A Normalizing Computation Rule for Propositional Extensionality in Higher-Order Minimal Logic

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

1 **Universitetet i Bergen, Institutt for Informatikk, Postboks 7800, N-5020 BERGEN, Norway**
{robin.adams,bezem}@ii.uib.no

2 **Chalmers tekniska högskola, Data- och informationsteknik, 412 96 Göteborg, Sweden**
coquand@chalmers.se

Abstract

The univalence axiom expresses the principle of extensionality for dependent type theory. However, if we simply add the univalence axiom to type theory, then we lose the property of *canonicity* — that every closed term computes to a canonical 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. So we wish to find a way to compute with the univalence axiom. While this problem has been solved with the formulation of cubical type theory, where the computations are expressed using a nominal extension of lambda-calculus, it may be interesting to explore alternative solutions, which do not require such an extension.

As a first step, we present here a system of propositional higher-order minimal logic (PHOML). There are three kinds of typing judgement in PHOML. There are *terms* which inhabit *types*, which are the simple types over Ω. There are *proofs* which inhabit *propositions*, which are the terms of type Ω. The canonical propositions are those constructed from $\bot$ by implication $\supset$. Thirdly, there are *paths* which inhabit *equations* $M =_A N$, where M and N are terms of type A. There are two ways to prove an equality: reflexivity, and *propositional extensionality* — logically equivalent propositions are equal. This system allows for some definitional equalities that are not present in cubical type theory, namely that transport along the trivial path is identity.

We present a call-by-name reduction relation for this system, and prove that the system satisfies canonicity: every closed typable term head-reduces to a canonical form. This work has been formalised in Agda.

1998 ACM Subject Classification Dummy classification – please refer to `http://www.acm.org/about/class/ccs98-html`

Keywords and phrases Dummy keyword – please provide 1–5 keywords

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

The *univalence axiom* of Homotopy Type theory (HoTT) [7] postulates a constant

$$\mathsf{isotoid} : A \simeq B \to A = B$$

that is an inverse to the obvious function $A = B \to A \simeq B$. However, if we simply add this constant to Martin-Löf type theory, then we lose the important property of *canonicity* — that every closed term of type A computes to a unique canonical object of type A. When a computation reaches a point where we eliminate a path (proof of equality) formed by isotoid, it gets 'stuck'.

As possible solutions to this problem, we may try to do with a weaker property than canonicity, such as *propositional canonicity*: that every closed term of type $\mathbb{N}$ is *propositionally* equal to a numeral, as conjectured by Voevodsky. Or we may attempt to change the definition of equality to make isotoid definable [5], or add a nominal extension to the syntax of the type theory (e.g. Cubical Type Theory [2]).

We could also try a more conservative approach, and simply attempt to find a reduction relation for a type theory involving isotoid that satisfies all three of the properties above. There seems to be no reason *a priori* to believe this is not possible, but it is difficult to do because the full Homotopy Type Theory is a complex and interdependent system. We can tackle the problem by adding univalence to a much simpler system, finding a well-behaved reduction relation, then doing the same for more and more complex systems, gradually approaching the full strength of HoTT.

In this paper, we present a system we call PHOML, or predicative higher-order minimal logic. It is a type theory with three kinds of typing judgement. There are *terms* which inhabit *types*, which are the simple types over Ω. There are *proofs* which inhabit *propositions*, which are the terms of type Ω. The canonical propositions are those constructed from $\bot$ by implication $\supset$. Thirdly, there are *paths* which inhabit *equations* $M =_A N$, where M and N are terms of type A.

There are two canonical forms for proofs of $M =_\Omega N$. For any term $\varphi : \Omega$, we have $\mathrm{ref}\,(\varphi) : \varphi =_\Omega \varphi$. We also add univalence for this system, in this form: if $\delta : \varphi \supset \psi$ and $\epsilon : \psi \supset \varphi$, then $\mathrm{univ}_{\varphi,\psi}\,(\delta, \epsilon) : \varphi =_\Omega \psi$.

This entails that in PHOML, two propositions that are logically equivalent are equal. Every function of type $\Omega \to \Omega$ that can be constructed in PHOML must therefore respect logical equivalence. That is, for any F and logically equivalent x, y we must have that Fx and Fy are logically equivalent. Moreover, if for $x : \Omega$ we have that Fx is logically equivalent to Gx, then $F =_{\Omega\to\Omega} G$. Every function of type $(\Omega \to \Omega) \to \Omega$ must respect this equality; and so on. This is the manifestation in PHOML of the principle that only homotopy invariant constructions can be performed in homotopy type theory. (See Section 3.1.)

We present a call-by-name reduction relation for this system, and prove that every typable term reduces to a canonical form. From this, it follows that the system is consistent.

For the future, we wish to include the equations in Ω, allowing for propositions such as $M =_A N \supset N =_A M$. We wish to expand the system with universal quantification, and expand it to a 2-dimensional system (with equations between proofs). We then wish to add more inductive types and more dimensions, getting ever closer to full homotopy type theory.

Another system with many of the same aims is cubical type theory [2]. The system PHOML is almost a subsystem of cubical type theory. We can attempt to embed PHOML into cubical type theory, mapping Ω to the universe U, and an equation $M =_A N$ to either the type $\mathsf{Path}\,A\,M\,N$ or to $\mathrm{Id}\ A\ M\ N$. However, PHOML has more definitional equalities than the relevant fragment of cubical type theory; that is, there are definitionally equal terms in PHOML that are mapped to terms that are not definitionally equal in cubical type theory. In particular, $\mathrm{ref}\,(x)^+\,p$ and p are definitionally equal, whereas the terms $\mathrm{comp}^i x[]p$ and p are not definitionally equal in cubical type theory (but they are propositionally equal). See Section 3.2.1 for more information.

The proofs in this paper have been formalized in Agda. The formalization is available at `https://github.com/radams78/TYPES2016`.

2 Predicative Higher-Order Minimal Logic with Extensional Equality

We call the following type theory PHOML, or *predicative higher-order minimal logic with extensional equality*.

2.1 Syntax

Fix three disjoint, infinite sets of variables, which we shall call *term variables*, *proof variables* and *path variables*. We shall use x and y as term variables, p and q as proof variables, e as a path variable, and z for a variable that may come from any of these three sets.

The syntax of PHOML is given by the grammar:

$$
\begin{array}{lrcl}
\text{Type} & A, B, C & ::= & \Omega \mid A \to B \\
\text{Term} & L, M, N, \varphi, \psi, \chi & ::= & x \mid \bot \mid \varphi \supset \psi \mid \lambda x : A.M \mid MN \\
\text{Proof} & \delta, \epsilon & ::= & p \mid \lambda p : \varphi.\delta \mid \delta\epsilon \mid P^+ \mid P^- \\
\text{Path} & P, Q & ::= & e \mid \mathrm{ref}\,(M) \mid P \supset^* Q \mid \mathrm{univ}_{\varphi,\psi}\,(P, Q) \mid \\
 & & & \lambda\!\!\lambda\!\!\lambda e : x =_A y.P \mid P_{MN}Q \\
\text{Context} & \Gamma, \Delta, \Theta & ::= & \langle\rangle \mid \Gamma, x : A \mid \Gamma, p : \varphi \mid \Gamma, e : M =_A N \\
\text{Judgement} & \mathbf{J} & ::= & \Gamma \vdash \mathrm{valid} \mid \Gamma \vdash M : A \mid \Gamma \vdash \delta : \varphi \mid \\
 & & & \Gamma \vdash P : M =_A N
\end{array}
$$

In the path $\lambda\!\!\lambda\!\!\lambda e : x =_A y.P$, the term variables x and y must be distinct. (We also have $x \not\equiv e \not\equiv y$, thanks to our stipulation that term variables and path variables are disjoint.) The term variable x is bound within M in the term $\lambda x : A.M$, and the proof variable p is bound within δ in $\lambda p : \varphi.\delta$. The three variables e, x and y are bound within P in the path $\lambda\!\!\lambda\!\!\lambda e : x =_A y.P$. We identify terms, proofs and paths up to α-conversion. We write $E[z := F]$ for the result of substituting F for z within E, using α-conversion to avoid variable capture.

We shall use the word 'expression' to mean either a type, term, proof, path, or equation (an equation having the form $M =_A N$). We shall use r, s, t, S and T as metavariables that range over expressions.

Note that we use both Roman letters M, N and Greek letters φ, ψ, χ to range over terms. Intuitively, a term is understood as either a proposition or a function, and we shall use Greek letters for terms that are intended to be propositions. Formally, there is no significance to which letter we choose.

Note also that the types of PHOML are just the simple types over Ω; therefore, no variable can occur in a type.

The intuition behind the new expressions is as follows (see also the rules of deduction in Figure 2). For any object $M : A$, there is the trivial path $\mathrm{ref}\,(M) : M =_A M$. The constructor $\supset^*$ ensures congruence for $\supset$ — if $P : \varphi =_\Omega \varphi'$ and $Q : \psi =_\Omega \psi'$ then $P \supset^* Q : \varphi \supset \psi =_\Omega \varphi' \supset \psi'$. The constructor univ gives 'univalence' (propositional extensionality) for our propositions: if $\delta : \varphi \supset \psi$ and $\epsilon : \psi \supset \varphi$, then $\mathrm{univ}_{\varphi,\psi}\,(\delta, \epsilon)$ is a path of type $\varphi =_\Omega \psi$. The constructors $^+$ and $^-$ are the converses, which denote the action of transport along a path: if P is a path of type $\varphi =_\Omega \psi$, then P^+ is a proof of $\varphi \supset \psi$, and P^- is a proof of $\psi \supset \varphi$.

