Syntax Directed Type Checking for Pure Type Systems

Markus Klinik

Radboud University Nijmegen

Type Theory and Proof Assistants 2012

Outline

What?

- What is Type Checking?
- What Does “Syntax Directed” Mean?
- What are PTSs?

Syntax Directed Type Checking for PTSs

- The Plan
- The Problem
- The Solution

What is Type Checking?

- Type Checking: Given Γ, M and A, decide whether the judgement $\Gamma \vdash M : A$ is derivable
- Type Synthesis: Given Γ and M, compute an A such that $\Gamma \vdash M : A$
- We need a type system
  - Syntax
  - Typing rules

What Does "Syntax Directed" Mean?

$$M ::= x \mid MM \mid \lambda x.M$$

$$\begin{aligned} FV(x) &= \{x\} \\ FV(MN) &= FV(M) \cup FV(N) \\ FV(\lambda x.M) &= FV(M) \backslash \{x\} \end{aligned}$$

A Syntax Directed Function

```
data Term
  = Var Char
  | App Term Term
  | Abs Char Term

freeVars :: Term -> [Char]
freeVars (Var x)   = [x]
freeVars (App m n) = freeVars m `union` freeVars n
freeVars (Abs x m) = freeVars m \\ [x]
```

What are PTSs?

- Generalized type systems
  - Identical syntax
  - Parametrized typing rules
- PTSs: Berardi (1988), Terlouw (1989)
- λ-cube: Barendregt (1991)
- λ-zoo $\rightarrow$ λ-cube $\rightarrow$ PTSs

The Ingredients to Generalized Typing

- Introduce *dependent product types*
- One syntactic category for terms and types
- Introduce *sorts*

The λ-Cube: Ingredients

$$M ::= x \mid \star \mid \Box \mid MM \mid \lambda x : M \,.\, M \mid \Pi x : M \,.\, M$$

$$\frac{M : (\Pi x : \sigma \,.\, \tau(x)) \qquad N : \sigma}{MN : \tau(N)}$$

$$\frac{\text{zeros} : (\Pi n : \text{nat} \,.\, \text{vec } n) \qquad m : \text{nat}}{\text{zeros } m : \text{vec } m}$$

$$\frac{\lambda n : \text{nat} \,.\, Sn : (\Pi n : \text{nat} \,.\, \text{nat}) \qquad m : \text{nat}}{(\lambda n : \text{nat} \,.\, Sn) m : \text{nat}}$$

λ-Cube Typing Rules

$$\text{(axiom)}\ \frac{}{\Gamma \vdash \star : \Box} \qquad \text{(start)}\ \frac{\Gamma \vdash A : s \quad x \notin \Gamma}{\Gamma, x : A \vdash x : A}$$

$$\text{(weakening)}\ \frac{\Gamma \vdash M : B \quad \Gamma \vdash A : s \quad x \notin \Gamma}{\Gamma, x : A \vdash M : B}$$

$$\text{(application)}\ \frac{\Gamma \vdash M : (\Pi x : A \,.\, B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : [N/x]B}$$

$$\text{(abstraction)}\ \frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash (\Pi x : A \,.\, B) : s \quad x \notin \Gamma}{\Gamma \vdash (\lambda x : A \,.\, M) : (\Pi x : A \,.\, B)}$$

$$\text{(product)}\ \frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2 \quad x \notin \Gamma}{\Gamma \vdash (\Pi x : A \,.\, B) : s_2}$$

$$\text{(conversion)}\ \frac{\Gamma \vdash M : A \quad A =_\beta B \quad \Gamma \vdash B : s}{\Gamma \vdash M : B}$$

λ-Cube Rules: Axiom, Start

$$\text{(axiom)}\ \frac{}{\Gamma \vdash \star : \Box}$$

$$\text{(start)}\ \frac{\Gamma \vdash A : s \qquad x \notin \Gamma}{\Gamma, x : A \vdash x : A}$$

