Logical Frameworks

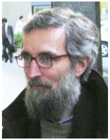
Mark Price

University of Bath, UK

Workshop on Deep Inference
Paris
1st-2nd December 2006

Judgements

Immanuel Kant (1800)
Introduces the notion of judgement



Per Martin-Löf (1983)
Develops Kant's notion of judgement to hypothetico-general judgement

Examples

Hypothetical Judgement:

$$\mathsf{J} \vdash \mathsf{K}$$

General Judgement:

$$\Lambda_{x \in \mathsf{C}} \mathsf{K}(x)$$

Hypothetico-General Judgement:

$$\Lambda_{x \in \mathsf{C}} \mathsf{J}(x) \vdash \mathsf{K}$$

Logical Framework

- Logical frameworks formalize Martin-Löf's notion of hypothetico-general judgement
- 'Logical Framework = Language + Representation'
- LF $= \lambda\Pi +$ judgements-as-types

$\lambda\Pi$-calculus

- $\lambda\Pi$ is a first-order dependent type theory with signature Σ
- $\lambda\Pi$ has three levels:
  - Objects;
  - Types (which classify objects);
  - Kinds (which classify families of types).
- Terms:
  - $\lambda x : A . M : \Pi x : A . B$;
  - $\lambda x : A . B : \Pi x : A . K$.

Judgements-as-types

Judgements can have the form:

$$\mathsf{true} : o \longrightarrow \mathrm{Type}$$

Inference rule:

$$\color{red}{\prod \phi_i} \quad \color{black}{\dfrac{\begin{matrix} \vdots \color{red}{\cup} \\ \mathsf{j}_1(\phi_1) \end{matrix} \qquad \begin{matrix} \color{red}{\supset} \\ \cdots \end{matrix} \qquad \begin{matrix} \vdots \color{red}{\cup} \\ \mathsf{j}_n(\phi_n) \end{matrix}}{\mathsf{j}(\phi)}} \ \color{red}{\cup}$$

Example

$$\dfrac{\begin{matrix}\vdots & & \vdots \\ \phi & & \phi \supset \psi\end{matrix}}{\psi}\, MP$$

Encoded as:

$$MP : \Pi_{\phi,\psi\,:\,o}\,.\,\mathsf{true}(\phi) \longrightarrow \mathsf{true}(\phi \supset \psi) \longrightarrow \mathsf{true}(\psi)$$

Instantiate:

$$(MP)(p)(q) : \mathsf{true}(p) \longrightarrow \mathsf{true}(p \supset q) \longrightarrow \mathsf{true}(q)$$

Given proofs Φ and Ψ, we have:

$$(MP)(p)(q)(\Phi)(\Psi) : \mathsf{true}(q)$$

Judgements-as-types Correspondence

Given a derivation Φ:

$$(X) \quad \mathsf{j}_1(\phi_1), \ldots, \mathsf{j}_n(\phi_n) \vdash_{\mathcal{O}_T} \mathsf{j}(\phi)$$

this corresponds to a derivation

$$\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \ldots, x_n : \mathsf{j}_n(\phi_n) \vdash_{\Sigma_{\mathcal{O}_T}} M_\Phi : \mathsf{j}(\phi)$$

in $\lambda\Pi$

Representation Theorem 1

The above example illustrated the correspondence between the derivation of a proof (term) in $\lambda\Pi$ and a proof in first order natural deduction system.

Theorem (Harper, Honsell and Plotkin (87))

For every proof

$$\Phi : (X) \quad \mathsf{true}(\phi_1), \ldots, \mathsf{true}(\phi_n) \vdash_{ND} \mathsf{true}(\phi)$$

in first-order natural deduction, there exists a (proof) term M_Φ *such that*

$$\Gamma_X, x_1 : \mathsf{true}(\phi_1), \ldots, x_n : \mathsf{true}(\phi_n) \vdash_{\Sigma_{ND}} M_\Phi : \mathsf{true}(\phi)$$

is a derivation in $\lambda\Pi$.

