

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:

Type Theory [1970's] \longrightarrow 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 $CC\omega$?

- 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 $Type_1 = Type$)
 - + **cumulativity rules** to enforce $Prop \subset Type_1$, $Type_i \subset Type_{i+1}$(but no inductive datatypes)
- The intended use of $CC\omega$ is slightly different from that of `CC`:
 - `Prop` is reserved 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 $CC\omega$

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

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

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

Judgments $\Gamma \vdash$ ' Γ is a well-formed context'
 $\Gamma \vdash M : T$ 'under Γ , the term M has type T '

Reduction $(\lambda x : T . M)N \rightarrow_\beta M\{x := N\}$

Notations $FV(M)$ (free variables), $DV(\Gamma)$ (declared variables),
 $M\{x := N\}$ (external substitution), $M_1 =_\beta M_2$ (β -conversion)

Typing rules (1/2)

(Context)	$\frac{}{[] \vdash} \quad \frac{\Gamma \vdash T : s}{\Gamma; [x : T] \vdash}$	$x \notin DV(\Gamma)$
(Var)	$\frac{\Gamma \vdash}{\Gamma \vdash x : T}$	$(x:T) \in \Gamma$
(Sort)	$\frac{\Gamma \vdash}{\Gamma \vdash \text{Prop} : \text{Type}_1} \quad \frac{\Gamma \vdash}{\Gamma \vdash \text{Type}_i : \text{Type}_{i+1}}$	
(Lam)	$\frac{\Gamma; [x : T] \vdash M : U \quad \Gamma \vdash \Pi x : T . U : s}{\Gamma \vdash \lambda x : T . M : \Pi x : T . U}$	
(App)	$\frac{\Gamma \vdash M : \Pi x : T . U \quad \Gamma \vdash N : T}{\Gamma \vdash MN : U\{x := N\}}$	

Typing rules (2/2)

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

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

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

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

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

A remark about dependent products

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

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

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

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

$$\frac{\Gamma \vdash T : \text{Type}_i \quad \Gamma; [x : T] \vdash U : \text{Type}_j}{\Gamma \vdash T : \text{Type}_{\max(i,j)} \quad \Gamma; [x : T] \vdash U : \text{Type}_{\max(i,j)}} \quad \text{(same idea for } (\text{Prop}, \text{Type}_i))$$

\vdots (CUM) \vdots (CUM)

Properties of $CC\omega$

- **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 $\Pi A : \text{Prop} . A$)
- (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

$\text{CC}\omega$ is a very expressive formalism in which we can define:

- **Intuitionistic connectives**

$$\begin{aligned}\perp &\equiv \Pi X : \text{Prop} . X \\ \top &\equiv \Pi X : \text{Prop} . X \rightarrow X \\ A \wedge B &\equiv \Pi X : \text{Prop} . (A \rightarrow B \rightarrow X) \rightarrow X \\ A \vee B &\equiv \Pi X : \text{Prop} . (A \rightarrow X) \rightarrow (B \rightarrow X) \rightarrow X \\ A \Rightarrow B &\equiv A \rightarrow B\end{aligned}$$

- **Quantifiers, Leibniz equality**

$$\begin{aligned}\forall x : T . A(x) &\equiv \Pi x : T . A(x) \\ \exists x : T . A(x) &\equiv \Pi X : \text{Prop} . (\Pi x : T . A(x) \rightarrow X) \rightarrow X \\ M_1 =_T M_2 &\equiv \Pi P : (T \rightarrow \text{Prop}) . P M_1 \rightarrow P M_2\end{aligned}$$

- **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 $\mathbf{CC}\omega$) as a set $\llbracket M \rrbracket$ in order to ensure the **soundness** property:

if $M : T$ is **derivable** (in $\mathbf{CC}\omega$)

then

$\llbracket M \rrbracket \in \llbracket T \rrbracket$ is **provable** (in set theory)

Type theory	Set theory
$\lambda x : T . M_x$	$(x \in T \mapsto M_x)$
MN	$M(N)$
$\prod x : T . U_x$	$\prod_{x \in T} U_x$
Type _{<i>i</i>}	\mathcal{U}_i (<i>i</i> th ZF-universe)
Prop	$\{0; 1\}$ (booleans)

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

Does the falsity $\prod A : \mathbf{Prop} . A$ have an inhabitant (in the empty context) ?

