Canonicity for cubical type theory

Introduction

We consider two versions of cubical type theory (on distributive lattices[1]). In both cases, the syntax has a filling and a "gluing" operation[2] and a hierarchy of cumulative universes. In version S_1 however, filling is treated as a *constant*, and in version S_2, filling over a type has computation rules given by case on this type. So version S_2 is an extension of S_1 by giving further computation rules.

Version S_1 has a model in simplicial sets and is compatible with classical logic, while it is not clear at this point if this is the case for version S_2.

Simon Huber has proved canonicity for S_2: any closed term of type N_2 (Booleans) reduces to 0 or reduces to 1. In particular it is convertible to 0 or 1.

The goal of this note is to prove canonicity for the system S_1 in the following form: if we have a closed term t of type N_2 then we either have a path from 0 to t or a path from 1 to t. This corresponds to Voevodsky's conjecture.

A corollary of this result is that in *any* extension of S_1 where we can compute (e.g. S_2, or the system based on de Morgan algebra, or the system based on Boolean algebras) a closed term of type N_2 will compute to the *same* value, independently of the extension. Also, since simplicial sets form a model of S_1, we know that, e.g. if we compute $\pi_4(S^3)$ in any of these systems, we get the same value as the one we get for topological spaces.

1 Proof relevant reducibility predicate

We consider the cubical set model of S_1. In this model we have a hierarchy of universes of fibrant types U_n, and on top of this hierarchy a universe U_ω. We also have (following A. Swan) a type $\mathsf{Id}\ A\ a_0\ a_1$.

We can in this model consider the term model[3] of the system system S_1. We have a type of contexts Con in U_0 and if $\Gamma : \mathsf{Con}$ a type $|\Gamma|$ of closed instances of Γ. We have a type $Type_n(\Gamma)$ in U_0 of types in Γ type of level n. If $A : Type_n(\Gamma)$ we have a type $\mathsf{Term}(\Gamma, A)$ of closed terms of type A. We have the empty context () and we write simply $\mathsf{Term}(A)$ for $\mathsf{Term}((), A)$ if A is in $Type_n()$ and we write simply $Type_n$ for $Type_n()$. We have a term $U_n : Type_{n+1}$ such that $\mathsf{Term}(U_n) = Type_n$.

We have a term $N_2 : Type_0$ with two canonical elements $0\ 1 : \mathsf{Term}(N_2)$ and we have a coding function $q : \mathsf{N}_2 \to \mathsf{Term}(N_2)$ defined by $q(0) = 0$ and $q(1) = 1$.

We form then a new model of S_1. In this model a context is given by a context $\Gamma : \mathsf{Con}$ and a "predicate" $C_\Gamma : |\Gamma| \to \mathsf{U}_\omega$. A type at level n in Γ, C_Γ is given by a term $T : Type_n(\Gamma)$ and a "predicate" $C_T(\rho, \rho') : \mathsf{Term}(T\rho) \to \mathsf{U}_n$. Context extension is then obtained by taking $\Gamma.T$ and

$$C_{\Gamma.T}(\rho, t) = \Sigma(\rho' : C_\Gamma(\rho))C_T(\rho, \rho')(t)$$

A substitution in $\Delta, C_\Delta \to \Gamma, C_\Gamma$ is given by a pair σ, σ' with σ in $\Delta \to \Gamma$ and σ' in $\Pi(\nu \in |\Delta|)C_\Delta(\nu) \to C_\Gamma(\sigma\nu)$. We define then $\mathsf{Term}((\Gamma, C_\Gamma), (A, C_A))$ to be a pair a, a' with a in $\mathsf{Term}(\Gamma, A)$ and $a'\rho\rho'$ is in $C_A(\rho, \rho')(a\rho)$ if ρ is in $|\Gamma|$ and ρ' in $C_\Gamma(\rho)$.

The extension operation is defined by $(\Gamma, C_\Gamma).(A, C_A) = \Gamma.A, C_{\Gamma.A}$ where $C_{\Gamma.A}(\rho, u)$ is the set of pairs ρ', u' with $\rho' \in C_\Gamma(\rho)$ and u' in $C_A(\rho, \rho')(u)$.