Representation Theorem 2

- The previous theorem only showed a correspondence between proofs of a first-order natural deduction system and (proof) terms of $\lambda\Pi$.
- The correspondence between (proof) terms of $\lambda\Pi$ and proofs in the object-logic is also of interest.
- We have the following theorem:

Theorem (Harper, Honsell and Plotkin (87))

For every canonical term of type $\mathsf{true}(\phi)$ *in* $\lambda\Pi$*, with derivation*

$$\Gamma_X, x_1{:}\mathsf{true}(\phi_1), \ldots, x_n{:}\mathsf{true}(\phi_n) \vdash_{\Sigma_{ND}} M_\Phi{:}\mathsf{true}(\phi)$$

there is a proof

$$\Phi{:}(X) \quad \mathsf{true}(\phi_1), \ldots, \mathsf{true}(\phi_n) \vdash_{ND} \mathsf{true}(\phi)$$

of ϕ *in first-order natural deduction.*

Dealing with Modalities

Judgements:

$$\mathsf{valid} : o \longrightarrow \mathrm{Type} \qquad \mathsf{true} : o \longrightarrow \mathrm{Type}$$

Natural Deduction rules:

$$\frac{\mathsf{valid}(\phi)}{\mathsf{valid}(\Box\phi)} \, \Box \, I$$

Encoded rules:

$$\Box \, I : \Pi_{\phi \,:\, o} \, . \, \mathsf{valid}(\phi) \longrightarrow \mathsf{valid}(\Box\phi)$$

Representation Theorem for $S4$

Theorem (Avron, Honsell and Mason (87))

There is a surjection between proofs of ϕ

$$\Phi : (X) \quad \mathsf{valid}(\phi_1), \ldots, \mathsf{valid}(\phi_n) \vdash_{S4} \mathsf{valid}(\phi)$$

and derivations of (proof) terms M_Φ

$$\Gamma_X, x_1 : \mathsf{valid}(\phi_1), \ldots, \mathsf{valid}(\phi_n) \vdash_{\Sigma_{S4}} M_\Phi : \mathsf{valid}(\phi)$$

- Here we do not have a bijection between proofs in $S4$ and (proof) terms in $\lambda\Pi$. We only have a surjection.

Towards a Theory of Representation 1

- Proving that a proof Φ in the object-logic corresponds to a derivation of a (proof) term M_Φ in $\lambda\Pi$ is 'easy' proof-theoretically.
- Proving that a derivation of a (proof) term M_Φ in $\lambda\Pi$ corresponds to a proof Φ in the object-logic is 'hard' proof-theoretically.
- Proofs usually involve analysing normal forms for derivations in the framework and have been done in an 'ad hoc' way.
- A semantic approach to these proofs is more natural (and 'easier').
- This approach will provide a uniform methodology to obtain these results.

Towards a Theory of Representation 2

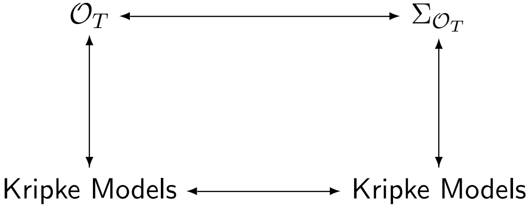
- Want a semantic framework which will provide a uniform way to obtain representation results.
- We sketch the development of this framework.
- Semantic models need to capture consequence.
- LF captures judged consequence, *i.e.* hypothetico-general judgements.
- Thus we always work in judged proof systems, $\mathcal{O}_T$.

Towards a Theory of Representation 3

- We need a model over both the object-logic and the encoded logic.
- The encoding should induce a morphism between these models.
- This means our models must live in the same category.
- Our models are (strict) indexed categories.
- They are 'in-between' the contextual categories of Cartmell (86) and their reformulation by Pitts (00).

Towards a Theory of Representation 4

This leads to the following diagram:



Kripke Models 1

- We use a slightly unusual notion of Kripke partiality.
- We usually think of a proof being defined at all worlds in the Kripke model.
- Here the world tells us which proofs are defined at that world.
- Thus not all proofs are defined at every world.

Kripke Models 2

- We exploit the constructive nature of $\lambda\Pi$.
- The construction of a (proof) term $M_\Phi : \mathsf{j}(\phi)$ at a world w corresponds to having a proof for $\mathsf{j}(\phi)$ in the object-logic.
- We interpret

$$[\![\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \ldots, x_n : \mathsf{j}_n(\phi_n) \vdash_{\Sigma_{\mathcal{O}_T}} M_\Phi : \mathsf{j}(\phi)]\!]^w$$

  as follows.
- The worlds in the Kripke model tell us which x's get interpreted. The interpretation of x_i corresponds to a proof of $\mathsf{j}_i(\phi_i)$ and thus the interpretation of $[\![x_i : \mathsf{j}_i(\phi)_i]\!]^{w_i}$.
- Thus at the world where all the x's get interpreted (and hence we have proofs of all the $\mathsf{j}_i(\phi_i)$'s), we get

$$[\![\Gamma_X, x_1 : \mathsf{j}_1(\phi_1), \ldots, x_n : \mathsf{j}_n(\phi_n) \vdash_{\Sigma_{\mathcal{O}_T}} M_\Phi : \mathsf{j}(\phi)]\!]^w$$

  interpreted and thus the proof of $\mathsf{j}(\phi)$ defined.

