TYPE SETUPS for Predicate Logic

Peter Aczel
Manchester

(with some help from Joao Belo - MSc thesis)

Types 2004
Paris
December 2004

Type Setups – Motivation 1

- There are advantages to <u>logic-enriched</u> type theories where predicate logic is simply added on top of a type theory.

  This allows the freedom to interprete or reinterprete the logic in different ways; e.g.

  - props-as-types
  - props as objects of type Prop (in CC)
  - sheaf reinterpretations
  - $\neg\neg$ reinterpretation

    ⋮

- What is needed from a type theory in order to logic-enrich it?

  Answer : A type setup

Type Setups - Motivation 2

- We want to generalise <u>many-sorted logic</u> to <u>dependently-sorted logic</u> (DS-logic)
- The notion of a <u>type setup</u> captures the needed properties of <u>sorts</u>, <u>terms</u> and <u>atomic formulae</u> for DS-logic
- The details of how expressions are formed using a <u>signature</u> of <u>sort/term/prop-constructors</u> is abstracted away
- We keep variables and contexts of variable declarations as in dependent type theory.
- We axiomatize how substitutions work.

$$\begin{tikzcd}
\text{a type theory with predicate symbols} \arrow[dr] & & \text{a signature of sort/term/prop constructors} \arrow[dl] \\
& \text{a type setup} \arrow[d] & \\
& \text{a ?-category with atomic props} &
\end{tikzcd}$$

$$\text{?-category} = \begin{cases} \text{category with families} \\ \text{contextual category} \\ \text{category with attributes} \\ \text{type category} \\ \vdots \end{cases}$$

Many roughly equivalent category theoretic notions with many authors.

Some References

J. Cartmell (Ph.D thesis, 1978
APAL, 1986)
Generalised Algebraic Theories and Contextual categories.

P. Dybjer (Types, 1995)
Internal Type Theory
(categories with families)

M. Makkai (unpublished? monograph, 1995)
FOLDS
(DS-logic without function symbols)

A type setup for DS-logic

consists of

- a category of contexts Γ and substitution maps $\sigma : \Delta \to \Gamma$,
- for each context Γ
  - a set $Ty(\Gamma)$ of Γ-types
  - a set $Tm(\Gamma, A)$ of Γ-terms of type A for each Γ-type A
  - a set $At(\Gamma)$ of atomic Γ-formulae
- for each substitution map $\sigma : \Delta \to \Gamma$ substition operations

$$A \in Ty(\Gamma) \longmapsto A\sigma \in Ty(\Delta)$$
$$a \in Tm(\Gamma, A) \longmapsto a\sigma \in Tm(\Delta, A\sigma)$$
$$\theta \in At(\Gamma) \longmapsto \theta\sigma \in At(\Delta)$$

These must satisfy TS1 – TS4

TSI : Substitution Action

- For each identity $\mathrm{id}_\Gamma : \Gamma \to \Gamma$

$$A\,\mathrm{id}_\Gamma = A \qquad (A \in \mathrm{Ty}(\Gamma))$$
$$a\,\mathrm{id}_\Gamma = a \qquad (a \in \mathrm{Tm}(\Gamma, A))$$
$$\theta\,\mathrm{id}_\Gamma = \theta \qquad (\theta \in \mathrm{At}(\Gamma))$$

- For maps $\sigma : \Delta \to \Gamma$, $\tau : \Lambda \to \Delta$

$$A(\sigma \circ \tau) = (A\sigma)\tau \qquad (A \in \mathrm{Ty}(\Gamma))$$
$$a(\sigma \circ \tau) = (a\sigma)\tau \qquad (a \in \mathrm{Tm}(\Gamma, A))$$
$$\theta(\sigma \circ \tau) = (\theta\sigma)\tau \qquad (\theta \in \mathrm{At}(\Gamma))$$

TS2 : Contexts

- Contexts Γ have the form

$$x_1 : A_1, \ldots, x_n : A_n$$

where

$x_1, \ldots, x_n$ are distinct variables

$$A_i \in Ty(\underbrace{x_1 : A_1, \ldots, x_{i-1} : A_{i-1}}_{\Gamma_{<i}}) \quad (i = 1, \ldots, n)$$

- Then

$$A_i \in Ty(\Gamma) \qquad (i = 1, \ldots, n)$$
$$x_i \in Tm(\Gamma, A_i)$$

TS3 : Substitution Maps

- Substitution maps $\sigma : \Delta \to \Gamma$ have the form

$$[x_1 := a_1, \ldots, x_n := a_n]_{\Delta \to \Gamma}$$

where

$$a_i \in Tm(\Delta, A_i[x_1 := a_1, \ldots, x_{i-1} := a_{i-1}]_{\Delta \to \Gamma_{<i}})$$

- Then

$$x_i \sigma = a_i \qquad (i = 1, \ldots, n)$$

and is the unique such map.

