Models of Type Theory

Thierry Coquand

August. 24, 2007

Some proofs in $\lambda*$

$$N = \Pi X : *.\ X \to (X \to X) \to X$$

$$0 = \lambda X : *.\lambda a : X.\lambda f : X \to X.\ a$$

$$S = \lambda n : N.\lambda X : *.\lambda a : X.\lambda f : X \to X.\ f\ (n\ X\ f\ a)$$

$$N_0 = \Pi X : *.\ X, \qquad N_1 = \Pi X : *.\ X \to X, \qquad \neg\ X = X \to N_0$$

Some proofs in $\lambda*$

$Id\ A\ x\ y = \Pi C : A \to *.\ C\ x \to C\ y$

We can define $C : N \to *$ such that $C\ 0 = N_1, \quad C\ (S\ x) = N_0$

We take $C\ n = n\ *\ N_1\ (\lambda X : *.N_0)$

Hence we have a proof of $\Pi x : N.\ \neg\ (Id\ N\ 0\ (S\ n))$

Interpretation of $\lambda*$

If we have

$$B : \mathsf{U},\ \epsilon : \mathsf{U} \to B,\ T : B \to \mathsf{U}$$

with $T\ (\epsilon\ A) = A$ then we can define

$$\pi\ :\ \Pi a : B.(T\ a \to B) \to B,\quad \pi\ a\ F : B = \epsilon\ (\Pi x : T\ a.T\ (F\ x))$$

such that $T\ (\pi\ a\ F) = \Pi x : T\ a.T\ (F\ x)$ and $b : B = \epsilon\ B$ such that $T\ b = B$

Interpretation of $\lambda*$

$$[*] = b, \quad [\Pi x : A.B] = \pi\ [A]\ (\lambda x : T\ [A].[B])$$

$$[\lambda x : A.M] = \lambda x : [A].[M], \qquad [N\ M] = [N]\ [M], \qquad [x] = x$$

If $M : A$ in $\lambda*$ we then have $[M] : T\ [A]$

Interpretation of $\lambda*$

For instance we cannot have $\Sigma X : U.B : U$ for $B : U\ [X : U]$ because we can then define

$$B : \mathsf{U} = \Sigma X : U.\mathsf{N}_1, \qquad \epsilon\ X = (X, 0), \qquad T\ (X, 0) = X$$

with $T\ (\epsilon\ X) = X$

Context and interpretation between contexts

One concrete example was the context

$$\Gamma = B : \mathsf{U},\ \epsilon : \mathsf{U} \to B,\ T : B \to \mathsf{U},\ h : \Pi A : \mathsf{U}.\ \ A \leftrightarrow T\ (\epsilon\ A)$$

and the context

$$\Delta = axC :\ \Pi : \mathsf{U}.\ A + \neg A$$

we described an interpretation $\Delta \to \Gamma$

Context and interpretation between contexts

In general if $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ and $\Delta = y_1 : B_1, \ldots, y_m : B_m$ an interpretation

$$(a_1, \ldots, a_n) : \Delta \to \Gamma$$

will be given by n terms $a_1, \ldots, a_n$ such that

$$\Delta \vdash a_1 : A_1,\ \Delta \vdash a_2 : A_1[a_1], \ldots,\ \Delta \vdash a_n : A_n[a_1, \ldots, a_{n-1}]$$

If $\Gamma \vdash a : A$ we have $\Delta \vdash a[a_1, \ldots, a_n] : A[a_1, \ldots, a_n]$

Theory and theory interpretation

What is a model of type theory?

Models of simple type theory: Henkin (1950) for a completeness proof for higher-order logic

$A \to B$ is *not* interpreted as a set-theoretic function space

We can have models where all types are interpreted by countable sets

Trying to precise what is a model suggests new improved syntactical presentation of the formal system

First: what is a model of simply typed lambda calculus with $\to, \times$ and β, η equality

Problem with defining model of lambda-calculus

General remark before starting the notion of model of type theory

If $\Gamma \vdash M : A$ and ρ an environment we try to define $[\![M]\!]_\rho$

For defining $[\![\lambda x.M]\!]_\rho$ it is not enough to know the set theoretic function

$$a \longmapsto [\![M]\!]_{\rho, x=a}$$

For instance, for domain models, we need to know also that this function is continuous. The same problem holds for the D-set model.

What is a model of type theory?

