

Proving with Computer Assistance

Lecture

Simple Type theory and *Formulas-as-Types* for propositional logic

Herman Geuvers

Typed λ calculus as the basis for a Proof Assistant

Typed λ calculus forms the basis for a variety of proof Assistants, e.g. Rocq (and Lean, Agda, Nuprl, Matita).

λ -term	type
program	specification
proof	formula

Integrated system for proving and programming

Types are not sets

Types are a bit like sets, but types give **syntactic information**, e.g.

$$3 + (7 \times 8)^5 : \text{nat}$$

whereas sets give **semantic information**, e.g.

$$3 \in \{n \in \mathbb{N} \mid \forall x, y, z \in \mathbb{N}^+ (x^n + y^n \neq z^n)\}.$$

- ▶ $3 + (7 \times 8)^5$ is of type `nat` because `3, 7, 8` are natural numbers and `×`, `+` and power are operations on natural numbers.
- ▶ $3 \in \{n \in \mathbb{N} \mid \forall x, y, z \in \mathbb{N}^+ (x^n + y^n \neq z^n)\}$ because there are no positive x, y, z such that $x^3 + y^3 = z^3$, which is an instance of **Fermat's last Theorem**, proved by Wiles.
- ▶ To establish that `3` is an element of the given set, we need a **proof**, we can't just read it off from the components of the statement.
- ▶ To establish $3 + (7 \times 8)^5 : \text{nat}$ we don't need a proof but a simple **computation** (the "reading the type of of the term").

Decidability of $:$, undecidability of \in

- ▶ Membership is undecidable in set theory, as it requires a proof to establish $a \in A$.
- ▶ Type checking is decidable: Verifying whether M is of type A requires purely syntactic methods, which can be cast into a typing algorithm.

$$3 + (7 \times 8)^5 : \text{nat} \quad \text{versus} \quad \frac{1}{2} \sum_{n=0}^{\infty} 2^{-n} \in \mathbb{N}$$

Question: Can we turn (e.g.)

$$\{n \in \mathbb{N} \mid \forall x, y, z \in \mathbb{N}^+ (x^n + y^n \neq z^n)\}$$

into a (syntactic) type, with decidable type checking?

Phrased differently: can we talk about this set as a “subtype of nat ”?

Formulas are also types; proofs are terms

$$\{n \in \text{nat} \mid \forall x, y, z \in \mathbb{N}^+(x^n + y^n \neq z^n)\}$$

is a type.

Its terms are **pairs** $\langle n, p \rangle$ where

- ▶ $n : \text{nat}$
- ▶ $p : \forall x, y, z \in \mathbb{N}^+(x^n + y^n \neq z^n)$

So p is a proof, and we view the formula

$\forall x, y, z \in \mathbb{N}^+(x^n + y^n \neq z^n)$ as the **type of its proofs**.

If we have **decidable proof checking**, then it is decidable whether a given pair $\langle n, p \rangle$ is typable with the above type or not.

We summarize:

- ▶ proof checking = type checking,
- ▶ type checking is decidable (so proof checking is decidable),
- ▶ proof finding is not decidable (proof finding is required to check an \in -judgment).

Simple Type Theory

Simplest system: $\lambda \rightarrow$ or **simple type theory**, STT. Just **arrow types**

$$\text{Typ} := \text{TVar} \mid (\text{Typ} \rightarrow \text{Typ})$$

- ▶ Examples: $(\alpha \rightarrow \beta) \rightarrow \alpha$, $(\alpha \rightarrow \beta) \rightarrow ((\beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \gamma))$
- ▶ Brackets associate to the right and outside brackets are omitted:
 $(\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \alpha \rightarrow \gamma$
- ▶ Types are denoted by σ, τ, \dots

Terms:

- ▶ typed variables $x_1^\sigma, x_2^\sigma, \dots$, countably many for every σ .
- ▶ application: if $M : \sigma \rightarrow \tau$ and $N : \sigma$, then $(MN) : \tau$
- ▶ abstraction: if $P : \tau$, then $(\lambda x^\sigma. P) : \sigma \rightarrow \tau$