The constructor $\lambda\!\!\lambda\!\!\lambda$ gives functional extensionality. Let F and G be functions of type $A \to B$. If $Fx =_B Gy$ whenever $x =_A y$, then $F =_{A\to B} G$. More formally, if P is a path of type $Fx =_B Gy$ that depends on $x : A$, $y : A$ and $e : x =_A y$, then $\lambda\!\!\lambda\!\!\lambda e : x =_A y.P$ is a path of type $F =_{A\to B} G$. The proofs P^+ and P^- represent transport along the path P.

Finally, if P is a path of type $F =_{A\to B} G$, and Q is a path $M =_A N$, then $P_{MN}Q$ is a path $FM =_B GN$.

2.1.1 Substitution and Path Substitution

Intuitively, if N and N' are equal then $M[x := N]$ and $M[x := N']$ should be equal. To handle this syntactically, we introduce a notion of *path substitution*. If N, M and M' are terms, x a term variable, and P a path, then we shall define a path $N\{x := P : M = M'\}$. The intention is that, if $\Gamma \vdash P : M =_A M'$ and $\Gamma, x : A \vdash N : B$ then $\Gamma \vdash N\{x := P : M = M'\} : N[x := M] =_B N[x := M']$ (see Lemma 17).

▶ **Definition 1** (Path Substitution)**.** Given terms $M_1, \ldots, M_n$ and $N_1, \ldots, N_n$; paths $P_1, \ldots, P_n$; term variables $x_1, \ldots, x_n$; and a term L, define the path

$$L\{x_1 := P_1 : M_1 = N_1, \ldots, x_n := P_n : M_n = N_n\}$$

as follows.

$$\begin{aligned}
x_i\{\vec{x} := \vec{P} : \vec{M} = \vec{N}\} &\stackrel{\text{def}}{=} P_i \\
y\{\vec{x} := \vec{P} : \vec{M} = \vec{N}\} &\stackrel{\text{def}}{=} \mathrm{ref}\,(y) \qquad (y \not\equiv x_1, \ldots, x_n) \\
\bot\{\vec{x} := \vec{P} : \vec{M} = \vec{N}\} &\stackrel{\text{def}}{=} \mathrm{ref}\,(\bot) \\
(LL')\{\vec{x} := \vec{P} : \vec{M} = \vec{N}\} & \\
&\stackrel{\text{def}}{=} L\{\vec{x} := \vec{P} : \vec{M} = \vec{N}\}_{L'[\vec{x}:=\vec{M}]L'[\vec{x}:=\vec{N}]}L'\{\vec{x} := \vec{P} : \vec{M} = \vec{N}\} \\
(\lambda y : A.L)\{\vec{x} := \vec{P} : \vec{M} = \vec{N}\} & \\
&\stackrel{\text{def}}{=} \lll\!\!\lll e : a =_A a'.L\{\vec{x} := \vec{P} : \vec{M} = \vec{N}, y := e : a = a'\} \\
(\varphi \supset \psi)\{\vec{x} := \vec{P} : \vec{M} = \vec{N}\} &\stackrel{\text{def}}{=} \varphi\{\vec{x} := \vec{P} : \vec{M} = \vec{N}\} \supset^* \psi\{\vec{x} := \vec{P} : \vec{M} = \vec{N}\}
\end{aligned}$$

We shall often omit the endpoints $\vec{M}$ and $\vec{N}$.

▶ **Note 2.** The case $n = 0$ is permitted, and we shall have that, if $\Gamma \vdash M : A$ then $\Gamma \vdash M\{\} : M =_A M$. There are thus two paths from a term M to itself: $\mathrm{ref}\,(M)$ and $M\{\}$. There are not always equal; for example, $(\lambda x : A.x)\{\} \equiv \lll\!\!\lll e : x =_A y.e$, which (after we define the reduction relation) will not be convertible with $\mathrm{ref}\,(\lambda x : A.x)$.

The following lemma shows how substitution and path substitution interact.

▶ **Lemma 3.** *Let $\vec{y}$ be a sequences of variables and x a distinct variable. Then*

1. $M[x := N]\{\vec{y} := \vec{P} : \vec{L} = \vec{L'}\}$
   $\equiv M\{x := N\{\vec{y} := \vec{P} : \vec{L} = \vec{L'}\} : N[\vec{y} := \vec{L}] = N[\vec{y} := \vec{L'}], \vec{y} := \vec{P} : \vec{L} = \vec{L'}\}$
2. $M\{\vec{y} := \vec{P} : \vec{L} = \vec{L'}\}[x := N]$
   $\equiv M\{\vec{y} := \vec{P}[x := N] : \vec{L}[x := N] = \vec{L'}[x := N], x := \mathrm{ref}\,(N) : N = N\}$

Proof. An easy induction on M in all cases. ◀

▶ **Note 4.** The familiar substitution lemma also holds as usual: $t[\vec{z_1} := \vec{s_1}][\vec{z_2} := \vec{s_2}] \equiv t[\vec{z_1} := \vec{s_1}[\vec{z_2} := \vec{s_2}], \vec{z_2} := \vec{s_2}]$. We cannot form a lemma about the fourth case, simplifying $M\{\vec{x} := \vec{P}\}\{\vec{y} := \vec{Q}\}$, because $M\{\vec{x} := \vec{P}\}$ is a path, and path substitution can only be applied to a term.

We introduce a notation for simultaneous substitution and path substitution of several variables:

▶ **Definition 5.** A *substitution* is a function that maps term variables to terms, proof variables to proofs, and path variables to paths. We write $E[\sigma]$ for the result of substituting the expression $\sigma(z)$ for z in E, for each variable z in the domain of σ.

A *path substitution* τ is a function whose domain is a finite set of term variables, and which maps each term variable to a path. Given a path substitution τ and substitutions ρ, σ with the same domain $\{x_1, \ldots, x_n\}$, we write

$$M\{\tau : \rho = \sigma\} \text{ for } M\{x_1 := \tau(x_1) : \rho(x_1) = \sigma(x_1), \ldots, \tau(x_n) : \rho(x_n) = \sigma(x_n)\} \ .$$

2.1.2 Call-By-Name Reduction

▶ **Definition 6** (Call-By-Name Reduction)**.** Define the relation of *call-by-name reduction* $\to$ on the expressions. The inductive definition is given by the rules in Figure 1

▶ **Lemma 7** (Confluence)**.** *If $E \twoheadrightarrow F$ and $E \twoheadrightarrow G$, then there exists H such that $F \twoheadrightarrow H$ and $G \twoheadrightarrow H$.*

Proof. The proof is given in Appendix B. ◀

▶ **Lemma 8** (Reduction respects path substitution)**.** *If $M \to N$ then $M\{\tau : \rho = \sigma\} \to N\{\tau : \rho = \sigma\}$.*

Proof. Induction on $M \to N$. The only difficult case is β-contraction. We have

$$\begin{aligned}
&((\lambda x : A.M)N)\{\tau : \rho = \sigma\} \\
\equiv &(\lambda\!\!\!\lambda e : x =_A x'.M\{\tau : \rho = \sigma, x := e : x = x'\})_{N[\rho]N[\sigma]}N\{\tau : \rho = \sigma\} \\
\to &M\{\tau : \rho = \sigma, x := N\{\tau\} : N[\rho] = N[\sigma]\} \\
\equiv &M[x := N]\{\tau : \rho = \sigma\} \qquad \text{(Lemma 3)}
\end{aligned}$$

◀

We write $\to^?$ for the reflexive closure of $\to$, we write $\twoheadrightarrow$ for the reflexive transitive closure of $\to$, and we write $\simeq$ for the reflexive symmetric transitive closure of $\to$. We say an expression E is in *normal form* iff there is no expression F such that $E \to F$.

▶ **Note 9.**

1. Reduction on proofs and paths does *not* respect substitution. For example, let $F \equiv \lambda x : \Omega.x$. Then we have

$$\begin{aligned}
\text{ref}\,(\lambda y : \Omega.z)_{\perp\perp}\,\text{ref}\,(\perp) &\to z\{y := \text{ref}\,(\perp) : \perp = \perp\} \equiv \text{ref}\,(z) \\
(\text{ref}\,(\lambda y : \Omega.z)_{\perp\perp}\,\text{ref}\,(\perp))[z := F] &\equiv \text{ref}\,(\lambda x : \Omega.F)_{\perp\perp}\,\text{ref}\,(\perp) \qquad (1) \\
\text{ref}\,(z)\,[z := F] &\equiv \text{ref}\,(F) \equiv \text{ref}\,(\lambda x : \Omega.x) \qquad (2)
\end{aligned}$$

   Expression (1) does not reduce to (2). Instead, (1) reduces to

$$F\{y := \text{ref}\,(\perp) : \perp = \perp\} \equiv \lambda\!\!\!\lambda e : x =_\Omega x'.e \ .$$

2. Reduction on terms does respect substitution: if $M \to N$ then $M[x := P] \to N[x := P]$, as is easily shown by induction on $M \to N$.

