Advanced Type Systems, Lecture I

The Boolean Model of Type Theory

Alexandre Miquel (miquel@cs.chalmers.se)

SUMMER 02: Proofs as Programs – Eugene, Oregon

Introduction

Motivation: Relating type theory with set theory by translating the former into the latter:

$$\text{Type Theory [1970's]} \quad \longrightarrow \quad \text{ZFC [1913]}$$

- We will focus on the Calculus of Constructions with universes ($CC\omega$) [Luo 1984], i.e. Coq's formalism (CIC) without any inductive datatypes
- We will build a model of $CC\omega$ into ZFC (+ some extra axiom) such that:
  - propositions are interpreted as booleans (classical interpretation)
  - typing relation is interpreted as membership (validity, or soundness)
- Thanks to the soundness property, we will deduce:
  - the logical consistency of $CC\omega$ (without assuming the SN property)
  - the consistency of some usual axioms (excluded middle, axiom of choice, etc.)

What is CCω ?

- The Calculus of Constructions with universes is:

  The Calculus of Constructions, based on sorts Prop, Type

  \+ infinitely many predicative universes Type_i, $i \geq 1$ (with $\mathsf{Type}_1 = \mathsf{Type}$)

  \+ cumulativity rules to enforce $\mathsf{Prop} \subset \mathsf{Type}_1$, $\mathsf{Type}_i \subset \mathsf{Type}_{i+1}$

  (but no inductive datatypes)

- The intended use of CCω is slightly different from that of CC:
  - Prop is reseved for propositions
  - datatypes are put in the Type_i's (universes $\Rightarrow$ predicative polymorphism)

  [in CC, Prop is used both for datatypes and propositions]

- This set-theoretic model can also interpret inductive datatypes, fixpoints, cases. . .
  . . . but not the impredicative sort Set of Coq (non-conservative extension)

A formal presentation of CCω

Sorts $\quad s \;::=\; \mathsf{Prop} \mid \mathsf{Type}_i \quad (i \geq 1)$

Terms $\quad M, N, T, U \;::=\; x \mid s \mid \Pi x : T \,.\, U \mid \lambda x : T \,.\, M \mid M N$

Contexts $\quad \Gamma, \Delta \;::=\; [] \mid \Gamma; [x : T]$

Judgments

$\Gamma \vdash$ — 'Γ is a well-formed context'

$\Gamma \vdash M : T$ — 'under Γ, the term M has type T'

Reduction $\quad (\lambda x : T \,.\, M) N \;\to_\beta\; M\{x := N\}$

Notations $\quad FV(M)$ (free variables), $\quad DV(\Gamma)$ (declared variables),

$M\{x := N\}$ (external substitution), $\quad M_1 =_\beta M_2$ (β-conversion)

Typing rules (1/2)

(Context)

$$\frac{}{[\,] \vdash} \qquad \frac{\Gamma \vdash T : s}{\Gamma; [x : T] \vdash} \quad {\scriptstyle x \notin DV(\Gamma)}$$

(Var)

$$\frac{\Gamma \vdash}{\Gamma \vdash x : T} \quad {\scriptstyle (x:T) \in \Gamma}$$

(Sort)

$$\frac{\Gamma \vdash}{\Gamma \vdash \mathsf{Prop} : \mathsf{Type}_1} \qquad \frac{\Gamma \vdash}{\Gamma \vdash \mathsf{Type}_i : \mathsf{Type}_{i+1}}$$

(Lam)

$$\frac{\Gamma; [x : T] \vdash M : U \qquad \Gamma \vdash \Pi x : T \,.\, U : s}{\Gamma \vdash \lambda x : T \,.\, M : \Pi x : T \,.\, U}$$

(App)

$$\frac{\Gamma \vdash M : \Pi x : T \,.\, U \qquad \Gamma \vdash N : T}{\Gamma \vdash M N : U\{x := N\}}$$

Typing rules (2/2)

$\Pi(\mathsf{Prop}, \mathsf{Prop})$ $$\frac{\Gamma \vdash T : \mathsf{Prop} \qquad \Gamma; [x : T] \vdash U : \mathsf{Prop}}{\Gamma \vdash \Pi x : T \,.\, U : \mathsf{Prop}}$$

