Herman Geuvers

Type Theory, Fall 2009
Lecture Meta theory and Decidability of Type Checking

Pure Type Systems

Determined by a triple $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ with

- $\mathcal{S}$ the set of sorts
- $\mathcal{A}$ the set of axioms, $\mathcal{A} \subseteq \mathcal{S} \times \mathcal{S}$
- $\mathcal{R}$ the set of rules, $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S}$

If $s_2 = s_3$ in $(s_1, s_2, s_3) \in \mathcal{R}$, we write $(s_1, s_2) \in \mathcal{R}$.

pseudoterms:

$$\mathsf{T} ::= \mathcal{S} \mid \mathsf{Var} \mid (\Pi \mathsf{Var}{:}\mathsf{T}.\mathsf{T}) \mid (\lambda \mathsf{Var}{:}\mathsf{T}.\mathsf{T}) \mid \mathsf{T}\mathsf{T}.$$

(sort) $\vdash s_1 : s_2$ if $(s_1, s_2) \in \mathcal{A}$ (var) $\dfrac{\Gamma \vdash A : s}{\Gamma, x{:}A \vdash x : A}$ if $x \notin \Gamma$

(weak) $\dfrac{\Gamma \vdash A : s \quad \Gamma \vdash M : C}{\Gamma, x{:}A \vdash M : C}$ if $x \notin \Gamma$

(Π) $\dfrac{\Gamma \vdash A : s_1 \quad \Gamma, x{:}A \vdash B : s_2}{\Gamma \vdash \Pi x{:}A.B : s_3}$ if $(s_1, s_2, s_3) \in \mathcal{R}$

(λ) $\dfrac{\Gamma, x{:}A \vdash M : B \quad \Gamma \vdash \Pi x{:}A.B : s}{\Gamma \vdash \lambda x{:}A.M : \Pi x{:}A.B}$

(app) $\dfrac{\Gamma \vdash M : \Pi x{:}A.B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B[N/x]}$

(conv$_\beta$) $\dfrac{\Gamma \vdash M : A \quad \Gamma \vdash B : s}{\Gamma \vdash M : B}$ $A =_\beta B$

Examples of PTSs

CC	
$\mathcal{S}$	Prop, Type
$\mathcal{A}$	Prop : Type
$\mathcal{R}$	(Prop, Prop), (Prop, Type), (Type, Prop), (Type, Type)

λPREDω	
$\mathcal{S}$	Set, Type^s, Prop, Type
$\mathcal{A}$	Set : Type^s, Prop : Type
$\mathcal{R}$	(Set, Set), (Set, Type), (Type, Type), (Prop, Prop), (Set, Prop), (Type, Prop)

λP	
$\mathcal{S}$	Prop, Type
$\mathcal{A}$	Prop : Type
$\mathcal{R}$	(Prop, Prop), (Type, Prop)

$\lambda\star$	
$\mathcal{S}$	$\star$
$\mathcal{A}$	$\star : \star$
$\mathcal{R}$	$(\star, \star)$

A PTS-morphism from $\lambda(\mathcal{S}, \mathcal{A}, \mathcal{R})$ to $\lambda(\mathcal{S}', \mathcal{A}', \mathcal{R}')$ is an $f : \mathcal{S} \to \mathcal{S}'$ that preserves the axioms and rules:

- if $(s_1, s_2) \in \mathcal{A}$ then $(f(s_1), f(s_2)) \in \mathcal{A}'$
- if $(s_1, s_2, s_3) \in \mathcal{R}$ then $(f(s_1), f(s_2), f(s_3)) \in \mathcal{R}'$

f extends to pseudoterms and contexts and we have

Proposition

If $\Gamma \vdash M : A$ then $f(\Gamma) \vdash f(M) : f(A)$

Properties of PTSs.

