A proof theory for model checking and arithmetic

Dale Miller

Inria Saclay & LIX, École Polytechnique
Palaiseau, France

Proof Theory for Automated Deduction,
Automated Deduction for Proof Theory

Funchal, Madeira, 24 October 2019

Goals of this talk:

1. Describe lessons learned from linear logic that have been useful in the proof theory of classical and intuitionistic logics.
2. Describe our first steps in applying those lessons to arithmetic. (Work in progress. Joint with Matteo Manighetti.)

Linear Logic

Girard proposed linear logic in 1987. Broadly speaking, it has had two kinds of impact.

As a new logic, it provided

- the λ-calculus (and functional programs) with *new* types;
- logic programming with *new* programs; and
- *new* proof structures, such as proof nets.

As the "logic behind (computational) logic", it introduced into classical and intuitionistic proof systems

- *polarization*,
- *focused* proofs, and
- new controls on contraction and weakening.

In the theme of this workshop

If you can control contractions, you can often turn proof theory results into algorithms.

- Herbrand disjunctions: The outermost $\exists$ quantifier is contracted *n* times (for some *n*). No other formulas are contracted.
- Expansion trees: Describe the contractions for all $\exists$ subformulas in the formula.

$$\textit{Contraction}: \quad \frac{\Gamma, !\,B, !\,B \vdash \Delta}{\Gamma, !\,B \vdash \Delta} \qquad \frac{\Gamma \vdash \Delta, ?\,B, ?\,B}{\Gamma \vdash \Delta, ?\,B}$$

$$\textit{Promotion}: \quad \frac{!\,\Gamma \vdash B, ?\,\Delta}{!\,\Gamma \vdash !\,B, ?\,\Delta} \qquad \frac{!\,\Gamma, B \vdash ?\,\Delta}{!\,\Gamma, ?\,B \vdash ?\,\Delta}$$

My linear logic tool box

Notation: $\mathbf{1}$, $\otimes$, $\mathbf{0}$, $\oplus$, $\top$, $\&$, $\bot$, $⅋$, $\multimap$, $!$, $?$, $(-)^\perp$

Terminology:

- *additive* connectives: $\mathbf{0}$, $\oplus$, $\top$, $\&$
- *multiplicative* connectives: $\mathbf{1}$, $\otimes$, $\bot$, $⅋$, $\multimap$
- *exponentials*: $!$, $?$
- *positive polarity*: $\mathbf{1}$, $\otimes$, $\mathbf{0}$, $\oplus$, $!$, $\exists$
- *negative polarity*: $\top$, $\&$, $\bot$, $⅋$, $?$, $\forall$, $\multimap$

Consider the right introduction rule of a logical connective.

- If it is invertible, the connective has *negative* polarity.
- If it is not invertible, the connective has *positive* polarity.

Linear logic negation flips polarities!

My linear logic tool box

Notation: t^+, $\wedge^+$, f^+, $\vee^+$, t^-, $\wedge^-$, f^-, $\vee^-$, $\multimap$, $!$, $?$, $(-)^\perp$

Terminology:

- *additive* connectives: f^+, $\vee^+$, t^-, $\wedge^-$
- *multiplicative* connectives: t^+, $\wedge^+$, f^-, $\vee^-$, $\multimap$
- *exponentials*: $!$, $?$
- *positive polarity*: t^+, $\wedge^+$, f^+, $\vee^+$, $!$, $\exists$
- *negative polarity*: t^-, $\wedge^-$, f^-, $\vee^-$, $?$, $\forall$, $\multimap$

Consider the right introduction rule of a logical connective.

- If it is invertible, the connective has *negative* polarity.
- If it is not invertible, the connective has *positive* polarity.

Linear logic negation flips polarities!

Example: Linear logic behind the LK vs LJ distinction

LJ is LK restricted: at most one formula on the right.

$$\Gamma \vdash \Delta \quad \text{where } \Delta \text{ is empty or a singleton.}$$

This restriction is equivalence to the following 2 conditions.

1. No contraction on the right.
2. In the (multiplicative) implication-left rule,

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

   the formula occurrence C cannot appear in the left premise.

In linear logic terms, (*i*) Δ is linear, (*ii*) Γ is intuitionistic (i.e., $!\,\Gamma$), and (*iii*) $A \supset B$ is encoded using two connectives $(!\,A) \multimap B$.

