An overview of Structural Proof Theory and Computing

Dale Miller

INRIA-Saclay & LIX, École Polytechnique
Palaiseau, France

Madison, Wisconsin, 2 April 2012

Part of the
Special Session in Structural Proof Theory and Computing
2012 ASL annual meeting

Outline

This special session

Alexis Saurin, University of Paris 7
Proof search and the logic of interaction

David Baelde, ITU Copenhagen
A proof theoretical journey from programming to model checking and theorem proving

Stefan Hetzl, Vienna University of Technology
Which proofs can be computed by cut-elimination?

Marco Gaboardi, University of Pennsylvania
Light Logics for Polynomial Time Computations

Some themes within proof theory

- Ordinal analysis of consistency proofs (Gentzen, Schütte, Pohlers, etc)
- Reverse mathematics (Friedman, Simpson, etc)
- Proof complexity (Cook, Buss, Krajíček, Pudlák, etc)
- Structural Proof Theory (Gentzen, Girard, Prawitz, etc)
  - Focus on the combinatorial and structural properties of proof.
  - Proofs and their constituent are elements of computation

Many roles of logic in computation

Computation-as-model: Computations happens, *i.e.*, states change, communications occur, *etc*. Logic is used to make statements *about* computation. *E.g.*, Hoare triples, modal logics.

Computation-as-deduction: Elements of logic are used to model elements of computation directly.

Many roles of logic in computation

Computation-as-model: Computations happens, *i.e.*, states change, communications occur, *etc.* Logic is used to make statements *about* computation. *E.g.*, Hoare triples, modal logics.

Computation-as-deduction: Elements of logic are used to model elements of computation directly.

Proof normalization. Programs are proofs and computation is proof normalization (λ-conversion, cut-elimination). A foundations for functional programming. Curry-Howard Isomorphism.

Proof search. Programs are theories and computation is the search for sequent proofs. A foundations for logic programming, model checking, and theorem proving.

Computing as proof reduction

Example: Church numerals.

$$1 = \lambda f \lambda x.fx : (i \to i) \to i \to i$$
$$2 = \lambda f \lambda x.f(fx) : (i \to i) \to i \to i$$
$$+ = \lambda n \lambda m \lambda f \lambda x.(nf)((mf)x) :$$
$$((i \to i) \to i \to i) \to ((i \to i) \to i \to i) \to (i \to i) \to i \to i$$

Compute $2 + 2$ using β-reduction: $(\lambda x.t)s \longrightarrow t[s/x]$.

$$(\lambda n \lambda m \lambda f \lambda x.(nf)((mf)x))(\lambda f \lambda x.f(fx))(\lambda h \lambda u.h(hu))$$
$$(\lambda m \lambda f \lambda x.((\lambda f \lambda x.f(fx))f)((mf)x))(\lambda h \lambda u.h(hu))$$
$$(\lambda m \lambda f \lambda x.(\lambda x.f(fx))((mf)x))(\lambda h \lambda u.h(hu))$$
$$(\lambda f \lambda x.(\lambda x.f(fx))(((\lambda h \lambda u.h(hu))f)x))$$
$$(\lambda f \lambda x.(\lambda x.f(fx))((\lambda u.f(fu))x))$$
$$(\lambda f \lambda x.(\lambda x.f(fx)))(f(fx))$$
$$(\lambda f \lambda x.f(f(f(fx))))$$

Proof normalization: functional programming

Types are (propositional) formulas and λ-terms are proofs.

Computation is repeatedly applying β-reductions

Typing generally guarantees termination. More expressive types can guarantee more properties about computation.

A β-normal form is *the value*.

Proof search: logic programming

A *logic program* is a set of formulas Γ and a *query* G and computation is the *search for a cut-free proof* of $\Gamma \vdash G$.

During search, the collection of open sequents (those still requiring a proof) change and that change captures a computation.

Comparing proof-normalization and proof-search

	Functional Prog.	**Logic Prog.**
Proofs are	complete	incomplete
Proofs	may contain cuts	are cut-free
Cut-elimination	powers computation	is *about* computation
Computation is	determinate	non-deterministic
Programs define	functions	relations

Many ideas from the proof theory have been applied to these two computing paradigms, *e.g.*,

- higher-order quantification
- linear logic
- game semantics

The gap between these paradigms has remained robust.