Reduction on Terms

$$\frac{}{(\lambda x : A.M)N \to M[x := N]} \quad \frac{M \to M'}{MN \to M'N} \quad \frac{\varphi \to \varphi'}{\varphi \supset \psi \to \varphi' \supset \psi} \quad \frac{\psi \to \psi'}{\varphi \supset \psi \to \varphi \supset \psi'}$$

Reduction on Proofs

$$\frac{}{(\lambda p : \varphi.\delta)\epsilon \to \delta[p := \epsilon]} \quad \frac{}{\mathrm{ref}\,(\varphi)^+\,\delta \to \delta} \quad \frac{}{\mathrm{ref}\,(\varphi)^-\,\delta \to \delta}$$

$$\frac{\delta \to \delta'}{\delta\epsilon \to \delta'\epsilon} \quad \frac{}{\mathrm{univ}_{\varphi,\psi}\,(\delta,\epsilon)^+ \to \delta} \quad \frac{}{\mathrm{univ}_{\varphi,\psi}\,(\delta,\epsilon)^- \to \epsilon}$$

$$\frac{P \to Q}{P^+ \to Q^+} \quad \frac{P \to Q}{P^- \to Q^-}$$

$$\frac{\phi \to \phi'}{\lambda p : \phi.\delta \to \lambda p : \phi'.\delta}$$

Reduction on Paths

$$\frac{}{(\lambda\!\!\lambda\!\!\lambda e : x =_A y.P)_{MN}Q \to P[x := M, y := N, e := Q]}$$

$$\frac{}{\mathrm{ref}\,(\lambda x : A.M)_{NN'}\,P \to M\{x := P : N = N'\}}$$

$$\frac{}{\mathrm{ref}\,(\varphi) \supset^* \mathrm{ref}\,(\psi) \to \mathrm{ref}\,(\varphi \supset \psi)}$$

$$\frac{}{\mathrm{ref}\,(\varphi) \supset^* \mathrm{univ}_{\psi,\chi}\,(\delta,\epsilon) \to \mathrm{univ}_{\varphi\supset\psi,\varphi\supset\chi}\,(\lambda p : \varphi \supset \psi.\lambda q : \varphi.\delta(pq), \lambda p : \varphi \supset \chi.\lambda q : \varphi.\epsilon(pq))}$$

$$\frac{}{\mathrm{univ}_{\varphi,\psi}\,(\delta,\epsilon) \supset^* \mathrm{ref}\,(\chi) \to \mathrm{univ}_{\varphi\supset\chi,\psi\supset\chi}\,(\lambda p : \varphi \supset \chi.\lambda q : \psi.p(\epsilon q), \lambda p : \psi \supset \chi.\lambda q : \varphi.p(\delta q))}$$

$$\frac{}{\begin{array}{l}\mathrm{univ}_{\varphi,\psi}\,(\delta,\epsilon) \supset^* \mathrm{univ}_{\varphi',\psi'}\,(\delta',\epsilon') \\ \to \mathrm{univ}_{\varphi\supset\varphi',\psi\supset\psi'}\,(\lambda p : \varphi \supset \varphi'.\lambda q : \psi.\delta'(p(\epsilon q)), \lambda p : \psi \supset \psi'.\lambda q : \varphi.\epsilon'(p(\delta q)))\end{array}}$$

$$\frac{P \to P'}{P_{MN}Q \to P'_{MN}Q} \quad \frac{M \to N}{\mathrm{ref}\,(M) \to \mathrm{ref}\,(N)} \quad \frac{P \to P'}{P \supset^* Q \to P' \supset^* Q} \quad \frac{Q \to Q'}{P \supset^* Q \to P \supset^* Q'}$$

$$\frac{\phi \to \phi'}{\mathrm{univ}_{\phi,\psi}\,(\delta,\epsilon) \to \mathrm{univ}_{\phi',\psi}\,(\delta,\epsilon)} \quad \frac{\psi \to \psi'}{\mathrm{univ}_{\phi,\psi}\,(\delta,\epsilon) \to \mathrm{univ}_{\phi,\psi'}\,(\delta,\epsilon)}$$

$$\frac{\delta \to \delta'}{\mathrm{univ}_{\phi,\psi}\,(\delta,\epsilon) \to \mathrm{univ}_{\phi,\psi}\,(\delta',\epsilon)} \quad \frac{\epsilon \to \epsilon'}{\mathrm{univ}_{\phi,\psi}\,(\delta,\epsilon) \to \mathrm{univ}_{\phi,\psi}\,(\delta,\epsilon')}$$

Figure 1 Reduction in PHOML

Contexts

$$(\langle\rangle)\ \frac{}{\langle\rangle \vdash \text{valid}} \qquad (\text{ctx}_T)\ \frac{\Gamma \vdash \text{valid}}{\Gamma, x : A \vdash \text{valid}} \qquad (\text{ctx}_P)\ \frac{\Gamma \vdash \varphi : \Omega}{\Gamma, p : \varphi \vdash \text{valid}}$$

$$(\text{ctx}_E)\ \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : A}{\Gamma, e : M =_A N \vdash \text{valid}}$$

$$(\text{var}_T)\ \frac{\Gamma \vdash \text{valid}}{\Gamma \vdash x : A}\ (x : A \in \Gamma) \qquad (\text{var}_P)\ \frac{\Gamma \vdash \text{valid}}{\Gamma \vdash p : \varphi}\ (p : \varphi \in \Gamma)$$

$$(\text{var}_E)\ \frac{\Gamma \vdash \text{valid}}{\Gamma \vdash e : M =_A N}\ (e : M =_A N \in \Gamma)$$

Terms

$$(\bot)\ \frac{\Gamma \vdash \text{valid}}{\Gamma \vdash \bot : \Omega} \qquad (\supset)\ \frac{\Gamma \vdash \varphi : \Omega \quad \Gamma \vdash \psi : \Omega}{\Gamma \vdash \varphi \supset \psi : \Omega}$$

$$(\text{app}_T)\ \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B} \qquad (\lambda_T)\ \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \to B}$$

Proofs

$$(\text{app}_P)\ \frac{\Gamma \vdash \delta : \varphi \supset \psi \quad \Gamma \vdash \epsilon : \varphi}{\Gamma \vdash \delta\epsilon : \psi} \qquad (\lambda_P)\ \frac{\Gamma, p : \varphi \vdash \delta : \psi}{\Gamma \vdash \lambda p : \varphi.\delta : \varphi \supset \psi}$$

$$(\text{conv}_P)\ \frac{\Gamma \vdash \delta : \varphi \quad \Gamma \vdash \psi : \Omega}{\Gamma \vdash \delta : \psi}\ (\varphi \simeq \psi)$$

Paths

$$(\text{ref})\ \frac{\Gamma \vdash M : A}{\Gamma \vdash \text{ref}\,(M) : M =_A M} \qquad (\supset^*)\ \frac{\Gamma \vdash P : \varphi =_\Omega \varphi' \quad \Gamma \vdash Q : \psi =_\Omega \psi'}{\Gamma \vdash P \supset^* Q : \varphi \supset \psi =_\Omega \varphi' \supset \psi'}$$

$$(\text{univ})\ \frac{\Gamma \vdash \delta : \varphi \supset \psi \quad \Gamma \vdash \epsilon : \psi \supset \varphi}{\Gamma \vdash \text{univ}_{\varphi,\psi}\,(\delta, \epsilon) : \varphi =_\Omega \psi}$$

$$(\text{plus})\ \frac{\Gamma \vdash P : \varphi =_\Omega \psi}{\Gamma \vdash P^+ : \varphi \supset \psi} \qquad (\text{minus})\ \frac{\Gamma \vdash P : \psi =_\Omega \psi}{\Gamma \vdash P^- : \psi \supset \varphi}$$

$$(\lambda\!\!\lambda\!\!\lambda)\ \frac{\begin{array}{c}\Gamma, x : A, y : A, e : x =_A y \vdash P : Mx =_B Ny \\ \Gamma \vdash M : A \to B \quad \Gamma \vdash N : A \to B\end{array}}{\Gamma \vdash \lambda\!\!\lambda\!\!\lambda e : x =_A y.P : M =_{A\to B} N}$$

$$(\text{app}_E)\ \frac{\Gamma \vdash P : M =_{A\to B} M' \quad \Gamma \vdash Q : N =_A N' \quad \Gamma \vdash N : A \quad \Gamma \vdash N' : A}{\Gamma \vdash P_{NN'}Q : MN =_B M'N'}$$

$$(\text{conv}_E)\ \frac{\Gamma \vdash P : M =_A N \quad \Gamma \vdash M' : A \quad \Gamma \vdash N' : A}{\Gamma \vdash P : M' =_A N'}\ (M \simeq M', N \simeq N')$$

Figure 2 Rules of Deduction of λoe

2.2 Rules of Deduction

The rules of deduction of PHOML are given in Figure 2.

2.2.1 Metatheorems

In the lemmas that follow, the letter $\mathcal{J}$ stands for any of the expressions that may occur to the right of the turnstile in a judgement, i.e. valid, $M : A$, $\delta : \varphi$, or $P : M =_A N$.

▶ **Lemma 10** (Context Validity). *Every derivation of* $\Gamma, \Delta \vdash \mathcal{J}$ *has a subderivation of* $\Gamma \vdash$ valid.

