LF$_{\mathcal{P}}$ – A Logical Framework with External Predicates

Petar Maksimović

in collaboration with

Furio Honsell, Marina Lenisa, Luigi Liquori, and Ivan Scagnetto

Mathematical Institute of the Serbian Academy of Sciences and Arts, Serbia
Faculty of Technical Sciences, University of Novi Sad, Serbia
INRIA Sophia Antipolis Méditerranée, France
Università di Udine, Italy

Fourth Workshop on Formal and Automated Theorem Proving and Applications, February 2-3, 2012, Belgrade, Serbia

Briefly about LF

The Harper-Honsell-Plotkin Logical Framework

- LF – a logical framework based on the $\lambda\Pi$-calculus
- Dependent types - types depending on terms
- Based on the Curry-Howard isomorphism
- Basis for the proof assistant Twelf

The ideas behind $LF_{\mathcal{P}}$

The main ideas

- Develop a way of easily and smoothly encoding logics with arbitrary structural side-conditions in LF,
- Separate derivation from verification/computation,
- Increase modularity, and
- Optimize performance.

The pseudo-syntax of LF$_{\mathcal{P}}$

$$
\begin{array}{rclcrcll}
\Sigma & \in & \mathcal{S} & & \Sigma & ::= & \emptyset \mid \Sigma, a{:}K \mid \Sigma, c{:}\sigma & \textit{Signatures} \\
\Gamma & \in & \mathcal{C} & & \Gamma & ::= & \emptyset \mid \Gamma, x{:}\sigma & \textit{Contexts} \\
K & \in & \mathcal{K} & & K & ::= & \mathrm{Type} \mid \Pi x{:}\sigma.K & \textit{Kinds} \\
\sigma, \tau, \rho & \in & \mathcal{F} & & \sigma & ::= & a \mid \Pi x{:}\sigma.\tau \mid \sigma\, N \mid \mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho] & \textit{Families} \\
M, N & \in & \mathcal{O} & & M & ::= & c \mid x \mid \lambda x{:}\sigma.M \mid M\, N \mid & \\
 & & & & & & \mid \mathcal{L}^{\mathcal{P}}_{N,\sigma}[M] \mid \mathcal{U}^{\mathcal{P}}_{N,\sigma}[M] & \textit{Objects}
\end{array}
$$

Figure: The pseudo-syntax of LF$_{\mathcal{P}}$

So, what is new?

- Predicates on derivable typing judgements $\mathcal{P}(\Gamma \vdash_{\Sigma} N : \sigma)$
  - Truth verified via an external call to a logical system,
  - Can inspect the signature, context, term, and the type.

So, what is new?

- Predicates on derivable typing judgements $\mathcal{P}(\Gamma \vdash_\Sigma N : \sigma)$
  - Truth verified via an external call to a logical system,
  - Can inspect the signature, context, term, and the type.
- Locked types ($\mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho]$), locked objects ($\mathcal{L}^{\mathcal{P}}_{N,\sigma}[M]$), and unlocked objects ($\mathcal{U}^{\mathcal{P}}_{N,\sigma}[M]$),

So, what is new?

- Predicates on derivable typing judgements $\mathcal{P}(\Gamma \vdash_\Sigma N : \sigma)$
  - Truth verified via an external call to a logical system,
  - Can inspect the signature, context, term, and the type.
- Locked types ($\mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho]$), locked objects ($\mathcal{L}^{\mathcal{P}}_{N,\sigma}[M]$), and unlocked objects ($\mathcal{U}^{\mathcal{P}}_{N,\sigma}[M]$),
- Introduction rules:

$$\frac{\Gamma \vdash_\Sigma \rho : \mathrm{Type} \quad \Gamma \vdash_\Sigma N : \sigma}{\Gamma \vdash_\Sigma \mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho] : \mathrm{Type}} \qquad \frac{\Gamma \vdash_\Sigma M : \rho \quad \Gamma \vdash_\Sigma N : \sigma}{\Gamma \vdash_\Sigma \mathcal{L}^{\mathcal{P}}_{N,\sigma}[M] : \mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho]}$$

So, what is new?

- Predicates on derivable typing judgements $\mathcal{P}(\Gamma \vdash_\Sigma N : \sigma)$
  - Truth verified via an external call to a logical system,
  - Can inspect the signature, context, term, and the type.
- Locked types ($\mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho]$), locked objects ($\mathcal{L}^{\mathcal{P}}_{N,\sigma}[M]$), and unlocked objects ($\mathcal{U}^{\mathcal{P}}_{N,\sigma}[M]$),
- Introduction rules:

$$\frac{\Gamma \vdash_\Sigma \rho : \mathrm{Type} \quad \Gamma \vdash_\Sigma N : \sigma}{\Gamma \vdash_\Sigma \mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho] : \mathrm{Type}} \qquad \frac{\Gamma \vdash_\Sigma M : \rho \quad \Gamma \vdash_\Sigma N : \sigma}{\Gamma \vdash_\Sigma \mathcal{L}^{\mathcal{P}}_{N,\sigma}[M] : \mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho]}$$

- Elimination rule:

$$\frac{\Gamma \vdash_\Sigma M : \mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho] \quad \Gamma \vdash_\Sigma N : \sigma \quad \mathcal{P}(\Gamma \vdash_\Sigma N : \sigma)}{\Gamma \vdash_\Sigma \mathcal{U}^{\mathcal{P}}_{N,\sigma}[M] : \rho}$$

So, what is new?