We use the theory of *cartesian closed category*

Types $A \ ::= \ X, \ A \times A, \ A \Rightarrow A$

$(f, g) : C \to A \times B$ if $f : C \to A, \ g : C \to B$

$\mathsf{p} : A \times B \to A, \quad \mathsf{q} : A \times B \to B$

$\lambda(f) : A \to B \Rightarrow C$ if $f : A \times B \to C$

$\mathsf{app} : (A \Rightarrow B) \times A \to B$

C-monoid

If we forget the types in the equations, we get the notion of C-monoid

A monoid with special constants $\mathsf{app}, \mathsf{p}, \mathsf{q}$, operations $(x, y), \lambda(x)$ and equations

$$(xy)z = x(yz), \quad x1 = 1x = x$$
$$\mathsf{p}(x, y) = x, \quad \mathsf{q}(x, y) = y, \quad (x, y)z = (xz, yz), \quad (\mathsf{p}, \mathsf{q}) = 1$$
$$\mathsf{app}(\lambda(x), y) = x(1, y), \quad (\lambda(x))y = \lambda(x(y\mathsf{p}, \mathsf{q})), \qquad x = \lambda(\mathsf{app}(x\mathsf{p}, \mathsf{q}))$$

Category with families

Introduced by Peter Dybjer (1995)

Can be seen as a name free presentation of type theory

Inspired from previous work of J. Cartmell (1978) on *contextual categories* and *generalised algebraic theories*

What is fundamental is the *category* of contexts

We get the *same equations* as for the theory of cartesian closed categories

What is a category?

Cartmell observed that we can give an almost equational ("generalised algebraic") presentation of this notion

We have a collection of *objects* $\Gamma, \Delta, \ldots$

Given two objects Γ, Δ we have a collection of *maps* $\Gamma \to \Delta$ (dependent types!)

What is a category?

For each object Γ we have $1 \in \Gamma \to \Gamma$

We have a composition operator $\sigma\delta \in \Theta \to \Gamma$ if $\delta \in \Theta \to \Delta$ and $\sigma \in \Delta \to \Gamma$.

$$\sigma\ 1 = 1\ \sigma = \sigma, \qquad (\theta\sigma)\delta = \theta(\sigma\delta)$$

Syntactical remarks

The real *explicit* terms are first-order terms

For instance we should really write comp $A\ B\ C\ f\ g$ instead of $g\ f$

id A instead of 1

But the shorter notations are not ambiguous

Category with families

If $\Gamma \in \mathsf{Con}$ we have a collection $\mathsf{Type}(\Gamma)$ of *types* over Γ.

Substitution: we have an operation that takes $A \in \mathsf{Type}(\Gamma)$ and $\sigma \in \Delta \to \Gamma$ to $A\sigma \in \mathsf{Type}(\Delta)$ such that the following equations hold

$$A\ 1 = A, \qquad (A\sigma)\delta = A(\sigma\delta)$$

Category with families

If $\Gamma \in \mathsf{Con}$ and $A \in \mathsf{Type}(\Gamma)$ we have a collection $\mathsf{Elem}(\Gamma, A)$ of *elements* of type A. We have an operation that takes $u \in \mathsf{Elem}(\Gamma, A)$ and $\sigma \in \Delta \to \Gamma$ to $u\sigma \in \mathsf{Elem}(\Delta, A\sigma)$. Furthermore the following equations hold

$$u\ 1 = u, \qquad (u\sigma)\delta = u(\sigma\delta)$$

Category with families

We have a context extension operation: if $A \in \mathsf{Type}(\Gamma)$ then we have a new context $\Gamma.A \in \mathsf{Con}$. Furthermore there is a projection morphism $\mathsf{p} \in \Gamma.A \to \Gamma$ and a special element $\mathsf{q} \in \mathsf{Elem}(\Gamma.A, A\ \mathsf{p})$. If $\sigma \in \Delta \to \Gamma$ and $A \in \mathsf{Type}(\Gamma)$ and $u \in \mathsf{Elem}(\Delta, A\sigma)$ we have also an extension operation $(\sigma, u) \in \Delta \to \Gamma.A$. This should satisfy the equations

$$\mathsf{p}\ (\sigma, u) = \sigma, \qquad \mathsf{q}\ (\sigma, u) = u, \qquad (\sigma, u)\delta = (\sigma\delta, u\delta), \qquad (\mathsf{p}, \mathsf{q}) = 1$$

Notations

If $u \in \mathsf{Elem}(\Gamma, A)$ we write $[u] \in \Gamma \to \Gamma.A$ the substitution $(1, u)$. Thus if $B \in \mathsf{Type}(\Gamma.A)$ we have $B[u] \in \mathsf{Type}(\Gamma)$, and if $v \in \mathsf{Elem}(\Gamma.A, B)$ we have $v[u] \in \mathsf{Elem}(\Gamma, B[u])$.

