Dependent Types Simplified

Tristan Bice

Institute of Mathematics of the Czech Academy of Sciences

Syntax and Semantics of Type Theory

June 4th 2026

Motivation

- Me: Background in set theory (ZFC = Zermelo-Fraenkel set theory with the Axiom of Choice) and model theory. Interested in type theories of proof assistants like Rocq and Lean.
- Can we trust such type theories? No – large ad hoc combinations of inference rules with only vague semantic justification inevitably produce inconsistent logical systems.
- Rocq has been plagued by inconsistencies for decades ($\sim$ one proof of False discovered per year). Now also happening with Lean, e.g. Carneiro's proof of False last year.
- Solution: A type theory with a simple language and minimal inference rules based on precise set theoretic semantics.
- Any such type theory will automatically be consistent, as long as ZFC is (so far no inconsistency found in over 100 years).

The Language

- Alphabet: $\{\mathsf{v}, \mathsf{c}, ', \beta, \lambda\}$. Grammar:

$$\begin{aligned} \mathsf{V} &::= \mathsf{v} \mid \mathsf{V}' && \text{(Variables)} \\ \mathsf{C} &::= \mathsf{c} \mid \mathsf{C}' && \text{(Constants)} \\ \mathsf{T} &::= \mathsf{V} \mid \mathsf{C} \mid \beta\mathsf{TT} \mid \lambda\mathsf{VTT} && \text{(Terms)} \end{aligned}$$

- So we have countably infinite collections of variables $\mathsf{V} = \{\mathsf{v}, \mathsf{v}', \mathsf{v}'', \ldots\}$ and constants $\mathsf{C} = \{\mathsf{c}, \mathsf{c}', \mathsf{c}'', \ldots\}$.
- Usually we use metavariables, e.g. $x, y, z \in \mathsf{V}$ and $a, b, c \in \mathsf{C}$.
- Likewise we use metavariables for the terms, e.g. $R, S, T \in \mathsf{T}$.
- We can also write application βRS and abstraction λxRS terms with the standard syntactic sugar, i.e.

$$\begin{aligned} \beta\beta RST &=: RST \\ \beta R\beta ST &=: R(ST) \\ \lambda xRS &=: \lambda x : R.S \text{ or } (x : R) \mapsto S \end{aligned}$$

Interpretations (a la Aczel)

- First we assign sets to the variables and constants.

Definition

An assignment is a map $[\![\cdot]\!] : \mathsf{V} \cup \mathsf{C} \to \mathbf{Set}$.

- Denote the assignment changed just at $x \in \mathsf{V}$ to $s \in \mathbf{Set}$ by

$$[\![t]\!]_{\langle s,x\rangle} = \begin{cases} s & \text{if } t = x \\ [\![t]\!] & \text{otherwise} \end{cases}$$

- Extend $[\![\cdot]\!]$ to an interpretation on T by defining

$$[\![RS]\!] = [\![R]\!]([\![S]\!])$$
$$[\![\lambda xRS]\!] = \{\langle [\![S]\!]_{\langle r,x\rangle}, r\rangle \mid r \in [\![R]\!]\}$$

- If $[\![R]\!]$ is a function and $[\![S]\!] \in \mathrm{dom}[\![R]\!]$ then $[\![R]\!]([\![S]\!])$ is the value of $[\![R]\!]$ at $[\![S]\!]$. Otherwise set e.g. $[\![R]\!]([\![S]\!]) = \emptyset$.

Free Variables

Definition

Define $\mathsf{F} : \mathsf{T} \to \mathsf{V}$ mapping terms to their free variables by

$$\begin{aligned} \mathsf{F}(a) &= \emptyset \quad \text{when } a \in \mathsf{C} \\ \mathsf{F}(x) &= \{x\} \quad \text{when } x \in \mathsf{V} \\ \mathsf{F}(RS) &= \mathsf{F}(R) \cup \mathsf{F}(S) \\ \mathsf{F}(\lambda xRS) &= \mathsf{F}(R) \cup (\mathsf{F}(S) \setminus \{x\}) \end{aligned}$$

- What is the semantic intuition behind this? Free variables are the only ones that can possibly affect the interpretation.