$\Pi(\mathsf{Type}_i, \mathsf{Type}_i)$ $$\frac{\Gamma \vdash T : \mathsf{Type}_i \qquad \Gamma; [x : T] \vdash U : \mathsf{Type}_i}{\Gamma \vdash \Pi x : T \,.\, U : \mathsf{Type}_i}$$

$\Pi(\mathsf{Type}_i, \mathsf{Prop})$ $$\frac{\Gamma \vdash T : \mathsf{Type}_i \qquad \Gamma; [x : T] \vdash U : \mathsf{Prop}}{\Gamma \vdash \Pi x : T \,.\, U : \mathsf{Prop}}$$

(Conv) $$\frac{\Gamma \vdash M : T \qquad \Gamma \vdash T' : s}{\Gamma \vdash M : T'} \qquad T =_\beta T'$$

(Cum) $$\frac{\Gamma \vdash T : \mathsf{Prop}}{\Gamma \vdash T : \mathsf{Type}_1} \qquad \frac{\Gamma \vdash T : \mathsf{Type}_i}{\Gamma \vdash T : \mathsf{Type}_{i+1}}$$

A remark about dependent products

In this presentation, dependent products are introduced by the only rules

$$(\mathsf{Prop}, \mathsf{Prop}), \qquad (\mathsf{Type}_i, \mathsf{Type}_i) \quad \text{and} \quad (\mathsf{Type}_i, \mathsf{Prop}).$$

But thanks to the cumulativity rules, we get two additional Π-rules for free:

$$(\mathsf{Type}_i, \mathsf{Type}_j, \mathsf{Type}_{\max(i,j)}) \quad \text{and} \quad (\mathsf{Prop}, \mathsf{Type}_i)$$

$$\frac{\dfrac{\Gamma \vdash T : \mathsf{Type}_i}{\Gamma \vdash T : \mathsf{Type}_{\max(i,j)}}\,(\textsc{Cum}) \qquad \dfrac{\Gamma; [x : T] \vdash U : \mathsf{Type}_j}{\Gamma; [x : T] \vdash U : \mathsf{Type}_{\max(i,j)}}\,(\textsc{Cum})}{\Gamma \vdash \Pi x : T \,.\, U : \mathsf{Type}_{\max(i,j)}}$$

(same idea for $(\mathsf{Prop}, \mathsf{Type}_i)$)

Properties of CCω

- **Syntactical properties:**
  - substitutivity, weakening, strengthening, β-subject reduction
  - principal type, up to β-conversion (no unicity, due to cumulativity)

- **Semantical properties:**

  (1) consistency (i.e no proof of $\Pi A : \mathsf{Prop} \,.\, A$)

  (2) strong normalization (for well-typed terms)

In the following, we will assume neither (1) nor (2), because:

(1) is precisely what we want to prove, without using (2)

(2) is at least as complex as (1) (and in fact, much more difficult)

$\Rightarrow$ combines all the ingredients of (1) + reducibility techniques

Expressivity

CCω is a very expressive formalism in which we can define:

- **Intuitionistic connectives**

$$\begin{array}{lcl}
\bot & \equiv & \Pi X : \mathsf{Prop} \,.\, X \\
\top & \equiv & \Pi X : \mathsf{Prop} \,.\, X \to X \\
A \wedge B & \equiv & \Pi X : \mathsf{Prop} \,.\, (A \to B \to X) \to X \\
A \vee B & \equiv & \Pi X : \mathsf{Prop} \,.\, (A \to X) \to (B \to X) \to X \\
A \Rightarrow B & \equiv & A \to B
\end{array}$$

- **Quantifiers, Leibniz equality**

$$\begin{array}{lcl}
\forall x : T \,.\, A(x) & \equiv & \Pi x : T \,.\, A(x) \\
\exists x : T \,.\, A(x) & \equiv & \Pi X : \mathsf{Prop} \,.\, (\Pi x : T \,.\, A(x) \to X) \to X \\
M_1 =_T M_2 & \equiv & \Pi P : (T \to \mathsf{Prop}) \,.\, P M_1 \to P M_2
\end{array}$$

