Contexts in sequents as indexed storage

Dale Miller

Inria Saclay & LIX, École Polytechnique
Palaiseau, France

Ticamore Workshop 2021
17 June 2021

Pros and Cons of Sequent Calculus Proofs

Such proofs are great for proof checking since they contain a great deal of detail.

Most other aspects of these proofs are pretty bad: they are large, full of detail, difficult to communication and understand, etc.

Also, the abundance of permutations of inference rules hides interesting proof structure.

Example of different proof systems:

- Herbrand disjuncts (classical logic)
  - make explicit the instances of some existential quantifiers
  - remaining details can be checked by a simple decision procedure.
- Typed λ-terms (intuitionistic logic)
  - These proof skeletons allow for computations via normalization.
  - Type checking/inference checks that these skeletons are actual proofs.

How many proofs of $a \supset a \supset a$ are there?

Gentzen's original proof system contains two (cut-free) proofs, depending on where one places the exchange rules.

If we use multisets, then we have just one proof: there is only one initial rule proving the sequent

$$a,\ a \vdash a.$$

In typed λ-calculus, where contexts are declarations of distinct variables, there are two such proofs, namely, $\lambda x_1 \lambda x_2 . x_1$ and $\lambda x_1 \lambda x_2 . x_2$ based on two different invocations of the initial rule.

$$x_1 : a,\ x_2 : a \vdash x_1 : a \qquad x_1 : a,\ x_2 : a \vdash x_2 : a$$

All of these schemes have drawbacks. I shall motivate a more flexible approach, based on an analysis of focused proof systems.

Polarized first-order classical logic formulas

- polarized connectives: t^{-}, t^{+},f^{-}, f^{+}, $\vee^{-}$, $\vee^{+}$, $\wedge^{-}$, $\wedge^{+}$,
- first-order quantifiers $\forall$ (negative) and $\exists$ (positive),
- All polarized formulas are in *negation normal form* (i.e., negations have only atomic scope), and
- atomic formulas (atoms are given a fixed polarity assignment).

If a formula's top-level connective is t^{+}, f^{+}, $\vee^{+}$, $\wedge^{+}$, or $\exists$, then that formula is *positive*.

If a formula's top-level connective is t^{-}, f^{-}, $\vee^{-}$, $\wedge^{-}$, or $\forall$, then it is *negative*.

LKF: proof rules

Negative introduction rules

$$\frac{}{\vdash t^-, \Gamma \Uparrow \Theta} \qquad \frac{\vdash A, \Gamma \Uparrow \Theta \qquad \vdash B, \Gamma \Uparrow \Theta}{\vdash A \wedge^- B, \Gamma \Uparrow \Theta}$$

$$\frac{\vdash \Gamma \Uparrow \Theta}{\vdash f^-, \Gamma \Uparrow \Theta} \qquad \frac{\vdash A, B, \Gamma \Uparrow \Theta}{\vdash A \vee^- B, \Gamma \Uparrow \Theta} \qquad \frac{\vdash [y/x]B, \Gamma \Uparrow \Theta}{\vdash \forall x.B, \Gamma \Uparrow \Theta}$$

Positive introduction rules

$$\frac{}{\vdash t^+ \Downarrow \Theta} \qquad \frac{\vdash A \Downarrow \Theta \qquad \vdash B \Downarrow \Theta}{\vdash A \wedge^+ B \Downarrow \Theta} \qquad \frac{\vdash B_i \Downarrow \Theta}{\vdash B_1 \vee^+ B_2 \Downarrow \Theta} \qquad \frac{\vdash [s/x]B \Downarrow \Theta}{\vdash \exists x.B \Downarrow \Theta}$$

Non-introduction rules

$$\frac{}{\vdash L \Downarrow \neg L, \Theta}\ \mathit{init} \qquad \frac{\vdash \Gamma \Uparrow Q, \Theta}{\vdash Q, \Gamma \Uparrow \Theta}\ \mathit{store} \qquad \frac{\vdash N \Uparrow \Theta}{\vdash N \Downarrow \Theta}\ \mathit{release}$$

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

Here: L is a positive literal, P is positive, N is negative, Q is positive or a literal.

