Introduction

What Is a Model of the Lambda Calculus?

A. Meyer, Information and Control 52, 87-122 (1982)

Elementary definition of what is a model of untyped lambda calculus

We want to do the same for type theory: a definition of model *independent* of the syntax

Equational logic

Proof theory

Model theory

Initial model

Completeness theorem

Automation: paramodulation, Knuth-Bendix

Generalised algebraic theories

What is a category?

What is a cartesian closed category?

Cartmell's notion of generalised algebraic theory "abstracted from versions of Martin-Löf type theory"

A generalisation of the usual notion of many-sorted algebraic or equational theory

Other formulations

Lambek deductive systems

Freyd essentially algebraic theories, cartesian logic

"most important and fruitful observation ... the theory of topos is essentially algebraic ... it has the extremely important consequence that the models of the theory are closed under directed limits"

M. Tierney review of *Aspect of topoi*, P. Freyd

Models of generalised algebraic theories

What is a cartesian category?

Collection of objects Ob

If A, B objects, collection of morphisms $Hom\ A\ B$

An identity $1 \in Hom\ A\ A$ and composition operator

A product operation $A \times B \in Ob$ if $A, B \in Ob$

Projection maps $\mathsf{p} \in Hom\ (A \times B)\ A,\ \mathsf{q} \in Hom\ (A \times B)\ B$

Models of generalised algebraic theories

Pairing $(x, y) \in Hom\ C\ (A \times B)$ if $x \in Hom\ C\ A$ and $y \in Hom\ C\ B$

Equations $f1 = f,\ 1f = f,\ (fg)h = f(gh)$ for categories

$\mathsf{p}(x, y) = x,\ \mathsf{q}(x, y) = y,\ (\mathsf{p}z, \mathsf{q}z) = z$ for products

Notations

We write p $\in Hom\ (A{\times}B)\ A$ but it should really be p $A\ B \in Hom\ (A{\times}B)\ A$

Similarly $1 \in Hom\ A\ A$ should be id $A \in Hom\ A\ A$

gf should be comp $A\ B\ C\ f\ g$

Like in equational logic, we have a fixed collection of *constants* with arity: id, comp, p, $\ldots$

Morphisms of models

Given two models M_1 and M_2

We should have a map $F : Ob_1 \rightarrow Ob_2$

A map $G : Hom_1\ A\ B \rightarrow Hom_2\ (F\ A)\ (F\ B)$

$F\ (A \times B) = (F\ A) \times (F\ B)$

$G\ 1 = 1, \quad G\ (gf) = (G\ g)(G\ f)$

$G\ (x, y) = (G\ x, G\ y),\ G\ \mathsf{p} = \mathsf{p},\ G\ \mathsf{q} = \mathsf{q}$

Morphisms of models

We have a notion of *initial model* (we add a constant object in order to get a non trivial model)

This is not mentionned usually in category theory (papers by Lambek are exceptions)

In category theory, there is no way to state

$F\ (A \times B) = (F\ A) \times (F\ B)$

Not so clear what is the free cartesian closed category for instance

Models of type theory

Not easy at all to check that a type theory given by the usual typing rules has a model, even the set theoretic model

We "reverse" the situation and define *directly* what should be a model

Models of type theory

A model is given by a collection Con of *contexts*.

If $\Gamma, \Delta \in$ Con we have a set $\Delta \to \Gamma$ of *substitutions* from Δ to Γ.

This should form a category: we have a substitution id $\in \Gamma \to \Gamma$ and a composition operator $\sigma\delta \in \Theta \to \Gamma$ if $\delta \in \Theta \to \Delta$ and $\sigma \in \Delta \to \Gamma$.