- Natural numbers (in Type_2), and even Zermelo's set theory (intuitionistic fragment)

Set-theoretic model (big picture)

Idea: use the following dictionary to translate each term M (of CCω) as a set $[\![M]\!]$ in order to ensure the soundness property:

if $M : T$ is derivable (in CCω)

then

$[\![M]\!] \in [\![T]\!]$ is provable (in set theory)

Type theory	Set theory
$\lambda x : T \,.\, M_x$	$(x_{\in T} \mapsto M_x)$
$M\,N$	$M(N)$
$\Pi x : T \,.\, U_x$	$\prod_{x \in T} U_x$
Type_i	$\mathcal{U}_i$ (ith ZF-universe)
Prop	$\{0; 1\}$ (booleans)

Thanks to this property, we will be able to reduce the question

Does the falsity $\Pi A : \mathsf{Prop} \,.\, A$ *have an inhabitant (in the empty context) ?*

to the question

Does the empty set $\varnothing$ *have an element ?* (since $[\![\Pi A : \mathsf{Prop} \,.\, A]\!] = \varnothing$)

Functions in set theory

- A set f is a function if
  1. f is a set of pairs
  2. $\forall x, y, y' \quad (x, y) \in f \ \wedge \ (x, y') \in f \quad \Rightarrow \quad y = y'$

- If f is a function, then
$$\mathrm{Dom}(f) \triangleq \{x; \quad \exists y \quad (x, y) \in f\}$$
$$\mathrm{Ran}(f) \triangleq \{y; \quad \exists x \quad (x, y) \in f\}$$

Abstraction: if D is a set, and if $E[x]$ is an expression depending on x, then

$$x_{\in D} \mapsto E[x] \quad \triangleq \quad \big\{(x, E[x]); \quad x \in D\big\}$$

Application: if $x \in \mathrm{Dom}(f)$, then $\quad f(x) \triangleq$ the unique y s.t. $(x, y) \in f$

Dependent products in set theory

- If A is a set, and if $(B_x)_{x\in A}$ is a family of sets indexed by A, then

$$\prod_{x\in A} B_x \quad \triangleq \quad \{f \text{ function}; \quad \mathrm{Dom}(f) = A \quad \wedge \quad \forall x \in A \quad f(x) \in B_x\}$$

- Non-dependent case: $\displaystyle\prod_{x\in A} B \;=\; A \to B$ (also denoted B^A)

- The set-theoretical equivalents of typing rules (Lam) and (App) are:

$$\frac{\forall x \in A \qquad E[x] \in B_x}{(x_{\in A} \mapsto E[x]) \in \prod_{x\in A} B_x} \qquad\qquad \frac{f \in \prod_{x\in A} B_x \qquad a \in A}{f(a) \in B_a}$$

(In set-theory, they are not 'rules' but theorems.)

Interpreting predicative universes

To interpret the universe hierarchy $(\mathsf{Type}_i)_{i\geq 1}$, we want a family of sets $(\mathcal{U}_i)_{i\geq 1}$ such that:

(1) $\mathcal{U}_i \in \mathcal{U}_{i+1}$ (to interpret the axiom $\mathsf{Type}_i : \mathsf{Type}_{i+1}$)

(2) $\mathcal{U}_i \subset \mathcal{U}_{i+1}$ (to interpret cumulativity)

(3) Each $\mathcal{U}_i$ is Π-closed:

$$A \in \mathcal{U}_i \quad \wedge \quad (\forall x \in A \quad B_x \in \mathcal{U}_i) \quad \Rightarrow \quad \Big(\prod_{x\in A} B_x\Big) \in \mathcal{U}_i$$

Problem:

- Condition (3) induces a dramatic combinatorial explosion (if we assume $\omega \in \mathcal{U}_i$)
- Existence of such sets is not provable in ZFC
  $\Rightarrow$ Need a notion of set-theoretic universe (ZF-universe)

ZF-universes

A ZF-universe is a set (of sets) $\mathcal{U}$ such that:

(1) if $A \in \mathcal{U}$, then $A \subset \mathcal{U}$ ($\mathcal{U}$ is transitive)

