Linear logic as a logical framework
An overview

Dale Miller

Inria Saclay & LIX, École Polytechnique
Palaiseau, France

SD 2017: Structures and Deduction
8 September 2017, Oxford

What is a logical framework?

I would like to have a fancy definition.
Instead, let us characterize some properties.

- A formal system based on well considered foundations that has a mature literature.
- That literature should contain clear means to implement and to reason about that framework.
- Specifications are independent of technology.

A logic framework, as opposed to an algebraic or categoric framework, should

- contain inference rules, consistency, cut-elimination (normalization) and
- various well understood logics (e.g., propositional minimal logic) should be subsystems.

Polemics: Girard and de Bruijn

J.-Y. Girard

- “The word ‘meta-logic’ should not be used in front of small children.” (sometime prior to Nov 2009)
- "Logic 2.0” at TLLA 2017 (September 2017). My understanding: the rejection of all axioms. Build everything from very few primitive concepts. The latter are inspired by propositional (linear) logic and proof nets.

Polemics: Girard and de Bruijn

J.-Y. Girard

- "The word 'meta-logic' should not be used in front of small children." (sometime prior to Nov 2009)
- "Logic 2.0" at TLLA 2017 (September 2017). My understanding: the rejection of all axioms. Build everything from very few primitive concepts. The latter are inspired by propositional (linear) logic and proof nets.

De Bruijn: "A plea for weaker frameworks"

- He was interested in supplying tools to mathematicians.
- Implement something simple (say, dependently typed λ-calculus) and move forward even if you must write lots of axioms.
- Mathematics uses lots of axioms, anyway.

Polemics: my perspective

Logical frameworks are tools but they should not be too simple.

We should incorporate more logical principles when possible: e.g., moving from intuitionistic logic ($\supset$, $\forall$) to LL (adding lollipop) to full linear logic (adding par and negation).

First-order quantification is no problem. Higher-order predicate quantification is still debated in some circles.

I replace de Bruijn's "weak" with "well understood and modular extensions with a mature literature and multiple implementations."

I prefer a richer logic if I can do with fewer axioms.

One success of logical frameworks: bindings

The first logical frameworks appear in the late 1980's when the first intuitionistic logic frameworks were developed: Isabelle/Generic, λProlog, Edinburgh LF.

Those three frameworks all provided roughly the same solution to the problem of first-order quantification, substitution, and eigenvariable restrictions by using typed λ-calculi modulo $\alpha\beta\eta$-conversion and using generic and hypothetical proof principles.

I prefer this approach over the nominal approach to bindings since the later usually requires axiomizations.

Intuitionistic meta-logics for natural deduction

Natural deduction inference rules such as

$$\frac{A \quad B}{A \wedge B} \qquad \frac{A \vee B \quad \begin{matrix} A \\ \vdots \\ C \end{matrix} \quad \begin{matrix} B \\ \vdots \\ C \end{matrix}}{C}$$

Can be encoded as formulas in intuitionistic logic as:

$$\forall A, B\ [\mathtt{pv}(A) \supset \mathtt{pv}(B) \supset \mathtt{pv}(A \wedge B)]$$

$$\forall A, B, C[\mathtt{pv}(A \vee B) \supset (\mathtt{pv}(A) \supset \mathtt{pv}(C)) \supset (\mathtt{pv}(B) \supset \mathtt{pv}(C)) \supset \mathtt{pv}(C)]$$

Object-level connectives are black; meta-level connectives are red.
We have one meta-level predicate pv.

What have we learned from linear logic?

The *additive* and *multiplicative* distinction of inference rules is important.

What have we learned from linear logic?

The *additive* and *multiplicative* distinction of inference rules is important.

Focusing and *polarity* yield synthetic inference rules.

- invertible rules are applied together (asynchronous phase)
- non-invertible rules are applied together and possibly in parallel (synchronous phase)
- In classical and intuitionistic logics, contraction is applied only with the "decide" rule and only on positive formulas.
- Negative non-atomic formulas are not contracted and not weakened. logic formulas.
- Rich dualities can be expressed: e.g. left/right, cut/initial, introduction/elimination

A short bibliography: linear logic as a framework

1. Henriksen. Using LJF as a Framework for Proof Systems. Technical Report, University of Copenhagen, 2009.
2. Lellmann, Olarte, and Pimentel. A uniform framework for substructural logics with modalities. LPAR-21, 2017.
3. Lellmann and Pimentel. Proof search in nested sequent calculi. LPAR-20, 2015.
4. M and Pimentel. A formal framework for specifying sequent calculus proof systems. TCS, 2013.
5. Nigam and M. A framework for proof systems. *J. of Automated Reasoning*, 2010.
6. Nigam, Pimentel, and Reis. Specifying proof systems in linear logic with subexponentials. LSFA 2010.
7. Pimentel and M. On the specification of sequent systems. LPAR-12, 2005.

Encoding the object-logic

We shall only consider two object-logics here: (first-order) intuitionistic and classical logics.

Most (object-level) proof systems mention (object-level) formulas in *two senses*.

- Sequent calculus: left-hand-side, right-hand-side
- Natural deduction: hypothesis, conclusion
- Tableaux: positive or negative signed formulas

These two senses are represented as the two meta-level predicates $\lfloor \cdot \rfloor$ (left) and $\lceil \cdot \rceil$ (right), both of type $bool \to o$.

The two-sided, object-level sequent $B_1, \ldots, B_n \vdash C_1, \ldots, C_m$ as the one-sided, meta-level sequent $\vdash \lfloor B_1 \rfloor, \ldots, \lfloor B_n \rfloor, \lceil C_1 \rceil, \ldots, \lceil C_m \rceil$.