Proposition

For any interpretation $[\![\cdot]\!]$, variable x, term R and set s,

$$x \notin \mathsf{F}(R) \qquad \Rightarrow \qquad [\![R]\!] = [\![R]\!]_{\langle s,x \rangle}$$

Substitution

Definition

Substitute a term T for a variable x by defining

$$
\begin{aligned}
x_{[T/x]} &:= T \\
u_{[T/x]} &:= u \quad \text{when } x \neq u \in \mathsf{V} \cup \mathsf{C} \\
(RS)_{[T/x]} &:= R_{[T/x]} S_{[T/x]} \\
(\lambda y R S)_{[T/x]} &:= \begin{cases} \lambda y R_{[T/x]} S & \text{if } y = x \text{ or } x \notin \mathsf{F}(S) \\ \lambda z R_{[T/x]} S_{[z/y][T/x]} & \text{otherwise, where } z \notin \mathsf{F}(\lambda y T S) \end{cases}
\end{aligned}
$$

- What is the semantic intuition here? Substitution is the syntactic equivalent of changing an interpretation at x to $[\![T]\!]$.

Proposition

For any interpretation $[\![\cdot]\!]$, variable x and terms S and T,

$$[\![S_{[T/x]}]\!] = [\![S]\!]_{\langle [\![T]\!], x \rangle}$$

β-Reduction

- Terms of the form $(\lambda x R S)\,T$ are said to β-reduce to $S_{[T/x]}$.
- Why? Reduced terms 'often' have the same interpretation.
- Define well-formed terms w.r.t. $[\![\cdot]\!]$, written $[\![\cdot]\!]^{\mathsf{wf}}$, by

$$
\begin{aligned}
[\![u]\!]^{\mathsf{wf}} \quad & \text{for all } u \in \mathsf{V} \cup \mathsf{C} \\
[\![RS]\!]^{\mathsf{wf}} \quad & \Leftrightarrow \quad [\![R]\!]^{\mathsf{wf}}, [\![S]\!]^{\mathsf{wf}},\ [\![R]\!] \in \textbf{Fun} \text{ and } [\![S]\!] \in \mathrm{dom}[\![R]\!], \\
[\![\lambda x R S]\!]^{\mathsf{wf}} \quad & \Leftrightarrow \quad [\![R]\!]^{\mathsf{wf}} \text{ and } \forall r \in [\![R]\!]\ ([\![S]\!]^{\mathsf{wf}}_{\langle r,x\rangle})
\end{aligned}
$$

Proposition

For any interpretation $[\![\cdot]\!]$, variable x and terms R, S and T,

$$
[\![(\lambda x R S)\,T]\!]^{\mathsf{wf}} \quad \Rightarrow \quad [\![S_{[T/x]}]\!]^{\mathsf{wf}} \text{ and } [\![(\lambda x R S)\,T]\!] = [\![S_{[T/x]}]\!]
$$

η-Subreduction

- If $x \notin \mathsf{F}(S)$ then $\lambda x R(Sx)$ is said to 'η-reduce' to S.
- Semantically, however this is really a case of 'subreduction'.
- Indeed $[\![\lambda x R(Sx)]\!]^{\mathsf{wf}}$ means $[\![Sx]\!]^{\mathsf{wf}}_{\langle r,x\rangle}$, for all $r \in [\![R]\!]$.
- Thus $[\![S]\!]^{\mathsf{wf}}$ and $[\![R]\!] \subseteq \mathrm{dom}[\![S]\!]$ but possibly $[\![R]\!] \neq \mathrm{dom}[\![S]\!]$.
- So $[\![\lambda x R(Sx)]\!]$ is only a subfunction of $[\![S]\!]$.