- Predicates on derivable typing judgements $\mathcal{P}(\Gamma \vdash_\Sigma N : \sigma)$
  - Truth verified via an external call to a logical system,
  - Can inspect the signature, context, term, and the type.
- Locked types ($\mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho]$), locked objects ($\mathcal{L}^{\mathcal{P}}_{N,\sigma}[M]$), and unlocked objects ($\mathcal{U}^{\mathcal{P}}_{N,\sigma}[M]$),
- Introduction rules:

$$\frac{\Gamma \vdash_\Sigma \rho : \mathrm{Type} \quad \Gamma \vdash_\Sigma N : \sigma}{\Gamma \vdash_\Sigma \mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho] : \mathrm{Type}} \qquad \frac{\Gamma \vdash_\Sigma M : \rho \quad \Gamma \vdash_\Sigma N : \sigma}{\Gamma \vdash_\Sigma \mathcal{L}^{\mathcal{P}}_{N,\sigma}[M] : \mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho]}$$

- Elimination rule:

$$\frac{\Gamma \vdash_\Sigma M : \mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho] \quad \Gamma \vdash_\Sigma N : \sigma \quad \mathcal{P}(\Gamma \vdash_\Sigma N : \sigma)}{\Gamma \vdash_\Sigma \mathcal{U}^{\mathcal{P}}_{N,\sigma}[M] : \rho}$$

- $\mathcal{L}$-reduction: $\mathcal{U}^{\mathcal{P}}_{N,\sigma}[\mathcal{L}^{\mathcal{P}}_{N,\sigma}[M]] \to_{\mathcal{L}} M$.

Properties of $LF_{\mathcal{P}}$

The main properties

- Confluence, Strong Normalization - Yes, immediately.

Properties of $LF_{\mathcal{P}}$

The main properties

- Confluence, Strong Normalization - Yes, immediately.
- Subject Reduction - Yes, with certain conditions imposed on predicates (closure under signature and context weakening and permutation, substitution, and $\beta\mathcal{L}$-reduction.

Properties of $\mathrm{LF}_{\mathcal{P}}$

The main properties

- Confluence, Strong Normalization - Yes, immediately.
- Subject Reduction - Yes, with certain conditions imposed on predicates (closure under signature and context weakening and permutation, substitution, and $\beta\mathcal{L}$-reduction.
- Decidability - Yes, if predicates used are decidable.

Properties of $LF_{\mathcal{P}}$

The main properties

- Confluence, Strong Normalization - Yes, immediately.
- Subject Reduction - Yes, with certain conditions imposed on predicates (closure under signature and context weakening and permutation, substitution, and $\beta\mathcal{L}$-reduction.
- Decidability - Yes, if predicates used are decidable.
- A canonical version of the system ($LF_{\mathcal{P}}^{\mathcal{C}}$) was also developed, dealing only with $\beta\eta$-long normal forms, for easy formulation and proofs of adequacy of the encodings.

Encoded examples

- The untyped λ-calculus (using HOAS)

Encoded examples

- The untyped λ-calculus (using HOAS)
- The untyped λ-calculus with call-by-value reduction strategy
  - β-reduction in $M\,N$ occurs only if N is a value

Encoded examples

- The untyped λ-calculus (using HOAS)
- The untyped λ-calculus with call-by-value reduction strategy
  - β-reduction in $M\,N$ occurs only if N is a value
- Modal logics S_4 and S_5 in Hilbert and Natural deduction style
  - Rule applicable if formula does not depend on assumptions

Encoded examples

- The untyped λ-calculus (using HOAS)
- The untyped λ-calculus with call-by-value reduction strategy
  - β-reduction in $M\,N$ occurs only if N is a value
- Modal logics S_4 and S_5 in Hilbert and Natural deduction style
  - Rule applicable if formula does not depend on assumptions
- Non-commutative linear logic
  - Conditions on occurence and order of assumptions

Encoded examples

- The untyped λ-calculus (using HOAS)
- The untyped λ-calculus with call-by-value reduction strategy
  - β-reduction in $M\,N$ occurs only if N is a value
- Modal logics S_4 and S_5 in Hilbert and Natural deduction style
  - Rule applicable if formula does not depend on assumptions
- Non-commutative linear logic
  - Conditions on occurence and order of assumptions
- A simple imperative language with Hoare-like logic
  - Pre- and post-conditions

The end of the presentation

Thank you for your attention!

Any questions?