[1] The same method can probably be applied to the cartesian cube model.
[2] This expresses an equivalence extension property, which is one way to formulate univalence.
[3] Actually, what we present can be done for an arbitrary model of S_1 as explained in the next section.

We define C_T as follows.

If $T = U_n$, we take $C_T(X) = \mathsf{Term}(X) \to \mathsf{U}_n$, so that $C_{U_n} : \mathsf{Term}(U_n) \to \mathsf{U}_{n+1}$.

If $T = N_2$, we take $C_T(t) = \Sigma(b : \mathsf{N}_2)\ \mathsf{Id}\ \mathsf{Term}(N_2)\ q(b)\ t$.

If $T = \Pi\ A\ B$, we take $C_T(\rho,\rho')(t) = \Pi(u : \mathsf{Term}(A\rho))\Pi(u' : C_A(\rho,\rho')(u))C_B((\rho,u),(\rho',u'))(\mathsf{app}(t,u))$.

If $T = \Sigma\ A\ B$, we take $C_T(\rho,\rho')(t) = \Sigma(u' : C_A(\rho,\rho')(t.1))C_B((\rho,t.1),(\rho',u'))(t.2)$.

If $T = \mathsf{Path}\ A\ a_0\ a_1$, we take $C_T(\rho,\rho')(t) = \mathsf{Path}^i\ C_A(\rho,\rho')(t\ i)\ a_0'\ a_1'$ where $a_0' : C_A(\rho,\rho')(a_0\rho)$ and $a_1' : C_A(\rho,\rho')(a_1\rho)$.

If $T = \mathsf{Glue}\ [\psi \mapsto (B,w)]\ A$, given C_A, and C_B, $w' : \Pi(t : B\rho)C_B(\rho,\rho')(t) \to C_A(\rho,\rho')(w\rho\ t)$ partially defined on ψ, we define[4]

$$C_T(\rho,\rho')(u) = \mathsf{Glue}\ [\psi \mapsto (C_B(\rho,\rho')(u), w'\ u)]\ C_A(\rho,\rho')(\mathsf{unglue}\ u)$$

This defines a new model of S_1. If we compute the semantics of a closed term t of type N_2 in this model we get an element t' in $\Sigma(b : \mathsf{N}_2)\mathsf{Id}\ \mathsf{Term}(N_2)\ q(b)\ t$. But in this model N_2 is the constant Boolean presheaf, hence we get the canonicity result.

We use Id in order to interpret elimination rule over N_2: if we define $f : \Pi\ N_2\ T$ by cases $\mathsf{app}(f,0) = u_0$ and $\mathsf{app}(f,1) = u_1$, and given $u_0' : C_T(0,0')(u_0)$ and $u_1' : C_T(1,1')(u_1)$, we can define

$$f' : \Pi(t : \mathsf{Term}(N_2))\Pi(t' : C_{N_2}(t))C_T(t,t')(\mathsf{app}(f,t))$$

by Id elimination, such that $f'\ 0\ 0' = u_0'$ and $f'\ 1\ 1' = u_1'$.

Since the model is effective, we can produce, given t, an *actual* value 0 or 1 and a path in $\mathsf{Term}(N_2)$ between t and this value.

A concrete example is the following. We can consider the equivalence $\neg : N_2 \to N_2$ defined by the negation. We can then consider the type $T(i) = \mathsf{Glue}\ [i = 0 \mapsto (N_2, id),\ i = 1 \mapsto (N_2, \neg)]\ N_2$ which is a path between $T(0) = N_2$ and $T(1) = N_2$ and the term $t = \mathsf{comp}^i\ T(i)\ []\ 0$ of type N_2. We compute an element in $C_{N_2}(t)$ and this produces the element 1 and a path betwen 1 and t in $\mathsf{Term}(N_2)$. This computation is possible since C_{N_2} defines a fibration over N_2.

Note that the same argument cannot apply as such for S_2 (reproving Simon Huber's result). The problem is for defining $C_{N_2}(t)$: if we express *strict* equality to 0 or 1 then we get a *non fibrant* type. It is then not so simple to define $C_{U_n}(X)$. These problems disappear in the present version where $C_{N_2}(t)$ is expressed as a *fibrant* type.

