Relating ordered hypergraph rewriting and proof theory: Work in progress

Dale Miller

Inria Saclay &
LIX, Institut Polytechnique de Paris
Partout Team

TermGraph 2026, 19 July 2026

Art by Nadia Miller

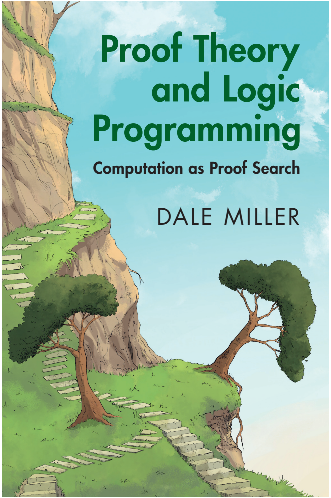
Proof Theory and Logic Programming: Computation as proof search, by Dale Miller

Published by Cambridge University Press, December 2025.

Preprint available from my web page. https://www.lix.polytechnique.fr/Labo/Dale.Miller/ptlp/

Organizes everything I learned about the intersection of proof theory and logic programming during four decades (1985-2025).

Uses classical, intuitionistic, and linear logic (first-order and higher-order) to design and reason about logic programs.



Art by Nadia Miller

Outline

The Abella proof assistant

Proof theory basics

Application of proof theory

Future work and conclusion

Abella

An interactive theorem prover well-suited for reasoning about the meta-theory of languages and logics involving binding.

- Various results on the λ-calculus involving big-step evaluation, small-step evaluation, and typing judgments
- Cut-admissibility for a sequent calculus
- Part 1a and Part 2a of the POPLmark challenge
- Several theorems about the π-calculus
- Takahashi's proof of the Church-Rosser theorem
- Tait's logical relations proof of weak normalization for STLC
- Girard's proof of strong normalization of STLC

Abella: A System for Reasoning about Relational Specifications by Baelde, Chaudhuri, Gacek, Miller, Nadathur, Tiu, and Wang. Journal of Formalized Reasoning 7(2), 2014, 1-89.

Abella is based on the sequent calculus

A goal in Abella is a sequent (à la Gentzen) displayed as

```
Variables: x1 ... xm
H1 : A1
...
Hn : An
============================
 C
```

- $x_1, \ldots, x_m$ are universally quantified variables (eigenvariables)
- `H1`, ..., `Hn` are hypothesis labels
- $A_1, \ldots, A_n$ are the hypotheses
- C is the conclusion

This goal denotes the sequent

$$x_1 : \tau_1, \ldots, x_m : \tau_m :: A_1, \ldots, A_n \vdash C.$$

The left-hand side is the context of this goal.

Abella is based on relations, not functions

Abella is based on quantificational logic: it contains

- relations (encoded as predicates) and
- term constructors.

Abella does not have a primitive notion of functions. These must be encoded as relations.
Example: Instead of writing $(a * x) + b \leq c$, write

$$\exists x_1.\exists x_2.\ \mathit{times}\ a\ x\ x_1 \wedge\ \mathit{plus}\ x_1\ b\ x_2 \wedge \mathit{lesseq}\ x_2\ c$$

Example sequents in Abella

Theorem: For all natural numbers d, n, and m, if $d \mid n$ and $d \mid m$ then $d \mid (n + m)$.

```
Theorem div_plus : forall D N M P,
  divides D N -> divides D M -> plus N M P ->
     divides D P.
```

Determinacy lemmas:

```
Theorem times_det : forall A B C D,
   times A B C -> times A B D -> C = D.
Theorem plus_det  : forall A B C D,
   plus  A B C -> plus  A B D -> C = D.
```

Totality lemmas:

```
Theorem times_total : forall A B,
   nat A -> nat B -> exists C, times A B C.
Theorem plus_total  : forall A B,
   nat A -> nat B -> exists C, times A B C.
```

An example goal

```
Variables: D N M P P1 P2 P6 Q XtY XtZ
H3: plus N M P
H4: times D P1 N
H5: times D P2 M
H12: plus P1 P2 Q
H13: times D Q P6
H14: times D P1 XtY
H15: times D P2 XtZ
H16: plus XtY XtZ P6
================================
 exists P1, times D P1 P
```

This goal specifies two ordered hypergraphs where variables denote nodes, e.g., the hypergraph for `times` contains the triples

```
(D,P1,N),(D,P2,M),(D,Q,P6),(D,P1,XtY),(D,P2,XtZ)
```

which, in turn, denote the equations

```
D*P1=N, D*P2=M, D*Q=P6, D*P1=XtY, D*P2=XtZ
```

Forward chaining as hypergraph rewriting

```
Theorem plus_comm  : forall M N K,
  plus M N K -> plus N M K.

Theorem plus_assoc : forall A B C AB ABC,
  plus A B AB -> plus AB C ABC ->
     exists BC, plus B C BC /\ plus A BC ABC.
```

How can such formulas be converted into hypergraph rewriting?

