Formal Construction of Set-Theoretic Models for an Extended Calculus of Constructions

joint work with Chad E. Brown

Jonas Kaiser

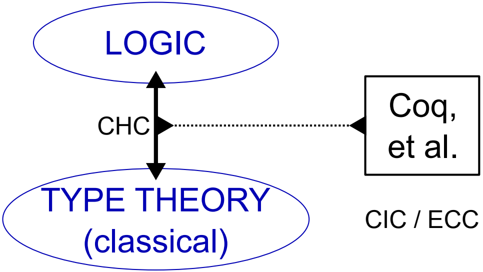
UdS, Graduate School of Computer Science

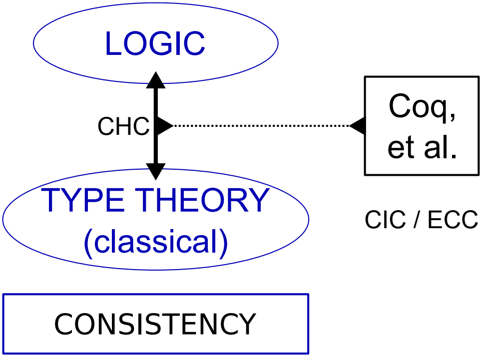
May 11, 2012

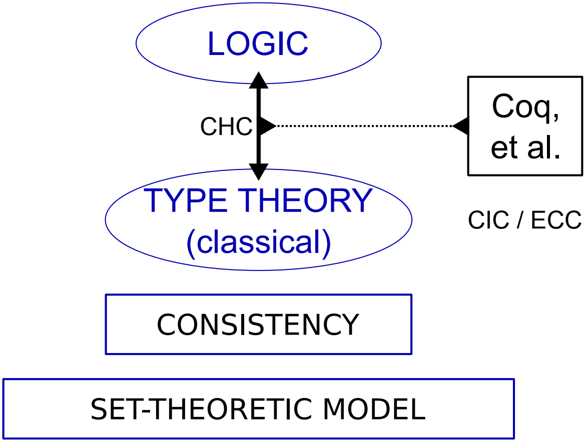
Outline

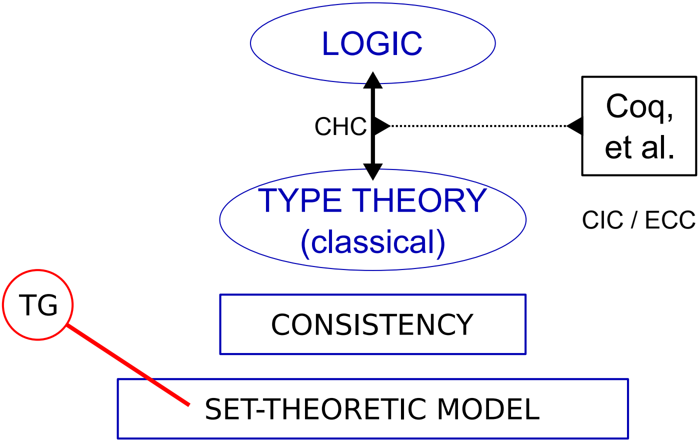
LOGIC

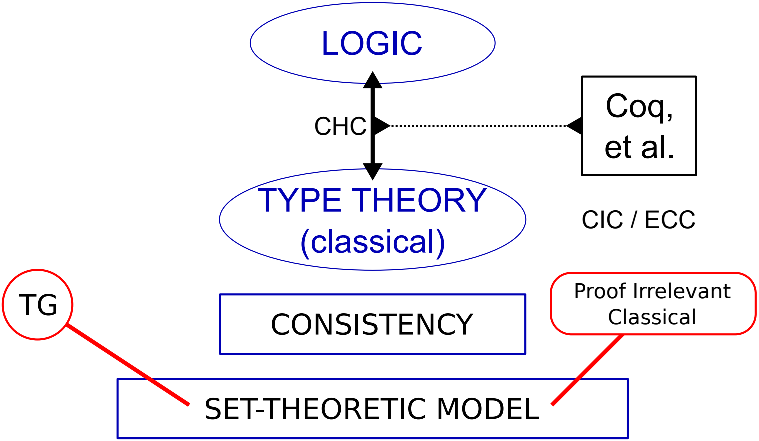
TYPE THEORY
(classical)