Example: Different information content in proofs

Classical, propositional logic with atoms, negated atoms, $\vee$, and $\wedge$.

Invertible rules

$$\frac{\vdash \Delta, B_1, B_2}{\vdash \Delta, B_1 \vee^- B_2}\ \vee^- \qquad \frac{\vdash \Delta, B_1 \quad \vdash \Delta, B_2}{\vdash \Delta, B_1 \wedge^- B_2}\ \wedge^-$$

Proof search proceeds by expanding into conjunctive normal form.

- Straightforward computation.
- Order of inference rules is not important.
- No contractions appear in proof.
- Weakening at leaves (only of literals).
- *Exponential* procedure.

Example: Different information content in proofs (con't)

Non-invertible rules

$$\frac{\vdash \Delta, B_i}{\vdash \Delta, B_1 \vee^+ B_2}\ \vee_i^+ \qquad \frac{\vdash \Delta_1, B_1 \quad \vdash \Delta_2, B_2}{\vdash \Delta_1, \Delta_2, B_1 \wedge^+ B_2}\ \wedge^+(\dagger)$$

The search for a proof of $\vdash B$ generates sequents of the form $\vdash B, C, \mathcal{L}$ where C is a subformula of B and $\mathcal{L}$ is a collection of literals.

- ($\dagger$) In classical logic, we can take $\Delta = \Delta_1 = \Delta_2 = \Delta_1, \Delta_2$.
- Contraction is needed but only on B.
- Proof construction consumes an external bit to decide $\vee_i^+$.

Proofs can be short since an oracle might contains some "clever" information.

Example: A short proof consuming three bits

Let C have several alternations of conjunction and disjunction and let $B = (p \vee C) \vee \neg p$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash B, p \quad , \neg p}\ \textit{init}}{\vdash B, p \vee C \quad , \neg p}\ *}{\vdash B, (p \vee C) \vee \neg p \quad , \neg p}\ *}{\vdash B, \neg p}\ \textit{contraction}}{\vdash B, (p \vee C) \vee \neg p}\ *}{\vdash B}\ \textit{contraction}
$$

The subformula C is avoided. Clever choices $*$ are injected at these points: right, left, left.

Focusing simply explained: proof search for $\Gamma \vdash \Delta$

- Do invertible introductions in any order, to exhaustion: positive connective on left; negative connective on right.
- Use the *decide* rule to pick a *focus* (includes the only case of contraction in intuitionistic logic).

$$\frac{\Gamma \Downarrow N \vdash \Delta}{\Gamma, N \vdash \Delta} \qquad \frac{\Gamma, N \Downarrow N \vdash \Delta}{\Gamma, N \vdash \Delta} \qquad \frac{\Gamma \vdash P \Downarrow}{\Gamma \vdash P}$$

- If the polarity flips in the focus, then use the *release* rule.

$$\frac{\Gamma, P \vdash \Delta}{\Gamma \Downarrow P \vdash \Delta} \qquad \frac{\Gamma \vdash N}{\Gamma \vdash N \Downarrow}$$

- Chose an *introduction* rule for non-atomic focus. Ask an oracle for help or consider backtracking. All premises are marked with $\Downarrow$.
- The remaining cases are the *initial* rules.

$$\frac{}{\Gamma \Downarrow N_a \vdash N_a}\ N_a \text{ neg atom} \qquad \frac{}{\Gamma, P_a \vdash P_a \Downarrow}\ P_a \text{ pos atom}$$

Atoms can have (non-canonical) polarity

Polarity can be assigned to atoms in a fixed but arbitrary fashion.

$$\dfrac{\dfrac{\Xi_1}{\Gamma \vdash Rab \Downarrow} \quad \dfrac{\dfrac{\Xi_2}{\Gamma \vdash Rbc \Downarrow} \quad \dfrac{\Xi_3}{\Gamma \Downarrow Rac \vdash \Delta}}{\Gamma \Downarrow Rbc \supset Rac \vdash \Delta} \supset L}{\dfrac{\Gamma \Downarrow Rab \supset Rbc \supset Rac \vdash \Delta}{\Gamma \Downarrow \forall x \forall y \forall z (Rxy \supset Ryz \supset Rxz) \vdash \Delta} \forall L \times 3} \supset L$$