$A \supset B$

$A_1 \supset A_2 \supset (B_1 \wedge B_2) \equiv (A_1 \wedge A_2) \supset (B_1 \wedge B_2)$

The following are less obvious.

$$A_1 \supset A_2 \supset (B_1 \vee B_2) \qquad A_1 \supset A_2 \supset x = y$$

$$A_1 \supset A_2 \supset \exists x.B \qquad A_1 \supset A_2 \supset ff$$

Or should we be using linear logic?

$A_1 \multimap A_2 \multimap B \equiv (A_1 \otimes A_2) \multimap B$

Outline

The Abella proof assistant

Proof theory basics

Application of proof theory

Future work and conclusion

Elements of the sequent calculus: the micro rules

There are three kinds of (micro) rules:

1. Structural rules: weakening and contraction
2. Identity rules: cut and initial
3. Introduction rules put logical connectives into the concluding sequents.

$$\frac{\Sigma :: \Gamma \vdash B \quad \Sigma :: \Gamma \vdash C}{\Sigma :: \Gamma \vdash B \wedge C} \wedge_r \qquad \frac{\Sigma :: \Gamma, B \vdash C}{\Sigma :: \Gamma \vdash B \supset C} \supset_r$$

$$\frac{\Sigma :: \Gamma \vdash B \quad \Sigma :: \Gamma, C \vdash E}{\Sigma :: \Gamma, B \supset C \vdash E} \supset_l$$

Elements of the sequent calculus: the macro rules

Focused proof systems permit the systematic construction of macro rules, usually called synthetic inference rules.
For example, the formula

$$\forall x \forall y \forall z\,(path(x,y) \supset path(y,z) \supset path(x,z))$$

(which specifies that the predicate *path* is transitive) can be converted into the following synthetic rules:

$$\frac{\Sigma :: \Gamma \vdash path(x,y) \quad \Sigma :: \Gamma \vdash path(y,z)}{\Sigma :: \Gamma \vdash path(x,z)}\ \text{back-chaining}$$

$$\frac{\Sigma :: path(x,y),\ path(y,z),\ path(x,z),\ \Gamma \vdash E}{\Sigma :: path(x,y),\ path(y,z), \Gamma \vdash E}\ \text{forward-chaining}$$

Elements of the sequent calculus: equality

Here, we treat equality as a logical connective: that is, it has a left and right introduction rule.

$$\frac{}{\Sigma :: \Gamma \vdash t = t} =_r$$

$$\frac{}{\Sigma :: s = t, \Gamma \vdash C} =_l \qquad\qquad \frac{\Sigma\theta :: \Gamma\theta \vdash C\theta}{\Sigma :: s = t, \Gamma \vdash C} =_l$$

where t and s not unifiable. where $\theta = \mathrm{mgu}(t, s)$.

If we have no constructors (i.e., terms are just eigenvariables), then these rules simplify to the following.

$$\frac{}{\Sigma :: \Gamma \vdash x = x} =_r \qquad\qquad \frac{\Sigma, x :: \Gamma[x/y] \vdash C[x/y]}{\Sigma, x, y :: x = y, \Gamma \vdash C} =_l$$

Elements of the sequent calculus: linear vs classical zones

Sequents can be split into two zones: one set Γ for classical assumptions, and one multiset Δ for linear logic assumptions.

$$\frac{\Gamma, B; \Delta \vdash C}{\Gamma; \Delta \vdash B \supset C} \qquad \frac{\Gamma; \Delta, B \vdash C}{\Gamma; \Delta \vdash B \multimap C}$$

$$\frac{\Gamma; \Delta_1 \vdash B \quad \Gamma; \Delta_2, C \vdash E}{\Gamma; \Delta_1, \Delta_2, B \multimap C \vdash E} \qquad \frac{\Gamma; \Delta_1 \vdash B \quad \Gamma; \Delta_2 \vdash C}{\Gamma; \Delta_1, \Delta_2 \vdash B \otimes C}$$

Subexponentials and more zones

In linear logic, the ! and ? exponentials are not canonical: that is, linear logic supports

- many subexponentials !, !, !, etc.
- many zones: $\Gamma; \Delta_1; \Delta_2; \Delta_3 \vdash E$

Thus, different multisets (and, by extension, hypergraphs) can be located in different places. [Nigam & M, PPDP, 2009].

