

A Strongly Normalizing Computation Rule for the Univalence Axiom in Higher-Order Propositional Logic

Robin Adams¹, Marc Bezem¹, and Thierry Coquard²

¹ Universitetet i Bergen, Bergen, Norway
`{robin.adams,marc}@uib.no`

² University of Gothenburg, Gothenburg, Sweden
`coquand@chalmers.se`

Homotopy type theory offers the promise of a formal system for the univalent foundations of mathematics. However, if we simply add the univalence axiom to type theory, then we lose the property of canonicity — that every term computes to a normal form. A computation becomes ‘stuck’ when it reaches the point that it needs to evaluate a proof term that is an application of the univalence axiom. We wish to find a way to compute with the univalence axiom.

As a first step towards such a system, we present here a system of higher-order propositional logic, with a universe Ω of propositions closed under implication and quantification over any simple type over Ω . We add a type $a =_A b$ for any terms a, b of type A (this type is not a proposition in Ω), and two ways to prove an equality: reflexivity, and the univalence axiom. We present reduction relations for this system, and prove the reduction confluent and strongly normalizing.

Predicative higher-order propositional logic with equality. We call the following type theory predicative higher-order propositional logic. Its rules of deduction are

$$\begin{array}{c}
 \frac{}{\langle \rangle \text{ valid}} \quad \frac{\Gamma \text{ valid}}{\Gamma, x : A \text{ valid}} \quad \frac{\Gamma \vdash \phi : \Omega}{\Gamma, p : \phi \text{ valid}} \quad \frac{\Gamma \text{ valid}}{\Gamma \vdash x : A} (x : A \in \Gamma) \quad \frac{\Gamma \text{ valid}}{\Gamma \vdash p : \phi} (p : \phi \in \Gamma) \\
 \frac{\Gamma \text{ valid}}{\Gamma \vdash \perp : \Omega} \quad \frac{\Gamma \vdash \phi : \Omega \quad \Gamma \vdash \psi : \Omega}{\Gamma \vdash \phi \rightarrow \psi : \Omega} \quad \frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B} \quad \frac{\Gamma \vdash \delta : \phi \rightarrow \psi \quad \Gamma \vdash \epsilon : \phi}{\Gamma \vdash \delta \epsilon : \psi} \\
 \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A. M : A \rightarrow B} \quad \frac{\Gamma, p : \phi \vdash \delta : \psi}{\Gamma \vdash \lambda p : \phi. \delta : \phi \rightarrow \psi} \quad \frac{\Gamma \vdash \delta : \phi \quad \Gamma \vdash \psi : \Omega}{\Gamma \vdash \delta : \psi} (\phi \simeq \psi)
 \end{array}$$

Extensional equality. On top of this system, we add an extensional equality relation. We extend the grammar with

$$\begin{array}{ll}
 \text{Path} & P ::= e \mid \text{ref}(M) \mid P \supset P \mid \text{univ}_{\phi, \phi}(\delta, \delta) \mid \mathbb{M}e : x =_A x.P \mid PP \\
 \text{Proof} & \delta ::= \dots \mid P^+ \mid P^- \\
 \text{Context} & \Gamma ::= \dots \mid \Gamma, e : M =_A M \\
 \text{Judgement} & \mathcal{J} ::= \dots \mid \Gamma \vdash P : M =_A M
 \end{array}$$

Note that, in the path $\mathbb{M}e : x =_A y.P$, the term variables x and y and the proof variable e are all bound within P .

We add the following rules of deduction

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : A}{\Gamma, e : M =_A N \text{ valid}} \quad \frac{\Gamma \text{ valid}}{\Gamma \vdash e : M =_A N} e : M =_A N \in \Gamma$$

$$\begin{array}{c}
\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathbf{ref}(M) : M =_A M} \quad \frac{\Gamma \vdash P : \phi =_\Omega \phi \quad \Gamma \vdash Q : \psi =_\Omega \psi}{\Gamma \vdash P \rightarrow Q : \phi \rightarrow \psi =_\Omega \phi' \rightarrow \psi'} \\
\\
\frac{\Gamma \vdash \delta : \phi \rightarrow \psi \quad \Gamma \vdash \epsilon : \psi \rightarrow \phi}{\Gamma \vdash \mathbf{univ}_{\phi, \psi}(\delta, \epsilon) : \phi =_\Omega \psi} \quad \frac{\Gamma \vdash P : \phi =_\Omega \psi}{\Gamma \vdash P^+ : \phi \rightarrow \psi} \quad \frac{\Gamma \vdash P : \psi =_\Omega \psi}{\Gamma \vdash P^- : \psi \rightarrow \phi} \\
\\
\frac{\Gamma, x : A, y : A, e : x =_A y \vdash Mx =_B Ny}{\Gamma \vdash \mathbb{M}e : x =_A y.P : M =_{A \rightarrow B} N} \\
\\
\frac{\Gamma \vdash P : M =_{A \rightarrow B} M' \quad \Gamma \vdash Q : N =_A N'}{\Gamma \vdash PQ : MN =_B M'N'} \\
\\
\frac{\Gamma \vdash P : M =_A N \quad \Gamma \vdash M' : A \quad \Gamma \vdash N' : A}{\Gamma \vdash P : M' =_A N'} (M \simeq_\beta M', N \simeq_\beta N')
\end{array}$$

Path substitution. Every term in the system is functional; that is, it maps equal terms to equal terms. That is, if $M : B$ depends on a variable $x : A$ and we have a proof of $N =_A N'$, then we can construct a proof of $[N/x]M =_B [N'/x]M$. We call this construction *path substitution*. The definition is as follows.