If $\sigma \in \Delta \to \Gamma$ and $A \in \mathsf{Type}(\Gamma)$ we have a square diagram with $\mathsf{p} \in \Delta.A\sigma \to \Delta$ and $\mathsf{p} \in \Gamma.A \to \Gamma$ and $(\sigma\mathsf{p}, \mathsf{q}) \in \Delta.A\sigma \to \Gamma.A$.

This is a pullback in the category of contexts.

Dependent products

We suppose furthermore one operation that takes $A \in \mathsf{Type}(\Gamma)$ and $B \in \mathsf{Type}(\Gamma.A)$ to $\Pi\ A\ B \in \mathsf{Type}(\Gamma)$. If $B \in \mathsf{Type}(\Gamma.A)$ and $\sigma \in \Delta \to \Gamma$ the following equation should hold

$$(\Pi\ A\ B)\sigma = \Pi\ (A\sigma)\ (B(\sigma\mathsf{p}, \mathsf{q}))$$

If $v \in \mathsf{Elem}(\Gamma.A, B)$ we have $\lambda v \in \mathsf{Elem}(\Gamma, \Pi\ A\ B)$. We have an application operation: if $w \in \mathsf{Elem}(\Gamma, \Pi\ A\ B)$ and $u \in \mathsf{Elem}(\Gamma, A)$ then $\mathsf{app}(w, u) \in \mathsf{Elem}(\Gamma, B[u])$.

Dependent products

These operations should satisfy the equations

$$\mathsf{app}(\lambda v, u) = v[u], \qquad w = \lambda(\mathsf{app}(w\ \mathsf{p}, \mathsf{q}))$$
$$(\lambda v)\sigma = \lambda v(\sigma\ \mathsf{p}, \mathsf{q}), \qquad \mathsf{app}(w, u)\sigma = \mathsf{app}(w\sigma, u\sigma)$$

Deductive system presentation

This describes what is a model of type theory with dependent product

Alternatively this can be presented in the form of *inference rules* with equations

These are called *deductive systems* in Lambek-Scott

Deductive system presentation

The *sorts* are Con and $\Delta \to \Gamma$ for Δ, Γ : Con and Type(Γ) for Γ : Con and Elem(Γ, A) for Γ : Con and A : Type(Γ).

$$\frac{\Gamma : \mathsf{Con}}{1 : \Gamma \to \Gamma} \qquad \frac{\sigma : \Delta \to \Gamma \quad \delta : \Theta \to \Delta}{\sigma\delta : \Theta \to \Gamma}$$

Equations:

$\sigma\ 1 = 1\ \sigma = \sigma$ where $\sigma : \Delta \to \Gamma$

$(\theta\sigma)\delta = \theta(\sigma\delta)$ where $\theta : \Gamma \to \Xi$, $\sigma : \Delta \to \Gamma,\ \delta : \Theta \to \Delta$

Generalised algebraic theory

In equational logic we have a clear notion of *model* and *maps* bewteen models. We have also a very pure form of *completeness theorem* which is proved by considering the *initial model* = term model

All this is still valid with *multi-sorted* equational logic

Cartmell's notion of generalised algebraic theory can be seen as equational logic with dependent sorts, and keep also these properties

Example of models

Set theoretic model: the *contexts* are sets

Type(Γ) is the collection of *families of sets* $A = (A_\gamma)_{\gamma \in \Gamma}$.

If $\sigma : \Delta \to \Gamma$ we define $A\sigma = (A_{\sigma(\delta)})_{\delta \in \Delta}$

Elem(Γ, A) is the collection of *sections*: family (a_γ) such that a_γ belongs to A_γ for γ in Γ

Example of models

We have dependent product and we can check that all equations are satisfied

If $A = (A_\gamma)$ in Type(Γ)

and $B = (B_{(\gamma,a)})$ in Type$(\Gamma.A)$

we can form $(\Pi\ A\ B)_\gamma = (\Pi x \in A_\gamma) B_{(\gamma,x)}$

Example of models

Domain model: the *contexts* are Scott domains

Type(Γ) is the collection of *families of domains* $A = (A_\gamma)$ the domain Γ, i.e. a *continuous* functor from Γ seen as a category to the category of domains with embedding-projection pairs as morphisms