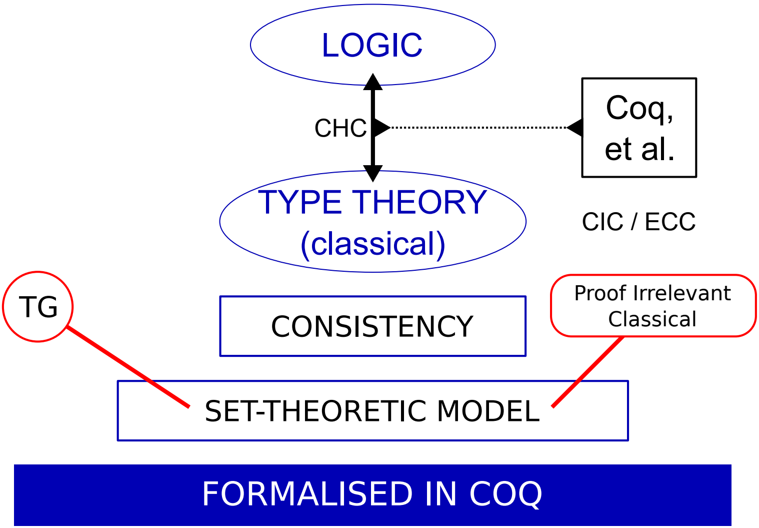

LOGIC
CHC
Coq,
et al.
TYPE THEORY
(classical)
CIC / ECC


LOGIC
CHC
Coq,
et al.
TYPE THEORY
(classical)
CIC / ECC
CONSISTENCY


LOGIC
CHC
Coq,
et al.
TYPE THEORY
(classical)
CIC / ECC
CONSISTENCY
SET-THEORETIC MODEL


LOGIC
CHC
Coq,
et al.
TYPE THEORY
(classical)
CIC / ECC
TG
CONSISTENCY
SET-THEORETIC MODEL


LOGIC
CHC
Coq,
et al.
TYPE THEORY
(classical)
CIC / ECC
TG
CONSISTENCY
Proof Irrelevant
Classical
SET-THEORETIC MODEL


LOGIC
CHC
Coq,
et al.
TYPE THEORY
(classical)
CIC / ECC
TG
CONSISTENCY
Proof Irrelevant
Classical
SET-THEORETIC MODEL
FORMALISED IN COQ

Luo's Extended Calculus of Constructions [4]

Term structure

- The kinds Prop and $\text{Type}_0, \text{Type}_1, \text{Type}_2, \ldots$ are terms
- Variables $(x, y, \ldots)$ are terms
- Let M, N, A and B be terms, then

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

are terms

Luo's Extended Calculus of Constructions [4]

Term structure

- The kinds Prop and $\text{Type}_0, \text{Type}_1, \text{Type}_2, \ldots$ are terms
- Variables $(x, y, \ldots)$ are terms
- Let M, N, A and B be terms, then

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

  are terms

Properties

- Strongly normalising
- No strong sums in Prop (would lead to inconsistency)
- Kinds are (fully) cumulative:

$$\text{Prop} \preceq \text{Type}_0$$
$$\text{Type}_n \preceq \text{Type}_{n+1}$$

Tarski-Grothendieck set theory: ZFC & GU

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

Choice: should follow from ε in meta theory

Tarski-Grothendieck set theory: ZFC & GU

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

Choice: should follow from ε in meta theory

Grothendieck Universes

- Transitive set ($X \in U,\ x \in X \implies x \in U$)
- Closed under above operators (e.g. $X \in U \implies \mathscr{P}(X) \in U$)
- For every X there is a *least* universe $U := \mathrm{G}_X$ such that $X \in \mathrm{G}_X$
- Implies infinity ($\mathrm{G}_\emptyset$ is inf.)