- Uniqueness of types
  If $\Gamma \vdash M : A$ and $\Gamma \vdash M : B$, then $A =_\beta B$.
  Holds if $\mathcal{A} \subseteq \mathcal{S} \times \mathcal{S}$ and $\mathcal{R} \subseteq (\mathcal{S} \times \mathcal{S}) \times \mathcal{S}$ are functions.
  Definition A PTS where $\mathcal{A}$ and $\mathcal{R}$ are functions is called a functional PTS (or singly sorted PTS).

- Subject Reduction
  If $\Gamma \vdash M : A$ and $M \longrightarrow_\beta N$, then $\Gamma \vdash N : A$.

- Strengthening
  If $\Gamma, x : B, \Delta \vdash M : A$ and $x \notin \mathsf{FV}(M, A, \Delta)$, then $\Gamma, \Delta \vdash M : A$.

Basic Properties of PTSs.

- Stripping or Generation
  - If $\Gamma \vdash x : A$, then $x : B \in \Gamma$ for some B with $B =_\beta A$.
  - If $\Gamma \vdash M\,N : A$, then $\Gamma \vdash M : \Pi x{:}B.C$ and $\Gamma \vdash N : B$ for some B, C, x with $C[x := N] =_\beta A$.
  - If $\Gamma \vdash \lambda x{:}B.M : A$, then $\Gamma, x{:}B \vdash M : C$ and $\Gamma \vdash \Pi x{:}B.C : s$ for some C, s with $\Pi x{:}B.C =_\beta A$.
  - If $\Gamma \vdash \Pi x{:}B.C : A$, then $\Gamma, x{:}B \vdash C : s_2$ and $\Gamma \vdash B : s_1$ for some $(s_1, s_2, s_3) \in \mathcal{R}$ with $s_3 =_\beta A$.

- Type correctness property

  If $\Gamma \vdash M : A$, then $A \equiv s$ or $\Gamma \vdash A : s$ for some $s \in \mathcal{S}$.

Properties of PTSs ctd.

- Substitution property
  If $\Gamma, x : B, \Delta \vdash M : A$, $\Gamma \vdash P : B$, then
  $\Gamma, \Delta[x := P] \vdash M[x := P] : A[x := P]$.

- Thinning
  If $\Gamma \vdash M : A$ and $\Gamma \subseteq \Delta$, Δ well-formed, then $\Delta \vdash M : A$.

Strong Normalization (SN)
If $\Gamma \vdash M : A$, then all β-reductions from M terminate.

SN holds for some PTSs, and for some not.
SN for CC is proved by a higher order extension of the saturated sets argument (for $\lambda 2$).

Some more examples of PTSs

CC^∞	
$\mathcal{S}$	Prop, $\{\text{Type}_i\}_{i\in\mathbb{N}}$
$\mathcal{A}$	Prop : Type, Type_i : Type_{i+1}
$\mathcal{R}$	$(\text{Prop}, \text{Prop}), (\text{Prop}, \text{Type}_i), (\text{Type}_i, \text{Prop})$
	$(\text{Type}_i, \text{Type}_j, \text{Type}_{\max(i,j)})$

Note that $(\text{Type}_1, \text{Type}_0, \text{Type}_0)$ is inconsistent (λU)
Similarly $(\text{Type}_{i+1}\text{Type}_i, \text{Type}_i))$ would be inconsistent.

First order Dependent Type theory, λP, adapted formulation

Derive judgements of the form

$$\Gamma \vdash M : B$$

- Γ is a context
- M and B are terms
  taken from the set of pseudoterms

$$\mathsf{T} ::= \mathsf{Var} \,|\, \mathbf{type} \,|\, \mathbf{kind} \,|\, (\mathsf{T}\mathsf{T}) \,|\, (\lambda x{:}\mathsf{T}.\mathsf{T}) \,|\, \Pi x{:}\mathsf{T}.\mathsf{T},$$

Auxiliary judgement

$$\Gamma \vdash$$

denoting that Γ is a correct context.

Derivation rules of λP

s ranges over $\{\mathbf{type}, \mathbf{kind}\}$.