Elem(Γ, A) is the collection of *continuous sections*: family (a_γ) such that a_γ belongs to A_γ for γ in Γ, and which is continuous in γ

We have dependent product and we can check that all equations are satisfied

Operations on models

Since these models are models of an essentially algebraic theory, we can (like in equational logic)

form the *product* of two models, or of a family of models

consider maps between models and the category of all models

this category has an intial model: the term model

consider inductive limits of models

This also holds for topos theory, but *not* for set theory

Impredicative universe

An impredicative universe is a type Prop with a dependent type $T\ x\ [x : \mathsf{Prop}]$ and an universal quantification $\forall : (A \to \mathsf{Prop}) \to \mathsf{Prop}$ with two maps

$$\mathsf{in} : (\Pi x : A.T\ (f\ x)) \to T\ (\forall\ f), \quad \mathsf{out} : T\ (\forall\ f) \to (\Pi x : A.T\ (f\ x))$$

such that $\mathsf{out}\ (\mathsf{in}\ u) = u \quad : \ \Pi x : A.T\ (f\ x)$

The classical intuition is that the types $T\ x$ of an impredicative universe have to be *small* (at most one element); for instance the type

$$T\ (\forall x : \mathsf{Prop}.\ x \Rightarrow (x \Rightarrow x) \Rightarrow x)$$

We are now showing a model with non trivial types $T\ x$

The D-set model

Due to M. Hyland, E. Moggi

We start from an arbitrary *combinatory algebra*

A set D with a binary application operation $b \cdot a \in D$ for $b, a \in D$ and two elements $k, s \in D$ such that

$$k \cdot a \cdot b = a, \qquad s \cdot a \cdot b \cdot c = a \cdot b \cdot (a \cdot c)$$

Combinatory algebra

D is *functionally complete*: if we have a term $t(x_1, \ldots, x_n)$ then there exists $f \in D$ such that $f \cdot x_1 \cdot \cdots \cdot x_n = t(x_1, \ldots, x_n)$.

We have elements $\pi_1,\ \pi_2$ such that $\pi_1 \cdot a \cdot b = a,\ \pi_2 \cdot a \cdot b = b$

We define $<a, b>$ such that $<a, b> \cdot c = c \cdot a \cdot b$

$<a, b> \cdot \pi_1 = a$ and $<a, b> \cdot \pi_2 = b$.

$\pi_1 = \pi_2$ iff D is trivial and $<a, b> = <a', b'>$ iff $a = a'$ and $b = b'$

Combinatory algebra

One can encode the natural numbers in any combinatory algebras

$d_n \cdot a \cdot f = (f\cdot)^n \cdot a$

Another way $c_0 = \mathsf{I}$ such that $\mathsf{I} \cdot x = x$ and

$$S \cdot x = <x, \pi_1>, \quad c_1 = <c_0, \pi_1>, \quad c_2 = <c_1, \pi_1>, \ldots$$

Then it is possible to show that $S\cdot$ is *injective* and that

$S \cdot x \neq c_0$ if D is not trivial

Two notions of types

D should be thought of as a collection of "untyped computations"

A (small) type can be interpreted as

a subset $X \subseteq D$ (intensional interpretation of types)

a Partial Equivalence Relation on D (extension interpretation of types)

A motivation for the second model comes from Gandy's model of extensionality: for representing a functional F we require $f_1 =_{N\to N} f_2$ implies $F \cdot f_1 =_N F \cdot f_2$ where $u_1 =_N u_2$ means $u_1 = u_2 = c_n$ for some $n \in \mathbb{N}$

The category of D-sets

A D-set is a pair $X, \Vdash_X$ where X is a(n arbitrary) set and $d \Vdash_X x$ a relation between D and X such that for each x in X there exists (at least) one d such that $d \Vdash_X x$

A map $(X, \Vdash_X) \to (Y, \Vdash_Y)$ is a set-theoretic function $f : X \to Y$ such that there exists d in D such that if $a \Vdash x$ then $d \cdot a \Vdash f(x)$

So a map is a set-theoretic function $f : X \to Y$ satisfying some conditions (like in domain theory, where the condition is that of being continuous)

The category of D-sets

The category of D-sets is cartesian closed

If $A = (X, \Vdash_X)$, $B = (Y, \Vdash_Y)$ then $A \Rightarrow B$ is $(A \to B, \Vdash)$ where $d \Vdash f$ iff $a \Vdash x$ implies $d \cdot a \Vdash f(x)$