Outline

Sequents

Sequents are pairs $\Gamma \vdash \Delta$ where

- Γ, the *left-hand-side*, is a multiset of formulas; and
- Δ, the *right-hand-side*, is a multiset of formulas.

NB: Gentzen used lists instead of multisets. (Sets are also another possible alternative.)

The formulas in Γ are "hypotheses" and the formulas in Δ are "possible conclusions".

There are three groups of inference rules: structural, identity, and introduction.

Inference rules: two structural rules

There are two sets of these: *contraction, weakening*.

$$\frac{\Gamma, B, B \vdash \Delta}{\Gamma, B \vdash \Delta}\ cL \qquad \frac{\Gamma \vdash \Delta, B, B}{\Gamma \vdash \Delta, B}\ cR$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, B \vdash \Delta}\ wL \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, B}\ wR$$

NB: Gentzen's use of lists of formulas required him to also have an *exchange* rule.

Inference rules: two identity rules

There are exactly two: *initial, cut*.

$$\frac{}{B \vdash B}\ \mathit{init} \qquad \frac{\Gamma_1 \vdash \Delta_1, B \qquad B, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\ \mathit{cut}$$

Notice the repeated use of the variable B in these rules.

In general: all instances of both of these rules can be *eliminated* except for *init* when B is atomic.

Inference rules: introduction rules (some examples)

$$\frac{\Gamma, B_i \vdash \Delta}{\Gamma, B_1 \wedge B_2 \vdash \Delta}\ \wedge\mathrm{L} \qquad \frac{\Gamma \vdash \Delta, B \qquad \Gamma \vdash \Delta, C}{\Gamma \vdash \Delta, B \wedge C}\ \wedge\mathrm{R}$$

$$\frac{\Gamma, B \vdash \Delta \qquad \Gamma, C \vdash \Delta}{\Gamma, B \vee C \vdash \Delta}\ \vee\mathrm{L} \qquad \frac{\Gamma \vdash \Delta, B_i}{\Gamma \vdash \Delta, B_1 \vee B_2}\ \vee\mathrm{R}$$

$$\frac{\Gamma_1 \vdash \Delta_1, B \qquad \Gamma_2, C \vdash \Delta_2}{\Gamma_1, \Gamma_2, B \supset C \vdash \Delta_1, \Delta_2}\ \supset\mathrm{L} \qquad \frac{\Gamma, B \vdash \Delta, C}{\Gamma \vdash \Delta, B \supset C}\ \supset\mathrm{R}$$

$$\frac{\Gamma, B[t/x] \vdash \Delta}{\Gamma, \forall x\ B \vdash \Delta}\ \forall\mathrm{L} \qquad \frac{\Gamma \vdash \Delta, B[y/x]}{\Gamma \vdash \Delta, \forall x\ B}\ \forall\mathrm{R}$$

$$\frac{\Gamma, B[y/x] \vdash \Delta}{\Gamma, \exists x\ B \vdash \Delta}\ \exists\mathrm{L} \qquad \frac{\Gamma \vdash \Delta, B[t/x]}{\Gamma \vdash \Delta, \exists x\ B}\ \exists\mathrm{R}$$

Single-conclusion and multi-conclusion sequents

- An arbitrary proof involving sequents is a proof in *classical logic*.
- A proof in which all sequents contain at most one formula on the right is an *intuitionistic proof*.

Equivalently: an intuitionistic (cut-free) proof

- has no contractions on the right and
- the implication left rule must be restricted as follows:

$$\frac{\Gamma_1 \vdash B \qquad \Gamma_2, C \vdash D}{\Gamma_1, \Gamma_2, B \supset C \vdash D} \supset \mathsf{L}$$

The first restriction cannot be stated using natural deduction.

Compare this characterization of classical vs intuitionistic logic with

- the presence or absence of the excluded middle,
- the use of Kripke semantics,
- references to construction reasoning, etc.

