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

Robin Adams[1], Marc Bezem[1], and Thierry Coquand[3]

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

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

Abstract

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. So we wish to find a way to compute with the univalence axiom.

As a first step, we present here a system of higher-order minimal propositional logic, with a universe Ω of propositions closed under implication. We add a type $M =_A N$ for any terms M, N of type A, and two ways to prove an equality: reflexivity, and *propositional extensionality* — logically equivalent propositions are equal. We present a head reduction relation for this system, and prove that the system satisfies canonicity: every closed typable term head-reduces to a canonical form.

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 rules of deduction of a type theory are traditionally justified by a *meaning explanation* [3], in which to know that a given term has a given type is to know that it computes to a *canonical object* of that type. A necessary condition for such a meaning explanation is that the type theory should have the following syntactic properties:

- **Confluence** — The reduction relation should be confluent.
- **Normalization** — Every well-typed term should reduce to a normal form.
- Every closed normal form of type A is a canonical object of type A.

From these three properties, we have:

- **Canonicity** — Every term of type A reduces to a unique canonical object of type A.

It is desirable to have, in addition, *strong normalization*, so that we know that we are free to choose whatever reduction strategy we please.

The *univalence axiom* of Homotopy Type theory (HoTT) [5] breaks the property of canonicity. It postulates a constant

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

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Acces; Article No. 23; pp. 23:1–23:9
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

that is an inverse to the canonical function $A = B \to A \simeq B$. 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*. We may attempt to prove 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[4], or extend the type theory with higher dimensions (e.g. Cubical Type Theory[1]).

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 λoe, or predicative higher-order minimal logic. It is a type theory with two universes: the universe Ω of *propositions*, and the universe of *types*. The propositions are closed under $\supset$ (implication) and include $\bot$ (falsehood), and an equality proposition $M =_A N$ for any type A and terms $M : A$ and $N : A$. The types include Ω itself and are closed under $\to$ (non-dependent function type).

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

We present a deteriministic head reduction relation for this system, and prove that every typable term head reduces to a canonical form. From this, it follows that the system is consistent. In the appendix, we present a proof of strong normalization for a different reduction relation.

For the future, we wish to expand the system with universal quantification, and expand it to a 2-dimensional system (with equations between proofs).

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

2 Predicative Higher-Order Minimal Logic with Extensional Equality

We call the following type theory λoe, 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 λoe is given by the grammar:

Type	A, B, C	$::=$	$\Omega \mid A \to B$
Term	$L, M, N, \phi, \psi, \chi$	$::=$	$x \mid \bot \mid \phi \supset \psi \mid \lambda x : A.M \mid MN$
Proof	δ, ϵ	$::=$	$p \mid \lambda p : \phi.\delta \mid \delta\epsilon \mid P^+ \mid P^-$
Path	P, Q	$::=$	$e \mid \mathrm{ref}\,(M) \mid P \supset^* Q \mid \mathrm{univ}_{\phi,\psi}\,(P, Q) \mid$ $\lambda\!\!\lambda\!\!\lambda e : x =_A y.P \mid P_{MN}Q$
Context	Γ, Δ, Θ	$::=$	$\langle\rangle \mid \Gamma, x : A \mid \Gamma, p : \phi \mid \Gamma, e : M =_A N$
Judgement	$\mathbf{J}$	$::=$	$\Gamma \vdash \mathrm{valid} \mid \Gamma \vdash M : A \mid \Gamma \vdash \delta : \phi \mid$ $\Gamma \vdash P : M =_A N$

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 $\lambda p : \phi.\delta$. 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 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 λoe 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 1). For any object $M : A$, there is the trivial path $\mathrm{ref}\,(M) : M =_A M$. The constructor $\supset^*$ ensures congruence for $\supset$ — if $P : \phi =_\Omega \phi'$ and $Q : \psi =_\Omega \psi'$ then $P \supset^* Q : \phi \supset \psi =_\Omega \phi' \supset \psi'$. The constructor univ gives univalence for our propositions: if $\delta : \phi \supset \psi$ and $\epsilon : \psi \supset \phi$, then $\mathrm{univ}_{\phi,\psi}\,(\delta, \epsilon)$ is a path of type $\phi =_\Omega \psi$. The constructors $^+$ and $^-$ are the converses: if P is a path of type $\phi =_\Omega \psi$, then P^+ is a proof of $\phi \supset \psi$, and P^- is a proof of $\psi \supset \phi$.

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 $F =_{A \to B} G$.

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 $FM =_B GN$.

2.2 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 $N\{x := P : M \sim M'\}$. The intention is that, if $\Gamma \vdash P : M =_A M'$ and $\Gamma, x : A \vdash N : B$ then $\Gamma \vdash N\{x := P : M \sim M'\} : N[x := M] =_B N[x := M']$ (see Lemma 13).