Preliminary work on encoding Milner's bigraphs in linear logic uses subexponentials [Chaudhuri & Reis, LPAR, 2015].

Outline

The Abella proof assistant

Proof theory basics

Application of proof theory

Future work and conclusion

Multiset rewriting as forward chaining

Focusing on formulas yields synthetic rules. For example:

$$\alpha : a \otimes b \multimap c \otimes d \qquad \frac{\Gamma; c, d, \Delta \vdash E}{\Gamma; a, b, \Delta \vdash E} \qquad \frac{\Gamma, a; c, d, \Delta \vdash E}{\Gamma, a; b, \Delta \vdash E}$$

$$\beta : e \multimap f \otimes g \qquad \frac{\Gamma; f, g, \Delta \vdash E}{\Gamma; e, \Delta \vdash E} \qquad \frac{\Gamma, e; f, g, \Delta \vdash E}{\Gamma, e; \Delta \vdash E}$$

We can do these actions in either order: $\alpha; \beta$ and $\beta; \alpha$ yield the same changes in sequents. <u>Multifocusing</u> also provides for $\alpha|\beta$.

Consider: $\gamma : a \otimes \mathit{lock} \multimap c \otimes \mathit{lock} \qquad \delta : b \otimes \mathit{lock} \multimap d \otimes \mathit{lock}$

Both $\gamma; \delta$ and $\delta; \gamma$ are possible but not their parallel composition, unless there is more than one occurrence of *lock* in Δ or $\mathit{lock} \in \Gamma$ (in which case, you don't really have a lock).

Naming and sharing

Administrative normal form (ANF) is a well known technique from compiling: apply functions only to named values and return a named value.

E.g., instead of $(a * x) + b \leq c$, write

```
name x1 = (a  * x) in
name x2 = (x1 + b) in
lesseq x2 c
```

This expression can be translated to the simple formula:

$$\exists x_1.\ \mathit{times}\ a\ x\ x_1 \wedge \exists x_2.\ \mathit{plus}\ x_1\ b\ x_2 \wedge \mathit{lesseq}\ x_2\ c$$

Naming and sharing (continued)

The use of names in this setting allows for sharing of common subexpressions.

```
name x1 = (f a a) in
name x2 = (f x1 x1) in
name x3 = (f x2 x2) in
name x4 = (f x3 x3) in
name x5 = (f x4 x4) in
x5;
```

If we remove the sharing, we get a big term containing, for example, 16 copies of the subexpression `(f a a)`.

If proofs are terms and formulas are types, then forward chaining in such proofs yield terms in ANF. See: [M & Wu, CSL 2023].

Forward chaining saturation

Geometric formulas: $\forall x_1 \ldots \forall x_m.\ A_1 \wedge \cdots \wedge A_n \supset G,$ where

- $A_1, \ldots, A_n$ are atomic
- G is built from atoms, $=$, $f\!f$, $t\!t$, $\wedge$, $\vee$, $\exists$.

Focused proof theory provides a notion of productive proofs.

Saturation occurs when a proof becomes unproductive.

Various syntactic characterizations can guarantee safe for saturation.

Using ANF, determinacy axioms, and saturation provides a proof theory (and proof search) account for congruence closure.

See: [Chaudhuri, Gantait, & M, Tableaux 2025]

Outline

Future work: Bindings

Hypergraphs and term graphs often encode expressions, and these expressions often contain binders.

The POPLmark challenge has revealed that the theorem proving community has not developed a canonical approach to encoding such binders: many proposals exist (de Bruijn numerals, nominal constants, HOAS, locally nameless, etc).

The term graph literature contains some approaches to capturing bindings: [Schneider, Kœhler, & Steuwer, PLDI 2025], [Tiurin, Ghica, & Hu, 2025], [Yasen & Ueda, 2017, 2018]

Future work: Bindings (continued)

The sequent calculus provides a natural, immediate, and powerful approach to binders, based on the notion of binder mobility.

> *There is no such thing as a free variable.*
> *(Perlis's epigram #47)*

The initial motivation for building the Abella prover was to capture this notion of binder mobility and the associated notion of λ-tree syntax and ∇-quantification.

Current situation: the notion of equality on λ-graph structures [Condoluci, Accattoli, Sacerdoti Coen, PPDP 2019] has a direct specification in Abella.

Future: Consider extending this work to Term Graphs more generally and compare with existing proposals.

Conclusions

- Proof-theoretic foundations has some overlaps with hypergraphs and term graphs.
- The implementations of hypergraph rewriting, e-graphs, etc, should find a home in Abella.
- I hope to force the proof theory approach to bindings to provide another approach to bindings in hypergraphs.

Thanks

Questions?

Art by Nadia Miller