Model Construction

Abstract Model

- Some generic set constructions:
  singletons, indexed unions, separation, ordered pairs

Concrete Model

Model Construction

Abstract Model

- Some generic set constructions:
  singletons, indexed unions, separation, ordered pairs
- Specific set constructions to reflect parts of ECC:

$$[\![\mathrm{Prop}]\!] := 2$$
$$[\![\mathrm{Type}_0]\!] := \mathrm{G}_\emptyset$$

Concrete Model

Model Construction

Abstract Model

- Some generic set constructions:
  singletons, indexed unions, separation, ordered pairs
- Specific set constructions to reflect parts of ECC:

$$[\![\mathrm{Prop}]\!] := 2$$
$$[\![\mathrm{Type}_0]\!] := G_\emptyset$$

- Inhabitance results to reflect validity of typing rules

Concrete Model

Model Construction

Abstract Model

- Some generic set constructions:
  singletons, indexed unions, separation, ordered pairs
- Specific set constructions to reflect parts of ECC:

$$[\![\mathrm{Prop}]\!] := 2$$
$$[\![\mathrm{Type}_0]\!] := \mathrm{G}_\emptyset$$

- Inhabitance results to reflect validity of typing rules

Concrete Model

- Formalise syntax, environments, typing rules

Model Construction

Abstract Model

- Some generic set constructions:
  singletons, indexed unions, separation, ordered pairs
- Specific set constructions to reflect parts of ECC:

$$[\![\mathrm{Prop}]\!] := 2$$
$$[\![\mathrm{Type}_0]\!] := G_\emptyset$$

- Inhabitance results to reflect validity of typing rules

Concrete Model

- Formalise syntax, environments, typing rules
- PTS-style or JE conversion formulation

Model Construction

Abstract Model

- Some generic set constructions:
  singletons, indexed unions, separation, ordered pairs
- Specific set constructions to reflect parts of ECC:

$$[\![\mathrm{Prop}]\!] := 2$$
$$[\![\mathrm{Type}_0]\!] := \mathrm{G}_\emptyset$$

- Inhabitance results to reflect validity of typing rules

Concrete Model

- Formalise syntax, environments, typing rules
- PTS-style or JE conversion formulation
- State soundness …

Model Construction

Abstract Model

- Some generic set constructions:
  singletons, indexed unions, separation, ordered pairs
- Specific set constructions to reflect parts of ECC:

$$[\![\mathrm{Prop}]\!] := 2$$
$$[\![\mathrm{Type}_0]\!] := \mathrm{G}_\emptyset$$

- Inhabitance results to reflect validity of typing rules

Concrete Model

- Formalise syntax, environments, typing rules
- PTS-style or JE conversion formulation
- State soundness …
- …and prove it?

Soundness

- If $\Gamma \vdash M : A$ is derivable, then it is valid in the model:

$$\forall \gamma \in [\![\Gamma]\!],\ [\![M]\!]_\gamma \in [\![A]\!]_\gamma$$

Soundness

- If $\Gamma \vdash M : A$ is derivable, then it is valid in the model:

$$\forall \gamma \in [\![\Gamma]\!],\ [\![M]\!]_\gamma \in [\![A]\!]_\gamma$$

- Soundness follows, if all typing rules preserve validity.

Soundness

- If $\Gamma \vdash M : A$ is derivable, then it is valid in the model:

$$\forall \gamma \in [\![\Gamma]\!],\ [\![M]\!]_\gamma \in [\![A]\!]_\gamma$$

- Soundness follows, if all typing rules preserve validity.
- Consider the Application rule:

$$\frac{\Gamma \vdash M : \Pi x : A,\ B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B\,[x := N]}$$

Soundness

- If $\Gamma \vdash M : A$ is derivable, then it is valid in the model:

$$\forall \gamma \in [\![\Gamma]\!],\ [\![M]\!]_\gamma \in [\![A]\!]_\gamma$$

- Soundness follows, if all typing rules preserve validity.
- Consider the Application rule:

$$\frac{\Gamma \vdash M : \Pi x : A,\ B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B\,[x := N]}$$

