On the Normative Power of Logical Frameworks

Frank Pfenning
In Memory of Gilles Dowek

Carnegie Mellon University

July 18, 2026

Collaboration

- Gilles' thesis on Automated Theorem Proving in the Calculus of Constructions [Dowek 1991]

Collaboration

- Gilles' thesis on Automated Theorem Proving in the Calculus of Constructions [Dowek 1991]
- Visited CMU January–June 1992

Collaboration

- Gilles' thesis on Automated Theorem Proving in the Calculus of Constructions [Dowek 1991]
- Visited CMU January–June 1992
- I was working on an implementation of the LF Logical Framework [Harper et al. 1987]
  - Type checking and inference for $\lambda\Pi$
  - Logic programming

Collaboration

- Gilles' thesis on Automated Theorem Proving in the Calculus of Constructions [Dowek 1991]
- Visited CMU January–June 1992
- I was working on an implementation of the LF Logical Framework [Harper et al. 1987]
  - Type checking and inference for $\lambda\Pi$
  - Logic programming
- Both employed higher-order unification
  - I assumed type inference to be undecidable, but couldn't find a reduction from higher-order unification (as I had for a form of polymorphic type inference)
  - Gilles found a surprising reduction from the *Post Correspondence Problem* to prove undecidability of type inference for $\lambda\Pi$ [Dowek 1993]

Collaboration, continued

- With Thérèse Hardin and Claude Kirchner [Dowek et al.'95], Gilles had developed an implementation of higher-order unification based on Abadi et al.'s calculus of explicit substitutions [Abadi et al.'91]

Collaboration, continued

- With Thérèse Hardin and Claude Kirchner [Dowek et al.'95], Gilles had developed an implementation of higher-order unification based on Abadi et al.'s calculus of explicit substitutions [Abadi et al.'91]
- Carsten Schürmann and I were working on the implementation of LF in the Twelf system, but realized that we needed unification over higher-order patterns [Miller'91] instead of higher-order unification.

Collaboration, continued

- With Thérèse Hardin and Claude Kirchner [Dowek et al.'95], Gilles had developed an implementation of higher-order unification based on Abadi et al.'s calculus of explicit substitutions [Abadi et al.'91]
- Carsten Schürmann and I were working on the implementation of LF in the Twelf system, but realized that we needed unification over higher-order patterns [Miller'91] instead of higher-order unification.
- Gilles, Thérèse, Claude, and I adapted their earlier work to higher-order patterns [Dowek'96]

Collaboration, continued

- With Thérèse Hardin and Claude Kirchner [Dowek et al.'95], Gilles had developed an implementation of higher-order unification based on Abadi et al.'s calculus of explicit substitutions [Abadi et al.'91]
- Carsten Schürmann and I were working on the implementation of LF in the Twelf system, but realized that we needed unification over higher-order patterns [Miller'91] instead of higher-order unification.
- Gilles, Thérèse, Claude, and I adapted their earlier work to higher-order patterns [Dowek'96]
- Carsten and I made this the algorithm at the core of Twelf [Pf. & Schürmann'99]
  - For type checking and reconstruction (see also [Pientka'13])
  - For logic programming (see also [Pientka & Pf.'03] [Pientka'06])
  - For meta-theorem proving and proof checking (see also [Schürmann & Pf.'98,'03] [Schürmann'00])

Collaboration, continued

- Talked and corresponded over the last 30 years
  - Background in Calculus of Constructions (Coq)
  - Developed interest in logical frameworks (Deduction Modulo)

Collaboration, continued

- Talked and corresponded over the last 30 years
  - Background in Calculus of Constructions (Coq)
  - Developed interest in logical frameworks (Deduction Modulo)
- Served on thesis committee of Thiago Felicissimo (2024)

Collaboration, continued

- Talked and corresponded over the last 30 years
  - Background in Calculus of Constructions (Coq)
  - Developed interest in logical frameworks (Deduction Modulo)
- Served on thesis committee of Thiago Felicissimo (2024)
- Serving on thesis committee of Thomas Traversié (2026)

What is a logical framework?

- Metalanguage for deductive systems
  - Specification
  - Algorithms
  - Metareasoning

What is a logical framework?