Proof. Induction on derivations. ◀

▶ **Lemma 11** (Weakening). *If* $\Gamma \vdash \mathcal{J}$, $\Gamma \subseteq \Delta$ *and* $\Delta \vdash$ valid *then* $\Delta \vdash \mathcal{J}$.

Proof. Induction on derivations. ◀

▶ **Lemma 12** (Type Validity).

1. *If* $\Gamma \vdash \delta : \varphi$ *then* $\Gamma \vdash \varphi : \Omega$.
2. *If* $\Gamma \vdash P : M =_A N$ *then* $\Gamma \vdash M : A$ *and* $\Gamma \vdash N : A$.

Proof. Induction on derivations. The cases where δ or P is a variable use Context Validity. ◀

▶ **Lemma 13** (Generation).

1. *If* $\Gamma \vdash x : A$ *then* $x : A \in \Gamma$.
2. *If* $\Gamma \vdash \bot : A$ *then* $A \equiv \Omega$.
3. *If* $\Gamma \vdash \varphi \supset \psi : A$ *then* $\Gamma \vdash \varphi : \Omega$, $\Gamma \vdash \psi : \Omega$ *and* $A \equiv \Omega$.
4. *If* $\Gamma \vdash \lambda x : A.M : B$ *then there exists* C *such that* $\Gamma, x : A \vdash M : C$ *and* $B \equiv A \to C$.
5. *If* $\Gamma \vdash MN : A$ *then there exists* B *such that* $\Gamma \vdash M : B \to A$ *and* $\Gamma \vdash N : B$.
6. *If* $\Gamma \vdash p : \varphi$, *then there exists* ψ *such that* $p : \psi \in \Gamma$ *and* $\varphi \simeq \psi$.
7. *If* $\Gamma \vdash \lambda p : \varphi.\delta : \psi$, *then there exists* χ *such that* $\Gamma, p : \varphi \vdash \delta : \chi$ *and* $\psi \simeq (\varphi \supset \chi)$.
8. *If* $\Gamma \vdash \delta\epsilon : \varphi$ *then there exists* ψ *such that* $\Gamma \vdash \delta : \psi \supset \varphi$ *and* $\Gamma \vdash \epsilon : \psi$.
9. *If* $\Gamma \vdash e : M =_A N$, *then there exist* M', N' *such that* $e : M' =_A N' \in \Gamma$ *and* $M \simeq M'$, $N \simeq N'$.
10. *If* $\Gamma \vdash \mathrm{ref}\,(M) : N =_A P$, *then we have* $\Gamma \vdash M : A$ *and* $M \simeq N \simeq P$.
11. *If* $\Gamma \vdash P \supset^* Q : \varphi =_A \psi$, *then there exist* φ_1, φ_2, ψ_1, ψ_2 *such that* $\Gamma \vdash P : \varphi_1 =_\Omega \psi_1$, $\Gamma \vdash Q : \varphi_2 =_\Omega \psi_2$, $\varphi \simeq (\varphi_1 \supset \psi_1)$, $\psi \simeq (\varphi_2 \supset \psi_2)$, *and* $A \equiv \Omega$.
12. *If* $\Gamma \vdash \mathrm{univ}_{\varphi,\psi}\,(\delta, \epsilon) : \chi =_A \theta$, *then we have* $\Gamma \vdash \delta : \varphi \supset \psi$, $\Gamma \vdash \epsilon : \psi \supset \varphi$, $\chi \simeq \varphi$, $\theta \simeq \psi$ *and* $A \equiv \Omega$.
13. *If* $\Gamma \vdash \lambda\!\!\lambda\!\!\lambda e : x =_A y.P : M =_B N$ *then there exists* C *such that* $\Gamma, x : A, y : A, e : x =_A y \vdash P : Mx =_C Ny$ *and* $B \equiv A \to C$.
14. *If* $\Gamma \vdash P_{MM'}Q : N =_A N'$, *then there exist* B, F *and* G *such that* $\Gamma \vdash P : F =_{B \to A} G$, $\Gamma \vdash Q : M =_B M'$, $N \simeq FM$ *and* $N' \simeq GM'$.
15. *If* $\Gamma \vdash P^+ : \varphi$, *then there exist* ψ, χ *such that* $\Gamma \vdash P : \psi =_\Omega \chi$ *and* $\varphi \simeq (\psi \supset \chi)$.
16. *If* $\Gamma \vdash P^- : \varphi$, *there exist* ψ, χ *such that* $\Gamma \vdash P : \psi =_\Omega \chi$ *and* $\varphi \simeq (\chi \supset \psi)$.

Proof. Induction on derivations. ◀

2.2.2 Substitutions

▶ **Definition 14.** Let Γ and Δ be contexts. A *substitution from* Γ *to* Δ[1], $\sigma : \Gamma \Rightarrow \Delta$, is a substitution whose domain is $\operatorname{dom}\Gamma$ such that:

- for every term variable $x : A \in \Gamma$, we have $\Delta \vdash \sigma(x) : A$;
- for every proof variable $p : \varphi \in \Gamma$, we have $\Delta \vdash \sigma(p) : \varphi[\sigma]$;
- for every path variable $e : M =_A N \in \Gamma$, we have $\Delta \vdash \sigma(e) : M[\sigma] =_A N[\sigma]$.

▶ **Lemma 15** (Well-Typed Substitution). *If* $\Gamma \vdash \mathcal{J}$, $\sigma : \Gamma \Rightarrow \Delta$ *and* $\Delta \vdash \text{valid}$, *then* $\Delta \vdash \mathcal{J}[\sigma]$.

Proof. Induction on derivations. ◀

▶ **Definition 16.** If $\rho, \sigma : \Gamma \Rightarrow \Delta$ and τ is a path substitution whose domain is the term variables in $\operatorname{dom}\Gamma$, then we write $\tau : \sigma = \rho : \Gamma \Rightarrow \Delta$ iff, for each variable $x : A \in \Gamma$, we have $\Delta \vdash \tau(x) : \sigma(x) =_A \rho(x)$.

▶ **Lemma 17** (Path Substitution). *If* $\tau : \sigma = \rho : \Gamma \Rightarrow \Delta$ *and* $\Gamma \vdash M : A$ *and* $\Delta \vdash \text{valid}$, *then* $\Delta \vdash M\{\tau : \sigma = \rho\} : M[\sigma] =_A M[\rho]$.

Proof. Induction on derivations. ◀

▶ **Proposition 18** (Subject Reduction). *If* $\Gamma \vdash s : T$ *and* $s \twoheadrightarrow t$ *then* $\Gamma \vdash t : T$.

Proof. It is sufficient to prove the case $s \to t$. The proof is by a case analysis on $s \to t$, using the Generation, Well-Typed Substitution and Path Substitution Lemmas. ◀

2.2.3 Canonicity

▶ **Definition 19** (Canonical Object).

- The canonical objects θ of Ω, or *canonical propositions*, are given by the grammar

$$\theta ::= \bot \mid \theta \supset \theta$$

- A canonical object of type $A \to B$ has the form $\lambda x : A.M$.

We define the *canonical proofs* of a canonical object θ of Ω as follows:

- There is no canonical proof of $\bot$.
- A canonical proof of $\varphi \supset \psi$ has the form $\lambda p : \varphi.\delta$.

We define the *canonical paths* of an equation $M =_A N$ as follows. (Note that this depends only on the type A, not on M and N.)

- A canonical path of $\varphi =_\Omega \psi$ is one of the form $\operatorname{ref}(M)$ or $\operatorname{univ}_{\varphi,\psi}(\delta, \epsilon)$.
- A canonical path of $F =_{A\to B} G$ is one of the form $\operatorname{ref}(M)$ or $\lambda\!\!\lambda\!\!\lambda e : x =_A y.P$.

▶ **Lemma 20.** *Suppose* φ *reduces to a canonical proposition* θ, *and* $\varphi \simeq \psi$. *Then* ψ *reduces to* θ.

Proof. This follows from the fact that $\to$ satisfies the diamond property, and every canonical proposition θ is a normal form. ◀

[1] These have also been called *context morphisms*, for example in Hoffman [3]. Note however that what we call a substitution from Γ to Δ is what Hoffman calls a context morphism from Δ to Γ.

2.2.4 Neutral Expressions

▶ **Definition 21** (Neutral). The *neutral* terms, paths and proofs are given by the grammar

$$\begin{array}{lrcl} \text{Neutral term} & M_n & ::= & x \mid M_n N \\ \text{Neutral proof} & \delta_n & ::= & p \mid P_n^+ \mid P_n^- \mid \delta_n \epsilon \\ \text{Neutral path} & P_n & ::= & e \mid P_n \supset^* Q \mid Q \supset^* P_n \mid (P_n)_{MN} Q \end{array}$$

3 Examples

We present two examples illustrating the way that proofs and paths behave in PHOML. In each case, we compare the example with the same construction performed in cubical type theory.

3.1 Functions Respect Logical Equivalence

As discussed in the introduction, every function of type $\Omega \to \Omega$ that can be constructed in PHOML must respect logical equivalence. This fact can actually be proved in PHOML, in the following sense: there exists a proof δ of

$$f : \Omega \to \Omega, x : \Omega, y : \Omega, p : x \supset y, q : y \supset x \vdash \delta : fx \supset fy$$

and a proof of $fy \supset fx$ in the same context. Together, these can be read as a proof of 'if $f : \Omega \to \Omega$ and x and y are logically equivalent, then fx and fy are logically equivalent'.