$A \times B$ is $(X \times Y, \Vdash)$ where $d \Vdash (x, y)$ iff $d = <a, b>$ with $a \Vdash x$ and $b \Vdash y$

Natural number object

We can define d_n such that $d_n \cdot f \cdot a = (f\cdot)^n a$

We define Nat $= \mathbb{N}, \Vdash$ with $d \Vdash n$ iff $d \cdot f \cdot a = (f\cdot)^n a$. In particular $d_n \Vdash n$

We have a map S : Nat $\to$ Nat since there exists s in D such that $s \cdot d \cdot f \cdot a = f \cdot (d \cdot f \cdot a)$

Given $A = X, \Vdash$ and $v : A \to A$ and $x \in X$ there is a unique map u : Nat $\to A$ such that $u\ 0 = x$ and u S $= v\ u$. We must have $u\ n = f^n\ x$, and if $f \Vdash v,\ a \Vdash x$ we have $d \cdot f \cdot a \Vdash u\ n$ if $d \Vdash n$

Family of D-sets

Given $\Gamma = (X, \Vdash)$ an element of Type(Γ) is a family $A = (Y_x, \Vdash_x)$ of D-sets over X

An element of Elem(Γ, A) is a family $b_x \in Y_x$ such that there exists d in D such that $d \cdot a \Vdash_x b_x$ whenever $a \Vdash x$

We define $\Gamma.A$ to be the D-set $\Sigma_{x \in X} Y_x, \Vdash$ where $d \Vdash (x, y)$ iff $d \cdot \pi_1 \Vdash x$ and $d \cdot \pi_2 \Vdash y$

Family of D-sets

An alternative definition, not equivalent a priori in general, is to define $\Gamma.A$ to be the D-set $\Sigma_{x \in X} Y_x, \Vdash$ where $d \Vdash (x, y)$ iff there exist d_1, d_2 such that $d = <d_1, d_2>$ and $d_1 \Vdash x$ and $d_2 \Vdash y$

The two possible D-sets are *isomorphic* in the category of D-sets, so this choice does not actually matter

Modest sets

A D-set $X, \Vdash$ is *modest* iff $d \Vdash x_1 \wedge d \Vdash x_2$ implies $x_1 = x_2$ in X

For instance Nat is modest if D is not trivial

Proposition: *If* B *in* $\mathrm{Type}(\Gamma.A)$ *is a family of modest sets then* $\Pi\ A\ B$ *in* $\mathrm{Type}(\Gamma)$ *is a family of modest sets*

Proposition: *If* B *in* $\mathrm{Type}(\Gamma.A)$ *is a family of modest sets* and $A \in \mathrm{Type}(\Gamma)$ *is a family of modest set then* $\Sigma\ A\ B$ *in* $\mathrm{Type}(\Gamma)$ *is a family of modest sets*

Large sets

If X is a set we write $\Delta(X)$ the D-set $X, \Vdash$ with $d \Vdash x$ for all d in D and x in X

Any set theoretic map $f : X \to Y$ is also a map $f : \Delta(X) \to \Delta(Y)$

Proposition: *If A is large and B is modest then any map $A \to B$ is constant*

Notice that Nat is *not* $\Delta(\mathbb{N})$

Modest sets

Proposition: *If* $A = (X, \Vdash)$ *is a modest set then the relation*

$$\exists x \in X. d_1 \Vdash x \wedge d_2 \Vdash x$$

is a symmetric and transitive relation

We let PER(D) be the set of symmetric and transitive relations on D. If R in PER(D) we write $[R]$ the set of equivalence class of R

If R is in PER(D) then $T(R)$ is the modest set $[R], \Vdash$ where $d \Vdash x$ iff d is in the equivalence class x

An impredicative universe

Prop is then the object $\Delta(\mathsf{PER}(D))$

If $f : A \to$ Prop then $T \circ f$ is a family of modest sets over A and $\Pi\ A\ (T \circ f)$ is a modest set

We can define $\forall f = [\Pi\ A\ (T \circ f)]$

If R, S are PERs we define $R \Rightarrow S = [T\ R \to T\ S]$

We have $d_1\ (R \Rightarrow S)\ d_2$ iff $a_1\ R\ a_2$ implies $d_1 \cdot a_1\ S\ d_2 \cdot a_2$

Products and intersections

Let A be the D-set $\Delta(X)$ where X is an arbitrary set

Let R_x be a family in PER(D)

Let B be the family of modest sets $T(R_x)$ for $x \in X$

Theorem: *The D-set $\Pi\ A\ B$ is* isomorphic *to the D-set $T(\bigcap_{x \in X} R_x)$*

Second application

We get the (relative) consistency of the type theory with an impredicative universe with a small type of natural numbers with large eliminations

It can be shown that such a type theory *contradicts* classical logic

Indeed we can check that the type

$$\Pi A : \mathsf{Prop}.\ \ \neg\neg A \to A$$

is empty in the D-set model

Encoding of Booleans

In impredicative type theory, one considers the type of Booleans

$$B = \Pi X : \mathsf{Prop}. X \to X \to X$$

What is this type in the D-set model?