If R-atoms have neg polarity, then Ξ_3 is initial and Δ is Rac. Also, Ξ_1 and Ξ_2 end with release. The synthetic rules is *back-chaining*.

$$\frac{\Gamma \vdash Rab \quad \Gamma \vdash Rbc}{\Gamma \vdash Rac}$$

If R-atoms have pos polarity, then Ξ_3 ends with release and Ξ_1, Ξ_2 are initial and Γ is Rab, Rbc, Γ'. The synthetic rules is *forward-chaining*.

$$\frac{\Gamma', Rab, Rbc, Rac \vdash \Delta}{\Gamma', Rab, Rbc \vdash \Delta}$$

Synthetic inference rules

In this way, geometric formulas yield inference rules that mention only atomic formulas: no logical connectives are visible in the rule.

See, for example, Negri's "from axioms to inference rules".

Synthetic rules built using focusing automatically satisfy cut-elimination.

Focused proofs provide a means for taking Gentzen's "atoms of inference rules" and building macro-level / synthetic inference rules ("molecules of inference").

Carry these observations to arithmetic

By arithmetic, I mean, logic extended with fixed points and induction rule. (I will not consider co-induction today.)

μMALL is a foundation for model-checking [Heath&M, JAR 2019].

The logic and much of the proof theory described here is part of the Abella theorem prover.

Abella

`http://abella-prover.org/`
`http://abella-prover.org/tutorial/try/`
runs in your browser

Arithmetic as a theory in logic

Peano's axioms fall into three groups.

- Equality is an equivalence relation.
- Zero and successors are constructors.
- Induction scheme

Peano Arithmetic is the classical logic treatment of these axioms.

Heyting Arithmetic is the intuitionistic logic treatment of these axioms.

Before we consider a linear logic treatment of arithmetic, it seems best to update this perspective on arithmetic more generally.

We first move away from Frege/Hilbert proofs to sequent calculus.

Arithmetic as a sequent calculus

We shall consider equality $=$ and the fixed point operator μ to be logical connectives with left and right introduction rules.

This approach was first considered by Girard and Schroeder-Heister independently [1991/92].

To capture *least* fixed points, an induction scheme is needed.

Various intuitionistic logics involving least and greatest fixed points have been consider in several papers during 1997-2011 by Gacek, McDowell, M, Momigliano, Nadathur, and Tiu.

Baelde and M have considered a linear logic variant as well.

Three ways to move beyond MALL

MALL has connectives $\wedge^+, t^+, \vee^+, f^+, \vee^-, f^-, \wedge^-, t^-$. It has no contraction and weakening. It is decidable and remains decidable if extended with first-order quantification.

Three notable extensions:

1. Girard [1987] added the *exponentials* $(!, ?)$ to get linear logic.
2. Girard [1991] and Liang & M [2009] added *classical and intuitionistic connectives* to get LU and LKU.
3. Baelde and M [2007] added *fixed points* to get μMALL.

We will illustrate that μMALL is well suited for describing model checking and inductive theorem proving. Note:

- Fixed point unfolding resembles contraction: $\mu B \bar{t} = B(\mu B)\bar{t}$.
- If B is *purely positive*, then $B \equiv\ !\, B$. In MALL: no interesting such formulas. In μMALL: a rich collection of such formulas.

Equality as a logical connective

$$\frac{}{\mathcal{X}\,;\vdash t = t}$$

$$\frac{}{\mathcal{X}\,;\Gamma, t = s \vdash \Delta}\ t \text{ and } s \text{ are not unifiable}$$

Here, $\mathcal{X}$ is the set of eigenvariables. Otherwise, set $\theta = \mathrm{mgu}(t, s)$:

$$\frac{\theta\mathcal{X}\,;\theta\Gamma \vdash \theta\Delta}{\mathcal{X}\,;\Gamma, t = s \vdash \Delta}$$

Here, $\theta\mathcal{X}$ is the result of removing from $\mathcal{X}$ variables in the domain of θ and then adding the variables free in the codomain of θ.

Unification is a black box attached to sequent calculus.

A failure (of unification) can be turned into a proof search success.

Proving the subset relation for two finite sets

Abbreviate z, $(s\ z)$, $(s\ (s\ z))$, $(s\ (s\ (s\ z)))$, etc by $\mathbf{0}$, $\mathbf{1}$, $\mathbf{2}$, $\mathbf{3}$, etc.