TS4: Weakening

- $\Gamma \subseteq \Delta$ if every declaration in Γ is in Δ

- If $\Gamma \subseteq \Delta$ then

$$Ty(\Gamma) \subseteq Ty(\Delta)$$
$$Tm(\Gamma, A) \subseteq Tm(\Delta, A) \qquad (A \in Ty(\Gamma))$$
$$At(\Gamma) \subseteq At(\Delta)$$

and

$$A\pi = A \qquad (A \in Ty(\Gamma))$$
$$a\pi = a \qquad (a \in Tm(\Gamma, A))$$
$$\theta\pi = \theta \qquad (\theta \in At(\Gamma))$$

when

$$\pi = [x_1 := x_1, \ldots, x_n := x_n]_{\Delta \to \Gamma} : \Delta \to \Gamma$$

where Γ is $(x_1 : A_1, \ldots, x_n : A_n)$

DS-logic over a type setup

- Γ- formulae φ

$$\varphi ::= \theta \mid \bot \mid (\varphi_1 \to \varphi_2) \mid (\forall x : A)\varphi_0$$

θ atomic Γ- formula
φ_1, φ_2 Γ- formulae
φ_0 $(\Gamma, x : A)$- formula

- Sequents $(\Gamma)\ \Phi \Rightarrow \varphi$

Φ list of Γ- formulae

- Inference Rules
standard for classical logic

e.g.

$$\frac{(\Gamma, x : A)\ \Phi \Rightarrow \varphi_0}{(\Gamma)\ \Phi \Rightarrow (\forall x : A)\varphi_0} \qquad \frac{(\Gamma)\ \Phi \Rightarrow (\forall x : A)\varphi_0}{(\Gamma)\ \Phi \Rightarrow \varphi_0[a/x]}$$

a is a Γ-term of type A

$$[a/x] = [x_1 := x_1, \ldots, x_n := x_n, x := a]_{\Gamma \to (\Gamma, x : A)}$$

$$: \Gamma \to (\Gamma, x : A)$$

where Γ is $(x_1 : A_1, \ldots, x_n : A_n)$

Semantics for DS-logic

- An interpretation $\mathcal{A}$ for a type setup consists of, for each context Γ
  - a set $[\![\Gamma]\!]^{\mathcal{A}}$ of tuples $\vec{u}$
  - a function $[\![\sigma]\!]^{\mathcal{A}} : [\![\Delta]\!]^{\mathcal{A}} \to [\![\Gamma]\!]^{\mathcal{A}}$ for each $\sigma : \Delta \to \Gamma$
  - for each $\vec{u} \in [\![\Gamma]\!]^{\mathcal{A}}$

$$\begin{cases} \text{a set } [\![A]\!]^{\mathcal{A}}_{\Gamma}(\vec{u}) & (A \in Ty(\Gamma)) \\ [\![a]\!]^{\mathcal{A}}_{\Gamma,A}(\vec{u}) \in [\![A]\!]^{\mathcal{A}}_{\Gamma}(\vec{u}) & (a \in Tm(\Gamma, A)) \\ [\![\theta]\!]^{\mathcal{A}}_{\Gamma} \subseteq [\![\Gamma]\!]^{\mathcal{A}} & (\theta \in At(\Gamma)) \end{cases}$$

satisfying . . .

- We can define "$\mathcal{A} \models_{\Gamma} \varphi[\vec{u}]$" as usual

e.g. $\mathcal{A} \models_{\Gamma} \theta[\vec{u}]$ iff $\vec{u} \in [\![\theta]\!]^{\mathcal{A}}_{\Gamma}$

⋮

$\mathcal{A} \models_{\Gamma} (\forall x : A)\varphi_0[\vec{u}]$ iff for all $v \in [\![A]\!]^{\mathcal{A}}_{\Gamma}(\vec{u})$ $\mathcal{A} \models_{\Gamma, x:A} \varphi[\vec{u}, v]$

III

Soundness-Completeness Theorem

$\Sigma \vdash \varphi$ iff $()\,\Phi \Rightarrow \varphi$ can be proved for some list Φ of sentences from Σ

$\Sigma \models \varphi$ iff $(\forall \psi \in \Sigma)(\mathcal{A} \models_{()} \psi[\,]) \Rightarrow \mathcal{A} \models_{()} \varphi[\,]$ for every interpretation $\mathcal{A}$

<u>Theorem</u> $\Sigma \vdash \varphi$ iff $\Sigma \models \varphi$

<u>Proof</u>: As usual e.g. use Henkin method via Extension Lemma:

If Σ is a consistent set of sentences of a type setup $\mathbb{T}$ and $(\exists x : A)\,\varphi_0 \in \Sigma$ then $\mathbb{T}$ can be extended to a type setup $\mathbb{T}[c:A]$ with c a new closed term of type A so that $\Sigma \cup \{\varphi_0[c/x]\}$ is a consistent set of sentences of $\mathbb{T}[c:A]$.

Some Signatures for DS-logic

- the notion of a category

$$
\begin{array}{l}
Obj \ \textbf{sort} \\
Hom(a,b : Obj) \ \textbf{sort} \\
Id(a : Obj) : Hom(a,a) \\
Comp(a,b,c : Obj, \\
\qquad f : Hom(b,c), \\
\qquad g : Hom(a,b) \\
\qquad) : Hom(a,c)
\end{array}
$$

- an unusual, but coherent, signature

1) $F(y : F(c))$ **sort**
2) $c : F(c)$

We can derive

i) $\Rightarrow c : F(c)$ by 2)

ii) $\Rightarrow F(c)$ **sort** by 1) and i)

iii) $y : F(c) \Rightarrow F(y)$ **sort** by 1) and ii)

1) is coherent by iii)

2) is coherent by ii)