Proposition

For any interpretation $[\![\cdot]\!]$, $R, S \in \mathsf{T}$ and $x \in \mathsf{V} \setminus \mathsf{F}(S)$,

$$[\![\lambda x R(Sx)]\!]^{\mathsf{wf}} \quad \Rightarrow \quad [\![S]\!]^{\mathsf{wf}} \text{ and } [\![\lambda x R(Sx)]\!] \subseteq [\![S]\!]$$

General (Sub)Reductions

- General (sub)reductions may only apply when changing the interpretation of variables, not constants.
- E.g. they may only apply when certain constants are interpreted as fixed algebraic structures and operations.
- For any interpretation $[\![\cdot]\!]$ and any $\psi : \mathsf{V} \to \mathbf{Set}$, define

$$[\![a]\!]_\psi = [\![a]\!] \quad \text{if } a \in \mathsf{C}$$
$$[\![x]\!]_\psi = \psi(x) \quad \text{if } x \in \mathsf{V}$$

  and extend to an interpretation on all terms as before.
- We say R reduces to S in $[\![\cdot]\!]$ if, for all $\psi : \mathsf{V} \to \mathbf{Set}$,

$$[\![R]\!]^{\mathsf{wf}}_\psi \quad \Rightarrow \quad [\![S]\!]^{\mathsf{wf}}_\psi \text{ and } [\![R]\!]_\psi = [\![S]\!]_\psi$$

- We say R subreduces to S in $[\![\cdot]\!]$ if, for all $\psi : \mathsf{V} \to \mathbf{Set}$,

$$[\![R]\!]^{\mathsf{wf}}_\psi \quad \Rightarrow \quad [\![S]\!]^{\mathsf{wf}}_\psi \text{ and } [\![R]\!]_\psi \subseteq [\![S]\!]_\psi$$

Statements

- We can express such properties in terms of formal statements consisting of pairs of terms separated by some symbol like

$$\begin{aligned} \mathsf{T} \blacktriangleright \mathsf{T} &\qquad \text{(Reduction Statements)} \\ \mathsf{T} \triangleright \mathsf{T} &\qquad \text{(Subreduction Statements)} \\ \mathsf{T} : \mathsf{T} &\qquad \text{(Typing Statements)} \end{aligned}$$

- We say the interpretation $[\![\cdot]\!]$ satsifies the statement X or that the statement is valid in the interpretation, written $[\![X]\!]$, when

$$\begin{aligned} [\![R \blacktriangleright S]\!] &\Leftrightarrow R \text{ reduces to } S \text{ in } [\![\cdot]\!] \\ [\![R \triangleright S]\!] &\Leftrightarrow R \text{ subreduces to } S \text{ in } [\![\cdot]\!] \\ [\![R : S]\!] &\Leftrightarrow [\![R]\!]^{\mathsf{wf}}, [\![S]\!]^{\mathsf{wf}} \text{ and } [\![R]\!] \in [\![S]\!]. \end{aligned}$$

Valid Statements

- Previous observations show all interpretations satisfy

$$(\lambda x R S)\, T \blacktriangleright S_{[T/x]} \quad \text{and} \quad \lambda x R (S x) \triangleright S.$$

- Contextual Closure: If $[\![R \blacktriangleright S]\!]$ then also $[\![R' \blacktriangleright S']\!]$ where S' is obtained from R' by replacing any/all R terms with S terms.
- In contrast, $[\![R \triangleright S]\!]$ only implies $[\![RT \blacktriangleright ST]\!]$.
- Statements are also related as follows.

$$\begin{array}{ccccc} & & [\![R \blacktriangleright S]\!] & \Rightarrow & ([\![R : T]\!] \Rightarrow [\![S : T]\!]) \\ [\![R \blacktriangleright S]\!] & \Rightarrow & [\![R \triangleright S]\!] & \Rightarrow & ([\![T : R]\!] \Rightarrow [\![T : S]\!]) \end{array}$$

- Thus reduction and subreduction play the role of 'definitional equality' and 'subtyping' respectively.