Models of type theory

$$\sigma\ \mathsf{id} = \mathsf{id}\ \sigma = \sigma, \qquad (\theta\sigma)\delta = \theta(\sigma\delta)$$

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

If $A \in$ Type(Γ) and $\sigma \in \Delta \to \Gamma$ we can form $A\sigma \in$ Type(Δ)

$$A\ \mathsf{id} = A, \qquad (A\sigma)\delta = A(\sigma\delta)$$

Models of type theory

If $\Gamma \in$ Con and $A \in$ Type(Γ) we have a collection Elem(Γ, A) of *elements* of type A.

If $u \in$ Elem(Γ, A) and $\sigma \in \Delta \rightarrow \Gamma$ we can form $u\sigma \in$ Elem$(\Delta, A\sigma)$.

$$u \ \mathsf{id} = u, \qquad (u\sigma)\delta = u(\sigma\delta)$$

Models of type theory

We have a context extension operation: if $A \in \mathsf{Type}(\Gamma)$ then we have a new context $\Gamma.A \in \mathsf{Con}$.

$\mathsf{p} \in \Gamma.A \to \Gamma, \mathsf{q} \in \mathsf{Elem}(\Gamma.A, A\ \mathsf{p})$

If $\sigma \in \Delta \to \Gamma$ and $A \in \mathsf{Type}(\Gamma)$ and $u \in \mathsf{Elem}(\Delta, A\sigma)$ we have $(\sigma, u) \in \Delta \to \Gamma.A$

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

Models of type theory

If $u \in \mathsf{Elem}(\Gamma, A)$ we write $[u] \in \Gamma \to \Gamma.A$ the substitution (id, 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 $v[u] \in \mathsf{Elem}(\Gamma, B[u])$.

Models of type theory with 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 $\sigma \in \Delta \rightarrow \Gamma$

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

If $v \in \mathsf{Elem}(\Gamma.A, B)$ we have $\lambda v \in \mathsf{Elem}(\Gamma, \Pi\ A\ B)$.

Models of type theory with dependent products

If $w \in \mathsf{Elem}(\Gamma, \Pi\ A\ B)$ and $u \in \mathsf{Elem}(\Gamma, A)$ we can form $\mathsf{app}(w, u) \in \mathsf{Elem}(\Gamma, B[u])$

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

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

Models of type theory with universe

To define a model of type theory with one universe, we assume that we have a special element $\mathsf{U} \in \mathsf{Type}(\Gamma)$ with the following equation

$$\mathsf{U}\sigma = \mathsf{U}$$

and such that $\mathsf{Elem}(\Gamma, \mathsf{U}) \subseteq \mathsf{Type}(\Gamma)$. We suppose also that U is closed by the product operation

$\Pi\ A\ B \in \mathsf{Elem}(\Gamma, \mathsf{U})$ if $A \in \mathsf{Elem}(\Gamma, \mathsf{U})$ and $B \in \mathsf{Elem}(\Gamma.A, \mathsf{U})$.

Models of type theory with universes

$\mathsf{U}_i \in \mathsf{Type}(\Gamma)$

$\mathsf{U}_i\sigma = \mathsf{U}_i$

$\mathsf{Elem}(\Gamma, \mathsf{U}_i) \subseteq \mathsf{Elem}(\Gamma, \mathsf{U}_{i+1}) \subseteq \mathsf{Type}(\Gamma)$

$\Pi\ A\ B \in \mathsf{Elem}(\Gamma, \mathsf{U}_i)$ if $A \in \mathsf{Elem}(\Gamma, \mathsf{U}_i)$ and $B \in \mathsf{Elem}(\Gamma.A, \mathsf{U}_i)$.