(base) $\emptyset \vdash$ (ctxt) $\dfrac{\Gamma \vdash A : \mathbf{s}}{\Gamma, x{:}A \vdash}$ if x not in Γ (ax) $\dfrac{\Gamma \vdash}{\Gamma \vdash \mathbf{type} : \mathbf{kind}}$

(proj) $\dfrac{\Gamma \vdash}{\Gamma \vdash x : A}$ if $x{:}A \in \Gamma$ (Π) $\dfrac{\Gamma, x{:}A \vdash B : \mathbf{s} \quad \Gamma \vdash A : \mathbf{type}}{\Gamma \vdash \Pi x{:}A.B : \mathbf{s}}$

(λ) $\dfrac{\Gamma, x{:}A \vdash M : B \quad \Gamma \vdash \Pi x{:}A.B : \mathbf{s}}{\Gamma \vdash \lambda x{:}A.M : \Pi x{:}A.B}$ (app) $\dfrac{\Gamma \vdash M : \Pi x{:}A.B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]}$

(conv) $\dfrac{\Gamma \vdash M : B \quad \Gamma \vdash A : \mathbf{s}}{\Gamma \vdash M : A} A =_{\beta} B$

Notation: write $A{\to}B$ for $\Pi x{:}A.B$ if $x \notin \mathsf{FV}(B)$.

Equivalence of 'new' λP with PTS definition of λP

- If $\Gamma \vdash^n M : A$, then $\Gamma \vdash M : A$
- If $\Gamma \vdash^n$, then $\Gamma \vdash \mathbf{type} : \mathbf{kind}$
- If $\Gamma \vdash M : A$, then $\Gamma \vdash^n M : A$

Properties of λP

- Uniqueness of types
  If $\Gamma \vdash M : \sigma$ and $\Gamma \vdash M : \tau$, then $\sigma =_\beta \tau$.

- Subject Reduction
  If $\Gamma \vdash M : \sigma$ and $M \longrightarrow_\beta N$, then $\Gamma \vdash N : \sigma$.

- Strong Normalization
  If $\Gamma \vdash M : \sigma$, then all β-reductions from M terminate.

Proof of SN is by defining a reduction preserving map from λP to $\lambda\!\rightarrow$.

Decidability Questions

$$\Gamma \vdash M : \sigma? \quad \text{TCP}$$
$$\Gamma \vdash M : ? \quad \text{TSP}$$
$$\Gamma \vdash ? : \sigma \quad \text{TIP}$$

For λP:

- TIP is undecidable
- TCP/TSP: simultaneously with Context checking

Type Checking

Define algorithms $\mathrm{Ok}(-)$ and $\mathrm{Type}_{\_}(-)$ simultaneously:

- $\mathrm{Ok}(-)$ takes a context and returns 'true' or 'false'
- $\mathrm{Type}_{\_}(-)$ takes a context and a term and returns a term or 'false'.

The type synthesis algorithm $\mathrm{Type}_{\_}(-)$ is sound if

$$\mathrm{Type}_{\Gamma}(M) = A \quad \Rightarrow \quad \Gamma \vdash M : A$$

for all Γ and M.

The type synthesis algorithm $\mathrm{Type}_{\_}(-)$ is complete if

$$\Gamma \vdash M : A \quad \Rightarrow \quad \mathrm{Type}_{\Gamma}(M) =_{\beta} A$$

for all Γ, M and A.