Definability

- With this notion of Kripke partiality in mind, we talk about when proofs are defined in our model.
- This naturally leads to the question being formulated in terms of logical relations.
- Have defined the correct notion of logical relation for the $\lambda\Pi$-calculus and proved the fundamental lemma.
- This has not yet been done for a judged object-logic.

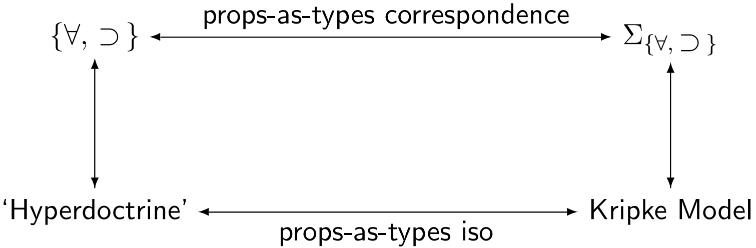
Main Result so far

- We take our object-logic as $\{\forall, \supset\}$-fragment of minimal first-order logic.
- This is the internal logic of $\lambda\Pi$, *i.e.* it is in propositions-as-types correspondence with $\lambda\Pi$.
- We use the propositions-as-types correspondence as our representation mechanism.
- This means that the encoding function is a bijection.
- This representation induces a morphism between the respective Kripke models.
- This morphism is an isomorphism, called the propositions-as-types isomorphism.

The Propositions-as-types Isomorphism

Theorem

Representing the $\{\forall, \supset\}$-fragment of minimal first-order logic with the propositions-as-types correspondence induces an isomorphism between the respective Kripke models.



Generalizing the Propositions-as-types Isomorphism

- The judgements-as-types correspondence is a generalization of the proposition-as-types correspondence.
- The morphism between models that the judgements-as-types correspondence induces is in general not an isomorphism.
- Morphism is an epimorphism in general, this corresponds to the notion of a uniform encoding, *i.e.* a surjective encoding function.

Uniform Encoding

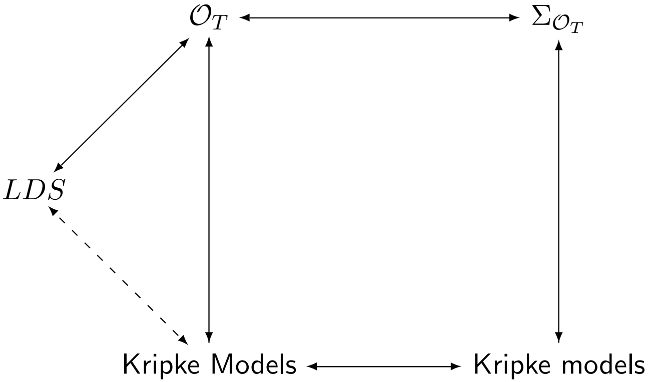
- In general, we want our encoding to be uniform.
- This means that the encoding function is surjective.
- The surjectivity means that each (proof) term M_Φ in $\lambda\Pi$ has come from a proof ϕ in the object-logic.
- This is the minimum condition with which we have something useful to work with.
- A uniform encoding will induce an epimorphism between the respective Kripke models.

Generating Judged Proof Systems

- We wish to prove representation results for classes of object-logics in a uniform way.
- Thus we need to have classes of specific object-logics.
- To obtain these classes of object-logics, we use labelled deductive systems.
- We use labelled deductive systems since they were developed with this rôle in mind and they provide a uniform characterization of a wide class of non-classical logics.

Introducing Labelled Deductive Systems

We extend our diagram to include labelled deductive systems.



Ongoing Work

- Prove representation theorems for classes of object-logics in a uniform way using the semantic framework just outlined.
- Study the effect of encoding on normalization.
- View this work in terms of logic programming. The encoded rules are essentially Horn Clauses. Can proof procedures in the object-logic be lifted to the encoded logic and exploited more generally?