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

Robin Adams[1], Marc Bezem[1], and Thierry Coquard[2]

[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.

1 Syntax and Rules of Deduction

We call the following type theory predicative higher-order propositional logic. Its syntax is given by the grammar

Proof	δ	::=	$p \mid \delta\delta \mid \lambda p : \phi.\delta$
Term	M, ϕ	::=	$x \mid \bot \mid MM \mid \lambda x : A.M \mid \phi \supset \phi$
Type	A	::=	$\Omega \mid A \to A$
Term Context	Γ	::=	$\langle\rangle \mid \Gamma, x : A$
Proof Context	Δ	::=	$\langle\rangle \mid \Delta, p : \phi$
Judgement	$\mathcal{J}$	::=	$\Gamma \text{ valid} \mid \Gamma \vdash M : A \mid \Gamma, \Delta \text{ valid} \mid \Gamma, \Delta \vdash \delta : \phi$

where p is a *proof variable* and x a *term variable*. Its rules of deduction are

$$\frac{}{\langle\rangle \text{ valid}} \qquad \frac{\Gamma \text{ valid}}{\Gamma, x : A \text{ valid}} \qquad \frac{\Gamma \vdash \phi : \Omega}{\Gamma, p : \phi \text{ valid}}$$

$$\frac{\Gamma \text{ valid}}{\Gamma \vdash x : A} \ (x : A \in \Gamma) \qquad \frac{\Gamma \text{ valid}}{\Gamma \vdash p : \phi} \ (p : \phi \in \Gamma)$$

$$\frac{\Gamma \text{ valid}}{\Gamma \vdash \bot : \Omega} \qquad \frac{\Gamma \vdash \phi : \Omega \quad \Gamma \vdash \psi : \Omega}{\Gamma \vdash \phi \to \psi : \Omega}$$

$$\frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B} \qquad \frac{\Gamma \vdash \delta : \phi \to \psi \quad \Gamma \vdash \epsilon : \phi}{\Gamma \vdash \delta\epsilon : \psi}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \to B} \qquad \frac{\Gamma, p : \phi \vdash \delta : \psi}{\Gamma \vdash \lambda p : \phi.\delta : \phi \to \psi}$$

$$\frac{\Gamma \vdash \delta : \phi \quad \Gamma \vdash \psi : \Omega}{\Gamma \vdash \delta : \psi} \ (\phi \simeq \phi)$$

1.1   Extensional Equality

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

Path	P	$::=$	$e \mid \mathsf{ref}\,(M) \mid P \supset P \mid \mathsf{univ}_{\phi,\phi}\,(\delta, \delta) \mid ⫴e : x =_A x.P \mid PP$
Proof	δ	$::=$	$\cdots \mid P^+ \mid P^-$
Context	Γ	$::=$	$\cdots \mid \Gamma, e : M =_A M$
Judgement	$\mathcal{J}$	$::=$	$\cdots \mid \Gamma \vdash P : M =_A M$

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

We add the following rules of deduction

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

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathsf{ref}\,(M) : M =_A M} \qquad \frac{\Gamma \vdash P : \phi =_\Omega \phi \quad \Gamma \vdash Q : \psi =_\Omega \psi}{\Gamma \vdash P \to Q : \phi \to \psi =_\Omega \phi' \to \psi}$$

$$\frac{\Gamma \vdash \delta : \phi \to \psi \quad \Gamma \vdash \epsilon : \psi \to \phi}{\Gamma \vdash \mathsf{univ}_{\phi,\psi}\,(\delta, \epsilon) : \phi =_\Omega \psi} \qquad \frac{\Gamma \vdash P : \phi =_\Omega \psi}{\Gamma \vdash P^+ : \phi \to \psi} \qquad \frac{\Gamma \vdash P : \psi =_\Omega \psi}{\Gamma \vdash P^- : \psi \to \phi}$$

$$\frac{\Gamma, x : A, y : A, e : x =_A y \vdash Mx =_B Ny}{\Gamma \vdash ⫴e : x =_A y.P : M =_{A \to B} N}$$

$$\frac{\Gamma \vdash P : M =_{A \to B} M' \quad \Gamma \vdash Q : N =_A N'}{\Gamma \vdash PQ : MN =_B M'N'}$$

$$\frac{\Gamma \vdash P : M =_A N \quad \Gamma \vdash M' : A \quad \Gamma \vdash N' : A}{\Gamma \vdash P : M' =_A N'} \ (M \simeq_\beta M', N \simeq_\beta N')$$

2   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 $P_1, \ldots, P_n$; term variales $x_1, \ldots, x_n$; and a term M, define the path $\{P_1/x_1, \ldots, P_n/x_n\}M$ as follows.

$$\{P_1/x_1, \ldots, P_n/x_n\}x_i \stackrel{\text{def}}{=} P_i$$

$$\{P_1/x_1, \ldots, P_n/x_n\}y \stackrel{\text{def}}{=} \mathsf{ref}\,(y) \qquad\qquad (y \not\equiv x_1, \ldots, x_n)$$

$$\{P_1/x_1, \ldots, P_n/x_n\}\bot \stackrel{\text{def}}{=} \mathsf{ref}\,(\bot)$$

$$\{\vec{P}/\vec{x}\}(MN) \stackrel{\text{def}}{=} \{\vec{P}/\vec{x}\}M\{\vec{P}/\vec{x}\}N$$

$$\{\vec{P}/\vec{x}\}(\lambda y : A.M) \stackrel{\text{def}}{=} \mathcal{W}e : a =_A a'.\{\vec{P}/\vec{x}, e/y\}M$$

$$\{\vec{P}/\vec{x}\}(\phi \to \psi) \stackrel{\text{def}}{=} \{\vec{P}/\vec{x}\}\phi \to \{\vec{P}/\vec{x}\}\psi$$

Lemma 1. *If* $\Gamma, x_1 : A_1, \ldots, x_n : A_n \vdash M : B$ *and* $\Gamma \vdash P_i : N_i =_{A_i} N_i'$ *for* $i = 1, \ldots, n$*, 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 $M : A \to B$ and $P : N =_A N'$, then $M \bullet P : MN =_B MN'$.