- We assume $\gamma \in [\![\Gamma]\!]$, $[\![M]\!]_\gamma \in [\![\Pi x : A,\ B]\!]_\gamma$ and $[\![N]\!]_\gamma \in [\![A]\!]_\gamma$

Soundness

- If $\Gamma \vdash M : A$ is derivable, then it is valid in the model:

$$\forall \gamma \in [\![\Gamma]\!],\ [\![M]\!]_\gamma \in [\![A]\!]_\gamma$$

- Soundness follows, if all typing rules preserve validity.
- Consider the Application rule:

$$\frac{\Gamma \vdash M : \Pi x : A,\ B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B\,[x := N]}$$

- We assume $\gamma \in [\![\Gamma]\!]$, $[\![M]\!]_\gamma \in [\![\Pi x : A,\ B]\!]_\gamma$ and $[\![N]\!]_\gamma \in [\![A]\!]_\gamma$
- We have to show $[\![M\ N]\!]_\gamma \in [\![B\,[x := N]]\!]_\gamma$

Soundness, JE vs. PTS

$$(\textsc{conv})\ \frac{\Gamma \vdash_{\mathrm{JE}} M : A \qquad \Gamma \vdash_{\mathrm{JE}} A = B : \mathrm{Type}_i}{\Gamma \vdash_{\mathrm{JE}} M : B}$$

$$(\textsc{conv})\ \frac{\Gamma \vdash_{\mathrm{PTS}} M : A \qquad \Gamma \vdash_{\mathrm{PTS}} B : \mathrm{Type}_i}{\Gamma \vdash_{\mathrm{PTS}} M : B}\ A \approx B$$

The JE case?

The PTS case?

Are they equivalent?

Soundness, JE vs. PTS

$$\text{(CONV)}\ \frac{\Gamma \vdash_{\mathrm{JE}} M : A \qquad \Gamma \vdash_{\mathrm{JE}} A = B : \mathrm{Type}_i}{\Gamma \vdash_{\mathrm{JE}} M : B} \qquad\qquad \text{(CONV)}\ \frac{\Gamma \vdash_{\mathrm{PTS}} M : A \qquad \Gamma \vdash_{\mathrm{PTS}} B : \mathrm{Type}_i}{\Gamma \vdash_{\mathrm{PTS}} M : B}\ A \approx B$$

The JE case?

Solved (e.g. Barras [2], Lee & Werner [3]).

The PTS case?

Are they equivalent?

Soundness, JE vs. PTS

$$(\textsc{conv})\ \frac{\Gamma \vdash_{\text{JE}} M : A \qquad \Gamma \vdash_{\text{JE}} A = B : \text{Type}_i}{\Gamma \vdash_{\text{JE}} M : B}$$

$$(\textsc{conv})\ \frac{\Gamma \vdash_{\text{PTS}} M : A \qquad \Gamma \vdash_{\text{PTS}} B : \text{Type}_i}{\Gamma \vdash_{\text{PTS}} M : B}\ A \approx B$$

The JE case?

Solved (e.g. Barras [2], Lee & Werner [3]).

The PTS case?

Unsolved, circularity problem (Miquel & Werner [5])

Are they equivalent?

Soundness, JE vs. PTS

$$(\textsc{conv})\ \frac{\Gamma \vdash_{\mathrm{JE}} M : A \qquad \Gamma \vdash_{\mathrm{JE}} A = B : \mathrm{Type}_i}{\Gamma \vdash_{\mathrm{JE}} M : B} \qquad\qquad (\textsc{conv})\ \frac{\Gamma \vdash_{\mathrm{PTS}} M : A \qquad \Gamma \vdash_{\mathrm{PTS}} B : \mathrm{Type}_i}{\Gamma \vdash_{\mathrm{PTS}} M : B}\ A \approx B$$

The JE case?