(2) if $A \in \mathcal{U}$, then $\mathfrak{P}(A) \in \mathcal{U}$ ($\mathcal{U}$ is $\mathfrak{P}$-closed)

(3) if $A \in \mathcal{U}$ and $\forall x \in A \quad B_x \in \mathcal{U}$, then $\bigcup_{x \in A} B_x \in \mathcal{U}$ ($\mathcal{U}$ is $\bigcup$-closed)

(4) $\omega \in \mathcal{U}$ (infinity)

- Such a set is closed under all the axioms of Zermelo-Fraenkel (+ choice) : pairing, powerset, comprehension, union, replacement, infinity

  $\Rightarrow$ Thus its existence cannot be proved in ZF (Gödel's argument)

- In particular, a ZF-universe is Π-closed (provided we postulate its existence)

ZF-universes and inaccessible cardinals

- A cardinal α is (strongly) inaccessible if:

  (1) if $\beta < \alpha$, then $2^\beta < \alpha$

  (2) if $\beta < \alpha$ and $\gamma_i < \alpha$ for all $i \in \beta$, then $\left(\sup_{i\in\beta} \gamma_i\right) < \alpha$

  (3) $\aleph_0 < \alpha$

  Intuitively, this definition expresses the same idea as the notion of ZF-universe, but only in terms of cardinality. In particular: *the cardinal of a ZF-universe is always inaccessible.*

- Conversely, inaccessible cardinals allow the construction of ZF-universes from the cumulative hierarchy (V_x), which is transfinitely defined by:

$$V_0 = \varnothing, \qquad V_{x+1} = \mathfrak{P}(V_x), \qquad V_x = \bigcup_{y<x} V_y \quad \text{(if } x \text{ limit ordinal)}$$

- **Lemma:** *If μ is inaccessible, then V_μ is a ZF-universe*

Building the universe hierarchy

We extend ZFC by adding the following axiom:

Axiom (SI$^\omega$): *There exists infinitely many inaccessible cardinals*

Then, using this (very strong!) axiom:

- Let: $\mu_i \triangleq i$th inaccessible cardinal, $\mathcal{U}_i \triangleq V_{\mu_i}$ (ith ZF-universe)
- From these definitions, one can easily check that for all $i \leq 1$:

$$\mathcal{U}_i \in \mathcal{U}_{i+1}, \qquad \mathcal{U}_i \subset \mathcal{U}_{i+1} \qquad \text{and} \qquad \mathcal{U}_i \text{ is } \Pi\text{-closed}$$

Remark: Inaccessible cardinals are not strictly needed to interpret universes:

– Some clever tricks [Melliès-Werner] permit to restrict the function spaces

– This prevents the combinatorial explosion $\Rightarrow$ the whole model fits in $V_{\omega 2}$

– But this method does not work anymore in presence of inductive datatypes

Interpreting Prop

Difficulty: how to interpret the impredicativity of Prop ?

$$\frac{\Gamma \vdash T : s \qquad \Gamma; [x : T] \vdash U : \mathsf{Prop}}{\Gamma \vdash \Pi x : T \,.\, U : \mathsf{Prop}}$$

Set theoretical translation:

$$(\forall x \in A \quad \mathit{Prop}(B_x)) \qquad \Rightarrow \qquad \mathit{Prop}\Big(\prod_{x \in A} B_x\Big)$$

Problem: How to define the predicate $\mathit{Prop}(X)$?

A simple solution: $\mathit{Prop}(X) \equiv X$ has at most one element (proof-irrelevance)

Proof-irrelevance

- The sort of propositions Prop will be interpreted as $\big\{\varnothing;\ \{\mathbf{prf}\}\big\}$ (i.e. booleans), where $\mathbf{prf}$ is an arbitrary (but small) object that will interpret any proof

- **Fact:** *if* $B_x \in \big\{\varnothing;\ \{\mathbf{prf}\}\big\}$ *for all* $x \in A$, *then:*

$$\prod_{x \in A} B_x \quad = \quad \begin{cases} \varnothing & \textit{if } B_x = \varnothing \ \textit{ for some } x \in A \\ \big\{(x_{\in A} \mapsto \mathbf{prf})\big\} & \textit{if } B_x = \{\mathbf{prf}\} \ \textit{ for all } x \in A \end{cases}$$