λ-Cube Rules: Weakening

$$\text{(start)}\ \frac{\Gamma \vdash A : s \quad x \notin \Gamma}{\Gamma, x : A \vdash x : A}$$

$$\text{(weakening)}\ \frac{\Gamma \vdash M : B \quad \Gamma \vdash A : s \quad x \notin \Gamma}{\Gamma, x : A \vdash M : B}$$

λ-Cube Rules: Application, Abstraction

$$\text{(application)}\ \frac{\Gamma \vdash M : (\Pi x : A \,.\, B) \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : [N/x]B}$$

$$\text{(abstraction)}\ \frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash (\Pi x : A \,.\, B) : s \qquad x \notin \Gamma}{\Gamma \vdash (\lambda x : A \,.\, M) : (\Pi x : A \,.\, B)}$$

λ-Cube Rules: Product

$$\text{(abstraction)}\ \frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash (\Pi x : A \,.\, B) : s \qquad x \notin \Gamma}{\Gamma \vdash (\lambda x : A \,.\, M) : (\Pi x : A \,.\, B)}$$

$$\text{(product)}\ \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2 \qquad x \notin \Gamma}{\Gamma \vdash (\Pi x : A \,.\, B) : s_2}$$

λ-Cube Rules: Conversion

$$\text{(application)}\ \frac{\Gamma \vdash M : (\Pi x : A \,.\, B) \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : [N/x]B}$$

$$\text{(conversion)}\ \frac{\Gamma \vdash M : A \qquad A =_\beta B \qquad \Gamma \vdash B : s}{\Gamma \vdash M : B}$$

λ-Cube Product Rule Again

$$\text{(product)}\ \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2 \qquad x \notin \Gamma}{\Gamma \vdash (\Pi x : A \,.\, B) : s_2}$$

$(\star, \star)$	terms depending on terms	$\lambda n : \mathrm{nat} \,.\, S n$
$(\Box, \star)$	terms depending on types	$\lambda \alpha : \star \,.\, \lambda x : \alpha \,.\, x$
$(\star, \Box)$	types depending on terms	$\Pi n : \mathrm{nat} \,.\, \mathrm{vec}\ n$
$(\Box, \Box)$	types depending on types	$\Pi \alpha : \star \,.\, \alpha \to \alpha$

λ-Cube Wrap Up

A type system in the λ-cube consists of:

- The set of pseudoterms
- The set of sorts $\mathcal{S} = \{\star, \Box\}$
- A set $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$ of *product rules*
- The typing rules

And Now For Something Slightly Different

A pure type system consists of:

- The set of pseudoterms
- An arbitrary set $\mathcal{S}$ of sorts
- A set $\mathcal{A} \subseteq \mathcal{S} \times \mathcal{S}$ of axioms
- A set $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S}$ of product rules
- The typing rules

$$\text{(axiom)}\ \frac{\mathcal{A}(s_1, s_2)}{\Gamma \vdash s_1 : s_2}$$

$$\text{(product)}\ \frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2 \quad \mathcal{R}(s_1, s_2, s_3) \quad x \notin \Gamma}{\Gamma \vdash (\Pi x : A\,.\,B) : s_3}$$

The Plan

$$\text{(product)}\ \frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2 \quad \mathcal{R}(s_1, s_2, s_3) \quad x \notin \Gamma}{\Gamma \vdash (\Pi x : A \,.\, B) : s_3}$$

```
data Term
  = Prod Char Term Term
  ...

typeSynthesis :: Env -> Term -> Maybe Term
typeSynthesis gamma (Prod x A B) =
  let s1 = typeSynthesis gamma A
      s2 = typeSynthesis (gamma ++ [(x,A)]) B
  in
    if (rel s1 s2 s3) then s3 else Nothing
```

The Problem

$$\text{(conversion)}\ \frac{\Gamma \vdash M : A \quad A =_\beta B \quad \Gamma \vdash B : s}{\Gamma \vdash M : B}$$

```
...
typeSynthesis gamma M           = ...
...
typeSynthesis gamma (App M N)   = ...
typeSynthesis gamma (Abs x A B) = ...
...
typeSynthesis gamma M           = ...
```