to the question

Does the empty set \emptyset have an element ? (since $\llbracket \prod A : \mathbf{Prop} . A \rrbracket = \emptyset$)

Functions in set theory

- A set f is a **function** if
 1. f is a set of pairs
 2. $\forall x, y, y' \quad (x, y) \in f \wedge (x, y') \in f \Rightarrow y = y'$
- If f is a function, then
$$\text{Dom}(f) \triangleq \{x; \exists y \ (x, y) \in f\}$$
$$\text{Ran}(f) \triangleq \{y; \exists x \ (x, y) \in f\}$$

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

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

Application: if $x \in \text{Dom}(f)$, then $f(x) \triangleq$ 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

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

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

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

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

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

Interpreting predicative universes

To interpret the universe hierarchy $(\mathbf{Type}_i)_{i \geq 1}$, 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 $\mathbf{Type}_i : \mathbf{Type}_{i+1}$)
- (2) $\mathcal{U}_i \subset \mathcal{U}_{i+1}$ (to interpret cumulativity)
- (3) Each \mathcal{U}_i is Π -closed:

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

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 $\forall x \in A \ B_x \in \mathcal{U}$, then $\bigcup_{x \in A} B_x \in \mathcal{U}$ (\mathcal{U} is **\cup -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 $(\sup_{i \in \beta} \gamma_i) < \alpha$
 - (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:

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

- **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^ω): *There exists infinitely many inaccessible cardinals*

Then, using this (very strong!) axiom:

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

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

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_{ω^2}
- But this method does not work anymore in presence of inductive datatypes

Interpreting Prop

Difficulty: how to interpret the impredicativity of Prop ?

$$\frac{\Gamma \vdash T : s \quad \Gamma; [x : T] \vdash U : \text{Prop}}{\Gamma \vdash \prod x : T . U : \text{Prop}}$$

Set theoretical translation:

$$(\forall x \in A \text{ Prop}(B_x)) \quad \Rightarrow \quad \text{Prop}\left(\prod_{x \in A} B_x\right)$$

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

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

Proof-irrelevance

- The sort of propositions **Prop** will be interpreted as $\{\emptyset; \{\mathbf{prf}\}\}$ (i.e. **booleans**), where **prf** is an arbitrary (but small) object that will interpret **any proof**
- **Fact:** if $B_x \in \{\emptyset; \{\mathbf{prf}\}\}$ for all $x \in A$, then:

$$\prod_{x \in A} B_x = \begin{cases} \emptyset & \text{if } B_x = \emptyset \text{ for some } x \in A \\ \{(x \in A \mapsto \mathbf{prf})\} & \text{if } B_x = \{\mathbf{prf}\} \text{ for all } x \in A \end{cases}$$

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

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

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

Model and valuations

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

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

- A **valuation** is a function $\rho : \mathcal{V} \rightarrow \mathcal{M}$ associating a value $\rho(x)$ to each variable $x \in \mathcal{V}$
- The set of all valuations is denoted by $\mathbf{Val}_{\mathcal{M}} (= \mathcal{V} \rightarrow \mathcal{M})$
- 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:

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

Interpreting terms

Each term is interpreted as a **partial function** $\llbracket M \rrbracket = (\rho \mapsto \llbracket M \rrbracket_\rho) : \mathbf{Val}_{\mathcal{M}} \rightarrow \mathcal{M}$ defined by induction on M as follows:

$$\llbracket x \rrbracket_\rho = \rho(x)$$

$$\llbracket \mathbf{Prop} \rrbracket_\rho = \{\emptyset; \{\mathbf{prf}\}\}$$

$$\llbracket \mathbf{Type}_i \rrbracket_\rho = \mathcal{U}_i (= V_{\mu_i})$$

$$\llbracket \Pi x : T . U \rrbracket_\rho = \prod_{v \in \llbracket T \rrbracket_\rho} \widehat{\llbracket U \rrbracket_{\rho; x \leftarrow v}} \quad (\text{provided it belongs to } \mathcal{M})$$

$$\llbracket \lambda x : T . M \rrbracket_\rho = \mathbf{lam}(v \in \llbracket T \rrbracket_\rho \mapsto \llbracket M \rrbracket_{\rho; x \leftarrow v}) \quad (\text{provided it belongs to } \mathcal{M})$$

$$\llbracket MN \rrbracket_\rho = \mathbf{app}(\llbracket M \rrbracket_\rho, \llbracket N \rrbracket_\rho) \quad (\text{may be undefined})$$