Given paths P_1, \dots, P_n ; term variables x_1, \dots, x_n ; and a term M , define the path $\{P_1/x_1, \dots, P_n/x_n\}M$ as follows.

$$\begin{aligned}
\{P_1/x_1, \dots, P_n/x_n\}x_i &\stackrel{\text{def}}{=} P_i \\
\{P_1/x_1, \dots, P_n/x_n\}y &\stackrel{\text{def}}{=} \mathbf{ref}(y) & (y \neq x_1, \dots, x_n) \\
\{P_1/x_1, \dots, P_n/x_n\}\perp &\stackrel{\text{def}}{=} \mathbf{ref}(\perp) \\
\{\vec{P}/\vec{x}\}(MN) &\stackrel{\text{def}}{=} \{\vec{P}/\vec{x}\}M\{\vec{P}/\vec{x}\}N \\
\{\vec{P}/\vec{x}\}(\lambda y : A.M) &\stackrel{\text{def}}{=} \mathbb{M}e : a =_A a'.\{\vec{P}/\vec{x}, e/y\}M \\
\{\vec{P}/\vec{x}\}(\phi \rightarrow \psi) &\stackrel{\text{def}}{=} \{\vec{P}/\vec{x}\}\phi \rightarrow \{\vec{P}/\vec{x}\}\psi
\end{aligned}$$

Lemma 1. *If $\Gamma, x_1 : A_1, \dots, x_n : A_n \vdash M : B$ and $\Gamma \vdash P_i : N_i =_{A_i} N'_i$ for $i = 1, \dots, n$, then $\Gamma \vdash \{\vec{P}/\vec{x}\}M : [\vec{N}/\vec{x}]M =_B [\vec{N}'/\vec{x}]M$.*

We also define $M \bullet P$ to be $\{P/x\}(Mx)$. Thus, if $M : A \rightarrow B$ and $P : N =_A N'$, then $M \bullet P : MN =_B MN'$.

The reduction relation. We define the following reduction relation on proofs and equality proofs.

$$\begin{aligned}
(\mathbf{ref}(\phi))^+ &\rightsquigarrow \lambda x : \phi.x & (\mathbf{ref}(\phi))^- &\rightsquigarrow \lambda x : \phi.x & \mathbf{univ}_{\phi, \psi}(\delta, \epsilon)^+ &\rightsquigarrow \delta & \mathbf{univ}_{\phi, \psi}(\delta, \epsilon)^- &\rightsquigarrow \epsilon \\
(\mathbf{ref}(\phi) \rightarrow \mathbf{univ}_{\psi, \chi}(\delta, \epsilon)) &\rightsquigarrow \mathbf{univ}_{\phi \rightarrow \psi, \phi \rightarrow \chi}(\lambda f : \phi \rightarrow \psi. \lambda x : \phi. \delta(fx), \lambda g : \phi \rightarrow \chi. \lambda x : \phi. \epsilon(gx)) \\
(\mathbf{univ}_{\phi, \psi}(\delta, \epsilon) \rightarrow \mathbf{ref}(\chi)) &\rightsquigarrow \mathbf{univ}_{\phi \rightarrow \chi, \psi \rightarrow \chi}(\lambda f : \phi \rightarrow \chi. \lambda x : \psi. f(\epsilon x), \lambda g : \psi \rightarrow \chi. \lambda x : \phi. g(\delta x)) \\
(\mathbf{univ}_{\phi, \psi}(\delta, \epsilon) \rightarrow \mathbf{univ}_{\phi', \psi'}(\delta', \epsilon')) &\rightsquigarrow \mathbf{univ}_{\phi \rightarrow \phi', \psi \rightarrow \psi'}(\lambda f : \phi \rightarrow \phi'. \lambda x : \psi. \delta'(f(\epsilon x)), \lambda g : \psi \rightarrow \psi'. \lambda y : \phi. \epsilon'(g(\delta y))) \\
(\mathbf{ref}(\phi) \rightarrow \mathbf{ref}(\psi)) &\rightsquigarrow \mathbf{ref}(\phi \rightarrow \psi) & \mathbf{ref}(M) \mathbf{ref}(N) &\rightsquigarrow \mathbf{ref}(MN) \\
(\mathbf{ref}(\lambda x : A.M))P &\rightsquigarrow \{P/x\}M & (\mathbb{M}e : x =_A y.P)Q &\rightsquigarrow [M/x, N/y, Q/e]P
\end{aligned}$$

where, in the last clause, we have $Q : M =_A N$.