Remove the arrows

Negative introduction rules

$$\frac{}{\vdash t^-, \Theta} \qquad \frac{\vdash A, \Theta \quad \vdash B, \Theta}{\vdash A \wedge^- B, \Theta}$$

$$\frac{\vdash \Gamma, \Theta}{\vdash f^-, \Theta} \qquad \frac{\vdash A, B, \Theta}{\vdash A \vee^- B, \Theta} \qquad \frac{\vdash [y/x]B, \Theta}{\vdash \forall x.B, \Theta}$$

Positive introduction rules

$$\frac{}{\vdash t^+, \Theta} \qquad \frac{\vdash A, \Theta \quad \vdash B, \Theta}{\vdash A \wedge^+ B, \Theta} \qquad \frac{\vdash B_i, \Theta}{\vdash B_1 \vee^+ B_2, \Theta} \quad \frac{\vdash [s/x]B, \Theta}{\vdash \exists x.B, \Theta}$$

Non-introduction rules

$\dfrac{}{\vdash p, \neg p, \Theta}\ \mathit{init}$ $\quad$ $\dfrac{\vdash \Gamma, Q, \Theta}{\vdash Q, \Gamma, \Theta}$ ~~store~~ noop $\quad$ $\dfrac{\vdash N, \Theta}{\vdash N, \Theta}$ ~~release~~ noop

$\dfrac{\vdash P, P, \Theta}{\vdash P, \Theta}$ ~~decide~~ contraction

NB: $\wedge^+$ and $\wedge^-$ coincide.

Observations

In LKF, contraction is on positive formulas only.

There is multiplicative and additive form of disjunction.

If LKF is generalized to the *multifocus* setting, then conjunction has distinct multiplicative and additive forms.

Drop plus/minus and the arrows:

- Some rules are no-ops.
- You get soundness immediate.
- You also lose the restriction on contraction.

Synthetic inference rules

A *synthetic inference rule* is an inference rule of the form

$$\frac{\vdash \cdot \Uparrow \Gamma_1 \quad \ldots \quad \vdash \cdot \Uparrow \Gamma_n}{\vdash \cdot \Uparrow \Gamma}$$

which is *justified* by a derivation of the form

$$\begin{array}{c} \vdash \cdot \Uparrow \Gamma_1 \quad \ldots \quad \vdash \cdot \Uparrow \Gamma_n \\ \Pi \\ \vdash \cdot \Uparrow \Gamma \end{array}$$

Here, $n \geq 0$ and the inference rules of derivation Π are such that no positive rule application occurs above a negative rule application.

Theorem: If you extend LK (LJ) with synthetic rules based on bipole formula (e.g., geometric formula), the resulting proof system satisfies cut-elimination.

Proof: See “From axioms to synthetic inference rules via focusing” by Marin, Miller, Pimentel, and Volpe.

Zones and storage

There are two zones in LKF:

$$\vdash \text{internal} \Uparrow \text{external} \qquad \text{and} \qquad \vdash \text{internal} \Downarrow \text{external}$$

The internal zone is where introduction rules take place. From the perspective of synthetic rules, the internal zone is not visible.

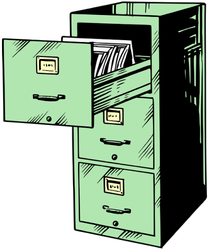
The external zone can be either be under linear or classical maintenance (i.e., contraction and weakening are available).

In this talk, I will not deal with linear maintenance. The difference between additive and multiplicative treatments of such contexts disappears.

I'll use the term "storage" to mean the external, classical zone.

Storage as an index relation

Instead of viewing contexts in sequents as lists / multisets / sets, we shall instead see them as *relations* between *indexes* and (polarized) formulas.



LKF has three operations on storage.

store inserts a formula with an index

decide extracts a formula with an index

init checks if the focused formula matches a formula with a specific index

Let $\mathcal{I}$ be the set of indexes and $\mathcal{B}$ the set of polarized formulas. Storage Θ is a finite subset of $\mathcal{I} \times \mathcal{B}$.

Indexing does not need to be functional: the same index can be related to different formulas.

Storage as an index relation (continued)

Examples:

1. Indexes as formulas: $\mathcal{I}$ is the same as $\mathcal{B}$ and Θ is always some subset of the equality relation on $\mathcal{I} \times \mathcal{B}$: that is, a subset of $\{\langle B, B\rangle \mid B \text{ is a polarized formula}\}$.
2. Trivial indexing: $\mathcal{I}$ is a singleton.
3. Variables as indexes: $\{x_1 : B_1, \ldots, x_n : B_n\}$. These are common in typed λ-calculi and are usually functional.