Examples of simply typed terms

$$\begin{aligned}\lambda x^\sigma . \lambda y^\tau . x & : \sigma \rightarrow \tau \rightarrow \sigma \\ \lambda x^{\alpha \rightarrow \beta} . \lambda y^{\beta \rightarrow \gamma} . \lambda z^\alpha . y(xz) & : (\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \alpha \rightarrow \gamma \\ \lambda x^\alpha . \lambda y^{(\beta \rightarrow \alpha) \rightarrow \alpha} . y(\lambda z^\beta . x) & : \alpha \rightarrow ((\beta \rightarrow \alpha) \rightarrow \alpha) \rightarrow \alpha\end{aligned}$$

For every type there is a term of that type:

$$x^\sigma : \sigma$$

Not for every type there is a **closed term** of that type:

$$(\alpha \rightarrow \alpha) \rightarrow \alpha \text{ is not } \mathbf{inhabited}$$

[That is: there is no closed term of type $(\alpha \rightarrow \alpha) \rightarrow \alpha$.]

Church' simple type theory

Church formulation of simple type theory: terms with type information.

Inductive definition of the terms:

- ▶ typed variables $x_1^\sigma, x_2^\sigma, \dots$, countably many for every σ .
- ▶ application: if $M : \sigma \rightarrow \tau$ and $N : \sigma$, then $(MN) : \tau$
- ▶ abstraction: if $P : \tau$, then $(\lambda x^\sigma. P) : \sigma \rightarrow \tau$

Alternative: Inductive definition of the terms in rule form:

$$\frac{}{x^\sigma : \sigma} \qquad \frac{M : \sigma \rightarrow \tau \quad N : \sigma}{MN : \tau} \qquad \frac{P : \tau}{\lambda x^\sigma. P : \sigma \rightarrow \tau}$$

Advantage: We also have a derivation tree, a proof of the fact that the term has that type.

We can reason over derivations.

Simple type theory à la Church with contexts

Formulation with **contexts** to declare the free variables:

$$x_1 : \sigma_1, x_2 : \sigma_2, \dots, x_n : \sigma_n$$

is a **context**, usually denoted by Γ .

Derivation rules of $\lambda \rightarrow$ (à la Church):

$$\frac{x:\sigma \in \Gamma}{\Gamma \vdash x : \sigma} \quad \frac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} \quad \frac{\Gamma, x:\sigma \vdash P : \tau}{\Gamma \vdash \lambda x:\sigma. P : \sigma \rightarrow \tau}$$

$\Gamma \vdash_{\lambda \rightarrow} M : \sigma$ if there is a derivation using these rules with conclusion $\Gamma \vdash M : \sigma$

Reading the typing rules top down

Inductive definition of the “derivable judgments”

$$\frac{x:\sigma \in \Gamma}{\Gamma \vdash x : \sigma} \quad \frac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} \quad \frac{\Gamma, x:\sigma \vdash P : \tau}{\Gamma \vdash \lambda x:\sigma. P : \sigma \rightarrow \tau}$$

Deriving

$$\vdash \lambda x:\alpha. \lambda y:(\beta \rightarrow \alpha) \rightarrow \alpha. y(\lambda z:\beta. x) : \alpha \rightarrow ((\beta \rightarrow \alpha) \rightarrow \alpha) \rightarrow \alpha$$

Reading the typing rules bottom up

Trying to solve a typing problem / an inhabitation problem

$$\frac{x:\sigma \in \Gamma}{\Gamma \vdash x : \sigma}$$

$$\frac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

$$\frac{\Gamma, x:\sigma \vdash P : \tau}{\Gamma \vdash \lambda x:\sigma. P : \sigma \rightarrow \tau}$$

Formulas-as-Types (Curry, Howard)

There are **two readings** of a judgement $M : \sigma$

1. term as **algorithm/program**, type as **specification**:
 M is a function of type σ

2. type as a **proposition**, term as its **proof**:
 M is a proof of the proposition σ

► There is a **one-to-one correspondence**:

typable terms in $\lambda \rightarrow \simeq$ **derivations** in minimal proposition
logic

► $x_1 : \tau_1, x_2 : \tau_2, \dots, x_n : \tau_n \vdash M : \sigma$ can be read as
 M is a **proof** of σ from the **assumptions** $\tau_1, \tau_2, \dots, \tau_n$.