Let the sets $A = \{\mathbf{0}, \mathbf{1}\}$ and $B = \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$ be encoded as

$$\lambda x.\ x = \mathbf{0} \vee x = \mathbf{1} \quad \text{and} \quad \lambda x.\ x = \mathbf{0} \vee x = \mathbf{1} \vee x = \mathbf{2}.$$

To prove that A is a subset of B requires proving the formula $\forall x.Ax \supset Bx$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\cdot\,;\,\cdot \vdash \mathbf{0} = \mathbf{0}}}{\cdot\,;\,\cdot \vdash \mathbf{0} = \mathbf{0} \vee \mathbf{0} = \mathbf{1} \vee \mathbf{0} = \mathbf{2}}}{x\,;\, x = \mathbf{0} \vdash x = \mathbf{0} \vee x = \mathbf{1} \vee x = \mathbf{2}} \quad \dfrac{\dfrac{\dfrac{}{\cdot\,;\,\cdot \vdash \mathbf{1} = \mathbf{1}}}{\cdot\,;\,\cdot \vdash \mathbf{1} = \mathbf{0} \vee \mathbf{1} = \mathbf{1} \vee \mathbf{1} = \mathbf{2}}}{x\,;\, x = \mathbf{1} \vdash x = \mathbf{0} \vee x = \mathbf{1} \vee x = \mathbf{2}}}{\dfrac{x\,;\, x = \mathbf{0} \vee x = \mathbf{1} \vdash x = \mathbf{0} \vee x = \mathbf{1} \vee x = \mathbf{2}}{\cdot\,;\,\cdot \vdash \forall x.(x = \mathbf{0} \vee x = \mathbf{1}) \supset (x = \mathbf{0} \vee x = \mathbf{1} \vee x = \mathbf{2})}}$$

Exercise: Prove $\neg\forall x.Bx \supset Ax$.

Fixed points

The least fixed point μ is a series of operators indexed by their arity. We leave this arity implicit. Unfolding $\mu B t_1 \ldots t_n$ yields $B(\mu B) t_1 \ldots t_n$. Also, μ has positive polarity.

$$\frac{\Gamma \vdash B(\mu B)\bar{t}, \Delta}{\Gamma \vdash \mu B \bar{t}, \Delta}\ \mu R \qquad \frac{\Gamma, B(\mu B)\bar{t} \vdash \Delta}{\Gamma, \mu B \bar{t} \vdash \Delta}\ \mu L$$

The induction rule scheme (S is a higher-order variable).

$$\frac{\Gamma, S\bar{t} \vdash \Delta \qquad BS\bar{x} \vdash S\bar{x}}{\Gamma, \mu B \bar{t} \vdash \Delta}\ \mathit{Ind}$$

The rule for μL rule is admissible given the *Ind* rule.

Baelde [ToCL 2012] proved that μMALL satisfies cut-elimination and has a focused proof system μMALLF.

We set aside the induction rule (*Ind*) until the very end.

Examples of fixed point definitions

As a Horn clause theory

```
nat z.
nat (s X) :- nat X.
plus z X X.
plus (s X) Y (s Z) :- plus X Y Z.
```

We can convert these clauses into the following μ-expressions.

As fixed point definitions

$$nat = \mu\lambda N\lambda n(n = \mathbf{0} \vee^{+} \exists n'(n = s\ n' \wedge^{+} N\ n'))$$

$$\begin{aligned} plus =& \mu\lambda P\lambda n\lambda m\lambda p.(n = \mathbf{0} \wedge^{+} m = p) \vee^{+} \\ & \exists n'\exists p'(n = s\ n' \wedge^{+} p = s\ p' \wedge^{+} P\ n'\ m\ p') \end{aligned}$$

Note that μ and $=$ are positive, as are $\wedge^{+}$, $\vee^{+}$, and $\exists$. These are *purely positive* expressions.