This notion of index and storage was developed in the *Foundational Proof Certificate* project which dealt with the Computer Science problem of representing and checking proofs. See Chihani, Miller, and Renaud, JAR 2017.

I will use the terms *index* and *label* probably interchangeably. Dov Gabbay's Labeled Deduction Systems is much more general.

Proof Terms and Proof Certificates

In typing of λ-terms:

$$x_1 : \alpha_1, \ldots, x_n : \alpha_n \vdash t : \alpha_0$$

Bottom-up proof construction is type checking of the term t.
Information in t guides the proof construction.
The variables $x_1, \ldots, x_n$ can be free in Ξ.

In classical logic:

$$\Xi \vdash \cdot \Uparrow l_1 : B_1, \ldots, l_n : B_n$$

Bottom-up proof construction is proof checking of the certificate Ξ. Information in Ξ guides the proof construction.
The indexes $l_1, \ldots, l_n$ can be appear in t.

Indexes: a well motivated and flexible proof theoretic tool

Notice that indexes are not parts of formulas:

- no need to extend the signature of formulas and terms to support them
- no need to give them logical (model theoretic) interpretations

Contrast this with *Tseitin constants* and *Skolem functions*.

In encodings of *modal logics*, worlds can be use as indexes.

Indexes for proof assistants, such as Coq and Abella.

- Local assumptions: `H1`, `H2`, `IH`, etc.
- Theorems proved in the current session: `plus_comm`, `plus_assoc`, etc.
- Archived proved theorems: a url.

  `https://abella-prover.org/examples/process-calculi/pic_lambda/picalc_str_eq_is_bisimulation.thm`

  `ipfs:Qm67c1714291237a037d2f1d17ea1d662eaa32734bbeaa659ea05549c3266e76e0`

Matings

Polarize a propositional formula using $\wedge^-$ and $\vee^-$.

The negative phase with conclusion $\vdash B \Uparrow \cdot$ has leaves of the form

$$\vdash \cdot \Uparrow l_1 : L_1, \ldots, l_n : L_n,$$

where $l_1, \ldots, l_n$ are *paths* to the literals $L_1, \ldots, L_n$ in B.

There can be an exponential number positive phases appearing on top of such sequents.

$$\dfrac{\dfrac{}{\vdash L_i \Downarrow l_1 : L_1, \ldots, l_n : L_n}\ \textit{init} \text{ on } l_j}{\vdash \cdot \Uparrow l_1 : L_1, \ldots, l_n : L_n}\ \textit{decide} \text{ on } l_i$$

Let $\mathcal{M}$ be the set of all such (ordered) pairs $\langle l_i, l_j \rangle$.

A *mating* is a set $\mathcal{M}$ of pairs of occurrences of literal in a formula [Andrews, JACM 1981]. Matings can be seen as *proof certificates*.

More about the structure of synthetic rules

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\cdots \quad \vdash \cdot \Uparrow \Gamma, P, \Delta_i \quad \cdots}{\cdots \Uparrow \cdots \Uparrow \cdots}}{\vdash N \Uparrow \Gamma, P}\ \mathit{release}}{\cdots \Downarrow \cdots \Downarrow \cdots}}{\vdash P \Downarrow \Gamma, P}}{\vdash \cdot \Uparrow \Gamma, P}\ \mathit{decide}$$

The formulas in Δ_i will have indexes.

Deciding on a literal stops the proof with an *init*.

Deciding on a non-literal formula starts a new synthetic inference rule.

Delays: $\partial_-(B)$ and $\partial_+(B)$

Delays can be used to break negative and positive phases: $\partial_-(B)$ is always negative and $\partial_+(B)$ is always positive, independent of B.

Delays can be defined by any one of the following choices.

Delay	*multiplicative*	*additive*	*quantification*
$\partial_-(B)$	$B \vee^- f^-$	$B \wedge^- t^-$	$\forall x.B$
$\partial_+(B)$	$B \wedge^+ t^+$	$B \vee^+ f^+$	$\exists x.B$

(where x is not free in B)

Equivalently:
make $\partial_+(\cdot)$ the 1-ary version of either the binary $\vee^+$ or $\wedge^+$ and
make $\partial_-(\cdot)$ the 1-ary version of either the binary $\vee^-$ or $\wedge^-$.

Tseitin Constants

Let B be a polarized propositional formula using only $\wedge^-$ and $\vee^-$.