Specifically, take

$$\delta \stackrel{\text{def}}{=} (\mathrm{ref}\,(f)_{xy}\,\mathrm{univ}_{x,y}\,(p,q))^+ \ .$$

Note that this is not possible in Martin-Löf Type Theory.

In cubical type theory, we can construct a term δ such that

$$f : \mathsf{Prop} \to \mathsf{Prop}, x : \mathsf{Prop}, y : \mathsf{Prop}, p : x.1 \to y.1, q : y.1 \to x.1 \vdash \delta : (fx).1 \to (fy).1$$

In fact, we can go further and prove that equality of propositions is equal to logical equivalence. That is, we can prove

$$\mathsf{Path}\,U\,(\mathsf{Path}\,\mathsf{Prop}\,x\,y)\,((x.1 \to y.1) \times (y.1 \to x.1)) \ .$$

3.2 Computation with Paths

Let $\top \stackrel{\text{def}}{=} \bot \supset \bot$. Using propositional extensionality, we can construct a path of type $\top = \top \supset \top$, and hence a proof of $\top \supset (\top \supset \top)$. But which of the canonical proofs of $\top \supset (\top \supset \top)$ have we constructed?

We define

$$\top := \bot \supset \bot, \quad \iota := \lambda p : \bot.p, \quad I := \lambda x : \Omega.x, \quad F := \lambda x : \Omega.\top \supset x, \quad H := \lambda h.h\top \ .$$

Let Γ be the context

$$\Gamma \stackrel{\text{def}}{=} x : \Omega, y : \Omega, e : x =_\Omega y \ .$$

Then we have

$$\begin{array}{ll} \Gamma \vdash \lambda p : \top \supset x.e^+(p\iota) & :(\top \supset x) \supset y \\ \Gamma \vdash \lambda m : y.\lambda n : \top.e^- m & :y \supset (\top \supset x) \\ \Gamma \vdash \mathsf{univ}\left(\lambda p : \top \supset x.e^- m, \lambda m : y.\lambda n : \top.e^- m\right) & :(\top \supset x) =_\Omega y \end{array}$$

Let $P \equiv \mathsf{univ}\,(\lambda p : \top \supset x.e^{+}(p\iota), \lambda m : y.\lambda n : \top.e^{-}m)$. Then

$$\therefore \vdash \lll e : x =_\Omega y.P \qquad :F =_{\Omega\to\Omega} I \tag{3}$$
$$\therefore \vdash (\mathrm{ref}\,(H))_{FI}(\lll e : x =_\Omega y.P) \qquad :(\top \supset \top) =_\Omega \top \tag{4}$$
$$\therefore \vdash ((\mathrm{ref}\,(H))_{FI}(\lll e : x =_\Omega y.P))^{-} \qquad :\top \supset (\top \supset \top) \tag{5}$$

And now we compute:

$$\begin{aligned}
&((\mathrm{ref}\,(H))_{FI}(\lll e : x =_\Omega y.P))^{-}\\
\to&((h\top)\{h := \lll e : x =_\Omega y.P : F = I\})^{-}\\
\equiv&((\lll e : x =_\Omega y.P)_{\top\top}(\mathrm{ref}\,(\top)))^{-}\\
\to&(P[x := \top, y := \top, e := \mathrm{ref}\,(\top)])^{-}\\
\equiv&\mathsf{univ}\left(\lambda p : \top \supset \top.\mathrm{ref}\,(\top)^{+}\,(p\iota), \lambda m : \top.\lambda n : \top.\mathrm{ref}\,(\top)^{-}\,m\right)^{-}\\
\to&\lambda m : \top.\lambda n : \top.\mathrm{ref}\,(\top)^{-}\,m
\end{aligned}$$

Therefore, given proofs $\delta, \epsilon : \top$, we have

$$((\mathrm{ref}\,(H))_{FI}(\lll e : x =_\Omega y.P))^{-}\delta\epsilon \to \delta \ .$$

Thus, the construction gives a proof of $\top \supset (\top \supset \top)$ which, given two proofs of $\top$, selects the first. We could have anticipated this: consider the context $\Delta \stackrel{\mathrm{def}}{=} X : \Omega, Y : \Omega, p : X$. By replacing in our example some occurrences of $\top$ with X and others with Y, and replacing ι with p, we can obtain a path

$$Y =_\Omega X \supset Y$$

and hence a proof of $Y \supset (X \supset Y)$. By parametricity, any proof that we can construct in the context Δ of this proposition must return the left input.

3.2.1 Comparison with Cubical Type Theory

In cubical type theory, we say that a type A is a *proposition* iff any two terms of type A are propositionally equal; that is, there exists a path between any two terms of type A. Let

$$\mathsf{isProp}\,(A) \stackrel{\mathrm{def}}{=} \Pi x, y : A.\mathsf{Path}\,A\,x\,y$$

and let Prop be the type of all types in U that are propositions:

$$\mathsf{Prop} \stackrel{\mathrm{def}}{=} \Sigma X : U.\mathsf{isProp}\,(X) \ .$$

Let $\bot$ be any type in the universe U that is a proposition; that is, there exists a term of type $\mathsf{isProp}\,(\bot)$. ($\bot$ may be the empty type, but we do not require this in what follows.)

Define

$$\top := \bot \to \bot$$

Then there exists a term $\top_{\mathsf{Prop}}$ of type $\mathsf{isProp}\,(\top)$ (we omit the details). Define

$$I := \lambda X : \mathsf{Prop}.X.1, \quad F := \lambda X : \mathsf{Prop}.\top \to X.1, \quad H := \lambda h.h(\top, \top_{\mathsf{Prop}})$$

Then we have

$$\vdash \top : U \quad \vdash I : \mathsf{Prop} \to U \quad \vdash F : \mathsf{Prop} \to U \quad \vdash H : (\mathsf{Prop} \to U) \to U$$

From the fact that univalence is provable in cubical type theory [2], we can construct a term Q such that

$$\vdash Q : \mathsf{Path}\,(\mathsf{Prop} \to U)\,I\,F \quad .$$

Hence we have

$$\vdash \langle i \rangle H(Qi) : \mathsf{Path}\,U\,H I\,H F$$

which is definitionally equal to

$$\vdash \langle i \rangle H(Qi) : \mathsf{Path}\,U\,\top \to \top\,\top$$

From this, we can apply transport to create a term of type $\top \to \top \to \top$. Applying this to any terms $\delta, \epsilon : \top$ gives a term that is definitionally equal to

$$Q\delta\epsilon = \mathsf{mapid}_\top\,\mathsf{mapid}_\top\,\delta$$

where mapid represents transport across the trivial path:

$$\mathsf{mapid}_A\,t \stackrel{\text{def}}{=} \mathsf{comp}^i\,A\,[]\,t \qquad (i \text{ does not occur in } A) \quad .$$

(For the details of the calculation, see Appendix A.)

In the version of cubical type theory given in [1], we have $\mathsf{mapid}_X\,x$ is definitionally equal to x, and therefore $Q\delta\epsilon = \delta$, just as in PHOML. This is no longer true in the version of cubical type theory given in [2].

4 Computable Expressions

We now proceed with the proof of canonicity for PHOML. Our proof follows the lines of the Girard-Tait reducibility method [6]: we define what it means to be a *computable* term (proof, path) of a given type (proposition, equation), and prove: (1) every typable expression is computable (2) every computable expression reduces to either a neutral or a canonical expression. In particular, every closed computable expression reduces to a canonical expression.

In this section, we use E, F, S and T as metavariables that range over expressions. In each case, either E and F are terms and S and T are types; or E and F are proofs and S and T are propositions; or E and F are paths and S and T are equations.

▶ **Definition 22** (Computable Expression)**.** We define the relation $\models E : T$, read 'E is a computable expression of type T', as follows.

- $\models \delta : \bot$ iff δ reduces to a neutral proof.
- For φ and ψ canonical propositions, $\models \delta : \varphi \supset \psi$ iff, for all ϵ such that $\models \epsilon : \varphi$, we have $\models \delta\epsilon : \psi$.
- If φ reduces to the canonical proposition ψ, then $\models \delta : \varphi$ iff $\models \delta : \psi$.
- $\models P : \varphi =_\Omega \psi$ iff $\models P^+ : \varphi \supset \psi$ and $\models P^- : \psi \supset \varphi$.
- $\models P : M =_{A \to B} M'$ iff, for all Q, N, N' such that $\models N : A$ and $\models N' : A$ and $\models Q : N =_A N'$, then we have $\models P_{NN'}Q : MN =_B M'N'$.
- $\models M : A$ iff $\models M\{\} : M =_A M$.

Note that the last three clauses define $\models M : A$ and $\models P : M =_A N$ simultaneously by recursion on A.

▶ **Definition 23** (Computable Substitution)**.** Let σ be a substitution with domain $\mathrm{dom}\,\Gamma$. We write $\models \sigma : \Gamma$ and say that σ is a *computable* substitution on Γ iff, for every entry $z : T$ in Γ, we have $\models \sigma(z) : T[\sigma]$.