Outline

A chemistry for inference

Girard's linear logic (1987) strengthen our understanding of structural and introduction rules.

The sequent calculi of Gentzen and Girard provides the *atoms of inference*.

The computer scientist wishing to use inference generally finds these atoms to be far too tiny and unstructured.

Recent work in structural proof theory has been developing a *chemistry for inference* so that we can engineer a rich set of tailor-made *molecules of inference*.

Classical logic and one-sided sequents

Two conventions for dealing with classical logic.

- Formulas are in *negation normal form*.
  - $B \supset C$ is replaced with $\neg B \vee C$,
  - negations are pushed to the atoms
- Sequents will be one-sided. In particular, the two sided sequent

$$B_1, \ldots, B_n \vdash C_1, \ldots, C_m$$

will be converted to

$$\vdash \neg B_1, \ldots, \neg B_n, C_1, \ldots, C_m.$$

LKF: Focusing for Classical Logic

The connectives are *polarized*: $\wedge^-$, $\wedge^+$, $\vee^-$, $\vee^+$, t^-, t^+, f^-, f^+.

A formula is *positive* if it is a top-level $\wedge^+$, $\vee^+$, t^+, f^+ or an atom.
A formula is *negative* if it is a top-level $\wedge^-$,$\vee^-$, t^-, f^- or a negated atom.

LKF is a focused, one-sided sequent calculus with the sequents

$$\vdash \Theta \Uparrow \Gamma \qquad \text{and} \qquad \vdash \Theta \Downarrow \Gamma$$

Here, Γ is a multiset of formulas and Θ is a multiset of positive formulas and negated atoms.

LKF : focused proof systems for classical logic

$$\frac{}{\vdash \Theta \Uparrow \Gamma, t^{-}} \qquad \frac{\vdash \Theta \Uparrow \Gamma, B \quad \vdash \Theta \Uparrow \Gamma, C}{\vdash \Theta \Uparrow \Gamma, B \wedge^{-} C} \qquad \frac{\vdash \Theta \Uparrow \Gamma}{\vdash \Theta \Uparrow \Gamma, f^{-}} \qquad \frac{\vdash \Theta \Uparrow \Gamma, B, C}{\vdash \Theta \Uparrow \Gamma, B \vee^{-} C}$$

LKF : focused proof systems for classical logic

$$\frac{}{\vdash \Theta \Uparrow \Gamma, t^-} \qquad \frac{\vdash \Theta \Uparrow \Gamma, B \quad \vdash \Theta \Uparrow \Gamma, C}{\vdash \Theta \Uparrow \Gamma, B \wedge^- C} \qquad \frac{\vdash \Theta \Uparrow \Gamma}{\vdash \Theta \Uparrow \Gamma, f^-} \qquad \frac{\vdash \Theta \Uparrow \Gamma, B, C}{\vdash \Theta \Uparrow \Gamma, B \vee^- C}$$

$$\frac{}{\vdash \Theta \Downarrow t^+} \qquad \frac{\vdash \Theta \Downarrow \Gamma_1, B_1 \qquad \vdash \Theta \Downarrow \Gamma_2, B_2}{\vdash \Theta \Downarrow \Gamma_1, \Gamma_2, B_1 \wedge^+ B_2} \qquad \frac{\vdash \Theta \Downarrow \Gamma, B_i}{\vdash \Theta \Downarrow \Gamma, B_1 \vee^+ B_2}$$

LKF : focused proof systems for classical logic

$$\frac{}{\vdash \Theta \Uparrow \Gamma, t^-} \qquad \frac{\vdash \Theta \Uparrow \Gamma, B \quad \vdash \Theta \Uparrow \Gamma, C}{\vdash \Theta \Uparrow \Gamma, B \wedge^- C} \qquad \frac{\vdash \Theta \Uparrow \Gamma}{\vdash \Theta \Uparrow \Gamma, f^-} \qquad \frac{\vdash \Theta \Uparrow \Gamma, B, C}{\vdash \Theta \Uparrow \Gamma, B \vee^- C}$$

$$\frac{}{\vdash \Theta \Downarrow t^+} \qquad \frac{\vdash \Theta \Downarrow \Gamma_1, B_1 \qquad \vdash \Theta \Downarrow \Gamma_2, B_2}{\vdash \Theta \Downarrow \Gamma_1, \Gamma_2, B_1 \wedge^+ B_2} \qquad \frac{\vdash \Theta \Downarrow \Gamma, B_i}{\vdash \Theta \Downarrow \Gamma, B_1 \vee^+ B_2}$$

