A formal set-theoretic model for the (extended) Calculus of Constructions

Jonas Kaiser

UdS, Graduate School of Computer Science

Feb 3, 2012

Outline

Goals

Step 1

Formally develop Tarski-Grothendieck set theory in Coq.

Goals

Step 1

Formally develop Tarski-Grothendieck set theory in Coq.

Step 2

Construct classical TG models for ECC where PI, PE, DN and related properties hold.

Goals

Step 1

Formally develop Tarski-Grothendieck set theory in Coq.

Step 2

Construct classical TG models for ECC where PI, PE, DN and related properties hold.

Step 3

Investigate further properties, mainly Prop $\leq$ Type_0 and later Inductive Propositions.

Background

Related work

- Set theoretic semantics since Church, 1940
- *Polymorphism is not Set-Theoretic* (Reynolds, 1984): PI-models
- large body of work by Werner, Lee & Miquel on issues of impredicativity, cumulativity and the conversion rule. (various type and set theories, mostly not formalised in Coq) [5, 3]
- Barras: Fully formalised IZF / HFDS models for CC_ω [2]

Tarski-Grothendieck set theory: ZF & GU

$$
\begin{aligned}
\forall x,\ x \notin \emptyset & \\
x \in \{a, b\} & \iff x = a \lor x = b \\
x \in \bigcup A & \iff \exists X \in A,\ x \in X \\
y \in \{F\,x \mid x \in X\} & \iff \exists z,\ z \in X \land y = F\,z \\
X \in \mathscr{P}(A) & \iff X \subseteq A \\
X = Y & \iff X \subseteq Y \land Y \subseteq X \\
(\forall X,\ (\forall x \in X,\ P\,x) & \longrightarrow P\,X) \longrightarrow \forall X,\ P\,X
\end{aligned}
$$

Tarski-Grothendieck set theory: ZF & GU

$$
\begin{aligned}
&\forall x,\ x \notin \emptyset \\
x \in \{a, b\} &\iff x = a \lor x = b \\
x \in \bigcup A &\iff \exists X \in A,\ x \in X \\
y \in \{F\,x \,|\, x \in X\} &\iff \exists z,\ z \in X \land y = F\,z \\
X \in \mathscr{P}(A) &\iff X \subseteq A \\
X = Y &\iff X \subseteq Y \land Y \subseteq X \\
(\forall X,\ (\forall x \in X,\ P\,x) &\longrightarrow P\,X) \longrightarrow \forall X,\ P\,X
\end{aligned}
$$

Grothendieck Universes

- a transitive set ($X \in G,\ x \in X \implies x \in G$)
- closed under above operators (e.g. $x \in G \implies \mathscr{P}(x) \in G$)
- for every set X there is a least universe G_X such that $X \in G_X$
- implies infinity

Luo's Extended Calculus of Constructions [4]

Term structure

- the kinds Prop and Type_j, $j \in \omega$ are terms
- variables $(x, y, \ldots)$ are terms
- let M, N, A and B be terms, then

$$\begin{aligned} &\Pi x : A,\ B \,|\, \lambda x : A.\, N \,|\, M\, N \,| \\ &\Sigma x : A,\ B \,|\, \mathbf{pair}_{\Sigma x:A,\, B}(M, N) \,|\, \pi_1(M) \,|\, \pi_2(M) \end{aligned}$$

  are terms

Properties

- strongly normalizing
- kinds are cumulative:

$$\begin{aligned} \text{Prop} &\preceq \text{Type}_0 \\ \text{Type}_n &\preceq \text{Type}_{n+1} \end{aligned}$$