We write $\models \tau : \rho = \sigma : \Gamma$, and say τ is a *computable* path substitution between ρ and σ, iff, for every term variable entry $x : A$ in Γ, we have $\models \tau(x) : \rho(x) =_A \sigma(x)$.

▶ **Lemma 24** (Conversion)**.** *If* $\models E : S$ *and* $S \simeq T$ *then* $\models E : T$*.*

Proof. This follows easily from the definition and Lemma 20. ◀

▶ **Lemma 25** (Expansion)**.** *If* $\models F : T$ *and* $E \to F$ *then* $\models E : T$*.*

Proof. An easy induction, using the fact that call-by-name reduction respects path substitution (Lemma 8). ◀

▶ **Lemma 26** (Reduction)**.** *If* $\models E : T$ *and* $E \to F$ *then* $\models F : T$*.*

Proof. An easy induction, using the fact that call-by-name reduction is confluent (Lemma 7). ◀

▶ **Definition 27.** We introduce a closed term c_A for every type A such that $\models c_A : A$.

$$c_\Omega \stackrel{\text{def}}{=} \bot$$
$$c_{A\to B} \stackrel{\text{def}}{=} \lambda x : A.c_B$$

▶ **Lemma 28.** $\models c_A : A$

Proof. An easy induction on A. ◀

▶ **Lemma 29.** *If* $\models E : T$ *then* E *reduces to either a neutral term or canonical expression of* T*.*

Proof. We prove by induction on the canonical proposition θ that, if $\models \delta : \theta$, then δ reduces to a neutral proof or a canonical proof of θ.

If $\models \delta : \bot$ then δ reduces to a neutral proof. Now, suppose $\models \delta : \theta \supset \theta'$. Then $\models \delta p : \theta'$, so δp reduces to either a neutral proof or canonical proof by the induction hypothesis. This reduction must proceed either by reducing δ to a neutral proof, or reducing δ to a λ-proof then β-reducing.

We then prove by induction on the type A that, if $\models P : M =_A N$, then P reduces to a neutral path or a canonical path. The two cases are straightforward.

Now, suppose $\models M : A$, i.e. $\models M\{\} : M =_A M$. Let $A \equiv A_1 \to \cdots \to A_n \to \Omega$. Then

$$\models M\{\}_{c_{A_1}c_{A_1}} c_{A_1}\{\}_{c_{A_2}c_{A_2}} \cdots c_{A_n}\{\} : Mc_{A_1}\cdots c_{A_n} =_\Omega Mc_{A_1}\cdots c_{A_n} \ .$$

Therefore, $Mc_{A_1}\cdots c_{A_n}$ reduces to a canonical proposition. The reduction must consist either in reducing M to a canonical proposition (if $n = 0$), or reducing M to a λ-expression then performing a β-reduction. In either case, M reduces to a canonical term. ◀

▶ **Lemma 30.** *If* $\models M : A \to B$ *then* M *reduces to a* λ*-expression.*

Proof. Similar to the last paragraph of the previous proof. ◀

▶ **Lemma 31.** *For any term* φ *that reduces to a canonical proposition, we have* $\models \mathrm{ref}\,(\varphi) : \varphi =_\Omega \varphi$*.*

Proof. In fact we prove that, for any terms M and φ such that φ reduces to a canonical proposition, we have $\models \mathrm{ref}\,(M) : \varphi =_\Omega \varphi$.

It is sufficient to prove the case where φ is a canonical proposition. We must show that $\models \mathrm{ref}\,(M)^+ : \varphi \supset \varphi$ and $\models \mathrm{ref}\,(M)^- : \varphi \supset \varphi$. So let $\models \delta : \varphi$. Then $\models \mathrm{ref}\,(M)^+\,\delta : \varphi$ and $\models \mathrm{ref}\,(M)^-\,\delta : \varphi$ by Expansion (Lemma 25), as required. ◀

▶ **Lemma 32.** *$\models \varphi : \Omega$ if and only if φ reduces to a canonical proposition.*

Proof. If $\models \varphi : \Omega$ then $\models \varphi\{\}^+ : \varphi \supset \varphi$. Therefore $\varphi \supset \varphi$ reduces to a canonical proposition, and so φ must reduce to a canonical proposition.

Conversely, suppose φ reduces to a canonical proposition θ. We have $\varphi\{\} \twoheadrightarrow \theta\{\}$, and $\theta\{\} \twoheadrightarrow \mathrm{ref}\,(\theta)$ for every canonical proposition θ. Therefore, $\models \varphi\{\} : \varphi =_\Omega \varphi$ by Expansion (Lemma 25). Hence $\models \varphi : \Omega$. ◀

▶ **Lemma 33.** *If δ is a neutral proof and φ reduces to a canonical proposition, then $\models \delta : \varphi$.*

Proof. It is sufficient to prove the case where φ is a canonical proposition. The proof is by induction on φ.

If $\varphi \equiv \bot$, then $\models \delta : \bot$ immediately from the definition.

If $\varphi \equiv \psi \supset \chi$, then let $\models \epsilon : \psi$. We have that $\delta\epsilon$ is neutral, hence $\models \delta\epsilon : \chi$ by the induction hypothesis. ◀

▶ **Lemma 34.** *Let $\models M : A$ and $\models N : A$. If P is a neutral path, then $\models P : M =_A N$.*

Proof. The proof is by induction on A.

For $A \equiv \Omega$: we have that P^+ and P^- are neutral proofs, and M and N head reduce to canonical propositions, so $\models P^+ : M \supset N$ and $\models P^- : N \supset M$ by the previous lemma, as required.

For $A \equiv B \to C$: let $\models L : B$, $\models L' : B$ and $\models Q : L =_B L'$. Then we have $\models ML : C$, $\models NL' : C$ and $P_{LL'}Q$ is a neutral path, hence $\models P_{LL'}Q : ML =_C NL'$ by the induction hypothesis, as required. ◀

▶ **Lemma 35.** *If $\models M : A$ then $\models \mathrm{ref}\,(M) : M =_A M$.*

Proof. If $A \equiv \Omega$, this is just Lemma 31.

So suppose $A \equiv B \to C$. Using Lemma 30, Reduction (Lemma 26) and Expansion (Lemma 25), we may assume that M is a λ-term. Let $M \equiv \lambda y : D.N$.

Let $\models L : B$ and $\models L' : B$ and $\models P : L =_B L'$. We must show that

$$\models \mathrm{ref}\,(\lambda y : D.N)_{LL'}\,P : (\lambda y : D.N)L =_C (\lambda y : D.N)L' \ .$$

By Expansion and Conversion, it is sufficient to prove

$$\models N\{y := P : L = L'\} : N[y := L] =_C N[y := L'] \ .$$

We have that $\models (\lambda y : D.N)\{\} : \lambda y : D.N =_{B\to C} \lambda y : D.N$, and so

$$\models (\lambda\!\!\lambda e : y =_D y'.N\{y := e : y = y'\})_{LL'}P : (\lambda y : D.N)L =_C (\lambda y : D.N)L' \ ,$$

and the result follows by Reduction and Conversion. ◀

▶ **Lemma 36.** *If $\models P : \varphi =_\Omega \varphi'$ and $\models Q : \psi =_\Omega \psi'$ then $\models P \supset^* Q : \varphi \supset \psi =_\Omega \varphi' \supset \psi'$.*

Proof. By Reduction (Lemma 26) and Expansion (Lemma 25), we may assume that P and Q are either neutral, or have the form $\mathrm{ref}\,(-)$ or $\mathrm{univ}_{-,-}\,(-,-)$.

If either P or Q is neutral then $P \supset^* Q$ is neutral, and the result follows from Lemma 34.

Otherwise, let $\models \delta : \varphi \supset \psi$ and $\epsilon \models \varphi'$. We must show that $\models (P \supset^* Q)^+ \delta\epsilon : \psi'$.

If $P \equiv \mathrm{ref}\,(M)$ and $Q \equiv \mathrm{ref}\,(N)$, then we have

$$(P \supset^* Q)^+ \delta\epsilon \to \mathrm{ref}\,(M \supset N)^+\, \delta\epsilon \to \delta\epsilon \ .$$

Now, $\models P^-\epsilon : \varphi$, hence $\models \epsilon : \varphi$ by Reduction, and so $\models \delta\epsilon : \psi$. Therefore, $\models Q^+(\delta\epsilon) : \psi'$, and hence by Reduction $\models \delta\epsilon : \psi'$ as required.

If $P \equiv \mathrm{ref}\,(M)$ and $Q \equiv \mathrm{univ}_{N,N'}\,(\chi, \chi')$, then we have

$$\begin{aligned}(P \supset^* Q)^+ \delta\epsilon &\to \mathrm{univ}_{M\supset N, M\supset N'}\,(\lambda pq.\chi(pq), \lambda pq.\chi'(pq))^+\, \delta\epsilon \\ &\to (\lambda pq.\chi(pq))\delta\epsilon \\ &\twoheadrightarrow \chi(\delta\epsilon)\end{aligned}$$

We have $\models P^-\epsilon : \varphi$, hence $\models \epsilon : \varphi$ by Reduction, and so $\models \delta\epsilon : \psi$. Therefore, $\models Q^+(\delta\epsilon) : \psi'$, and hence by Reduction $\models \chi(\delta\epsilon) : \psi'$ as required.