Universes

- Modern type systems single out a sequence of constants $(\mathsf{u}_n) \subseteq \mathsf{C}$ to use as domains of polymorphic functions.
- As in previous work (e.g. by Carneiro), it would be natural to interpret these as larger and larger models of ZFC.
- But this presupposes the existence of inaccessible cardinals.
- To avoid this we consider sets $H(\kappa)$ of hereditary cardinality less than some infinite regular κ, which still model ZFC$-$P.
- This suffices for the vast majority of inductive specifications (Exception: Aczel's specification of ZFC itself).
- Specifically, take some (κ_n) increasing 'sufficiently fast' and require all interpretations to satisfy $[\![\mathsf{u}_n]\!] = H(\kappa_n)$.

Polymorphic Product Operators

- We can then single out further constants $(\mathsf{p}_m^n) \subseteq \mathsf{C}$ to use as polymorphic product operators, interpreting them as functions such that, for all $D \in [\![\mathsf{u}_m]\!]$ and $\phi : D \to [\![\mathsf{u}_n]\!]$,

$$[\![\mathsf{p}_m^n]\!](D)(\phi) = \prod_{d \in D} \phi(d)$$

- In other words, we require interpretations to satisfy

$$[\![\mathsf{p}_m^n]\!] = \{\langle \{ \langle \prod_{d \in D} \phi(d), \phi \rangle \mid \phi \in [\![\mathsf{u}_n]\!]^D \}, D \rangle \mid D \in [\![\mathsf{u}_m]\!] \}$$

- Can drop the indices when they clear from the context.
- Further introduce standard syntactic sugar:

$$\begin{aligned} \pi x R S &:= \mathsf{p} R \lambda x R S. \\ (x : R) \to S &:= \pi x R S. \\ R \to S &:= \pi x R S, \text{ where } x \notin \mathsf{F}(S). \end{aligned}$$

Typing Products

- As $[\![\mathsf{u}_n]\!] = H(\kappa_n)$ for increasing (κ_n),

$$[\![\mathsf{u}_n : \mathsf{u}_{n+1}]\!]$$

- As (κ_n) increases 'sufficiently fast',

$$[\![\mathsf{p}^n_m : (x : \mathsf{u}_m) \to (x \to \mathsf{u}_n) \to \mathsf{u}_{\max(m+1,n)}]\!]$$

- So now we do not need separate rules for product terms – these are already covered by typing statements.
- In particular, for products of 'propositions' u_0

$$[\![\mathsf{p}^0_m : (x : \mathsf{u}_m) \to (x \to \mathsf{u}_0) \to \mathsf{u}_{m+1}]\!]$$

- If (κ_n) are inaccessible then we can bump this up to

$$[\![\mathsf{p}^0_m : (x : \mathsf{u}_m) \to (x \to \mathsf{u}_0) \to \mathsf{u}_m]\!]$$

- However, our interpretations are NEVER 'impredicative', i.e. $[\![\cdot]\!]$ does not satisfy $\mathsf{p}^0_m : (x : \mathsf{u}_m) \to (x \to \mathsf{u}_0) \to \mathsf{u}_0$.
- '~~Polymorphism~~ Impredicativity is not set theoretic' (Reynolds).