Model Construction

We need a *useful* set theory

Singleton sets:

$$\{x\} ::= \{x, x\}$$

$$y \in \{x\} \iff y = x$$

We need a *useful* set theory

Singleton sets:

$$\{x\} ::= \{x, x\}$$
$$y \in \{x\} \iff y = x$$

Ordered Pairs a lá Kuratowski

$$(x, y) ::= \{\{x\}, \{x, y\}\}$$
$$(a, b) = (c, d) \iff a = c \wedge b = d$$

We need a *useful* set theory

Singleton sets:

$$\{x\} ::= \{x, x\}$$
$$y \in \{x\} \iff y = x$$

Ordered Pairs a lá Kuratowski

$$(x, y) ::= \{\{x\}, \{x, y\}\}$$
$$(a, b) = (c, d) \iff a = c \wedge b = d$$

The 'Axiom' of Separation

$$y \in \{x \in X \,|\, Px\} \iff y \in X \wedge Py$$

First steps towards the model

For starters, this should give us PI, PE and DN:

$$[\![\mathrm{Prop}]\!] ::= 2 = \{\emptyset, \{\emptyset\}\}$$
$$[\![\mathrm{Type}_0]\!] ::=^? \mathrm{G}_2 (= \mathrm{G}_\emptyset)$$

First steps towards the model

For starters, this should give us PI, PE and DN:

$$[\![\mathrm{Prop}]\!] ::= 2 = \{\emptyset, \{\emptyset\}\}$$
$$[\![\mathrm{Type}_0]\!] ::=^{?} \mathrm{G}_2 (= \mathrm{G}_\emptyset)$$

Prop should be closed under function spaces:

$$0 : \mathrm{Prop},\ 1 : \mathrm{Prop}$$
$$\Rightarrow^{?} \begin{cases} 0 \longrightarrow 0 : \mathrm{Prop}, & 0 \longrightarrow 1 : \mathrm{Prop}, \\ 1 \longrightarrow 0 : \mathrm{Prop}, & 1 \longrightarrow 1 : \mathrm{Prop} \end{cases}$$

First steps towards the model

For starters, this should give us PI, PE and DN:

$$
\begin{aligned}
[\![\mathrm{Prop}]\!] &::= 2 = \{\emptyset, \{\emptyset\}\} \\
[\![\mathrm{Type}_0]\!] &::=^? \mathrm{G}_2 (= \mathrm{G}_\emptyset)
\end{aligned}
$$

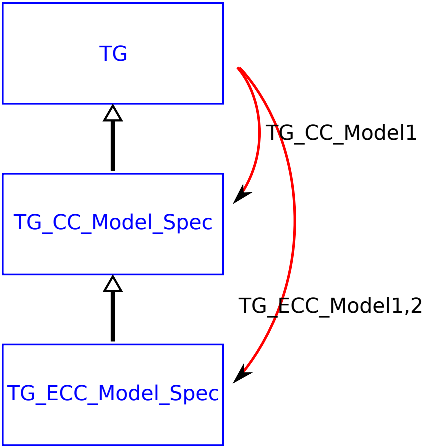
Prop should be closed under function spaces:

$$
\begin{aligned}
& 0 : \mathrm{Prop},\ \ 1 : \mathrm{Prop} \\
\Rightarrow^? & \left\{ \begin{array}{ll} 0 \longrightarrow 0 : \mathrm{Prop}, & 0 \longrightarrow 1 : \mathrm{Prop}, \\ 1 \longrightarrow 0 : \mathrm{Prop}, & 1 \longrightarrow 1 : \mathrm{Prop} \end{array} \right.
\end{aligned}
$$

For this we need Aczel's non-standard encoding of functions [1]:

$$
\begin{aligned}
[\![\mathtt{ap}\ f\ x]\!] &::= \{y \mid (x, y) \in f\} \\
[\![\mathtt{lam}\ X\ F]\!] &::= \{(x, y) \mid x \in X \wedge y \in F\ x\} \\
[\![\mathtt{Pi}\ X\ Y]\!] &::= \{[\![\mathtt{lam}\ X\ F]\!] \mid \forall x \in X,\ F\ x \in Y\ x\}
\end{aligned}
$$