- **Problem:** the constant function $(x_{\in A} \mapsto \mathbf{prf})$ is not equal to $\mathbf{prf}$

  $\Rightarrow$ must introduce some trick to identify them

Identifying singletons

- We introduce a simple mechanism of encoding/decoding:

$$\mathbf{lam}(f) \triangleq \begin{cases} \mathbf{prf} & \text{if } f = (x_{\in A} \mapsto \mathbf{prf}) \text{ for some } A \\ f & \text{otherwise} \end{cases}$$

$$\mathbf{app}(h, x) \triangleq \begin{cases} \mathbf{prf} & \text{if } h = \mathbf{prf} \\ h(x) & \text{otherwise} \end{cases}$$

- Create a new cartesian product which keeps functions in their encoded form only:

$$\widehat{\prod_{x \in A}} B_x \triangleq \left\{ \mathbf{lam}(f); \quad f \in \prod_{x \in A} B_x \right\}$$

- In all cases, we have: $\mathbf{app}(\mathbf{lam}(f), x) = f(x)$ (provided $x \in \mathrm{Dom}(f)$)

Model and valuations

- The model (i.e. the set of all values) is defined by:

$$\mathcal{M} \quad = \quad \bigcup_{i \geq 1} V_{\mu_i} \quad = \quad V_{\sup_{i \geq 1} \mu_i}$$

- A valuation is a function $\rho : \mathcal{V} \to \mathcal{M}$ associating a value $\rho(x)$ to each variable $x \in \mathcal{V}$

- The set of all valuations is denoted by $\mathbf{Val}_{\mathcal{M}}$ $(= \mathcal{V} \to \mathcal{M})$

- For all $\rho \in \mathbf{Val}_{\mathcal{M}}$, $x \in \mathcal{V}$ and $v \in \mathcal{M}$ we define $(\rho; x \leftarrow v)$ by setting:

$$(\rho; x \leftarrow v)(y) \quad \triangleq \quad \begin{cases} v & \text{if } y = x \\ \rho(y) & \text{otherwise} \end{cases}$$

Interpreting terms

Each term is interpreted as a partial function $[\![M]\!] \;=\; (\rho \mapsto [\![M]\!]_\rho) \;:\; \mathbf{Val}_{\mathcal{M}} \nrightarrow \mathcal{M}$ defined by induction on M as follows:

$$
\begin{array}{lcll}
[\![x]\!]_\rho & = & \rho(x) & \\
[\![\mathsf{Prop}]\!]_\rho & = & \{\varnothing;\ \{\mathbf{prf}\}\} & \\
[\![\mathsf{Type}_i]\!]_\rho & = & \mathcal{U}_i \quad (= \quad V_{\mu_i}) & \\
[\![\Pi x : T \,.\, U]\!]_\rho & = & \widehat{\prod_{v \in [\![T]\!]_\rho}} [\![U]\!]_{\rho; x \leftarrow v} & \text{(provided it belongs to } \mathcal{M}\text{)} \\
[\![\lambda x : T \,.\, M]\!]_\rho & = & \mathbf{lam}\big(v_{\in [\![T]\!]_\rho} \mapsto [\![M]\!]_{\rho; x \leftarrow v}\big) & \text{(provided it belongs to } \mathcal{M}\text{)} \\
[\![M N]\!]_\rho & = & \mathbf{app}\big([\![M]\!]_\rho, [\![N]\!]_\rho\big) & \text{(may be undefined)}
\end{array}
$$

Remark: Application introduces partiality, as well as Π and λ (that may not fit in $\mathcal{M}$)

Interpreting contexts

- A valuation $\rho : \mathcal{V} \to \mathcal{M}$ is adapted to a context $\Gamma = [x_1 : T_1; \ldots; x_n : T_n]$ if:

$$\forall i \in [1..n] \qquad \rho(x) \in [\![T]\!]_\rho$$

- The interpretation of a context is the set of all its adapted valuations:

$$[\![\Gamma]\!] \quad \triangleq \quad \{\rho \in \mathbf{Val}_{\mathcal{M}}; \quad \rho \text{ is adapted to } \Gamma\}$$