- Metalanguage for deductive systems
  - Specification
  - Algorithms
  - Metareasoning
- Consisting of two parts
  - A formal metalanguage
  - A representation methodology

The LF Logical Framework

- Metalanguage: (canonical) $\lambda\Pi$

The LF Logical Framework

- Metalanguage: (canonical) $\lambda\Pi$
- Representation methodology
  - Judgments-as-types
  - Hypothetical judgments as function types
  - Adequacy of representation via a compositional bijection between proofs and terms
  - Reduces proof checking to type checking

Example: Natural Deduction

- Judgment $A\ \mathit{true}$

$$\left[\ \frac{}{\Gamma, x{:}A\ \mathit{true} \vdash A}\ \mathrm{hyp}_x\ \right] \qquad \frac{\Gamma, x{:}A\ \mathit{true} \vdash B\ \mathit{true}}{\Gamma \vdash A \supset B} \supset I^x \qquad \frac{\Gamma \vdash A \supset B\ \mathit{true} \quad \Gamma \vdash A\ \mathit{true}}{\Gamma \vdash B\ \mathit{true}} \supset E$$

- Type $\mathit{true}\ \ulcorner A \urcorner$

```
true : prop -> type.
impI : (true A -> true B) -> true (imp A B).
impE : true (imp A B) -> true A -> true B.
```

- In LF

$$\ulcorner\Gamma\urcorner \vdash \ulcorner\mathcal{D}\urcorner : \mathit{true}\ \ulcorner A\urcorner$$

- These are not inductive types (in the usual sense)

Hypothetical Judgments

- $\Gamma \vdash A\ \mathit{true}$ (ND), or

Hypothetical Judgments

- $\Gamma \vdash A \; \mathit{true}$ (ND), or
- $\ulcorner \Gamma \urcorner \vdash M : \mathit{true} \ulcorner A \urcorner$ (LF)

Hypothetical Judgments

- $\Gamma \vdash A\ \mathit{true}$ (ND), or
- $\ulcorner\Gamma\urcorner \vdash M : \mathit{true}\ulcorner A\urcorner$ (LF)
- Semantically defined by substitution [Martin-Löf]

  If $\Gamma \vdash J$ *and* $\Delta, J \vdash J'$ *then* $\Delta, \Gamma \vdash J'$

  (approximately)

Hypothetical Judgments

- $\Gamma \vdash A \; \mathit{true}$ (ND), or
- $\ulcorner\Gamma\urcorner \vdash M : \mathit{true}\,\ulcorner A\urcorner$ (LF)
- Semantically defined by substitution [Martin-Löf]

  If $\Gamma \vdash J$ and $\Delta, J \vdash J'$ then $\Delta, \Gamma \vdash J'$

  (approximately)
- Hypotheses are persistent

Hypothetical Judgments

- $\Gamma \vdash A$ *true* (ND), or
- $\ulcorner\Gamma\urcorner \vdash M : \mathit{true}\ulcorner A\urcorner$ (LF)
- Semantically defined by substitution [Martin-Löf]

  If $\Gamma \vdash J$ *and* $\Delta, J \vdash J'$ *then* $\Delta, \Gamma \vdash J'$

  (approximately)
- Hypotheses are persistent
- Hypotheses need not be used

Example: Sequent Calculus

- Judgments $x : A\ \mathit{ante}$ and $A\ \mathit{succ}$

Example: Sequent Calculus

- Judgments $x : A$ *ante* and A *succ*
- Sequents formed as $\quad x_1{:}A_1\ \mathit{ante}, \ldots, x_n{:}A_n\ \mathit{ante} \vdash C\ \mathit{succ}$