Solved (e.g. Barras [2], Lee & Werner [3]).

The PTS case?

Unsolved, circularity problem (Miquel & Werner [5])

Are they equivalent?

Unknown in the general case, approximations exist.

Results so Far . . .

- Adams [1]: equivalent, given uniqueness of types

Results so Far …

- Adams [1]: equivalent, given uniqueness of types
- Miquel & Werner [5]: circumvent problems in PTS case with syntactic annotations to ensure well-sortedness

Results so Far . . .

- Adams [1]: equivalent, given uniqueness of types
- Miquel & Werner [5]: circumvent problems in PTS case with syntactic annotations to ensure well-sortedness
- Pagano, Coquand et al. (02/2012): equivalent, when dropping impredicativity (norm. by eval.)

Model Properties of Interest

Consistency

- A statement is consistent when we can exhibit a satisfying model
- We construct a proof-irrelevant, classical model
- PI, DN, PE, FE, XM, … should be satisfied in the model

Model Properties of Interest

Consistency

- A statement is consistent when we can exhibit a satisfying model
- We construct a proof-irrelevant, classical model
- PI, DN, PE, FE, XM, … should be satisfied in the model

Independence

- A statement is independent when it is consistent but not provable
- To refute provability, exhibit a non-satisfying model
- E.g. existence of an infinite type in Type_0, formally

$$\exists X : \mathrm{Type}_0,\ \exists f : X \to X, (\exists x : X,\ \forall y : X,\ f y \neq x) \wedge (\forall y\, z : X,\ f y = f z \to y = z)$$

Mutual Consistency of Standard Library

Mutual Consistency

- $(\Gamma,\ A\ \textbf{consistent}) \land (\Gamma,\ B\ \textbf{consistent}) \nRightarrow (\Gamma,\ A,\ B\ \textbf{consistent})$
- XM and ¬PI are both *separately* consistent with CiC …
- … but {XM, ¬PI} is inconsistent with CiC

Mutual Consistency of Standard Library

Mutual Consistency

- $(\Gamma,\ A\ \textbf{consistent}) \wedge (\Gamma,\ B\ \textbf{consistent}) \not\Rightarrow (\Gamma,\ A,\ B\ \textbf{consistent})$
- XM and ¬PI are both *separately* consistent with CiC …
- … but {XM, ¬PI} is inconsistent with CiC

ECC representation of CiC axioms

- Consider $P : \mathrm{Prop}_{\mathrm{CiC}}$, find suitable $Q : \mathrm{Prop}_{\mathrm{ECC}}$
- such that $\vdash_{\mathrm{CiC}} Q \leftrightarrow P$ (write $Q \overset{\mathrm{ECC}}{\multimapdotboth} P$)

Mutual Consistency of Standard Library

Mutual Consistency

- $(\Gamma,\ A\ \textbf{consistent}) \wedge (\Gamma,\ B\ \textbf{consistent}) \not\Rightarrow (\Gamma,\ A,\ B\ \textbf{consistent})$
- XM and ¬PI are both *separately* consistent with CiC …
- … but {XM, ¬PI} is inconsistent with CiC

ECC representation of CiC axioms

- Consider $P : \mathrm{Prop}_{\mathrm{CiC}}$, find suitable $Q : \mathrm{Prop}_{\mathrm{ECC}}$
- such that $\vdash_{\mathrm{CiC}} Q \leftrightarrow P$ (write $Q \overset{\mathrm{ECC}}{\circ\!\!-\!\!\bullet} P$)

Example: Proof Irrelevance

- In Coq / CiC: `forall (P:Prop) (p1 p2:P), p1 = p2.`
- In ECC: $\Pi P : \mathrm{Prop},\ \Pi u : P,\ \Pi v : P,\ u =_P v$
- Where $u =_P v := \Pi R : P \to \mathrm{Prop},\ R\ u \to R\ v$
- The abstract version of $u =_P v$ an Coq's = provably coincide.