- Inductive characterization:

$$[\![[]]\!] \quad = \quad \mathbf{Val}_{\mathcal{M}}\,, \qquad\qquad [\![\Gamma; [x : T]]\!] \quad = \quad \{\rho \in [\![\Gamma]\!]; \quad \rho(x) \in [\![T]\!]_\rho\}$$

$\Rightarrow$ The longest the context, the smallest its interpretation

Remark: the interpretation of a context may be empty (i.e. $[\![\Gamma]\!] = \varnothing$ for some Γ)

Soundness

- **Variable dependence:** $[\![M]\!]_\rho$ only depends on the values $\rho(x)$ for $x \in FV(M)$

  $\Rightarrow$ If M is closed, then $[\![M]\!]_\rho$ does not depend on ρ (usually denoted $[\![M]\!]$)

- **Substitutivity:** $[\![M\{x := N\}]\!]_\rho = [\![M]\!]_{(\rho; x \leftarrow [\![N]\!]_\rho)}$ (for all M, N, x, ρ)

  [Notice that the lefthand side is defined iff the righthand side is defined too]

- **Soundness of typing:** if $\Gamma \vdash M : T$, then for all $\rho \in [\![\Gamma]\!]$

  $$[\![M]\!]_\rho,\ [\![T]\!]_\rho \text{ are well defined} \quad \text{and} \quad \underline{[\![M]\!]_\rho \in [\![T]\!]_\rho}$$

- **Soundness of β-reduction:** if $\Gamma \vdash M : T$ and $M \to_\beta M'$, then:

  $$\underline{[\![M]\!]_\rho = [\![M']\!]_\rho} \quad (\text{for all } \rho \in [\![\Gamma]\!])$$

Problem for proving soundness. . .

- Soundness of typing

$$\Gamma \vdash M : T, \quad \rho \in [\![\Gamma]\!] \qquad \Rightarrow \qquad [\![M]\!]_\rho \in [\![T]\!]_\rho$$

  cannot be simply proven by induction on $\Gamma \vdash M : T$.

- Almost all the typing rules (Var, Sort, Prod, Lam, App, Cum) sucessfully pass the test. . .
  . . . but the typing rule Conv fails, because the implication

$$M \to_\beta M' \qquad \not\Rightarrow \qquad [\![M]\!]_\rho = [\![M']\!]_\rho$$

  does not hold for raw-terms M, M'.

- We should prove soundness of typing and soundness of (typed) β-reduction simultaneously. . .
  . . . but it seems hard to do it simply (and correctly).

… and how to fix it

- A simple idea:
  - introduce an explicit error: $[\![M]\!] \;:\; \mathbf{Val}_{\mathcal{M}} \to \mathcal{M} \cup \{\mathbf{err}\}$
  - prove that: $M \to_\beta M', \quad [\![M]\!]_\rho \neq \mathbf{err} \quad \Rightarrow \quad [\![M']\!]_\rho = [\![M]\!]_\rho \quad (*)$
  - reformulate soundness as: *if* $\Gamma \vdash M : T$ *and* $\rho \in [\![\Gamma]\!]$, *then:*

$$[\![M]\!]_\rho \neq \mathbf{err}, \quad [\![T]\!]_\rho \neq \mathbf{err} \quad \textit{and} \quad [\![M]\!]_\rho \in [\![T]\!]_\rho$$

$\Rightarrow$ Does not work due to the impredicativity of Prop ($(*)$ does not hold)

- Consider typed applications + typed redexes [Altenkirch, Melliès-Werner]

$$@_A(\lambda x : A \,.\, M,\ N) \quad \to_\beta \quad M\{x := N\} \quad \text{only if types } (A) \text{ match}$$

- Replace untyped conversion by an equality judgment [Martin-Löf]

Consistency

- The intuitionistic falsity $\bot \triangleq \Pi X : \mathsf{Prop} \,.\, X$ is interpreted as

$$[\![\Pi X : \mathsf{Prop} \,.\, X]\!]_\rho \quad = \quad \widehat{\prod_{a \in [\![\mathsf{Prop}]\!]_\rho}} [\![X]\!]_{\rho; X \leftarrow a} \quad = \quad \widehat{\prod_{a \in \{\varnothing; \{\mathbf{prf}\}\}}} a \quad = \quad \varnothing$$