The Concern

$$\text{(product)}\ \frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2 \quad \mathcal{R}(s_1, s_2, s_3) \quad x \notin \Gamma}{\Gamma \vdash (\Pi x : A \,.\, B) : s_3}$$

- Derivations are trees
- Context correctness is checked on every branch

The Solution

- Product rule and axioms might be ambiguous
  - Require $\mathcal{A}$ and $\mathcal{R}$ to be *functional* i.e. right-unique
  - $\mathcal{R}(s_1, s_2, s_3)$ becomes
    `s3 = R(s1, s2) ::  Maybe Term`
- Computations on type level
  - Require strong normalization
- Contexts are checked repeatedly
  - Keep track of context correctness
- Weakening and Conversion rules not syntax directed
  - Absorb into other rules

PTS Rules Again

$$(\text{axiom})\ \frac{\mathcal{A}(s_1, s_2)}{\Gamma \vdash s_1 : s_2} \qquad (\text{start})\ \frac{\Gamma \vdash A : s \qquad x \notin \Gamma}{\Gamma, x : A \vdash x : A}$$

$$(\text{weakening})\ \frac{\Gamma \vdash M : B \qquad \Gamma \vdash A : s \qquad x \notin \Gamma}{\Gamma, x : A \vdash M : B}$$

$$(\text{product})\ \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2 \qquad \mathcal{R}(s_1, s_2, s_3) \qquad x \notin \Gamma}{\Gamma \vdash (\Pi x : A \,.\, B) : s_3}$$

$$(\text{abstraction})\ \frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash (\Pi x : A \,.\, B) : s \qquad x \notin \Gamma}{\Gamma \vdash (\lambda x : A \,.\, M) : (\Pi x : A \,.\, B)}$$

$$(\text{application})\ \frac{\Gamma \vdash M : (\Pi x : A \,.\, B) \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : [N/x]B}$$

$$(\text{conversion})\ \frac{\Gamma \vdash M : A \qquad A =_\beta B \qquad \Gamma \vdash B : s}{\Gamma \vdash M : B}$$

Maintaining Context Validity

$$\text{(valid-nil)}\ \frac{}{\vdash_{vc} \emptyset} \qquad \text{(valid-cons)}\ \frac{\Gamma \vdash_{vt} A : s \quad x \notin \Gamma}{\vdash_{vc} \Gamma, x : A}$$

$$\text{(axiom)}\ \frac{\vdash_{vc} \Gamma \quad \mathcal{A}(s_1, s_2)}{\Gamma \vdash_{vt} s_1 : s_2} \qquad \text{(var)}\ \frac{\vdash_{vc} \Gamma \quad x : A \in \Gamma}{\Gamma \vdash_{vt} x : A}$$

$$\text{(product)}\ \frac{\Gamma \vdash_{vt} A : s_1 \quad \Gamma, x : A \vdash_{vt} B : s_2 \quad \mathcal{R}(s_1, s_2, s_3) \quad x \notin \Gamma}{\Gamma \vdash_{vt} (\Pi x : A \,.\, B) : s_3}$$

$$\text{(abstraction)}\ \frac{\Gamma, x : A \vdash_{vt} M : B \quad \Gamma \vdash_{vt} (\Pi x : A \,.\, B) : s \quad x \notin \Gamma}{\Gamma \vdash_{vt} (\lambda x : A \,.\, M) : (\Pi x : A \,.\, B)}$$

$$\text{(application)}\ \frac{\Gamma \vdash_{vt} M : (\Pi x : A \,.\, B) \quad \Gamma \vdash_{vt} N : A}{\Gamma \vdash_{vt} MN : [N/x]B}$$

$$\text{(conversion)}\ \frac{\Gamma \vdash_{vt} M : A \quad A =_\beta B \quad \Gamma \vdash_{vt} B : s}{\Gamma \vdash_{vt} M : B}$$