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

Interpreting contexts

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

$$\forall i \in [1..n] \quad \rho(x) \in \llbracket T \rrbracket_\rho$$

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

$$\llbracket \Gamma \rrbracket \triangleq \{ \rho \in \mathbf{Val}_{\mathcal{M}}; \rho \text{ is adapted to } \Gamma \}$$

- Inductive characterization:

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

\Rightarrow The longer the context, the smaller its interpretation

Remark: the interpretation of a context **may be empty** (i.e. $\llbracket \Gamma \rrbracket = \emptyset$ for some Γ)

Soundness

- **Variable dependence:** $\llbracket M \rrbracket_\rho$ only depends on the values $\rho(x)$ for $x \in FV(M)$
 \Rightarrow If M is **closed**, then $\llbracket M \rrbracket_\rho$ **does not depend on ρ** (usually denoted $\llbracket M \rrbracket$)

- **Substitutivity:** $\llbracket M\{x := N\} \rrbracket_\rho = \llbracket M \rrbracket_{(\rho; x \leftarrow \llbracket N \rrbracket_\rho)}$ (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 $\rho \in \llbracket \Gamma \rrbracket$

$$\llbracket M \rrbracket_\rho, \llbracket T \rrbracket_\rho \text{ are well defined} \quad \text{and} \quad \underline{\llbracket M \rrbracket_\rho \in \llbracket T \rrbracket_\rho}$$

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

$$\underline{\llbracket M \rrbracket_\rho = \llbracket M' \rrbracket_\rho} \quad (\text{for all } \rho \in \llbracket \Gamma \rrbracket)$$

Problem for proving soundness...

- Soundness of typing

$$\Gamma \vdash M : T, \quad \rho \in \llbracket \Gamma \rrbracket \quad \Rightarrow \quad \llbracket M \rrbracket_\rho \in \llbracket T \rrbracket_\rho$$

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

- Almost all the typing rules (VAR, SORT, PROD, LAM, APP, CUM) successfully pass the test. . .
. . . but the typing rule **CONV** fails, because the implication

$$M \rightarrow_\beta M' \quad \not\Rightarrow \quad \llbracket M \rrbracket_\rho = \llbracket M' \rrbracket_\rho$$

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**: $\llbracket M \rrbracket : \mathbf{Val}_{\mathcal{M}} \rightarrow \mathcal{M} \cup \{\mathbf{err}\}$

– prove that: $M \rightarrow_{\beta} M', \llbracket M \rrbracket_{\rho} \neq \mathbf{err} \Rightarrow \llbracket M' \rrbracket_{\rho} = \llbracket M \rrbracket_{\rho} \quad (*)$

– reformulate soundness as: *if $\Gamma \vdash M : T$ and $\rho \in \llbracket \Gamma \rrbracket$, then:*

$$\llbracket M \rrbracket_{\rho} \neq \mathbf{err}, \llbracket T \rrbracket_{\rho} \neq \mathbf{err} \text{ and } \llbracket M \rrbracket_{\rho} \in \llbracket T \rrbracket_{\rho}$$

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

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

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

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

Consistency

- The intuitionistic falsity $\perp \triangleq \Pi X : \text{Prop} . X$ is interpreted as

$$\llbracket \Pi X : \text{Prop} . X \rrbracket_\rho = \prod_{a \in \llbracket \text{Prop} \rrbracket_\rho} \llbracket X \rrbracket_{\rho; X \leftarrow a} = \prod_{a \in \{\emptyset; \{\mathbf{prf}\}\}} a = \emptyset$$

- Assume there is some M such that $\Box \vdash M : \Pi X : \text{Prop} . X$
 - Take an arbitrary $\rho \in \llbracket \Box \rrbracket = \mathbf{Val}_{\mathcal{M}}$
 - From soundness we get: $\llbracket M \rrbracket_\rho \in \llbracket \Pi X : \text{Prop} . X \rrbracket_\rho = \emptyset$ (**absurdity**)
 - Hence the assumption is false \Rightarrow $\text{CC}\omega$ 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 ($\text{ZFC} + \text{SI}^\omega$)