Mutual Consistency of Standard Library

Axioms in the Library

Mutual Consistency of Standard Library

Axioms in the Library

Classical classical, functional_extensionality, Extensionality_Ensembles

Mutual Consistency of Standard Library

Axioms in the Library

Classical classical, functional_extensionality, Extensionality_Ensembles

Choice epsilon_statement, constructive_(in)definite_description, dependent_unique_choice, relational_choice

Mutual Consistency of Standard Library

Axioms in the Library

Classical classical, functional_extensionality, Extensionality_Ensembles

Choice epsilon_statement, constructive_(in)definite_description, dependent_unique_choice, relational_choice

PI proof_irrelevance, eq_rect_eq, JMeq_eq

Mutual Consistency of Standard Library

Axioms in the Library

- **Classical**: classical, functional_extensionality, Extensionality_Ensembles
- **Choice**: epsilon_statement, constructive_(in)definite_description, dependent_unique_choice, relational_choice
- **PI**: proof_irrelevance, eq_rect_eq, JMeq_eq
- **Reals**: archimed, completeness, Rplus_assoc, …

Thesis Aims

For my thesis I want to . . .

- complete two abstract models
- formalise ECC concretely with JE conversion
- proof soundness for this scenario
- and show mutual consistency of the standard library

Given spare time . . .

- I'd like to investigate the PTS vs. JE problem

References

Robin Adams.
Pure type systems with judgemental equality.
J. of Functional Programming, 16(2):219–246, March 2006.

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.

Appendix

Consistency

- Certain types are not inhabited
- A **provable** $::= \exists \mathscr{D}, \vdash \mathscr{D} : A$
- Γ **consistent** $::= \neg\exists \mathscr{D}, \Gamma \vdash \mathscr{D} : \bot$
- We take $\bot : \mathrm{Prop} := \forall P : \mathrm{Prop},\ P$
- ECC is consistent (Proof via Strong Normalisation)
- XM, PE, FE, PI are consistent additions to ECC & CiC
- Set-theoretic Models: $[\![A]\!] \neq \emptyset$

Mutual Consistency & Independence

Independence

- A **consistent** $[\nvdash \neg A]$, not (A **provable**) $[\nvdash A]$
- To refute provability: provide a model where $[\![A]\!] = \emptyset$
- XM is independent from ECC
- (Type_0 contains inf. types, like $\mathbb{N}$) is independent from ECC

Mutual Consistency

- $(\Gamma,\ A$ **consistent**$) \wedge (\Gamma,\ B$ **consistent**$) \nRightarrow (\Gamma,\ A,\ B$ **consistent**$)$
- XM and ¬PI are both *separately* consistent with CiC ...
- ... but {XM, ¬PI} is inconsistent with CiC

Constructions in TG & Meta Theory

- Singleton Sets: $\{x\}$
- 1: $\{\emptyset\} = \mathscr{P}(\emptyset)$
- 2: $\{\emptyset, 1\} = \mathscr{P}(1)$
- Indexed Union: $\bigcup_{i \in I} X_i$
- Separation: $\{x \in X \mid Px\}$
- Ordered Pairs (Kuratowski): $(a, b)_{\mathrm{k}}$
- Cartesian Product: $A \times B$

Related Lemmas

Introduction and elimination rules, correctness statements and useful equalities with respect to the special sets 0, 1 and 2.

Meta Theory

We use classical CiC with extensionality principles and Hilbert's ε.

The ECC Model

Kinds:

$$
\begin{aligned}
[\![\mathrm{Prop}]\!] &:= 2 \\
[\![\mathrm{Type}_0]\!] &:= \mathrm{G}_\emptyset
\end{aligned}
$$

Functions (using Aczel's encoding):

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

Strong Sums & Pairs:

- $\mathtt{Sig}\ d\ Y := \mathtt{lam}\ d\ Y$
- Pairs: $(a, b) := \{\{a\}, \{a, b\}\}$

Preliminary Results

True and False

- Both are in $[\![\text{Prop}]\!]$
- $[\![\text{FALSE}]\!] = 0$
- $[\![\text{TRUE}]\!] = 1$