Semantic Consequences

- We say a statement X is a consequence of some other statements Γ if X valid whenever every statement in Γ is.
- Define the consequence relation $\vDash$ on sets of statements by

$$\Gamma \vDash \Delta \quad \Leftrightarrow \quad \text{Every statement in } \Delta \text{ is a consequence of } \Gamma$$

- Then $\vDash$ is a sequent in that

$$X \in \Gamma \text{ or } \Gamma \vDash \Delta \vDash X \quad \Rightarrow \quad \Gamma \vDash X \qquad \text{(Sequent)}$$

- We can then prove that

$$(R \blacktriangleright S), (R : T) \vDash (S : T) \qquad \text{(Reduction)}$$
$$(R \triangleright S), (T : R) \vDash (T : S) \qquad \text{(Subreduction)}$$
$$(R : S), (G : \mathrm{p}SH) \vDash (GR : HR) \qquad \text{(Application)}$$

- If $x \notin \mathrm{F}(\Gamma) \cup \mathrm{F}(Q)$ and $\Gamma \vDash (Q : \mathrm{u}_m)$ then also

$$\Gamma, (x : Q) \vDash (R : S), (S : \mathrm{u}_n) \;\Rightarrow\; \Gamma \vDash (\lambda xQR : \pi^n_m xQS) \qquad \text{(Abstraction)}$$

Syntactic Inferences

Definition

The inference relation $\vdash$ is the smallest sequent satisfying

$$(R \blacktriangleright S), (R : T) \vdash (S : T) \qquad \text{(Reduction)}$$

$$(R \triangleright S), (T : R) \vdash (T : S) \qquad \text{(Subreduction)}$$

$$(R : S), (G : \mathrm{p}SH) \vdash (GR : HR) \qquad \text{(Application)}$$

$$\Gamma, (x : Q) \vdash (R : S), (S : \mathrm{u}_n) \;\Rightarrow\; \Gamma \vdash (\lambda xQR : \pi^n_m xQS) \qquad \text{(Abstraction)}$$

whenever $x \notin \mathrm{F}(\Gamma) \cup \mathrm{F}(Q)$ and $\Gamma \vdash (Q : \mathrm{u}_m)$.

- The inference relation is automatically sound, i.e. $\vdash \subseteq \models$.

$\Rightarrow$ This logical system is consistent, as long as ZFC is consistent.

- Close to first order logic – (Application) and (Abstraction) are analogous to modus ponens and universal generalisation.

$\Rightarrow$ Proofs in ZFC should translate analogously.

- Future work: get desirable syntactic properties like unique typing, type inference, normalisation, etc. by restricting the Γ we allow in inferences $\Gamma \vdash X$ to appropriate 'legal contexts'.

Specifications

- Inductive specifications can be viewed as collections of statements, e.g. for Cartesian products we can consider

$$
\begin{aligned}
&(\mathrm{pr}_{m,n} : (v : \mathrm{u}_m) \to (w : \mathrm{u}_n) \to \mathrm{u}_{m \vee n}),\\
&(\mathrm{mk}_{m,n} : (v : \mathrm{u}_m) \to (w : \mathrm{u}_n) \to v \to w \to \mathrm{pr}_{m,n} v w),\\
&(\mathrm{rec}^{I}_{m,n} : (v : \mathrm{u}_m) \to (w : \mathrm{u}_n) \to (f : \mathrm{pr}_{m,n} v w \to \mathrm{u}_I)\\
&\qquad \to ((x : v) \to (y : w) \to f(\mathrm{mk}_{m,n} v w x y))\\
&\qquad \to (z : \mathrm{pr}_{m,n} v w) \to f z) \quad \text{and}\\
&(\mathrm{rec}^{I}_{m,n} V W F G(\mathrm{mk}_{m,n} V W X Y) \blacktriangleright G X Y).
\end{aligned}
$$

- Valid when $[\![\mathrm{pr}]\!](s)(t) = s \times t$, $[\![\mathrm{mk}]\!](s)(t)(q)(r) = \langle q, r \rangle$ and $[\![\mathrm{rec}]\!](s)(t)(\phi)(\theta) = \theta^{\times}$ where $\theta^{\times}(q, r) = \theta(q)(r)$.
- As these statements Γ are valid in some interpretation, $\Gamma \not\models (P : \mathrm{False})$ and hence $\Gamma \not\vdash (P : \mathrm{False})$.
- Future work: Find a more systematic way of simultaneously validating general classes of inductive specifications.