▶ **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 $L\{x_1 := P_1 : M_1 \sim$

$N_1, \ldots, x_n := P_n : M_n \sim N_n\}$ as follows.

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

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

Note

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, $(\lambda x : A.x)\{\} \equiv \lambda\!\!\!\lambda\!\!\!\lambda e : x =_A y.e$, which (after we define the reduction relation) will not be convertible with $\mathrm{ref}\,(\lambda x : A.x)$.

► **Lemma 2.**

$$M\{\vec{x} := \vec{P} : \vec{M} \sim \vec{N}\} \equiv M\{\vec{x} := \vec{P} : \vec{M} \sim \vec{N}, y := \mathrm{ref}\,(y) : y \sim y\}$$

Proof. An easy induction on M. ◄

The following lemma shows how substitution and path substitution interact.

► **Lemma 3** (Substitution). *Let $\vec{x}$ and $\vec{y}$ be a disjoint sequences of variables. Then*

1. $M[x := N]\{\vec{y} := \vec{P} : \vec{L} \sim \vec{L'}\}$

$\equiv M\{x := N\{\vec{y} := \vec{P} : \vec{L} \sim \vec{L'}\} : N[\vec{y} := \vec{L}] \sim N[\vec{y} := \vec{L'}], \vec{y} := \vec{P} : \vec{L} \sim \vec{L'}\}$

2. $M\{\vec{y} := \vec{P} : \vec{L} \sim \vec{L'}\}[x := N]$

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

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

Note

The familiar substitution lemma also holds: $t[\vec{z_1} := \vec{s_1}][\vec{z_2} := \vec{s_2}] \equiv t[\vec{z_1} := \vec{s_1}[\vec{z_2} := \vec{s_2}], \vec{z_2} := \vec{s_2}]$. 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.

► **Definition 4.** 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

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

Given substitutions σ, ρ, ρ' and a path substitution τ, let $\tau \bullet_{\rho,\rho'} \sigma$ be the path substitution defined by

$$(\tau \bullet_{\rho,\rho'} \sigma)(x) \stackrel{\text{def}}{=} \sigma(x)\{\tau : \rho \sim \rho'\}$$

▶ **Lemma 5.** $M[\sigma]\{\tau : \rho \sim \rho'\} \equiv M\{\tau \bullet_{\rho\rho'} \sigma : \rho \circ \sigma \sim \rho' \circ \sigma\}$

Proof. An easy induction on M. ◀

2.3 Rules of Deduction

The rules of deduction of λoe are given in Figure 1. In these rules, $\simeq_\beta$ denotes the usual relation of β-convertibility between terms.

2.4 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, $M : A$, $\delta : \phi$, or $P : M =_A N$.

▶ **Lemma 6** (Context Validity). *Every derivation of* $\Gamma, \Delta \vdash \mathcal{J}$ *has a subderivation of* $\Gamma \vdash \text{valid}$.

Proof. Induction on derivations. ◀

▶ **Lemma 7** (Weakening). *If* $\Gamma \vdash \mathcal{J}$, $\Gamma \subseteq \Delta$ *and* $\Delta \vdash \text{valid}$ *then* $\Delta \vdash \mathcal{J}$.

Proof. Induction on derivations. ◀

▶ **Lemma 8** (Type Validity).
1. *If* $\Gamma \vdash \delta : \phi$ *then* $\Gamma \vdash \phi : \Omega$.
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 9** (Generation).
1. *If* $\Gamma \vdash x : A$ *then* $x : A \in \Gamma$.
2. *If* $\Gamma \vdash \bot : A$ *then* $A \equiv \Omega$.
3. *If* $\Gamma \vdash \phi \supset \psi : A$ *then* $\Gamma \vdash \phi : \Omega$, $\Gamma \vdash \psi : \Omega$ *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* $\Gamma \vdash p : \phi$, *then there exists* ψ *such that* $p : \psi \in \Gamma$ *and* $\phi \simeq \psi$.
7. *If* $\Gamma \vdash \lambda p : \phi.\delta : \psi$, *then there exists* χ *such that* $\Gamma, p : \phi \vdash \delta : \chi$ *and* $\psi \simeq \phi \supset \chi$.
8. *If* $\Gamma \vdash \delta\epsilon : \phi$ *then there exists* ψ *such that* $\Gamma \vdash \delta : \psi \supset \phi$ *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* $\Gamma \vdash \text{ref}\,(M) : N =_A P$, *then we have* $\Gamma \vdash M : A$ *and* $M \simeq N \simeq P$.
11. *If* $\Gamma \vdash P \supset^* Q : \phi =_A \psi$, *then there exist* ϕ_1, ϕ_2, ψ_1, ψ_2 *such that* $\Gamma \vdash P : \phi_1 =_\Omega \psi_1$, $\Gamma \vdash Q : \phi_2 =_\Omega \psi_2$, $\phi \simeq \phi_1 \supset \psi_1$, $\psi \simeq \phi_2 \supset \psi_2$, *and* $A \equiv \Omega$.
12. *If* $\Gamma \vdash \text{univ}_{\phi,\psi}(P, Q) : \chi =_A \theta$, *then we have* $\Gamma \vdash P : \phi \supset \psi$, $\Gamma \vdash Q : \psi \supset \phi$, $\Gamma \vdash \chi \simeq_\Delta \phi : \Omega$, $\Gamma \vdash \theta \simeq_\Delta \psi : \Omega$ *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* $\Gamma \vdash P^+ : \phi$, *then there exist* ψ, χ *such that* $\Gamma \vdash P : \psi =_\Omega \chi$ *and* $\phi \simeq (\psi \supset \chi)$.
16. *If* $\Gamma \vdash P^- : \phi$, *there exist* ψ, χ *such that* $\Gamma \vdash P : \psi =_\Omega \chi$ *and* $\phi \simeq (\chi \supset \psi)$.