Init

$$\frac{}{\vdash \neg A, \Theta \Downarrow A}$$

Store

$$\frac{\vdash \Theta, C \Uparrow \Gamma}{\vdash \Theta \Uparrow \Gamma, C}$$

Release

$$\frac{\vdash \Theta \Uparrow \mathcal{N}}{\vdash \Theta \Downarrow \mathcal{N}}$$

Decide

$$\frac{\vdash \mathcal{P}, \Theta \Downarrow \mathcal{P}}{\vdash \mathcal{P}, \Theta \Uparrow \cdot}$$

$\mathcal{P}$ multiset of positives; $\mathcal{N}$ multiset of negatives;
A atomic; C positive formula or negated atom

Results about LKF

Let B be a first-order logic formula and let $\hat{B}$ result from B by placing $+$ or $-$ on t, f, $\wedge$, and $\vee$ (there are exponentially many such placements).

Theorem. B is a first-order theorem if and only if $\hat{B}$ has an LKF proof. [Liang & M, TCS 2009]

Thus the different polarizations do not change *provability* but can radically change the *proofs*.

One can easy move from a *linear-sized* proof to an *exponentially-sized* proof simply by changing the polarity of connectives.

Immediate by inspection of LKF

The only form of *contraction* is in the Decide rule.

$$\frac{\vdash \mathcal{P}, \Theta \Downarrow \mathcal{P}}{\vdash \mathcal{P}, \Theta \Uparrow \cdot}$$

Thus: only positive formulas are contracted.

The only occurrence of *weakening* is in the Init rule.

$$\frac{}{\vdash \neg A, \Theta \Downarrow A}$$

Thus formulas that are top-level $\wedge^-$,$\vee^-$, t^-, f^- are treated *linearly* (in the sense of linear logic).

The abstraction behind focused proofs

If we ignore the internal structure of phases and consider only their boundaries, we move from *micro-rules* (the atoms of inference) to *macro-rules* (pos or neg phases, the molecules of inference).

$$\frac{\vdash \Theta_1 \Uparrow \cdot \qquad \cdots \qquad \vdash \Theta_n \Uparrow \cdot}{\vdash \Theta \Uparrow \cdot}$$

An example

Let a, b, c be atoms and let Θ contain the formula $a \wedge^+ b \wedge^+ \neg c$.

$$\dfrac{\dfrac{}{\vdash \Theta \Downarrow a}\ \textit{Init} \quad \dfrac{}{\vdash \Theta \Downarrow b}\ \textit{Init} \quad \dfrac{\dfrac{\vdash \Theta, \neg c \Uparrow \cdot}{\vdash \Theta \Uparrow \neg c}\ \textit{Store}}{\vdash \Theta \Downarrow \neg c}\ \textit{Release}}{\dfrac{\vdash \Theta \Downarrow a \wedge^+ b \wedge^+ \neg c}{\vdash \Theta \Uparrow \cdot}\ \textit{Decide}}$$

This derivation is possible iff Θ is of the form $\neg a, \neg b, \Theta'$. Thus, the "macro-rule" is

$$\dfrac{\vdash \neg a, \neg b, \neg c, \Theta' \Uparrow \cdot}{\vdash \neg a, \neg b, \Theta' \Uparrow \cdot}$$

Conclusion

The sequent calculus of Gentzen stressed the use of *structural rules* in the specification of both intuitionistic and classical logics.

Girard's linear logic refined our understanding of the interplay between structural and introduction rules.

In general, the identity rules (initial and cut) can be eliminated.

For many applications of inference in computer science, these atoms of inference need to be organized into larger rules.

Focus proofs systems (which also exist for linear and intuitionistic logics) can be used to flexibly introduce such larger, molecular inference rules.