Example: Sequent Calculus

- Judgments $x : A\ \mathit{ante}$ and $A\ \mathit{succ}$
- Sequents formed as $\quad x_1{:}A_1\ \mathit{ante}, \ldots, x_n{:}A_n\ \mathit{ante} \vdash C\ \mathit{succ}$
- Rules

$$\frac{\Gamma, x{:}A\ \mathit{ante}, y{:}B\ \mathit{ante} \vdash C\ \mathit{succ}}{\Gamma, z{:}A \wedge B\ \mathit{ante} \vdash C\ \mathit{succ}}\ \wedge L \qquad \frac{\Gamma \vdash A\ \mathit{succ} \quad \Gamma \vdash B\ \mathit{succ}}{\Gamma \vdash A \wedge B\ \mathit{succ}}\ \wedge R$$

$$\frac{}{\Gamma, x{:}A\ \mathit{ante} \vdash A\ \mathit{succ}}\ \mathrm{id}_x \qquad \frac{\Gamma \vdash A\ \mathit{succ} \quad \Gamma, x{:}A\ \mathit{ante} \vdash C\ \mathit{succ}}{\Gamma \vdash C\ \mathit{succ}}\ \mathrm{cut}^x$$

Example: Sequent Calculus

- Judgments $x : A\ \mathit{ante}$ and $A\ \mathit{succ}$
- Sequents formed as $\quad x_1{:}A_1\ \mathit{ante}, \ldots, x_n{:}A_n\ \mathit{ante} \vdash C\ \mathit{succ}$
- Rules

$$\frac{\Gamma, x{:}A\ \mathit{ante}, y{:}B\ \mathit{ante} \vdash C\ \mathit{succ}}{\Gamma, z{:}A \wedge B\ \mathit{ante} \vdash C\ \mathit{succ}}\ \wedge L \qquad \frac{\Gamma \vdash A\ \mathit{succ} \quad \Gamma \vdash B\ \mathit{succ}}{\Gamma \vdash A \wedge B\ \mathit{succ}}\ \wedge R$$

$$\frac{}{\Gamma, x{:}A\ \mathit{ante} \vdash A\ \mathit{succ}}\ \mathrm{id}_x \qquad \frac{\Gamma \vdash A\ \mathit{succ} \quad \Gamma, x{:}A\ \mathit{ante} \vdash C\ \mathit{succ}}{\Gamma \vdash C\ \mathit{succ}}\ \mathrm{cut}^x$$

- In LF ?

```
andL : (ante A -> ante B -> succ C)
     -> (ante (and A B) -> succ C).
andR : succ A -> succ B
     -> succ (and A B).
id : ante A -> succ A.
cut : succ A -> (ante A -> succ C)
     -> succ C.
```

Limits of Hypothetical Judgments

- This rule is subtle:

$$\frac{\Gamma, x{:}A\; ante, y{:}B\; ante \vdash C\; succ}{\Gamma, z{:}A \wedge B\; ante \vdash C\; succ} \wedge L$$

Limits of Hypothetical Judgments

- This rule is subtle:

$$\frac{\Gamma, x{:}A\ \mathit{ante}, y{:}B\ \mathit{ante} \vdash C\ \mathit{succ}}{\Gamma, z{:}A \wedge B\ \mathit{ante} \vdash C\ \mathit{succ}}\ \wedge L$$

- Only hypothetical judgment if $z{:}A \wedge B\ \mathit{ante}$ is preserved from conclusion to premise in $\supset L$!

Limits of Hypothetical Judgments

- This rule is subtle:

$$\frac{\Gamma, x{:}A\ \mathit{ante}, y{:}B\ \mathit{ante} \vdash C\ \mathit{succ}}{\Gamma, z{:}A \wedge B\ \mathit{ante} \vdash C\ \mathit{succ}}\ \wedge L$$

- Only hypothetical judgment if $z{:}A \wedge B\ \mathit{ante}$ is preserved from conclusion to premise in $\supset L$!
- Otherwise, persistence of hypotheses is violated

Limits of Hypothetical Judgments

- This rule is subtle:

$$\frac{\Gamma, x{:}A\ \mathit{ante}, y{:}B\ \mathit{ante} \vdash C\ \mathit{succ}}{\Gamma, z{:}A \wedge B\ \mathit{ante} \vdash C\ \mathit{succ}}\ \wedge L$$

- Only hypothetical judgment if $z{:}A \wedge B\ \mathit{ante}$ is preserved from conclusion to premise in $\supset L$!
- Otherwise, persistence of hypotheses is violated
- Consequently no longer in bijection with LF representation

Limits of Hypothetical Judgments

- This rule is subtle:

$$\frac{\Gamma, x{:}A\ \mathit{ante}, y{:}B\ \mathit{ante} \vdash C\ \mathit{succ}}{\Gamma, z{:}A \wedge B\ \mathit{ante} \vdash C\ \mathit{succ}}\ \wedge L$$