Proof. Induction on derivations. ◀

Contexts

$$\frac{}{\langle\rangle \vdash \text{valid}} \qquad \frac{\Gamma \vdash \text{valid}}{\Gamma, x : A \vdash \text{valid}} \qquad \frac{\Gamma \vdash \phi : \Omega}{\Gamma, p : \phi \vdash \text{valid}} \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : A}{\Gamma, e : M =_A N \vdash \text{valid}}$$

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

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

Terms

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

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

Proofs

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

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

Paths

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

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

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

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

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

Figure 1 Rules of Deduction of λoe

2.4.1 Substitutions

▶ **Definition 10.** Let Γ and Δ be contexts. A *substitution from* Γ *to* Δ[1], $\sigma : \Gamma \Rightarrow \Delta$, is a substitution whose domain is $\mathrm{dom}\,\Gamma$ such that:

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

▶ **Lemma 11** (Well-Typed Substitution). *If* $\Gamma \vdash \mathcal{J}$*,* $\sigma : \Gamma \Rightarrow \Delta$ *and* $\Delta \vdash \mathrm{valid}$*, then* $\Delta \vdash \mathcal{J}[\sigma]$*.*

Proof. Induction on derivations. ◀

▶ **Definition 12.** If $\rho, \sigma : \Gamma \to \Delta$ and τ is a path substitution whose domain is the term variables in $\mathrm{dom}\,\Gamma$, then we write $\tau : \sigma \sim \rho : \Gamma \to \Delta$ iff, for each variable $x : A \in \Gamma$, we have $\Delta \vdash \tau(x) : \sigma(x) =_A \rho(x)$.

▶ **Lemma 13** (Path Substitution). *If* $\tau : \sigma \sim \rho : \Gamma \to \Delta$ *and* $\Gamma \vdash M : A$ *and* $\Delta \vdash \mathrm{valid}$*, then* $\Delta \vdash M\{\tau : \sigma \sim \rho\} : M[\sigma] =_A M[\rho]$*.*

Proof. Induction on derivations. ◀

▶ **Lemma 14.** *If* $\sigma : \Gamma \to \Delta$ *and* $\tau : \rho \sim \rho' : \Delta \to \Theta$ *then* $\tau \bullet_{\rho,\rho'} \sigma : \rho \circ \sigma \sim \rho' \circ \sigma : \Gamma \to \Theta$*.*

Proof. Let $x : A \in \Gamma$. We have $\Delta \vdash \sigma(x) : A$, hence $\Theta \vdash \sigma(x)\{\tau : \rho \sim \rho'\} : \sigma(x)[\rho] =_A \sigma(x)[\rho']$. ◀

▶ **Proposition 15** (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 Lemma. ◀

2.4.2 Canonicity

▶ **Definition 16** (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 $\lambda x : A.M$, where $x : A \vdash M : B$ and M is in normal form.

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 $\phi \supset \psi$ has the form $\lambda p : \phi.\delta$, where $p : \phi \vdash \delta : \psi$ and δ is in normal form.

We define the *canonical paths* of an equation $M =_A N$, where M and N are canonical objects of A, as follows:

- A canonical path of $\phi =_\Omega \psi$ is either $\mathrm{ref}\,(\phi)$ if $\phi \simeq \psi$, or $\mathrm{univ}_{\phi,\psi}\,(\delta, \epsilon)$, where δ is a canonical proof of $\phi \supset \psi$ and ϵ is a canonical proof of $\psi \supset \phi$.
- A canonical path of $F =_{A \to B} G$ is either $\mathrm{ref}\,(F)$ if $F \simeq G$, or $\lambda\!\!\lambda\!\!\lambda e : x =_A y.P$ where $x : A, y : A, e : x =_A y \vdash P : Fx =_B Gy$ and P is in normal form.