Overall Framework (part 1)

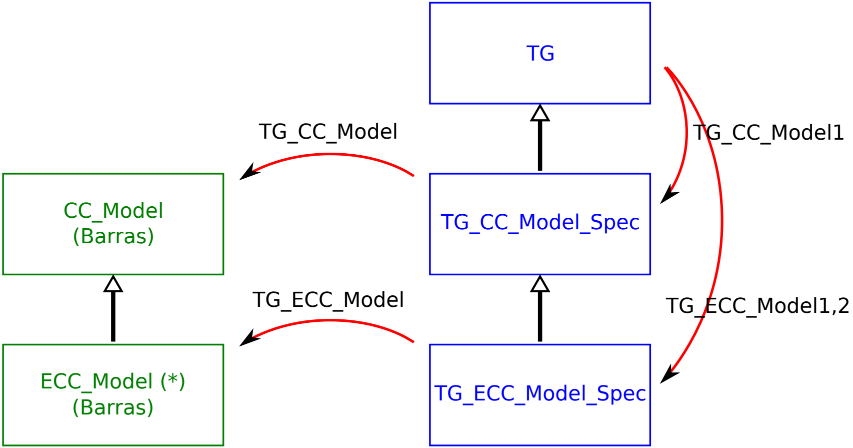


Relating to Barras' work

Overview of Barras' work [2]

- IZF / HFDS models for CC / CC_ω (called ECC in his Coq devl.)
- provides model specifications for CC & CC_ω
- changes from ECC to CC_ω:
  - no Σ-Types
  - no $\mathrm{Prop} \leq \mathrm{Type}_0$ (dropped to allow for flexible interpretations)
- claims some form of soundness result for models satisfying his specifications (requires judgmental equality in place of conversion rule)

Overall Framework with Barras (part 2)



Research Questions

Research Questions

- Can we extend the model to support inductive Propositions, or, respectively, why was this omitted from previous developments? (`Inductive True : Prop := I : True.`)
- Having Prop $\leq$ Type_0 and conversion seems to be problematic in some regards (e.g. PI). Why, and can we get around it?
- Does our interpretation of Type_0 contain infinite types?
- Our models should satisfy a large batch of axioms. Is it possible to simultaneously satisfy *all* the axioms in the Coq Standard Library, i.e. is the Library mutually consistent?

Q & A

Thank you

References

Peter Aczel.
On Relating Type Theories and Set Theories.
In *TYPES*, pages 1–18, 1998.

Bruno Barras.
Sets in Coq, Coq in Sets.
Formalized Reasoning, 3(1), 2010.

Gyesik Lee and Benjamin Werner.
Proof-Irrelevant Model of CC with Predicative Induction and Judgmental Equality.
Logical Methods in Computer Science, 7(4), 2011.

Zhaohui Luo.
ECC, an Extended Calculus of Constructions.
In *Logic in Computer Science (LICS)*, pages 386–395, 1989.

Alexandre Miquel and Benjamin Werner.
The Not So Simple Proof-Irrelevant Model of CC.
In *TYPES*, pages 240–258, 2002.

Backup

Infinite Type in Type_0, e.g. `nat`

$$\exists X : \text{Type}_0,\ \exists f : X \longrightarrow X, (\exists x : X,\ \forall y : X,\ fy \neq x) \wedge$$
$$(\forall y\, z : X,\ fy = fz \longrightarrow y = z)$$

What's wrong with the standard function encoding?

- The function space $1 \longrightarrow 1$ contains exactly one element, the function mapping $\emptyset$ to $\emptyset$.
- in the standard graph-encoding: $\{(\emptyset, \emptyset)\}$
- however, we want $[\![1 \longrightarrow 1]\!] = 1 = \{\emptyset\}$
- but $\emptyset \neq \{(\emptyset, \emptyset)\}$!
- with the alternative function encoding, the two sides match up.