- Only hypothetical judgment if $z{:}A \wedge B\ \mathit{ante}$ is preserved from conclusion to premise in $\supset L$!
- Otherwise, persistence of hypotheses is violated
- Consequently no longer in bijection with LF representation
- Then, the judgment is a linear hypothetical judgment where hypotheses do not need to be persistent

Limits of Hypothetical Judgments

- This rule is subtle:

$$\frac{\Gamma, x{:}A\ \mathit{ante}, y{:}B\ \mathit{ante} \vdash C\ \mathit{succ}}{\Gamma, z{:}A \wedge B\ \mathit{ante} \vdash C\ \mathit{succ}}\ \wedge L$$

- Only hypothetical judgment if $z{:}A \wedge B\ \mathit{ante}$ is preserved from conclusion to premise in $\supset L$!
- Otherwise, persistence of hypotheses is violated
- Consequently no longer in bijection with LF representation
- Then, the judgment is a linear hypothetical judgment where hypotheses do not need to be persistent
- Defined semantically by linear substitution

Substructural extensions and alternatives

- Linear Logical Framework LLF [Cervesato & Pf.'96]
  - Linear hypothetical judgments as linear function types
  - Adequacy: a compositional bijection between proofs and linear terms
  - Reduces proof checking to type checking

Substructural extensions and alternatives

- Linear Logical Framework LLF [Cervesato & Pf.'96]
  - Linear hypothetical judgments as linear function types
  - Adequacy: a compositional bijection between proofs and linear terms
  - Reduces proof checking to type checking
- Alternative: polarized focused linear logic [Nigam & Miller'10]

Substructural extensions and alternatives

- Linear Logical Framework LLF [Cervesato & Pf.'96]
  - Linear hypothetical judgments as linear function types
  - Adequacy: a compositional bijection between proofs and linear terms
  - Reduces proof checking to type checking
- Alternative: polarized focused linear logic [Nigam & Miller'10]
  - Levels of adequacy
    1. Provability
    2. Bijections with proofs
    3. Bijections with partial proofs ($\sim$ compositional bijection)

Substructural extensions and alternatives

- Linear Logical Framework LLF [Cervesato & Pf.'96]
  - Linear hypothetical judgments as linear function types
  - Adequacy: a compositional bijection between proofs and linear terms
  - Reduces proof checking to type checking
- Alternative: polarized focused linear logic [Nigam & Miller'10]
  - Levels of adequacy
    1. Provability
    2. Bijections with proofs
    3. Bijections with partial proofs ($\sim$ compositional bijection)
  - Proofs are not terms

What is Gilles' contribution?

- LF and related systems (ironically) deal poorly with equality and computation

What is Gilles' contribution?

- LF and related systems (ironically) deal poorly with equality and computation
- Examples

$$\frac{\Gamma \vdash e : \tau \quad \tau \leq \sigma}{\Gamma \vdash e : \sigma}\,\text{sub} \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash A \equiv B}{\Gamma \vdash M : B}\,\text{conv}$$

$$\frac{\Gamma \vdash M : \Pi x{:}A.\, B \quad \Gamma \vdash N : A \quad [N/x{:}A]B = B'}{\Gamma \vdash M\, N : B'}\,\text{app}$$

What is Gilles' contribution?

- LF and related systems (ironically) deal poorly with equality and computation
- Examples

$$\frac{\Gamma \vdash e : \tau \quad \tau \leq \sigma}{\Gamma \vdash e : \sigma}\,\text{sub} \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash A \equiv B}{\Gamma \vdash M : B}\,\text{conv}$$

$$\frac{\Gamma \vdash M : \Pi x{:}A.\, B \quad \Gamma \vdash N : A \quad [N/x{:}A]B = B'}{\Gamma \vdash M\, N : B'}\,\text{app}$$

- Foundational and pragmatic issues

What is Gilles' contribution?