Leibniz Equality

- defined in object logic, one for each type level
- coincides with Coq's equality on our meta type `set`
- asserts that domains are ok

Proof Irrelevance

- We have shown $[\![\text{PI}]\!] = 1$
- i.e. inhabited ...
- ... and in $[\![\text{Prop}]\!]$

Barras [2]

- Defines signatures for CC and CC_ω models
- Works in an intuitionistic setting
- Models are proof-irrelevant
- Implements his signatures using HF and IZF
- Proves soundness of his signatures when using JE
- Obtains soundness for CC with Conversion via Adams
- Won't work for CC_ω since we lack *uniqueness of types*
- Fully formalised in Coq
- Models for our signatures also satisfy his signatures.

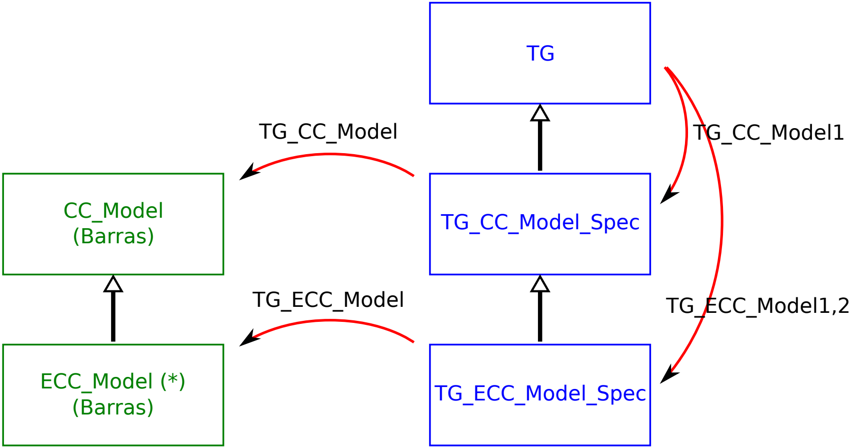
Werner, Lee & Miquel [5, 3]

- Initially compared proof theoretic strength of CiC and ZFC
- Models are proof-irrelevant
- Later mostly focused on the soundness problem.
- 'Solved' by syntactically annotating variables with sorts and dropping $\mathrm{Prop} \leq \mathrm{Type}_0$
- Partially formalised in Coq
- They aim for CiC but exclude Inductive Propositions

Remarks on the Meta theory

- We work in Coq: CiC
- Add `ClassicalFacts`: relates XM, PD, PE, PI, …
- Add `Classical_Prop`: XM, DN, Peirce, PI, …
- Add `FunctionalExtensionality`: FE, …
- Add `Epsilon`: Hilbert's ε, Church's ι
- CDP: $\forall P : \mathrm{Prop},\ P + (\neg P)$ follows from DN and ε.
- $\forall P : \mathrm{Type},\ P + (P \to \bot)$ follows from CDP and ε.

Barras' Framework



Infinite Types in Type_0

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

f is a function on X which is *injective* but *not surjective*.
This implies that X is infinite.

- Not satisfied in the $[\![\mathrm{Type}_0]\!] := \mathrm{G}_\emptyset$ model;
  (any injective f on X is also surjective – classical)
- Satisfied in the $[\![\mathrm{Type}_0]\!] := \mathrm{G}_{\mathrm{G}_\emptyset}$ model;
  Use $X := \mathrm{G}_\emptyset$ and $f := \mathscr{P}$

Encoding Functions

What's wrong with the standard graph encoding of functions?

- The function space $\top \to \top$ contains exactly one element, the function mapping $\emptyset$ to $\emptyset$.
- with standard graph-encoding: $[\![\top \to \top]\!] = \{\{(\emptyset, \emptyset)\}\} \notin 2$
- however, we want $[\![\top \to \top]\!] = 1 \in 2$
- but $\emptyset \neq \{(\emptyset, \emptyset)\}$!
- with the alternative function encoding, the two sides match up.