Convention: $\lfloor \Gamma \rfloor$ denotes $\{\lfloor F \rfloor \mid F \in \Gamma\}$, etc.

The theory $\mathcal{L}$: introduction rules

$$
\begin{array}{llll}
(\Rightarrow_L) & \lfloor A \Rightarrow B \rfloor^\perp \otimes (\lceil A \rceil \otimes \lfloor B \rfloor) & (\Rightarrow_R) & \lceil A \Rightarrow B \rceil^\perp \otimes (\lfloor A \rfloor \parr \lceil B \rceil) \\
(\wedge_L) & \lfloor A \wedge B \rfloor^\perp \otimes (\lfloor A \rfloor \oplus \lfloor B \rfloor) & (\wedge_R) & \lceil A \wedge B \rceil^\perp \otimes (\lceil A \rceil \,\&\, \lceil B \rceil) \\
(\vee_L) & \lfloor A \vee B \rfloor^\perp \otimes (\lfloor A \rfloor \,\&\, \lfloor B \rfloor) & (\vee_R) & \lceil A \vee B \rceil^\perp \otimes (\lceil A \rceil \oplus \lceil B \rceil) \\
(\forall_L) & \lfloor \forall B \rfloor^\perp \otimes \lfloor Bx \rfloor & (\forall_R) & \lceil \forall B \rceil^\perp \otimes \forall x \lceil Bx \rceil \\
(\exists_L) & \lfloor \exists B \rfloor^\perp \otimes \forall x \lfloor Bx \rfloor & (\exists_R) & \lceil \exists B \rceil^\perp \otimes \lceil Bx \rceil \\
(\perp_L) & \lfloor \perp \rfloor^\perp & (t_R) & \lceil t \rceil^\perp \otimes \top
\end{array}
$$

The meanings of the two senses for object-level connectives are supplied by these formulas.

Without polarization of the $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ atomic formulas, we do not know what proof system we are encoding.

The theory $\mathcal{L}$: structural and identity rules

$$
\begin{array}{llll}
(Id_1) & \lfloor B \rfloor^\perp \otimes \lceil B \rceil^\perp & (Id_2) & \lfloor B \rfloor \otimes \lceil B \rceil \\
(Str_L) & \lfloor B \rfloor^\perp \otimes ?\lfloor B \rfloor & (Str_R) & \lceil B \rceil^\perp \otimes ?\lceil B \rceil \\
(W_R) & \lceil C \rceil^\perp \otimes \perp & &
\end{array}
$$

Specification of the identity rules (*e.g.*, cut and initial), the structural rules (weakening and contraction), and just weakening (on the right).

Note: Mix would correspond to the formula $\perp \otimes \perp$, *i.e.*, the smallest positive formula B of MALL (without atoms) such that neither $\vdash B$ nor $\vdash B^\perp$.

Proving dualities

The Id_1 and Id_2 formulas can prove the duality of the $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ predicates: in particular, one can prove in linear logic that

$$\vdash \forall B(\lceil B \rceil \equiv \lfloor B \rfloor^{\perp}) \,\&\, \forall B(\lfloor B \rfloor \equiv \lceil B \rceil^{\perp}), Id_1, Id_2$$

Similarly, the formulas Str_L and Str_R allow us to prove the equivalences $\lfloor B \rfloor \equiv ?\lfloor B \rfloor$ and $\lceil B \rceil \equiv ?\lceil B \rceil$.

Three levels of adequacy

Roughly speaking

- if $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ formulas are both polarized negatively, we are encoding sequent calculus.
- if $\lceil \cdot \rceil$ is polarized negatively and $\lfloor \cdot \rfloor$ polarized positively, we are encoding natural deduction.

Level 0 / Relative completeness: the two systems have the same theorems.
Level -1 / Full completeness of proofs: The proofs of a formula in one proof system are in one-to-one correspondence with proofs in the other proof system.
Level -2 / Full completeness of derivations: The derivations (*i.e.*, open proofs) in one system are in one-to-one correspondence with the other proof system.

Good frameworks should aim for Level -2 encodings.

Asking more from our tools

- Is the logical framework mechanizable?
- Can we get (prototype) provers and proof checkers?
- Can we use the logical framework to prove cut elimination and initial elimination?
- Can we have simple checks that guarantee that cut is admissible? that non-atomic initials are admissible?

Next steps: Contexts

Are they lists or multisets? Something more specialized?

Beluga has an approach using contextual modal type theory.

Abella is moving incrementally.

- We construct them as lists but deconstruct them as multisets.
- Many properties about them need to be defined and proved (even though those properties appear to re-occur across many specifications).
- Cut-elimination and instantiation of object-level judgments is built into this approach to contexts.

Next steps: New perspective on terms

Term structure is too opaque. Term equality and term unification (in Abella, say) seems just too complex. It's a big black box.

The identification of the *mobility of bindings* (via pattern unification and β_0-conversion) has opened up this box more.

$$\lambda x.t = \lambda x.s \quad \text{iff} \quad \forall x.t = \forall x.s \quad (\text{or maybe } \nabla x.t = \nabla x.s)$$

Can we re-think typed terms? Why are they based always on intuitionistic natural deduction proof?

Recent papers explore focusing vis-a-vis term representation.

- Scherer's PhD and subsequent papers
- Brock-Nannestad, Guenot, and Gustafsson. PPDP 2015: $\lambda\kappa$-terms
- Gérard and M, CSL 2017: administrative normal forms

Thank you

Questions?