A model of type theory

Let U be a Grothendieck universe

We take $Con = U$ and Type$(\Gamma) = \Gamma \rightarrow U$ if $\Gamma \in U$

Elem$(\Gamma, F) = \Pi x \in \Gamma . F(x)$ if $\Gamma \in U,\ F \in$ Type(Γ)

$\Gamma . F = \Sigma x \in \Gamma . F(x)$

We get a model: we only have to check that all the equations hold

Another model of type theory

Since we have a generalised algebraic theory, there is an initial model, the *term* model

It is obtained by looking at the definition of the model syntactically (as inference rules)

The syntax is *a priori* very explicit

$\lambda\ \Gamma\ A\ B\ v$ instead of $\lambda\ v$

app $\Gamma\ A\ B\ w\ u$ instead of app(w, u)

Syntax

Can one justify the implicit syntax?

The solution is in

Th. Streicher *Semantics of type theory. Correctness, completeness and independence results.* 1991

One needs to prove: Π is one-to-one (forthcoming work with Andreas and Peter)

Justifies: term in β-normal form with untyped abstraction and any term with typed abstraction. We can decide if the term is correct and if it is, compute its semantics in an arbitrary model

Lambda-calculus notation

de Bruijn index

$\lambda x.\lambda y.x\ y$ instead of $\lambda\ \lambda$ $\mathsf{app}(\mathsf{q}\ \mathsf{p}, \mathsf{q})$

$\Pi A : \mathsf{U}_1.\Pi x : A.A$ instead of $\Pi\ \mathsf{U}_1\ (\Pi\ \mathsf{q}\ (\mathsf{q}\ \mathsf{p}))$

PER model

Any combinatory algebra extends to a model of type theory, where a type is interpreted as a Partial Equivalence Relation on the combinatory algebra

Three different models:

Set theoretic model

PER model

initial model

Internal model

Can one represent the set theoretical model in type theory??

What is a set? Type with equivalence relation (setoid or Bishop set)

What is a family of sets? The “right” definition seems to be (following Richman, Palmgren) a functor $X^{\#} \rightarrow Set$ where Set is the category of (Bishop) sets and $X^{\#}$ the discrete category associated to the set X

Extensional universe

A set $S, =_S$ with a family of sets $El\ A$ for $A : S$ such that S is closed under dependent products and sums

Checked in agda 2 (Ulf)

Extensional universe

```
data S : Set where
  nat   : S
  pi    : (A : S)(F : Fam A) -> S
  sigma : (A : S)(F : Fam A) -> S

data Fam (A : S) : Set where
  fam : (F : El A -> S) -> Map _==_ _=S_ F -> Fam A
```

Extensional universe

```
_=S'_ : rel S
El' : S -> Set
_=='_ : {A : S} -> rel (El A)
data _==_ {A : S}(x y : El A) : Set where
_=Fam_ : {A : S} -> rel (Fam A)
_!_ : {A : S} -> Fam A -> El A -> S
pFam : {A : S}(F : Fam A) -> Map _==_ _=S_ (_!_ F)
```

Extensional universe

```
_>>_ : {A B : S} -> Fam A -> A =S B -> Fam B
pfiFam : {A B : S}(p q : A =S B)(F : Fam A) ->
         F >> p =Fam F >> q
p<< : {A B : S}(A=B : A =S B) -> Map _==_ _==_ (_<<_ A=B)
refS : Refl _=S_
transS : Trans _=S_
symS : Sym _=S_
pfi : {A B : S}(p q : A =S B)(x : El B) -> p << x == q << x
```

Extensional universe

```
ref<< : {A : S}(x : El A) -> refS << x == x
trans<< : {A B C : S}(A=B : A =S B)(B=C : B =S C)(x : El C) ->
sym<< : {A B : S}(A=B : A =S B)(x : El B) ->
ref : {A:S} -> Refl {El A} _==_
trans : {A:S} -> Trans {El A} _==_
sym : {A:S} -> Sym {El A} _==_
```

Groupoid model

What should be the groupoid model?

Groupoids are categories, and the collection of groupoid is a 2-category

The notion of family of groupoid should be defined using this structure

Internalisation in type theory?

Use of universe to stratify the *complexity* of the objects (and not directly their size): U_1 for sets, U_2 for categories, ...