The other two cases are similar. ◀

▶ **Lemma 37.** *If* $\models \delta : \phi \supset \psi$ *and* $\models \epsilon : \psi \supset \phi$ *then* $\models \mathrm{univ}_{\phi,\psi}\,(\delta, \epsilon) : \phi =_\Omega \psi$.

Proof. We must show that $\models \mathrm{univ}_{\phi,\psi}\,(\delta,\epsilon)^+ : \phi \supset \psi$ and $\models \mathrm{univ}_{\phi,\psi}\,(\delta,\epsilon)^- : \psi \supset \phi$. These follow from the hypotheses, using Expansion (Lemma 25). ◀

5 Proof of Canonicity

▶ **Theorem 38.** *1. If* $\Gamma \vdash \mathcal{J}$ *and* $\models \sigma : \Gamma$, *then* $\models \mathcal{J}[\sigma]$.
2. If $\Gamma \vdash M : A$ *and* $\models \tau : \rho = \sigma : \Gamma$, *then* $\models M\{\tau : \rho = \sigma\} : M[\rho] =_A M[\sigma]$.

Proof. The proof is by induction on derivations. Most cases are straightforward, using the lemmas from Section 4. We deal with one case here, the rule (λ_T).

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \to B}$$

1. We must show that

$$\models \lambda x : A.M[\sigma] : A \to B \ .$$

So let $\models Q : N =_A N'$. Define the path substitution τ by

$$\tau(x) \equiv Q, \qquad \tau(y) \equiv \mathrm{ref}\,(\sigma(y)) \ \ (y \in \mathrm{dom}\,\Gamma)$$

Then we have $\models \tau : (\sigma, x := N) = (\sigma, x := N')$, and so the induction hypothesis gives

$$\models M\{\tau\} : M[\sigma, x := N] =_B M[\sigma, x := N']$$

We observe that $M\{\tau\} \equiv M[\sigma]\{x := Q : N = N'\}$ (Lemma 3), and so by Expansion (Lemma 25) and Conversion (Lemma 24) we have

$$\models (\lambda x : A.M[\sigma])\{\}_{NN'}Q : (\lambda x : A.M[\sigma])N =_B (\lambda x : A.M[\sigma])N'$$

as required.

2. We must show that

$$\models \lambda\!\!\lambda\!\!\lambda e : x =_A y.M\{\tau : \rho = \sigma, x := e : x = y\} : \lambda x : A.M[\rho] =_{A\to B} \lambda x : A.M[\sigma] \ .$$

So let $\models P : N =_A N'$. The induction hypothesis gives

$$\models M\{\tau : \rho = \sigma, x := P : N = N'\} : M[\rho, x := N] =_B M[\sigma, x := N'] \ ,$$

and so we have

$$\models (\lambda\!\!\lambda\!\!\lambda e : x =_A y.M\{\tau : \rho = \sigma, x := e : x = y\})_{NN'}P : (\lambda x : A.M[\rho])N =_B (\lambda x : A.M[\sigma])N'$$

by Expansion and Conversion, as required. ◀

▶ **Corollary 39.** *1. If $\Gamma \vdash \delta : \phi$ then δ reduces to a neutral proof or canonical proof.*
2. If $\Gamma \vdash P : M =_A N$ then P reduces to a neutral path or canonical path.

Proof. For every context Γ, define the substitution σ_Γ by

$$\sigma_\Gamma(x) \stackrel{\text{def}}{=} c_A \ (x : A \in \Gamma) \qquad \sigma_\Gamma(p) \stackrel{\text{def}}{=} p \qquad \sigma_\Gamma(e) \stackrel{\text{def}}{=} e \ .$$

If $\Gamma \vdash$ valid then $\models \sigma_\Gamma : \Gamma$ using Lemmas 33 and 34.

Therefore, if $\Gamma \vdash E : T$ then $\models E[\sigma_\Gamma] : T[\sigma_\Gamma]$, and so $E[\sigma_\Gamma]$ reduces to a neutral expression or canonical expression. Each reduction step is either a reduction that can be carried out in E, or a reduction within one of the terms c_A. By carrying out the same reduction steps in E, we see that E can be reduced to a neutral expression or canonical expression. ◀

▶ **Corollary 40** (Canonicity)**.**
*1. **(Consistency)** There is no δ such that $\vdash \delta : \bot$.*
2. If $\vdash \delta : \varphi \supset \psi$, then δ reduces to $\lambda p : \varphi'.\epsilon$ where $\varphi \simeq \varphi'$ and $p : \varphi' \vdash \epsilon : \psi$.
3. If $\vdash P : \varphi =_\Omega \psi$ then either P reduces to $\mathrm{ref}\,(\varphi')$ and $\varphi' \simeq \varphi \simeq \psi$; or P reduces to $\mathrm{univ}_{\varphi',\psi'}\,(\lambda p : \varphi'.\delta, \lambda q : \psi'.\epsilon)$ where $\varphi \simeq \varphi'$, $\psi \simeq \psi'$, $p : \varphi' \vdash \delta : \psi'$ and $q : \psi' \vdash \epsilon : \varphi'$.
4. If $\vdash P : F =_{A\to B} G$ then either P reduces to $\mathrm{ref}\,(M)$ and $M \simeq F \simeq G$; or P reduces to $\lambda\!\!\lambda\!\!\lambda e : x =_A y.Q$ where $x : A, y : A, e : x =_A y \vdash Q : Fx =_B Gy$.

Proof. A closed expression cannot be neutral, so from the previous corollary every typed closed expression must reduce to a canonical expression. We now apply case analysis to the possible forms of canonical expression, and use the Generation Lemma. ◀

6 Conclusion

We have presented a system with propositional extensionality, and shown that it satisfies the property of canonicity. This gives hope that it will be possible to find a computation rule for homotopy type theory that satisfies canonicity, and that does not involve extending the type theory, either with a nominal extension of the syntax as in cubical type theory or otherwise.

We now intend to do the same for stronger and stronger systems, getting ever closer to full homotopy type theory. The next steps will be:

- a system where the equations $M =_A N$ are objects of Ω, allowing us to form propositions such as $M =_A N \supset N =_A M$.
- a system with universal quantification over the types A, allowing us to form propositions such as $\forall x : A.x =_A x$ and $\forall x, y : A.x =_A y \supset y =_A x$

References

1 Marc Bezem, Thierry Coquand, and Simon Huber. A Model of Type Theory in Cubical Sets. In Ralph Matthes and Aleksy Schubert, editors, *19th International Conference on Types for Proofs and Programs (TYPES 2013)*, volume 26 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 107–128, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2014/4628`.

2 Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical type theory: a constructive interpretation of the univalence axiom. *CoRR*, abs/1611.02108, 2016. URL: `http://arxiv.org/abs/1611.02108`.

3 Martin Hofmann. Syntax and semantics of dependent types. In *Semantics and Logics of Computation*, pages 79–130. Cambridge University Press, 1997.

4 Zhaohui Luo. *Computation and Reasoning: A Type Theory for Computer Science*. Number 11 in International Series of Monographs on Computer Science. Oxford University Press, 1994.

5 Andrew Polonsky. Internalization of extensional equality. Preprint `http://arxiv.org/abs/1401.1148`, 2014.

6 W. W. Tait. Intensional iinterpretation of ffunctional of finite type i. *Journal of Symbolic Logic*, 32:198–212, 1967.

7 The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. `https://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.

A Calculation in Cubical Type Theory

We can prove that, if X is a proposition, then the type $\Sigma f : \top \to X.Path\, X\, x\, (fI)$ is contractible (we omit the details). Let $e[X, x, p]$ be the term such that

$$X : \mathsf{Prop}, x : X.1, p : \Sigma f : \top \to X.1.\mathsf{Path}\, X.1\, x\, (fI)$$
$$\vdash e[X, x, p] : \mathsf{Path}\, (\Sigma f : \top \to X.1.\mathsf{Path}\, X.1\, x\, (fI))\, \langle \lambda t : \top.x, 1_{X.1} \rangle\, p$$

Let $\mathsf{step}_2[X, x] \stackrel{\text{def}}{=} \langle \langle \lambda t : T.x, 1_{X.1} \rangle, \lambda p : \Sigma f : T \to X.1.\mathsf{Path}\, X.1\, x\, (fI).e[X, x, p] \rangle$. Then

$$X : \mathsf{Prop}, x : X.1 \vdash \mathsf{step}_2[X, x] : \mathsf{isContr}(\Sigma f : T \to X.1.\mathsf{Path}\, X.1\, x\, (fI)) \ .$$

Let $\mathsf{step}_3[X] \equiv \lambda x : X.1.\mathsf{step}_2[X, x]$. Then

$$X : \mathsf{Prop} \vdash \mathsf{step}_3[X] : \mathsf{isEquiv}\, (T \to X.1)\, X.1\, (\lambda f : T \to X.1.fI) \ .$$

Let $E[X] \equiv \langle \lambda f : \top \to X.1.fI, step3[X] \rangle$. Then

$$X : \mathsf{Prop} \vdash E[X] : \mathsf{Equiv}\, (\top \to X.1)\, X.1$$

From this equivalence, we want to get a path from $\top \to X.1$ to $X.1$ in U. We apply the proof of univalence in [2]