Example: computing during the invertible phase

Consider searching for a proof of $\Gamma, \mathit{plus}\ \mathbf{2}\ \mathbf{3}\ x \vdash (Q\ x)$.
Using μL yields

$$\Gamma, ((\mathbf{2} = \mathbf{0} \wedge^+ \mathbf{3} = x) \vee^+ \exists n' \exists x' (\mathbf{2} = s\ n' \wedge^+ x = s\ x' \wedge^+ \mathit{plus}\ n'\ \mathbf{3}\ x')) \vdash (Q\ x).$$

The disjunction introduction rule yields two premises:
(1) $\Gamma, (\mathbf{2} = \mathbf{0} \wedge^+ \mathbf{3} = x) \vdash (Q\ x)$ is proved immediately.
(2)

$$\dfrac{\dfrac{\Gamma, \mathit{plus}\ \mathbf{1}\ \mathbf{3}\ x' \vdash (Q\ (s\ x'))}{\Gamma, (\mathbf{2} = s\ n' \wedge^+ x = s\ x' \wedge^+ \mathit{plus}\ n'\ \mathbf{3}\ x') \vdash (Q\ x)}}{\Gamma, (\exists n' \exists x' (\mathbf{2} = s\ n' \wedge^+ x = s\ x' \wedge^+ \mathit{plus}\ n'\ \mathbf{3}\ x')) \vdash (Q\ x)}$$

The invertible phase terminates with the premise

$$\Gamma \vdash (Q\ \mathbf{5})$$

Abstracting away the invertible phase, we obtain the following synthetic rule:

$$\frac{\vdash Q(\mathbf{5})}{\mathit{plus}\ \mathbf{2}\ \mathbf{3}\ x \vdash Q(x)}$$

The polarity ambiguity of singleton sets

Let P be a predicate of one argument such that

$$\vdash (\exists x.P(x)) \wedge (\forall x \forall y.P(x) \supset P(y) \supset x = y)$$

Thus, $\exists x.P(x) \wedge^{+} Q(x) \equiv \forall x.P(x) \multimap Q(x) \equiv Q(\iota P)$.

Assume that P is a purely positive formula.

A proof of $\Gamma \vdash \exists x.(P(x) \wedge^{+} Q(x)) \Downarrow$ *guesses* a term t and then proves $\Gamma \vdash P(t) \Downarrow$ and $\Gamma \vdash Q(t) \Downarrow$.

A proof of $\Gamma \vdash \forall x.P(x) \multimap Q(x)$ *computes* the value that satisfies P, starting with proving $\Gamma, P(y) \vdash Q(y)$. The completed phase has the premise $\Gamma \vdash Q(t)$.

When relations denote functions, we have singletons

For example, the predicate (*plus* **2 3**) denotes the singleton set containing only **5**.

During the invertible phase, proof search computes functions. This approach is different from

- Church and Hilbert who used choice operators (ϵ, ι) to convert some predicates to functions.
- Dowek et. al. who use confluent term rewriting to compute functions in the deduction-modulo setting.

For more, see [Gérard & M, CSL 2017; Gérard, PhD 2019].

More examples: paths in graph

Horn clauses (Prolog) can be encoded as purely positive fixed point expressions. For example, for specifying a (tiny) graph and its transitive closure:

```
step a b.  step b c.  step c b.
path X Z :- step X Z.
path X Z :- step X Y, path Y Z.
```

Write the `step` as the least fixed point expression

$$\mu(\lambda A \lambda x \lambda y.\, (x = a \wedge^{+} y = b) \vee^{+} (x = b \wedge^{+} y = c) \vee^{+} (x = c \wedge^{+} y = b))$$

Likewise, `path` can be encoded as the relation $path(\cdot, \cdot)$:

$$\mu(\lambda A \lambda x \lambda z.\, \texttt{step}\ x\ z \vee^{+} (\exists y.\, \texttt{step}\ x\ y \wedge^{+} A\, y\, z)).$$

These expressions use only positive connectives and no non-logical predicates.

Examples: reachability

There is no proof that there is a step from a to c.

$$\dfrac{\dfrac{\mathit{fail}}{\vdash (a = a \wedge^+ c = b) \vee^+ (a = b \wedge^+ c = c) \vee^+ (a = c \wedge^+ c = b)}}{\vdash \texttt{step}\ a\ c}$$

There is a proof that there is a path from a to c.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash \texttt{step}\ a\ b \qquad \vdash \mathit{path}\, b\, c}{\vdash \texttt{step}\ a\ b \wedge^+ \mathit{path}\, b\, c}}{\vdash \exists y.\, \texttt{step}\ a\ y \wedge^+ \mathit{path}\, y\, c}}{\vdash \texttt{step}\ a\ c \vee^+ (\exists y.\, \texttt{step}\ a\ y \wedge^+ \mathit{path}\, y\, c)}}{\vdash \mathit{path}(a, c)}$$