Proof of strong normalization. The proof of strong normalization follows the method of Tait [1]. We define the set of *computable* proofs of each term, terms of each type and paths of each equation (in a context Γ) as follows.

$$\begin{aligned}
E_\Gamma(\Omega) &\stackrel{\text{def}}{=} \{M \mid \Gamma \vdash M : \Omega, M \in \mathbf{SN}\} \\
E_\Gamma(A \rightarrow B) &\stackrel{\text{def}}{=} \{M \mid \Gamma \vdash M : A \rightarrow B, M \in \mathbf{SN}, \\
&\quad \forall N, N' \in E_\Gamma(A). \forall P \in E_\Gamma(N =_A N'). M \bullet P \in E_\Gamma(MN =_B MN')\} \\
E_\Gamma(\perp) &\stackrel{\text{def}}{=} \{\delta \mid \Gamma \vdash \delta : \perp, \delta \in \mathbf{SN}\} \\
E_\Gamma(\phi \rightarrow \psi) &\stackrel{\text{def}}{=} \{\delta \mid \Gamma \vdash \delta : \phi \rightarrow \psi, \delta \in \mathbf{SN}, \\
&\quad \forall \epsilon \in E_\Gamma(\phi). \delta \epsilon \in E_\Gamma(\psi)\} \\
E_\Gamma(\phi) &\stackrel{\text{def}}{=} E_\Gamma(nf(\phi)) \quad (\phi \text{ a normalizable term of type } A) \\
E_\Gamma(\phi =_\Omega \psi) &\stackrel{\text{def}}{=} \{P \mid \Gamma \vdash P : \phi =_\Omega \psi, P \in \mathbf{SN}, \\
&\quad P^+ \in E_\Gamma(\phi \rightarrow \psi), P^- \in E_\Gamma(\psi \rightarrow \phi)\} \\
E_\Gamma(M =_{A \rightarrow B} M') &\stackrel{\text{def}}{=} \{P \mid \Gamma \vdash P : M =_{A \rightarrow B} M', P \in \mathbf{SN}, \\
&\quad \forall N, N' \in E_\Gamma(A), Q \in E_\Gamma(N =_A N'). PQ \in E_\Gamma(MN =_B M'N')\}
\end{aligned}$$

A *computable substitution* $\sigma : \Gamma \rightarrow \Delta$ is a function mapping variables to terms such that, for every $x : A$ in Γ , we have $\Delta \vdash \sigma(x) : A$.

A *path* between computable substitutions $\rho : \sigma \rightarrow \sigma' : \Gamma \rightarrow \Delta$ is a function mapping variables to paths such that, for every $x : A$ in Γ , we have $\Delta \vdash \rho(x) : \sigma(x) =_A \sigma'(x)$.

Our main theorem is the following:

Theorem 1. 1. If $\sigma : \Gamma \rightarrow \Delta$ and $\Delta \vdash M : A$, then $\sigma M \in E_\Gamma(A)$.

2. If $\sigma : \Gamma \rightarrow \Delta$ and $\Delta \vdash P : M =_A N$, then $\sigma P \in E_\Gamma(\sigma M =_A \sigma N)$.

3. If $\rho : \sigma \rightarrow \sigma' : \Gamma \rightarrow \Delta$ and $\Delta \vdash M : A$, then $\rho M \in E_\Gamma(\sigma M =_A \sigma' M)$.

This is proven by induction on derivations. The difficult part is proving that each term constructed is strongly normalizing. Tait's proof used confluence, which is not available to us here. Our reduction relation is not confluent in general. However, it is confluent on the computable terms. It turns out that, with this result, the induction hypothesis always gives us confluence of the terms required.

The remainder of the proof is straightforward: the substitution *id* mapping x to x is a computable substitution. Hence, if $\Gamma \vdash M : A$, then *id* $M \equiv M$ is computable, hence in \mathbf{SN} ; and if $\Gamma \vdash P : M = N : A$ then *id* $P \equiv P$ is computable, hence in \mathbf{SN} .

Future work. In the future, we wish to extend this work in several directions. We wish to add constructions $\text{sym}(P)$ and $\text{trans}(P, Q)$ that make our equality into an equivalence relation, and provide reduction rules for each of these. We wish to extend our logic by placing the equations $M =_A N$ inside Ω , and allow quantification over any type in A (including Ω) in our logic. This would move our system ever closer to homotopy type theory. Our ultimate goal is to provide a strongly normalizing reduction relation for homotopy type theory, including the terms involving the univalence axiom, and we believe we have taken the first step.

References

- [1] W. W. Tait. Intensional interpretation of functional of finite type i. *Journal of Symbolic Logic*, 32:198–212, 1967.