Let $P[X] \equiv \langle i \rangle \mathsf{Glue}[(i = 0) \mapsto (\top \to X.1, E[X]), (i = 1) \mapsto (X.1, equiv^k X.1)]X.1$. Then

$$X : \mathsf{Prop} \vdash P[X] : \mathsf{Path}\, U\, (\top \to X.1)\, X.1$$

Let $Q \equiv \langle i \rangle \lambda x : \mathsf{Prop}.P[X]i$. Then

$$\vdash Q : \mathsf{Path}\, (\mathsf{Prop} \to U)\, F\, I$$

This is the term in cubical type theory that corresponds to $\lll e : x =_\Omega y.P$ in PHOML (formula 3). We now construct terms corresponding to formulas (4) and (5):

$$\vdash \langle i \rangle H(Qi) : \mathsf{Path}\, U\, (\top \to \top)\, \top$$

$$\vdash \lambda x : \top.\mathsf{comp}^i(H(Q(1-i)))[]x : \top \to \top \to \top$$

Let us write output for this term:

$$\mathsf{output} \stackrel{\text{def}}{=} \lambda x : \top.\mathsf{comp}^i(H(Q(1-i)))[]x \ .$$

And we calculate (using the notation from [2] section 6.2):

$$\begin{aligned}
&\mathsf{output} \\
&= \lambda x : \top.\mathsf{comp}^i(Q(1-i)\top)[]x \\
&= \lambda x : \top.\mathsf{comp}^i(P[\top](1-i))[]x \\
&= \lambda x : \top.\mathsf{comp}^i(\mathsf{Glue}[(i=1) \mapsto (\top \to \top, E[\top]), (i=0) \mapsto (\top, \mathsf{equiv}^k \top)]\top)[]x \\
&= \lambda x : \top.\mathsf{glue}[1_{\mathbb{F}} \mapsto t_1]a_1 \\
&= \lambda x : \top.t_1 \\
&= \lambda x : \top.(\mathsf{equiv}\, E[\top]\, []\, \mathsf{mapid}_\top\, x).1 \\
&= \lambda x : \top.(\mathsf{contr}(\mathit{step}2[\top, \mathsf{mapid}_\top\, x])[]).1 \\
&= \lambda x : \top.(\mathsf{comp}^i \\
&\quad (\Sigma f : \top \to \top.\mathsf{Path}\, \top\, (\mathsf{mapid}_\top\, x)\, (fI)) \\
&\quad [] \\
&\quad \langle \lambda t : \top.\mathsf{mapid}_\top\, x, 1_{\mathit{mapid}_\top(x)} \rangle).1 \\
&= \lambda x : \top.\mathsf{mapid}_{\top\to\top}\, (\lambda y : \top.\mathsf{mapid}_\top\, x)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathsf{output}\, m\, n \\
&= \mathsf{mapid}_{\top\to\top}\, (\lambda y : \top.\mathsf{mapid}_\top\, m)n \\
&\equiv (\mathsf{comp}^i(\top \to \top)[](\lambda_.\top.\mathsf{mapid}_\top\, m))n \\
&= \mathsf{mapid}_\top\, \mathsf{mapid}_\top\, m
\end{aligned}$$

B Proof of Confluence

The proof follows the same lines as the proof given in [4].

▶ **Definition 41** (Parallel One-Step Reduction)**.** Define the notion of *parallel one-step reduction* $\rhd$ by the rules given in Figure 3. Let $\rhd^*$ be the transitive closure of $\rhd$.

▶ **Lemma 42.** *1. If $E \to F$ then $E \rhd F$.*
2. If $E \twoheadrightarrow F$ then $E \rhd^* F$.
3. If $E \rhd^* F$ then $E \twoheadrightarrow F$.

Proof. These are easily proved by induction. ◀

Our reason for defining $\rhd$ is that it satisfies the diamond property:

Reflexivity

$$\frac{}{E \rhd E}$$

Reduction on Terms

$$\frac{}{(\lambda x : A.M)N \rhd M[x := N]} \quad \frac{M \rhd M'}{MN \rhd M'N} \quad \frac{\varphi \rhd \varphi' \quad \psi \rhd \psi'}{\varphi \supset \psi \rhd \varphi' \supset \psi'}$$

Reduction on Proofs

$$\frac{}{(\lambda p : \varphi.\delta)\epsilon \rhd \delta[p := \epsilon]} \quad \frac{}{\mathrm{ref}\,(\varphi)^+ \rhd \lambda p : \varphi.p} \quad \frac{}{\mathrm{ref}\,(\varphi)^- \rhd \lambda p : \varphi.p}$$

$$\frac{\delta \rhd \delta'}{\delta\epsilon \rhd \delta'\epsilon} \quad \frac{}{\mathrm{univ}_{\varphi,\psi}\,(\delta,\epsilon)^+ \rhd \delta} \quad \frac{}{\mathrm{univ}_{\varphi,\psi}\,(\delta,\epsilon)^- \rhd \epsilon}$$

$$\frac{P \rhd Q}{P^+ \rhd Q^+} \quad \frac{P \rhd Q}{P^- \rhd Q^-}$$

$$\frac{\phi \rhd \phi'}{\lambda p : \phi.\delta \rhd \lambda p : \phi'.\delta}$$

Reduction on Paths

$$\frac{}{(\lambda\!\!\lambda e : x =_A y.P)_{MN}Q \rhd P[x := M, y := N, e := Q]}$$

$$\frac{}{\mathrm{ref}\,(\lambda x : A.M)_{NN'}\,P \rhd M\{x := P : N = N'\}}$$

$$\frac{}{\mathrm{ref}\,(\varphi) \supset^* \mathrm{ref}\,(\psi) \rhd \mathrm{ref}\,(\varphi \supset \psi)}$$

$$\frac{}{\mathrm{ref}\,(\varphi) \supset^* \mathrm{univ}_{\psi,\chi}\,(\delta,\epsilon) \rhd \mathrm{univ}_{\varphi\supset\psi,\varphi\supset\chi}\,(\lambda p : \varphi \supset \psi.\lambda q : \varphi.\delta(pq), \lambda p : \varphi \supset \chi.\lambda q : \varphi.\epsilon(pq))}$$

$$\frac{}{\mathrm{univ}_{\varphi,\psi}\,(\delta,\epsilon) \supset^* \mathrm{ref}\,(\chi) \rhd \mathrm{univ}_{\varphi\supset\chi,\psi\supset\chi}\,(\lambda p : \varphi \supset \chi.\lambda q : \psi.p(\epsilon q), \lambda p : \psi \supset \chi.\lambda q : \varphi.p(\delta q))}$$

$$\frac{}{\begin{array}{l}\mathrm{univ}_{\varphi,\psi}\,(\delta,\epsilon) \supset^* \mathrm{univ}_{\varphi',\psi'}\,(\delta',\epsilon') \\ \rhd \mathrm{univ}_{\varphi\supset\varphi',\psi\supset\psi'}\,(\lambda p : \varphi \supset \varphi'.\lambda q : \psi.\delta'(p(\epsilon q)), \lambda p : \psi \supset \psi'.\lambda q : \varphi.\epsilon'(p(\delta q)))\end{array}}$$

$$\frac{P \rhd P'}{P_{MN}Q \rhd P'_{MN}Q} \quad \frac{M \rhd N}{\mathrm{ref}\,(M) \rhd \mathrm{ref}\,(N)} \quad \frac{P \rhd P' \quad Q \rhd Q'}{P \supset^* Q \rhd P' \supset^* Q}$$

$$\frac{\phi \rhd \phi' \quad \psi \rhd \psi' \quad \delta \rhd \delta' \quad \epsilon \rhd \epsilon'}{\mathrm{univ}_{\phi,\psi}\,(\delta,\epsilon) \to \mathrm{univ}_{\phi',\psi'}\,(\delta',\epsilon')}$$

Figure 3 Parallel One-Step Reduction

▶ **Lemma 43** (Diamond Property). *If $E \rhd F$ and $E \rhd G$ then there exists an expression H such that $F \rhd H$ and $G \rhd H$.*

Proof. The proof is by case analysis on $E \rhd F$ and $E \rhd G$. We give the details for one case here:

$$\mathrm{ref}\,(\phi) \supset^* \mathrm{ref}\,(\psi) \rhd \mathrm{ref}\,(\phi \supset \psi) \quad \text{and} \quad \mathrm{ref}\,(\phi) \supset^* \mathrm{ref}\,(\psi) \rhd \mathrm{ref}\,(\phi') \supset^* \mathrm{ref}\,(\psi')$$

where $\phi \rhd \phi'$ and $\psi \rhd \psi'$. In this case, we have $\mathrm{ref}\,(\phi \supset \psi) \rhd \mathrm{ref}\,(\phi' \supset \psi')$ and $\mathrm{ref}\,(\phi') \supset^* \mathrm{ref}\,(\psi') \rhd \mathrm{ref}\,(\phi' \supset \psi')$. ◀

▶ **Corollary 44.** *If $E \rhd^* F$ and $E \rhd^* G$ then there exists H such that $F \rhd^* H$ and $G \rhd^* H$.*

▶ **Corollary 45.** *If $E \twoheadrightarrow F$ and $E \twoheadrightarrow G$ then $F \twoheadrightarrow H$ and $G \twoheadrightarrow H$.*

Proof. Immediate from the previous corollary and Lemma 42. ◀