Note also that we could instead have used the *simplicial set model* of S_1. The difference is that we would get only, given a closed term of type N_2, the *classical* existence of a Boolean 0 or 1 and a path between 0 or 1 and this term.

If we have a type of natural numbers N we define inductively

$$C_N(t) = \mathsf{Id}\ \mathsf{Term}(N)\ 0\ t + \Sigma(u : \mathsf{Term}(N))(u' : C_N(u))\ \mathsf{Id}\ \mathsf{Term}(N)\ (\mathsf{succ}\ u)\ t$$

[4] It can be shown that $w'\ u$ is an equivalence.

An algebraic presentation of S_1

We work in a presheaf topos where we have an interval $\mathbb{I}$ (with a distributive lattice structure) and a presheaf $\mathbb{F}$ of "cofibrant" truth values. In the following "set" means set in the internal language of this presheaf topos.

In this framework we can define what is a model of S_1. This is a generalization of the notion of category with families.

A model is given by a set of *contexts*. If Γ, Δ are two given contexts we have a set $\Delta \to \Gamma$ of *substitutions* from Δ to Γ. These collections of sets are equipped with operations that satisfy the laws of composition in a category: we have a substitution 1 in $\Gamma \to \Gamma$ and a composition operator $\sigma\delta$ in $\Theta \to \Gamma$ if δ is in $\Theta \to \Delta$ and σ in $\Delta \to \Gamma$. Furthermore we should have $\sigma 1 = 1\sigma = \sigma$ and $(\sigma\delta)\theta = \sigma(\delta\theta)$ if $\theta : \Theta_1 \to \Theta$.

We assume to have a "terminal" context $()$: for any other context, there is a unique substitution, also written $()$, in $\Gamma \to ()$. In particular we have $()\sigma = ()$ in $\Delta \to ()$ if σ is in $\Delta \to \Gamma$.

We write $|\Gamma|$ the set of substitutions $() \to \Gamma$.

If Γ is a context we have a cumulative sequence of sets $Type_n(\Gamma)$ of *types over* Γ at level n (where n is a natural number). If A in $Type_n(\Gamma)$ and σ in $\Delta \to \Gamma$ we should have $A\sigma$ in $Type_n(\Delta)$. Furthermore $A1 = A$ and $(A\sigma)\delta = A(\sigma\delta)$. If A in $Type_n(\Gamma)$ we also have a collection $\mathsf{Term}(\Gamma, A)$ of *elements of type* A. If a in $\mathsf{Term}(\Gamma, A)$ and σ is in $\Delta \to \Gamma$ we have $a\sigma$ in $\mathsf{Term}(\Delta, A\sigma)$. Furthermore $a1 = a$ and $(a\sigma)\delta = a(\sigma\delta)$. If A is in $Type_n()$ we write $|A|$ the set $\mathsf{Term}((), A)$.

We have a *context extension operation*: if A is in $Type_n(\Gamma)$ then we can form a new context $\Gamma.A$. Furthermore there is a projection p in $\Gamma.A \to \Gamma$ and a special element q in $\mathsf{Term}(\Gamma.A, A\mathsf{p})$. If σ is in $\Delta \to \Gamma$ and A in $Type_n(\Gamma)$ and a in $\mathsf{Term}(\Delta, A\sigma)$ we have an extension operation (σ, a) in $\Delta \to \Gamma.A$. We should have $\mathsf{p}(\sigma, a) = \sigma$ and $\mathsf{q}(\sigma, a) = a$ and $(\sigma, a)\delta = (\sigma\delta, a\delta)$ and $(\mathsf{p}, \mathsf{q}) = 1$.

If a is in $\mathsf{Term}(\Gamma, A)$ we write $\langle a\rangle = (1, a)$ in $\Gamma \to \Gamma.A$. Thus if B is in $Type_n(\Gamma.A)$ and a in $\mathsf{Term}(\Gamma, A)$ we have $B\langle a\rangle$ in $Type_n(\Gamma)$. If furthermore b is in $\mathsf{Term}(\Gamma.A, B)$ we have $b\langle a\rangle$ in $\mathsf{Term}(\Gamma, B\langle a\rangle)$.