- Assume there is some M such that $[] \vdash M : \Pi X : \mathsf{Prop} \,.\, X$
  - Take an arbitrary $\rho \in [\![[]]\!] = \mathbf{Val}_{\mathcal{M}}$
  - From soundness we get: $[\![M]\!]_\rho \in [\![\Pi X : \mathsf{Prop} \,.\, X]\!]_\rho = \varnothing$ (absurdity)
  - Hence the assumption is false $\Rightarrow$ CCω is logically consistent

- **Remark:** A very simple proof, which relies on the soundness property

$\Rightarrow$ But this result has been proved in a very strong set theory (ZFC + SI$^\omega$)

Interpretation of connectives

- Intuitionistic connectives are defined in CCω as:

$$\bot \;:\; \mathsf{Prop} \;\triangleq\; \Pi X : \mathsf{Prop} \,.\, X \qquad\qquad \top \;:\; \mathsf{Prop} \;\triangleq\; \Pi X : \mathsf{Prop} \,.\, X \to X$$

$$\begin{array}{lcll}
\neg & : & \mathsf{Prop} \to \mathsf{Prop} & \triangleq\; \lambda A : \mathsf{Prop} \,.\, A \to \bot \\
\wedge & : & \mathsf{Prop} \to \mathsf{Prop} \to \mathsf{Prop} & \triangleq\; \lambda A, B : \mathsf{Prop} \,.\, \Pi X : \mathsf{Prop} \,.\, (A \to B \to X) \to X \\
\vee & : & \mathsf{Prop} \to \mathsf{Prop} \to \mathsf{Prop} & \triangleq\; \lambda A, B : \mathsf{Prop} \,.\, \Pi X : \mathsf{Prop} \,.\, (A \to X) \to (B \to X) \to X \\
\Rightarrow & : & \mathsf{Prop} \to \mathsf{Prop} \to \mathsf{Prop} & \triangleq\; \lambda A, B : \mathsf{Prop} \,.\, A \to B
\end{array}$$

- Let $\mathbf{0} = \varnothing$ (false) and $\mathbf{1} = \{\mathbf{prf}\}$ (true). Thanks to soundness, we have:

$$[\![\bot]\!], [\![\top]\!] \in \{\mathbf{0}; \mathbf{1}\}; \qquad [\![\neg]\!] \in \{\mathbf{0}; \mathbf{1}\} \to \{\mathbf{0}; \mathbf{1}\}; \qquad [\![\wedge]\!], [\![\vee]\!], [\![\Rightarrow]\!] \in \{\mathbf{0}; \mathbf{1}\} \to \{\mathbf{0}; \mathbf{1}\} \to \{\mathbf{0}; \mathbf{1}\}$$

- Since the objects $[\![\bot]\!]$, $[\![\top]\!]$, $[\![\wedge]\!]$, $[\![\vee]\!]$ and $[\![\Rightarrow]\!]$ are finite, we can easily check that

$$[\![\neg]\!](\mathbf{0}) = \mathbf{1}, \quad [\![\neg]\!](\mathbf{1}) = \mathbf{0}, \quad [\![\wedge]\!](\mathbf{0}, \mathbf{0}) = \mathbf{0}, \quad \text{etc.} \qquad \text{(classical truth-values tables)}$$

[In the same way, intuitionistic quantifiers $\forall$ and $\exists$ become classical in the model]

Adding axioms in the context

- A context Γ is:
  - consistent if there is no M such that $\Gamma \vdash M : \bot$
  - satisfiable if there is some $\rho \in [\![\Gamma]\!]$

- **Lemma:** *Any satisfiable context is consistent* (same proof as for consistency)

- To extend a given satisfiable context Γ (with a given $\rho \in [\![\Gamma]\!]$):
  - Take a type T such that $[\![T]\!]_\rho \neq \varnothing$, and pick some $v \in [\![T]\!]_\rho$
  - Then $[\![\Gamma; [x : T]]\!]$ is satisfiable, with the valuation $(\rho; x \leftarrow v) \in [\![\Gamma; [x : T]]\!]$