Interpretation of connectives

- Intuitionistic connectives are defined in $\mathbf{CC}\omega$ as:

$$\begin{aligned}
 \perp & : \text{Prop} \triangleq \Pi X : \text{Prop} . X & \top & : \text{Prop} \triangleq \Pi X : \text{Prop} . X \rightarrow X \\
 \neg & : \text{Prop} \rightarrow \text{Prop} \triangleq \lambda A : \text{Prop} . A \rightarrow \perp \\
 \wedge & : \text{Prop} \rightarrow \text{Prop} \rightarrow \text{Prop} \triangleq \lambda A, B : \text{Prop} . \Pi X : \text{Prop} . (A \rightarrow B \rightarrow X) \rightarrow X \\
 \vee & : \text{Prop} \rightarrow \text{Prop} \rightarrow \text{Prop} \triangleq \lambda A, B : \text{Prop} . \Pi X : \text{Prop} . (A \rightarrow X) \rightarrow (B \rightarrow X) \rightarrow X \\
 \Rightarrow & : \text{Prop} \rightarrow \text{Prop} \rightarrow \text{Prop} \triangleq \lambda A, B : \text{Prop} . A \rightarrow B
 \end{aligned}$$

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

$$\llbracket \perp \rrbracket, \llbracket \top \rrbracket \in \{\mathbf{0}; \mathbf{1}\}; \quad \llbracket \neg \rrbracket \in \{\mathbf{0}; \mathbf{1}\} \rightarrow \{\mathbf{0}; \mathbf{1}\}; \quad \llbracket \wedge \rrbracket, \llbracket \vee \rrbracket, \llbracket \Rightarrow \rrbracket \in \{\mathbf{0}; \mathbf{1}\} \rightarrow \{\mathbf{0}; \mathbf{1}\} \rightarrow \{\mathbf{0}; \mathbf{1}\}$$

- Since the objects $\llbracket \perp \rrbracket$, $\llbracket \top \rrbracket$, $\llbracket \wedge \rrbracket$, $\llbracket \vee \rrbracket$ and $\llbracket \Rightarrow \rrbracket$ are **finite**, we can easily check that

$$\llbracket \neg \rrbracket(\mathbf{0}) = \mathbf{1}, \quad \llbracket \neg \rrbracket(\mathbf{1}) = \mathbf{0}, \quad \llbracket \wedge \rrbracket(\mathbf{0}, \mathbf{0}) = \mathbf{0}, \quad \text{etc.} \quad (\text{classical truth-values tables})$$

[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 $\Gamma \vdash M : \perp$
 - **satisfiable** if there is some $\rho \in \llbracket \Gamma \rrbracket$
- **Lemma:** *Any satisfiable context is consistent* (same proof as for consistency)
- To extend a given satisfiable context Γ (with a given $\rho \in \llbracket \Gamma \rrbracket$):
 - Take a type T such that $\llbracket T \rrbracket_\rho \neq \emptyset$, and pick some $v \in \llbracket T \rrbracket_\rho$
 - Then $\llbracket \Gamma; [x : T] \rrbracket$ is satisfiable, with the valuation $(\rho; x \leftarrow v) \in \llbracket \Gamma; [x : T] \rrbracket$
- **Morality:**
 - T provable $\Rightarrow \llbracket T \rrbracket \neq \emptyset$
 - $\llbracket T \rrbracket = \emptyset \Rightarrow T$ not provable
 - $\llbracket T \rrbracket \neq \emptyset \Rightarrow T$ consistent (can be safely added as an axiom)

Some valid propositions

- **Propositional axioms**

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

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

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

- **Functional extensionality:**

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

- **Axiom of choice:**

$$[\forall x : T . \exists y : U . R(x, y)] \Rightarrow \exists f : T \rightarrow U . \forall x : T . R(x, f(x))$$

- **Hilbert's epsilon** (for any inhabited type T):

$$\epsilon : (T \rightarrow \text{Prop}) \rightarrow T$$

$$\forall P : T \rightarrow \text{Prop} . [\exists x . P(x)] \Rightarrow P(\epsilon(P))$$

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: $V_0 = \emptyset$, $V_{n+1} = \mathfrak{P}(V_n)$, $V_\omega = \bigcup V_n$]
- This set is Π -closed, and contains $\{\emptyset; \{\mathbf{prf}\}\}$ (provided $\mathbf{prf} \in V_\omega$)
- Shift the whole hierarchy: $\mathcal{U}_1 = V_\omega$, $\mathcal{U}_2 = V_{\mu_1}$, $\mathcal{U}_3 = V_{\mu_2}$, 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)
⇒ 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.)

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

- **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):

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