$$\mathrm{Ok}(<>) \quad = \quad \text{‘true’}$$

$$\mathrm{Ok}(\Gamma, x{:}A) \quad = \quad \mathrm{Type}_\Gamma(A) \in \{\mathbf{type}, \mathbf{kind}\},$$

$$\mathrm{Type}_\Gamma(x) \quad = \quad \text{if } \mathrm{Ok}(\Gamma) \text{ and } x{:}A \in \Gamma \text{ then } A \text{ else ‘false’},$$

$$\mathrm{Type}_\Gamma(\mathbf{type}) \quad = \quad \text{if } \mathrm{Ok}(\Gamma) \text{then } \mathbf{kind} \text{ else ‘false’},$$

$$\begin{aligned}
\mathrm{Type}_\Gamma(MN) \quad = \quad & \text{if } \mathrm{Type}_\Gamma(M) = C \text{ and } \mathrm{Type}_\Gamma(N) = D \\
& \text{then} \quad \text{if } C \twoheadrightarrow_\beta \Pi x{:}A.B \text{ and } A =_\beta D \\
& \qquad\quad \text{then } B[x := N] \text{ else ‘false’} \\
& \text{else} \quad \text{‘false’},
\end{aligned}$$

$$
\begin{array}{rcl}
\mathrm{Type}_\Gamma(\lambda x{:}A.M) & = & \text{if } \mathrm{Type}_{\Gamma,x:A}(M) = B \\
& & \quad \text{then} \qquad \text{if } \mathrm{Type}_\Gamma(\Pi x{:}A.B) \in \{\mathbf{type}, \mathbf{kind}\} \\
& & \qquad\qquad\quad \text{then } \Pi x{:}A.B \text{ else 'false'} \\
& & \quad \text{else 'false'}, \\
\mathrm{Type}_\Gamma(\Pi x{:}A.B) & = & \text{if } \mathrm{Type}_\Gamma(A) = \mathbf{type} \text{ and } \mathrm{Type}_{\Gamma,x:A}(B) = s \\
& & \text{then } s \text{ else 'false'}
\end{array}
$$

Soundness

$$\text{Type}_\Gamma(M) = A \;\;\Rightarrow\;\; \Gamma \vdash M : A$$

Completeness

$$\Gamma \vdash M : A \;\;\Rightarrow\;\; \text{Type}_\Gamma(M) =_\beta A$$

As a consequence:

$$\text{Type}_\Gamma(M) = \text{'false'} \;\;\Rightarrow\;\; M \text{ is not typable in } \Gamma$$

NB 1. Completeness only makes sense if we have uniqueness of types (Otherwise: let $\text{Type}_{\_}(-)$ generate a set of possible types)

NB 2. Completeness implies that Type terminates on all well-typed terms. We want that Type terminates on all pseudo terms.

Termination

We want $\text{Type}_{\_}(-)$ to terminate on all inputs.
Interesting cases: λ-abstraction and application:

$$\begin{array}{rcl} \text{Type}_\Gamma(\lambda x{:}A.M) & = & \text{if } \text{Type}_{\Gamma,x:A}(M) = B \\ & & \text{then} \quad \text{if } \text{Type}_\Gamma(\Pi x{:}A.B) \in \{\mathbf{type}, \mathbf{kind}\} \\ & & \qquad\quad \text{then } \Pi x{:}A.B \text{ else 'false'} \\ & & \text{else 'false'}, \end{array}$$

! Recursive call is not on a smaller term!

Replace the side condition

$$\text{if } \text{Type}_\Gamma(\Pi x{:}A.B) \in \{\mathbf{type}, \mathbf{kind}\}$$

by

$$\text{if } \text{Type}_\Gamma(A) \in \{\mathbf{type}\}$$

Termination

We want $\mathrm{Type}_{\_}(-)$ to terminate on all inputs.
Interesting cases: λ-abstraction and application:

$$\begin{array}{rcl}\mathrm{Type}_\Gamma(MN) & = & \text{if } \mathrm{Type}_\Gamma(M) = C \text{ and } \mathrm{Type}_\Gamma(N) = D \\ & & \text{then} \quad \text{if } C \twoheadrightarrow_\beta \Pi x{:}A.B \text{ and } A =_\beta D \\ & & \qquad\quad \text{then } B[x := N] \text{ else 'false'} \\ & & \text{else} \quad \text{'false'},\end{array}$$

! Need to decide β-reduction and β-equality!

For this case, termination follows from soundness of Type and the decidability of equality on well-typed terms (using SN and CR).