Example of models

Truth-value model: the category of contexts is the poset $\{0, 1\}$

$\mathsf{Type}(0) = \mathsf{Type}(1) = \{0, 1\}$

$\mathsf{Elem}(\Gamma, A) = \{0\}$ if $\Gamma \leq A$

$\mathsf{Elem}(\Gamma, A) = \emptyset$ if $A = 0,\ \Gamma = 1$

We can define $\Gamma.A = \Gamma \wedge A$ and we have dependent products

In this model any inhabited type has at most one element

This model supports (intensional) identity types

First application

Proposition: *It is impossible to show*

$$\neg \mathsf{Id}\ 0\ (\mathsf{S}\ 0)$$

without using universes

Indeed the truth-value models is a model where the interpretation of this type is empty

Axiom of Choice

In impredicative type theory, one can define

$$\exists y : B.C = \Pi X : \mathsf{Prop}.(\Pi x : B.C \to X) \to X$$

but then it does not seem possible to prove AC

$$(\Pi x : A.\exists y : B.R\ x\ y) \to \exists f : A \to B.\Pi x : A.R\ x\ (f\ x)$$

A new model

We sketch a model construction, due to Stefano Berardi, Herman Geuvers and Thomas Streicher, which shows that

AC is *not* provable

there is no way to realise the context

$$B : \mathsf{Prop}, \mathsf{t} : B, \mathsf{f} : B$$

$$axI : \Pi C : B \to \mathsf{Prop}.\ C\ \mathsf{t} \to C\ \mathsf{f} \to \Pi x : B.C\ x$$

Two models

In the truth-value models AC is true and the previous context is inhabited

In any D-set model also, AC is true and the previous context is inhabited

Motivation

Given a domain D, model of lambda-calculus, so that we have an embedding of $[D \to D]$ in D it is natural to interpret a *type* as a *subset* of D

If $X, Y \subseteq D$ we define

$$X \to Y = \{d \in D \mid \forall x \in X. d \cdot x \in Y\} = \{d \in D \mid d \cdot X \subseteq Y\}$$

This will interpret *small* types

The category of contexts

A type will be a pair $X, a : X \to D$ where X is a(n arbitrary) set and a an arbitrary set-theoretic function

A context with be a tuple $n, X, a : X \to D^n$

A map $(n, X, a) \to (m, Y, b)$ is a pair (f, g) where f is a set-theoretic map $f : X \to Y$ and $g : D^n \to D^m$ is a *continuous* map such that $b\ f = g\ a$

Families and terms

Given $\Gamma = (n, X, a)$ an element A of Type(Γ) is a family $Y_x, b_x : Y_x \to D$ indexed by x in X

An element of Elem(Γ, A) is a pair $((s_x), f)$ where $s_x \in Y_x$ and f is continuous function $f : D^n \to D$ such that for all x in X we have $b_x(s_x) = f(a(x))$

We can define the substitution operation by composition

Impredicative universes

This model has two impredicative universes

$Pow(D), a : Pow(D) \to D$ where $a(X) = \bot$ we define $T(X) = X, i : X \to D$ where i is the inclusion

$P(D), a : P(D) \to D$ where $a(X) = \bot$ we define $T(X) = X, i : X \to D$ where i is the inclusion

where $P(D) = \{X \subseteq D \mid \forall a.\ a \in X \to X = D\}$

We shall use the second universe

Anaysis of this model

If we have u, v in D we define the function $C : D \to P(D)$ such that $C(e) = \{d \in D \mid e = u \vee e = v\}$

Notice that equality in D is in general undecidable but we have $C(e)$ in $P(D)$

Using C we see that if we have

$$axI : \Pi C : B \to \text{Prop.}\ \ C\ \text{t} \to C\ \text{f} \to \Pi x : B.C\ x$$

then we have $e = u \vee e = v$ for all e in D which implies that D is trivial