Maintaining Local Context Validity

$$\text{(valid-nil)}\ \frac{}{\vdash_{lvc} \emptyset} \qquad \text{(valid-cons)}\ \frac{\Gamma \vdash_{lvt} A : s \quad x \notin \Gamma}{\vdash_{lvc} \Gamma, x : A}$$

$$\text{(axiom)}\ \frac{\mathcal{A}(s_1, s_2)}{\Gamma \vdash_{lvt} s_1 : s_2} \qquad \text{(var)}\ \frac{x : A \in \Gamma}{\Gamma \vdash_{lvt} x : A}$$

$$\text{(product)}\ \frac{\Gamma \vdash_{lvt} A : s_1 \quad \Gamma, x : A \vdash_{lvt} B : s_2 \quad \mathcal{R}(s_1, s_2, s_3) \quad x \notin \Gamma}{\Gamma \vdash_{lvt} (\Pi x : A . B) : s_3}$$

$$\text{(abstraction)}\ \frac{\Gamma, x : A \vdash_{lvt} M : B \quad \Gamma \vdash_{lvt} (\Pi x : A . B) : s \quad x \notin \Gamma}{\Gamma \vdash_{lvt} (\lambda x : A . M) : (\Pi x : A . B)}$$

$$\text{(application)}\ \frac{\Gamma \vdash_{lvt} M : (\Pi x : A . B) \quad \Gamma \vdash_{lvt} N : A}{\Gamma \vdash_{lvt} MN : [N/x]B}$$

$$\text{(conversion)}\ \frac{\Gamma \vdash_{lvt} M : A \quad A =_\beta B \quad \Gamma \vdash_{lvt} B : s}{\Gamma \vdash_{lvt} M : B}$$

The Conversion Rule

$$\text{(conversion)}\ \frac{\Gamma \vdash M : A \quad A =_\beta B \quad \Gamma \vdash B : s}{\Gamma \vdash M : B}$$

$$\text{(application)}\ \frac{\Gamma \vdash_{lvt} M : (\Pi x : A \,.\, B) \quad \Gamma \vdash_{lvt} N : A}{\Gamma \vdash_{lvt} MN : [N/x]B}$$

$$\text{(application)}\ \frac{\Gamma \vdash M :\twoheadrightarrow \Pi x : A \,.\, B \quad \Gamma \vdash N : A' \quad A =_\beta A'}{\Gamma \vdash MN : [N/x]B}$$

Notation: We write $\Gamma \vdash M :\twoheadrightarrow A$ for $(\Gamma \vdash M : X \wedge X \twoheadrightarrow A)$

Weak head normal form: top level is not a redex

Finally Syntax Directed

$$\text{(valid-nil)}\ \frac{}{\vdash \emptyset} \qquad \text{(valid-cons)}\ \frac{\Gamma \vdash A :\twoheadrightarrow s \quad x \notin \Gamma}{\vdash \Gamma, x : A}$$

$$\text{(axiom)}\ \frac{\mathcal{A}(s_1, s_2)}{\Gamma \vdash s_1 : s_2} \qquad \text{(var)}\ \frac{x : A \in \Gamma}{\Gamma \vdash x : A}$$

$$\text{(product)}\ \frac{\Gamma \vdash A :\twoheadrightarrow s_1 \quad \Gamma, x : A \vdash B :\twoheadrightarrow s_2 \quad \mathcal{R}(s_1, s_2, s_3) \quad x \notin \Gamma}{\Gamma \vdash (\Pi x : A \,.\, B) : s_3}$$

$$\text{(abstraction)}\ \frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash (\Pi x : A \,.\, B) : s \quad x \notin \Gamma}{\Gamma \vdash (\lambda x : A \,.\, M) : (\Pi x : A \,.\, B)}$$

$$\text{(application)}\ \frac{\Gamma \vdash M :\twoheadrightarrow \Pi x : A \,.\, B \quad \Gamma \vdash N : A' \quad A =_\beta A'}{\Gamma \vdash MN : [N/x]B}$$

Questions