A *global* type of level n is given by a an element C in $Type_n()$. We write simply C instead of $C()$ in $Type_n(\Gamma)$ for $()$ in $\Gamma \to ()$. Given such a global element C, a global element of type C is given by an element c in $\mathsf{Term}((), C)$. We then write similarly simply c instead of $c()$ in $\mathsf{Term}(\Gamma, C)$.

Models are sometimes presented by giving a class of special maps (fibrations), where a type are modelled by a fibration and elements by a section of this fibration. In our case, the fibrations are the maps p in $\Gamma.A \to \Gamma$, and the sections of these fibrations correspond exactly to elements in $\mathsf{Term}(\Gamma, A)$. Any element a $\mathsf{Term}(\Gamma, A)$ defines a section $\langle a\rangle = (1, a) : \Gamma \to \Gamma.A$ and any such section is of this form.

1.1 Dependent product types

A category with families has *product types* if we furthermore have one operation $\Pi\ A\ B$ in $Type_n(\Gamma)$ for A is in $Type_n(\Gamma)$ and B is in $Type_n(\Gamma.A)$. We should have $(\Pi\ A\ B)\sigma = \Pi\ (A\sigma)\ (B\sigma^+)$ where $\sigma^+ = (\sigma\mathsf{p}, \mathsf{q})$. We have an abstraction operation λb in $\mathsf{Term}(\Gamma, \Pi\ A\ B)$ given b in $\mathsf{Term}(\Gamma.A, B)$. We have an application operation such that $\mathsf{app}(c, a)$ is in $\mathsf{Term}(\Gamma, B\langle a\rangle)$ if a is in $\mathsf{Term}(\Gamma, A)$ and c is in $\mathsf{Term}(\Gamma, \Pi\ A\ B)$. These operations should satisfy the equations

$$\mathsf{app}(\lambda b, a) = b\langle a\rangle \qquad c = \lambda(\mathsf{app}\ (c\mathsf{p}, \mathsf{q})) \qquad (\lambda b)\sigma = \lambda(b\sigma^+) \qquad \mathsf{app}(c, a)\sigma = \mathsf{app}(c\sigma, a\sigma)$$

where we write $\sigma^+ = (\sigma\mathsf{p}, \mathsf{q})$.

1.2 Cumulative universes

We assume to have global elements U_n in $Type_{n+1}(\Gamma)$ such that $Type_n(\Gamma) = \mathsf{Term}(\Gamma, U_n)$.

1.3 Filling operation

We assume to have a filling operation. This means that if A is in $Type_n(\Gamma)^{\mathbb{I}}$ and we have a partial family of elements in $\mathsf{Term}(\Gamma, A(i))$ for i satisfying $\psi \vee i = b$, where ψ is in $\mathbb{F}$ and $b = 0$ or 1 then we can extend this partial family to a total family of elements in $\mathsf{Term}(\Gamma, A(i))$.

The *new* component is that we don't assume any "computation rules" for these filling operations.

1.4 "Gluing" operation

This expresses that we can extend a partially defined equivalence to a totally defined equivalence (which is one way to formulate univalence). Given A in $Type_n(\Gamma)$ and T defined only on the extent ψ in $Type_n(\Gamma)$ and an equivalence w between T and A defined on the extent ψ, then we can find an element $G = \mathsf{Glue}\ [\psi \mapsto (T, w)]\ A$ in $Type_n(\Gamma)$ which restricts to T on ψ and a map unglue from G to A which restricts to w on ψ. An element of G is of the form $\mathsf{glue}\ [\psi \mapsto t]\ a$ with a in $\mathsf{Term}(\Gamma, A)$ and t partial element of $\mathsf{Term}(\Gamma, T)$ of extent ψ such that $\mathsf{app}(w, t) = a$ on ψ. It is then possible to show that the map $\mathsf{unglue} : G \to A$ is an equivalence and so univalence (equivalent to the equivalence extension property) is provable in this system.