[1] These have also been called *context morphisms*, for example in Hoffman [2]. Note however that what we call a substitution from Γ to Δ is what Hoffman calls a context morphism from Δ to Γ.

3 Head Reduction

▶ **Definition 17** (Head Reduction). Define the relation of *head reduction* $\to$ on the expressions as follows:

- $(\lambda x : A.M)N \to M[x := N]$.
- If $M \to M'$ then $MN \to M'N$.
- If $\phi \to \phi'$ and $\psi \to \psi'$ then $\phi \sup \psi \to \phi' \sup \psi'$.

Note that this relation is *deterministic*: i.e. for any E, there is at most one F such that $E \to F$.

▶ **Lemma 18.** *Suppose ϕ reduces to a canonical proposition ϕ', and $\phi \simeq_\beta \psi$. Then ψ reduces to ϕ'.*

Proof. Note that on terms, our head reduction is β-reduction; that is, if ϕ reduces to ϕ', then $\phi \twoheadrightarrow_\beta \phi'$. The result follows from the Church-Rosser theorem for β-reduction, and the fact that every canonical proposition is a β-normal form. ◀

4 Computable Expressions

▶ **Definition 19** (Computable Expression). Let Γ be a context in which term variables do not occur. We define the relation $\Gamma \models E : T$, read 'E is a computable expression of type T under Γ', as follows.

- $\Gamma \models \phi : \Omega$ iff ϕ reduces to a canonical proposition.
- $\Gamma \models M : A \to B$ iff, for all N such that $\Gamma \models N : A$, we have $\Gamma \models MN : B$.
- For ϕ and ψ canonical propositions, $\Gamma \models \delta : \phi \supset \psi$ iff, for all ϵ such that $\Gamma \models \epsilon : \phi$, we have $\Gamma \models \delta\epsilon : \psi$.
- If ϕ reduces to the canonical proposition ψ, then $\Gamma \models \delta : \phi$ iff $\Gamma \models \delta : \psi$.

▶ **Definition 20** (Computable Substitution). We write $\sigma : \Gamma \to_C \Delta$, and say σ is a *computable* substitution from Γ to Δ, iff, for every entry $z : T$ in Γ, we have $\Delta \models \sigma(z) : T[\sigma]$.

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

▶ **Lemma 21** (Well-typed Expansion). *If $\Gamma \models F : T$ and $E \to F$ then $\Gamma \models E : T$.*

Proof. Easy induction. ◀

▶ **Lemma 22.** *If $\Gamma \models \phi : \Omega$ and $\Gamma \models \psi : \Omega$ then $\Gamma \models \phi \supset \psi : \Omega$.*

Proof. If ϕ reduces to the canonical proposition ϕ' and ψ reduces to ψ' then $\phi \supset \psi$ reduces to $\phi' \supset \psi'$. ◀

▶ **Lemma 23.** *If $\Gamma \models \delta : \phi$ and $\phi \simeq_\beta \psi$, then $\Gamma \models \delta : \psi$.*

Proof. This follows from the definitions and Lemma 18. ◀

5 Proof of Canonicity

▶ **Theorem 24.** *1. If $\Gamma \vdash \mathcal{J}$ and $\sigma : \Gamma \to_C \Delta$, then $\Delta \models \mathcal{J}[\sigma]$.*
2. If $\Gamma \vdash M : A$ and $\tau : \rho = \sigma : \Gamma \to_C \Delta$, then $\Delta \vdash M\{\tau : \rho = \sigma\} : M[\rho] =_A M[\sigma]$.

Proof. The proof is by induction on derivations.

1. For the (var) rules, the result is immediate from the hypothesis. The case ($\bot$) is trivial.
   For the rule ($\supset$), we use Lemma 22.
   The case (app_T) is trivial.
   For the rule (λ_T), we must show that

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

   So let $\Delta \models N : A$. Then the induction hypothesis gives

   $$\Delta \models M[\sigma, x := N] : B$$

   and so by well-typed expansion we have

   $$\Delta \models (\lambda x : A.M[\sigma])N : B$$

   as required.
   The case (app_P) is trivial.
   The case (λ_P) is similar to (λ_T), also making use of Lemma 23.
2.

◀

6 Example

7 Conclusion

References

1 Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical type theory: a constructive interpretation of the univalence axiom. Preprint `http://www.math.ias.edu/$\sim$amortberg/papers/cubicaltt.pdf`, 2015.

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

3 Per Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, 1984.

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

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