As we noted, this leads to a focused proof with

- one negative phase with an exponential number of premises,
- only literals are stored,
- an exponential number of trivial positive phases (involving only *decide* and *init*),
- there is no alternation of phases.

We can sometimes avoid this exponential blowup by inserting $\partial_+(\cdot)$.

Such delayed item will be stored and given indexes. These indexes can play the role of Tseitin constants.

Such an explanation of Tseitin constants does not inject them into the signature of formulas: no model extension property is needed.

A simple example

Consider the polarized formula $((p \vee^- (q \wedge^- \neg r)) \wedge^- s)$.

Introduce the Tseitin constant tc as a propositional constant (with negative polarity) with the following axioms for $tc \equiv (q \wedge^- \neg r)$:

$$tc \wedge^+ (\neg q \vee^+ r) \qquad \text{and} \qquad (q \wedge^- \neg r) \wedge^+ \neg tc.$$

Let Γ be some set of polarized formulas and let $\Gamma^\star$ be Γ with the two axioms for tc added.

Consider the following two derivation, one ending with

$$\vdash (p \vee^- tc) \wedge^- s \Uparrow \Gamma^\star$$

and the other ending with

$$\vdash (p \vee^- \partial_+(q \wedge^- \neg r)) \wedge^- s \Uparrow \Gamma$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash \cdot \Uparrow q, '' \qquad \vdash \cdot \Uparrow \neg r, ''}{\vdash q \land^- \neg r \Uparrow ''}}{\vdash q \land^- \neg r \Downarrow ''} \qquad \dfrac{}{\vdash \neg tc \Downarrow ''}\,\text{init}}{\vdash (q \land^- \neg r) \land^+ \neg tc \Downarrow ''}\,\text{decide}}{\vdash \cdot \Uparrow p, tc, \Gamma^*} \qquad \vdash \cdot \Uparrow s, \Gamma^*}{\vdash (p \lor^- tc) \land s^- \Uparrow \Gamma^*}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash \cdot \Uparrow q, \text{''} \qquad \vdash \cdot \Uparrow \neg r, \text{''}}{\vdash q \wedge^- \neg r \Uparrow \text{''}}}{\vdash q \wedge^- \neg r \Downarrow \text{''}}}{\vdash \delta^+(q \wedge^- \neg r) \Downarrow \text{''}}}{\vdash \cdot \Uparrow p, tc: \delta^+(q \wedge^- \neg r], \Gamma} \qquad \vdash \cdot \Uparrow s, \Gamma}{\vdash (p \vee^- \delta^+(q \wedge^- \neg r)) \wedge^- s \Uparrow \Gamma}$$

Another example: beyond bi-poles

Switch to the intuitionistic setting for this example.

- Storage is on the left of the sequent arrow.
- $\vee^-$ and f^- are dropped and $\supset$ is added.

Assume that the context Γ contains the formula

$$\partial_-((p \supset q) \vee^+ (p \supset q))$$

and that p and q are positive.

Note that this formula is neg-pos-neg-pos (not a bipole).

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \left(\dfrac{\Delta, q \vdash E}{\Delta, p \vdash E}\,\ell\right)\\ \vdots\\ \Gamma, \ell : p \supset q \Uparrow \cdot \vdash C\end{array}}{\Gamma \Uparrow p \supset q \vdash C} \qquad \dfrac{\begin{array}{c}\vdots\\ \left(\dfrac{\Delta, p \vdash E}{\Delta, q \vdash E}\,r\right)\\ \vdots\\ \Gamma, r : q \supset p \Uparrow \cdot \vdash C\end{array}}{\Gamma, \Uparrow q \supset p \vdash C}}{\Gamma \Uparrow (p \supset q) \vee^{+} (q \supset p) \vdash C}}{\Gamma \Downarrow (p \supset q) \vee^{+} (q \supset p) \vdash C}}{\dfrac{\Gamma \Downarrow S^{-}[(p \supset q) \vee^{+} (q \supset p)] \vdash C}{\Gamma \Uparrow \cdot \vdash C}}$$

Conclusion

The design of focused proofs provides not only phases of inference rules but also the abstraction of “formula storage”.

A flexible way to exploit this abstraction is via a relation between indexes and polarized formulas.

Indexes can range widely and can be used to support the description of proof structures: such is their role in FPCs.

Indexes are purely proof theoretic and they do not need a model theoretic treatment to justify their use.