- **Morality:**

$$\begin{array}{lcl} T \text{ provable} & \Rightarrow & [\![T]\!] \neq \varnothing \\ [\![T]\!] = \varnothing & \Rightarrow & T \text{ not provable} \\ [\![T]\!] \neq \varnothing & \Rightarrow & T \text{ consistent} \quad \text{(can be safely added as an axiom)} \end{array}$$

Some valid propositions

- **Propositional axioms**

  $\forall A : \mathsf{Prop} .\quad A \ \vee\ \neg A$ (excluded middle)

  $\forall A : \mathsf{Prop} .\quad \forall x, y : A .\quad x =_A y$ (proof-irrelevance)

  $\forall A : \mathsf{Prop} .\quad A =_{\mathsf{Prop}} \bot \ \vee\ A =_{\mathsf{Prop}} \top$ (implies both E.M. and P.I.)

- **Functional extensionality:**

  $$\forall f, g : T \to U .\quad [\forall x : T .\quad f(x) =_U g(x)] \quad \Rightarrow \quad f =_{T \to U} g$$

- **Axiom of choice:**

  $$[\forall x : T . \exists y : U . R(x, y)] \quad \Rightarrow \quad \exists f : T{\to}U . \forall x : T . R(x, f(x))$$

- **Hilbert's epsilon** (for any <u>inhabited</u> type T):

  $$\boldsymbol{\varepsilon} \quad : \quad (T \to \mathsf{Prop}) \to T$$

  $$\forall P : T{\to}\mathsf{Prop} .\quad [\exists x . P(x)] \quad \Rightarrow \quad P(\boldsymbol{\varepsilon}(P))$$

About the interpretation of Type_1

- Inaccessible cardinals are not necessary to interpret Type_1

  $\Rightarrow$ We can interpret Type_1 by V_ω (set of all hereditarily finite sets)

  [Remember that: $V_0 = \varnothing,\quad V_{n+1} = \mathfrak{P}(V_n),\quad V_\omega = \bigcup V_n$]

- This set is Π-closed, and contains $\{\varnothing;\ \{\mathbf{prf}\}\}$ (provided $\mathbf{prf} \in V_\omega$)

- Shift the whole hierarchy: $\mathcal{U}_1 = V_\omega,\quad \mathcal{U}_2 = V_{\mu_1},\quad \mathcal{U}_3 = V_{\mu_2},$ etc.

- With this new construction, soundness still holds

- This model shows that there is no provably infinite datatype in Type_1

  (because the denotation of such a type would be infinite. . .)

  $\Rightarrow$ But this is no more true in Type_i for $i \geq 2$ (cf next lecture)

An advanced exercise

In your favorite functional language (here, Objective Caml), implement the finitary model of the Calculus of Constructions:

```
type term =
  | Rel of int              (* de Bruijn index *)
  | Prop | Type             (* sorts *)
  | Prd of term * term      (* dependent product *)
  | Lam of term * term      (* abstraction *)
  | App of term * term ;;   (* application *)

type denot =
  | Prf                              (* unique proof object *)
  | Set of denot list                (* finite set (type), as a list *)
  | Fun of (denot * denot) list ;;   (* function, as an association list *)

exception Undefined ;;

val interp : term -> denot list -> denot ;; (* may raise Undefined *)
```

Conclusion

- A simple way of checking consistency (independently from SN)

  $\Rightarrow$ Any proof of SN needs the same ingredients (+ reducibility)

- **Possible extensions** (without changing the model):
  - Inductive datatypes, record types
  - Subtyping with covariance (such as in ECC [Luo 84]):
  - All the 'classical' mathematics (quotients, reals, etc.)

$$\frac{B \leq B'}{A \to B \leq A \to B'}$$

- **Things that cannot be interpreted** (in this model)
  - Intuitionistic features (non-provability of E.M., sort Set of Coq)
  - Domain-free abstractions (i.e. $\lambda x \,.\, M$ of DFPTS [Barthe])
  - Subtyping with contravariance (in some versions of ECC):

$$\frac{A \leq A'}{A' \to B \leq A \to B}$$