Example

$$\frac{\frac{\frac{[\alpha \rightarrow \beta \rightarrow \gamma]^3 \quad [\alpha]^1}{\beta \rightarrow \gamma} \quad \frac{[\alpha \rightarrow \beta]^2 \quad [\alpha]^1}{\beta}}{\frac{\gamma}{\alpha \rightarrow \gamma} \quad 1} \quad 2}{(\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma} \quad 3}{(\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma}$$

\approx

$$\lambda x: \alpha \rightarrow \beta \rightarrow \gamma. \lambda y: \alpha \rightarrow \beta. \lambda z: \alpha. xz(yz) \\ : (\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma$$

Example

$$\frac{\frac{\frac{[x : \alpha \rightarrow \beta \rightarrow \gamma]^3 [z : \alpha]^1}{xz : \beta \rightarrow \gamma}}{\frac{[y : \alpha \rightarrow \beta]^2 [z : \alpha]^1}{yz : \beta}}}{\frac{xz(yz) : \gamma}{\frac{\lambda z : \alpha. xz(yz) : \alpha \rightarrow \gamma}{\lambda y : \alpha \rightarrow \beta. \lambda z : \alpha. xz(yz) : (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma}}}{\lambda x : \alpha \rightarrow \beta \rightarrow \gamma. \lambda y : \alpha \rightarrow \beta. \lambda z : \alpha. xz(yz) : (\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma} \quad 3$$

Exercise: Give the derivation that corresponds to

$$\lambda x : \gamma \rightarrow \varepsilon. \lambda y : (\gamma \rightarrow \varepsilon) \rightarrow \varepsilon. y(\lambda z. y x) : (\gamma \rightarrow \varepsilon) \rightarrow ((\gamma \rightarrow \varepsilon) \rightarrow \varepsilon) \rightarrow \varepsilon$$

Flag style deductions

The **Fitch** style (also: **flag** style) presentation of $\lambda \rightarrow$.

1			$x : \sigma$		
2			...		
3			...		
4			$M : \tau$		
5			$\lambda x : \sigma. M : \sigma \rightarrow \tau$	abs, 1, 4	
					abs-rule
1			...		
2			...		
3			$M : \sigma \rightarrow \tau$		
4			...		
5			...		
6			$N : \sigma$		
7			...		
8			$M N : \tau$	app, 3, 6	
					app-rule

Example

1		$x : \alpha \rightarrow \beta \rightarrow \gamma$
2		<hr/> $y : \alpha \rightarrow \beta$
3		$z : \alpha$
4		<hr/> $x z : \beta \rightarrow \gamma$
5		$y z : \beta$
6		$x z (y z) : \gamma$
7		$\lambda z : \alpha. x z (y z) : \alpha \rightarrow \gamma$
8		$\lambda y : \alpha \rightarrow \beta. \lambda z : \alpha. x z (y z) : (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma$
9		$\lambda x : \alpha \rightarrow \beta \rightarrow \gamma. \lambda y : \alpha \rightarrow \beta. \lambda z : \alpha. x z (y z) : (\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma$

Computation

► **β -reduction**: $(\lambda x:\sigma.M)P \rightarrow_{\beta} M[x := P]$

Cut-elimination

Cut-elimination in minimal logic = β -reduction in $\lambda \rightarrow$.

$$\frac{\frac{\frac{[\sigma]^1}{\mathcal{D}_1} \quad \tau}{\sigma \rightarrow \tau} \quad 1 \quad \frac{\mathcal{D}_2}{\sigma}}{\tau}}{\tau} \quad \longrightarrow \quad \frac{\mathcal{D}_2}{\sigma} \quad \frac{\mathcal{D}_1}{\tau}$$
$$\frac{\frac{\frac{[x : \sigma]^1}{\mathcal{D}_1} \quad M : \tau}{\lambda x : \sigma . M : \sigma \rightarrow \tau} \quad 1 \quad \frac{\mathcal{D}_2}{P : \sigma}}{(\lambda x : \sigma . M) P : \tau}}{\longrightarrow_{\beta}} \quad \frac{\mathcal{D}_2}{P : \sigma} \quad \frac{\mathcal{D}_1}{M[x := P] : \tau}$$

