rightarrow M\]

\[\text{hcom}[r \sim r']\{S1\}(M)[...] \Rightarrow \text{fhcom}[r \sim r'](M)[...]\]

formal homo. composition
Kan Circle

\[\text{coe}[r \sim r']\{_.S1\}(M) \Rightarrow M\]

\[\text{hcom}[r \sim r']\{S1\}(M)[... \Rightarrow \text{fhcom}[r \sim r'](M)[...]\]

\[\text{fhcom}[r \sim r](M)[...] \Rightarrow M\]

formal homo. composition
Kan Circle

coe[r\sim r']{_.S1}(M) \to M

\text{hcom}[r\sim r']{S1}(M)[... ] \Rightarrow \text{fhcom}[r\sim r'](M)[... ]

\text{fhcom}[r\sim r](M)[... ] \Rightarrow M

r! = r' \quad r_i = r'_i \quad (\text{the first } i)

\text{formal homo. composition}

\text{fhcom}[r\sim r'](M)[... , r_i = r'_i \to y. N_i, ... ] \Rightarrow N_i <r'/y>
Kan Circle

\[\text{coe} [r \sim r'] \{._{S1}\} (M) \Rightarrow M\]

\[\text{hcom} [r \sim r'] \{S1\} (M)[...] \Rightarrow \text{fhcom} [r \sim r'] (M)[...]\]

\[\text{fhcom} [r \sim r'] (M)[...] \Rightarrow M\]

\[r! = r' \quad r_i = r'_i \quad \text{(the first } i)\]

\[\text{fhcom} [r \sim r'] (M)[..., r_i = r'_i \rightarrow y.N_i, ...] \Rightarrow N_i <r'/y>\]

\[r! = r' \quad r_i! = r'_i \quad \text{for all } i\]

\[\text{fhcom} [r \sim r'] (M)[...] \text{ val}\]
Kan Circle

S1elim needs to handle fcom
Kan Circle

\( S_{1 \text{elim}} \) needs to handle \( f_{\text{com}} \)

\[
\begin{align*}
\text{if } r \neq r' & \quad r_i \neq r'_i \\
\end{align*}
\]

\[
\begin{align*}
\text{S}_{1 \text{elim}}(a.A, f_{\text{com}}[r \rightsquigarrow r'](M)[...], B, x.L) \\
\Rightarrow \text{com}[r \rightsquigarrow r']\{y.A[f_{\text{com}}[r \rightsquigarrow y](M)[...]/a} \\
(S_{1 \text{elim}}(M, B, x.L))[...]
\end{align*}
\]

\( S_{1 \text{elim}}(\text{composition}) \Rightarrow \text{composition}(S_{1 \text{elim}}) \)
Cubical Stability

Dimension subs. do not commute with evaluation!
Cubical Stability

Dimension subsrts. do not commute with evaluation!

\( \text{S}1\text{elim}(a.A, \text{loop}\{x\}, B, y.L) \rightarrow L<x/y> \rightarrow <0/x> \rightarrow L<0/y> \)
Cubical Stability

Dimension subssts. do not commute with evaluation!

\[ \text{S1elim}(a.A, \text{loop}\{x\}, B, y.L) \quad \overset{\text{Dimension subssts. do not commute with evaluation!}}{\longrightarrow} \quad L\langle x/y \rangle \]

\[ \downarrow \quad <0/x> \quad \quad \downarrow \quad <0/x> \]

\[ \text{S1elim}(a.A, \text{base}, B, y.L) \quad \overset{}{\longrightarrow} \quad B \quad \overset{\text{??}}{=} \quad L\langle 0/y \rangle \]
Cubical Stability

Dimension subssts. do not commute with evaluation!

\[ \text{S1elim}(a.A, \\text{loop}\{x\}, B, y.L) \quad \vdash \quad L^{(x/y)} \]

\[ \oplus \quad \downarrow \]

\[ \text{S1elim}(a.A, \text{base}, B, y.L) \quad \vdash \quad B \quad \Leftrightarrow \quad L^{(0/y)} \]

Restrict our theory to only cubically stable parts
Cubical Type Theory

stability: consider every substitution
Cubical Type Theory

stability: consider every substitution

\[ A \simeq B \text{ type } [\Psi] \]

A and B stably recognize the same stable values and have stably equal Kan structures

(see our arXiv papers)
Cubical Type Theory

stability: consider every substitution

\[ A \simeq B \text{ type } [\Psi] \]
A and B stably recognize the same stable values and have stably equal Kan structures

\[ M \simeq N \in A \ [\Psi] \]
A stably treats M' and N' as the same

(see our arXiv papers)
# Variables

<table>
<thead>
<tr>
<th>Nuprl/...</th>
<th>Coq/Agda/...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vars range over</td>
<td>Vars are indet.</td>
</tr>
<tr>
<td>closed terms</td>
<td></td>
</tr>
<tr>
<td>Defined by</td>
<td>Defined by</td>
</tr>
<tr>
<td>transition b/w</td>
<td>conversion b/w</td>
</tr>
<tr>
<td>closed terms</td>
<td>open terms</td>
</tr>
</tbody>
</table>

\[
\text{exp } \text{vars} \quad \text{dim } \text{vars}
\]

\[
\text{cubical computational TT}
\]
arXiv papers

CHTT Part I [AHW 2016]
Cartesian cubical + computational

CHTT Part II [AH 2017]
Dependent types

CHTT Part III [AFH 2017]
Univalent Kan universes
Strict equality

CHTT Part IV [AFH 2017]
Higher inductive types
Proof Assistants

RedPRL
In Nuprl style
redprl.org

redtt
(Work in progress)
github.com/RedPRL/redtt

yacctt
Proof of concept
modified from cubicaltt
github.com/mortberg/yacctt
Conclusion

We extended Nuprl semantics by cubical structure which justifies key features of HoTT
Conclusion

We extended Nuprl semantics by cubical structure which justifies key features of HoTT

*Best of the two worlds!*
Conclusion

We extended Nuprl semantics by cubical structure which justifies key features of HoTT

Best of the two worlds!

We also built proof assistants

redprl.org
github.com/RedPRL/redtt
github.com/mortberg/yacctt