Examples: reachability (con't)

Below is a proof that the node a is not adjacent to c.

$$
\dfrac{
\dfrac{\dfrac{}{a = a, c = b \vdash \cdot}}{a = a \wedge^{+} c = b \vdash \cdot}
\quad
\dfrac{\dfrac{}{a = b, c = c \vdash \cdot}}{a = b \wedge^{+} c = c \vdash \cdot}
\quad
\dfrac{\dfrac{}{a = c, c = b \vdash \cdot}}{a = c \wedge^{+} c = b \vdash \cdot}
}{
\dfrac{(a = a \wedge^{+} c = b) \vee^{+} (a = b \wedge^{+} c = c) \vee^{+} (a = c \wedge^{+} c = b) \vdash \cdot}{\texttt{step}\ a\ c \vdash \cdot}
}
$$

In general, proofs by negation-as-finite-failure yield sequent calculus proofs in this setting. (Hallnäs & S-H, 1990)

Example: simulation

Let $P \xrightarrow{A} Q$ be a labeled transition system between processes and actions. Assume it is defined as a purely positive expression.

Let ν be the de Morgan dual of μ. Since we are only unfolding fixed points, μ and ν are extensionally the same although the polarity of ν is negative.

The following expressions denote simulation and bisimulation for this label transition systems.

$$\nu\big(\lambda S \lambda p \lambda q. \forall a \forall p'.\, p \xrightarrow{a} p' \multimap \exists q'.\, q \xrightarrow{a} q' \wedge^{+} S\, p'\, q'\big)$$

$$\begin{aligned}\nu(\lambda B \lambda p \lambda q. \quad & (\forall a \forall p'.\, p \xrightarrow{a} p' \multimap \exists q'.\, q \xrightarrow{a} q' \wedge^{+} B\, p'\, q') \\ \wedge^{-} \; & (\forall a \forall q'.\, q \xrightarrow{a} q' \multimap \exists p'.\, p \xrightarrow{a} p' \wedge^{+} B\, q'\, p'))\end{aligned}$$

Note that bisimulation has both conjunctions. Stirling's games for bisimulation [1996] are directly encoded in these focused proofs.

An example of a synthetic inference rules

$$\dfrac{\dfrac{\cdots \qquad \dfrac{\dfrac{\dfrac{\vdash sim(p_i, q_i)}{\vdash sim(p_i, q_i) \Downarrow}}{\vdash \exists Q'.q_0 \xrightarrow{a_i} Q' \wedge^+ sim(p_i, Q') \Downarrow}\; C}{\vdash \exists Q'.q_0 \xrightarrow{a_i} Q' \wedge^+ sim(p_i, Q')} \qquad \cdots}{P', A : p_0 \xrightarrow{A} P' \vdash \exists Q'.q_0 \xrightarrow{A} Q' \wedge^+ sim(P', Q')}\; B}{\vdash sim(p_0, q_0)}\; A$$

A contain introduction rules for $\forall$ and $\multimap$.

B consists of left invertible rules which generate all a_i and p_i such that $p_0 \xrightarrow{a_i} p_i$.

C is a sequence of $\Downarrow$ rules that proves that $q_0 \xrightarrow{a_i} q_i$.

Finally, the top-most inference rule is a release rule.

A proof theory for model checking

μMALL can provide a proof theory for model checking.

See [Heath & M, J. Automated Reasoning 2018].

Focusing can be used to design proof certificates for some common model checking problems.

- A path in a graph can be proof certificate for *reachability*.
- Connected components can be a proof certificate for *non-reachability*.
- A bisimulation can be a proof certificate for bisimilarity.
- A Hennessy-Milner modal formula can be a proof certificate for *non-bisimilarity*.

Next steps

Turing machines are easy to code in (pure) Prolog. Thus, we can define predicates as purely positive expression which capture a universal notions of computability.

The next challenges involve the induction scheme.

- What predicates can be proved total?
- Relate the arithmetic hierarchy (involving quantifier alternations) to focusing polarity.
- Design μLJ and μLJF and prove cut-elimination and completeness of focusing (mostly done).
- Design μLK and μLKF and establish cut-elimination and completeness (maybe impossible). In the most natural settings, completeness of focusing for μLKF would provide a simple method for extracting computational content of classical proofs of Π^0_2 formulas (something we do not expect).