Example

Proof of $A \rightarrow A \rightarrow B, (A \rightarrow B) \rightarrow A \vdash B$

$$\frac{\frac{\frac{[A]^1 \quad A \rightarrow A \rightarrow B}{A \rightarrow B}}{B}}{A \rightarrow B} \quad \frac{\frac{\frac{[A]^1 \quad A \rightarrow A \rightarrow B}{A \rightarrow B}}{B}}{A \rightarrow B}}{(A \rightarrow B) \rightarrow A}}{B}$$

It contains a cut: a \rightarrow -i directly followed by an \rightarrow -e.

Example ctd

Proof of $A \rightarrow A \rightarrow B, (A \rightarrow B) \rightarrow A \vdash B$ after reduction

$$\begin{array}{c}
 \frac{\frac{\frac{[A]^1 \quad A \rightarrow A \rightarrow B}{A \rightarrow B}}{B}}{(A \rightarrow B) \rightarrow A \quad A \rightarrow B} \quad A \\
 \frac{\frac{\frac{[A]^1 \quad A \rightarrow A \rightarrow B}{A \rightarrow B}}{B}}{(A \rightarrow B) \rightarrow A \quad A \rightarrow B} \quad A \quad \frac{\frac{\frac{[A]^1 \quad A \rightarrow A \rightarrow B}{A \rightarrow B}}{B}}{(A \rightarrow B) \rightarrow A \quad A \rightarrow B} \quad A \rightarrow B}{A \rightarrow B} \quad A \rightarrow A \rightarrow B \\
 \hline
 B
 \end{array}$$

Example ctd

Proof of $A \rightarrow A \rightarrow B, (A \rightarrow B) \rightarrow A \vdash B$ with term information.

$$\frac{\frac{\frac{[y : A]^1 \quad p : A \rightarrow A \rightarrow B}{p y : A \rightarrow B}}{p y y : B} \quad \frac{\frac{[x : A]^1 \quad p : A \rightarrow A \rightarrow B}{p x : A \rightarrow B}}{p x x : B}}{q : (A \rightarrow B) \rightarrow A \quad \lambda x : A. p x x : A \rightarrow B}}{\lambda y : A. p y y : A \rightarrow B \quad q(\lambda x : A. p x x) : A}}{\lambda y : A. p y y)(q(\lambda x : A. p x x)) : B}$$

Term contains a β -redex: $(\lambda x : A. p x x)(q(\lambda x : A. p x x))$

Example ctd

Reduced proof of $A \rightarrow A \rightarrow B, (A \rightarrow B) \rightarrow A \vdash B$ with term info.

$$\frac{\frac{\frac{[x : A]^1 \quad p : A \rightarrow A \rightarrow B}{p x : A \rightarrow B}}{p x x : B} \quad q : (A \rightarrow B) \rightarrow A \quad \lambda x : A. p x x : A \rightarrow B}{q(\lambda x : A. p x x) : A} \quad \frac{\frac{\frac{[x : A]^1 \quad p : A \rightarrow A \rightarrow B}{p x : A \rightarrow B}}{p x x : B} \quad q : (A \rightarrow B) \rightarrow A \quad \lambda x : A. p x x : A \rightarrow B}{q(\lambda x : A. p x x) : A} \quad p : A \rightarrow A \rightarrow B}{p(q(\lambda x : A. p x x))(q(\lambda x : A. p x x)) : B}$$

Extension with other connectives: STT with **product types** \times
(proposition logic with **conjunction** \wedge)

$$\frac{\Gamma \vdash M : \sigma \times \tau}{\Gamma \vdash \pi_1 M : \sigma}$$

$$\frac{\Gamma \vdash M : \sigma \times \tau}{\Gamma \vdash \pi_2 M : \tau}$$

$$\frac{\Gamma \vdash P : \sigma \quad \Gamma \vdash Q : \tau}{\Gamma \vdash \langle P, Q \rangle : \sigma \times \tau}$$

With reduction rules

$$\pi_1 \langle P, Q \rangle \rightarrow P$$

$$\pi_2 \langle P, Q \rangle \rightarrow Q$$