3 The Reduction Relation

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

$$(\mathsf{ref}\,(\phi))^+ \rightsquigarrow \lambda x : \phi.x \qquad (\mathsf{ref}\,(\phi))^- \rightsquigarrow \lambda x : \phi.x \qquad \mathsf{univ}_{\phi,\psi}\,(\delta, \epsilon)^+ \rightsquigarrow \delta \qquad \mathsf{univ}_{\phi,\psi}\,(\delta, \epsilon)^- \rightsquigarrow \epsilon$$

$$(\mathsf{ref}\,(\phi) \to \mathsf{univ}_{\psi,\chi}\,(\delta, \epsilon)) \rightsquigarrow \mathsf{univ}_{\phi \to \psi, \phi \to \chi}\,(\lambda f : \phi \to \psi.\lambda x : \phi.\delta(fx), \lambda g : \phi \to \chi.\lambda x : \phi.\epsilon(gx))$$

$$(\mathsf{univ}_{\phi,\psi}\,(\delta, \epsilon) \to \mathsf{ref}\,(\chi)) \rightsquigarrow \mathsf{univ}_{\phi \to \chi, \psi \to \chi}\,(\lambda f : \phi \to \chi.\lambda x : \psi.f(\epsilon x), \lambda g : \psi \to \chi.\lambda x : \phi.g(\delta x))$$

$$(\mathsf{univ}_{\phi,\psi}\,(\delta, \epsilon) \to \mathsf{univ}_{\phi',\psi'}\,(\delta', \epsilon')) \rightsquigarrow \mathsf{univ}_{\phi \to \phi', \psi \to \psi'}\,(\lambda f : \phi \to \phi'.\lambda x : \psi.\delta'(f(\epsilon x)), \lambda g : \psi \to \psi'.\lambda y : \phi.\epsilon'(g(\delta y)))$$

$$(\mathsf{ref}\,(\phi) \to \mathsf{ref}\,(\psi)) \rightsquigarrow \mathsf{ref}\,(\phi \to \psi) \qquad \mathsf{ref}\,(M)\,\mathsf{ref}\,(N) \rightsquigarrow \mathsf{ref}\,(MN)$$

$$(\mathsf{ref}\,(\lambda x : A.M))P \rightsquigarrow \{P/x\}M \qquad (\mathcal{W}e : x =_A y.P)Q \rightsquigarrow [M/x, N/y, Q/e]P$$

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

4 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.

$$E_\Gamma(\Omega) \stackrel{\text{def}}{=} \{M \mid \Gamma \vdash M : \Omega, M \in \mathbf{SN}\}$$

$$E_\Gamma(A \to B) \stackrel{\text{def}}{=} \{M \mid \Gamma \vdash M : A \to B, M \in \mathbf{SN}, \qquad\qquad \forall N \in E_\Gamma(A).MN \in E_\Gamma($$
$$\forall N, N' \in E_\Gamma(A).\forall P \in E_\Gamma(N =_A N).M \bullet P \in E_\Gamma(MN =_B MN')\}$$

$$E_\Gamma(\bot) \stackrel{\text{def}}{=} \{\delta \mid \Gamma \vdash \delta : \bot, \delta \in \mathbf{SN}\}$$

$$E_\Gamma(\phi \to \psi) \stackrel{\text{def}}{=} \{\delta \mid \Gamma \vdash \delta : \phi \to \psi, \delta \in \mathbf{SN},$$
$$\forall \epsilon \in E_\Gamma(\phi).\delta\epsilon \in E_\Gamma(\psi)\}$$

$$E_\Gamma(\phi) \stackrel{\text{def}}{=} E_\Gamma(nf(\phi)) \qquad\qquad (\phi \text{ a normalizable term of type}$$

$$E_\Gamma(\phi =_\Omega \psi) \stackrel{\text{def}}{=} \{P \mid \Gamma \vdash P : \phi =_\Omega \psi, P \in \mathbf{SN},$$
$$P^+ \in E_\Gamma(\phi \to \psi), P^- \in E_\Gamma(\psi \to \phi)\}$$

$$E_\Gamma(M =_{A \to B} M') \stackrel{\text{def}}{=} \{P \mid \Gamma \vdash P : M =_{A \to B} M', P \in \mathbf{SN},$$
$$\forall N, N' \in E_\Gamma(A), Q \in E_\Gamma(N =_A N').PQ \in E_\Gamma(MN =_B M'N')\}$$

A *computable substitution* $\sigma : \Gamma \to \Delta$ 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 $\rho : \sigma \to \sigma' : \Gamma \to \Delta$ 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 $\sigma : \Gamma \to \Delta$ and $\Delta \vdash M : A$, then $\sigma M \in E_\Gamma(A)$.*

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

3. If $\rho : \sigma \to \sigma' : \Gamma \to \Delta$ 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 $\mathbf{SN}$.

5 Future Work

In the future, we wish to extend this work in several directions. We wish to add constructions sym(P) and 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 iinterpretation of ffunctional of finite type i. *Journal of Symbolic Logic*, 32:198–212, 1967.