- LF and related systems (ironically) deal poorly with equality and computation
- Examples

$$\frac{\Gamma \vdash e : \tau \quad \tau \leq \sigma}{\Gamma \vdash e : \sigma}\ \mathsf{sub} \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash A \equiv B}{\Gamma \vdash M : B}\ \mathsf{conv}$$

$$\frac{\Gamma \vdash M : \Pi x{:}A.\, B \quad \Gamma \vdash N : A \quad [N/x{:}A]B = B'}{\Gamma \vdash M\, N : B'}\ \mathsf{app}$$

- Foundational and pragmatic issues
- There are too many distinct derivations / terms
  Adequacy is no longer a bijection

What is Gilles' contribution?

- LF and related systems (ironically) deal poorly with equality and computation
- Examples

$$\frac{\Gamma \vdash e : \tau \quad \tau \leq \sigma}{\Gamma \vdash e : \sigma}\ \mathsf{sub} \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash A \equiv B}{\Gamma \vdash M : B}\ \mathsf{conv}$$

$$\frac{\Gamma \vdash M : \Pi x{:}A.\, B \quad \Gamma \vdash N : A \quad [N/x{:}A]B = B'}{\Gamma \vdash M\, N : B'}\ \mathsf{app}$$

- Foundational and pragmatic issues
- There are too many distinct derivations / terms
  Adequacy is no longer a bijection
- All computations (except substitution) become deductions / trace terms
  Representations are bloated

Deduction Modulo

- Supports a representation-specific notion of equality

Deduction Modulo

- Supports a representation-specific notion of equality
- Formal metalanguage
  - First-order logic [Dowek'01]
  - $\lambda\Pi/\equiv$ [Cousineau & Dowek'07]

Deduction Modulo

- Supports a representation-specific notion of equality
- Formal metalanguage
  - First-order logic [Dowek'01]
  - $\lambda\Pi/\equiv$ [Cousineau & Dowek'07]
- Generalized representation methodology
  - Computations from rewrite rules
  - Equalities from rewrite rules

Deduction Modulo

- Supports a representation-specific notion of equality
- Formal metalanguage
  - First-order logic [Dowek'01]
  - $\lambda\Pi/\equiv$ [Cousineau & Dowek'07]
- Generalized representation methodology
  - Computations from rewrite rules
  - Equalities from rewrite rules
- Give up a priori
  - Decidability of framework equality
  - Decidability of type checking

Deduction Modulo

- Supports a representation-specific notion of equality
- Formal metalanguage
  - First-order logic [Dowek'01]
  - $\lambda\Pi/\equiv$ [Cousineau & Dowek'07]
- Generalized representation methodology
  - Computations from rewrite rules
  - Equalities from rewrite rules
- Give up a priori
  - Decidability of framework equality
  - Decidability of type checking
- Research program
  - Establish for specific encodings
  - Design general criteria (confluence, termination)

My related attempts

- Constraints and rewriting for LF
  - Constraint logic programming [Michaylov & Pf.'93]
  - Higher-order superposition [Virga'96]
  - Higher-order rewriting with dependent types [Virga'99]

My related attempts

- Constraints and rewriting for LF
  - Constraint logic programming [Michaylov & Pf.'93]
  - Higher-order superposition [Virga'96]
  - Higher-order rewriting with dependent types [Virga'99]
- Higher-order pattern unification [Dowek, Hardin, Kirchner, Pf.'96]

My related attempts

- Constraints and rewriting for LF
  - Constraint logic programming [Michaylov & Pf.'93]
  - Higher-order superposition [Virga'96]
  - Higher-order rewriting with dependent types [Virga'99]
- Higher-order pattern unification [Dowek, Hardin, Kirchner, Pf.'96]
- Implementation of Twelf in ML using higher-order pattern unification with constraints with Carsten Schürmann, Brigitte Pientka, Roberto Virga, Kevin Watkins, and many others

Dedukti

- Dedukti is a logical framework (including implementation) based on $\lambda\Pi/\equiv$ [Cousineau & Dowek'07] [Assaf et al.'23]

Dedukti

- Dedukti is a logical framework (including implementation) based on $\lambda\Pi/\equiv$ [Cousineau & Dowek'07] [Assaf et al.'23]
- Supports high-level encodings of many proof systems
  - Pure type systems, including universes
  - HOL Light

Dedukti

- Dedukti is a logical framework (including implementation) based on $\lambda\Pi/\equiv$ [Cousineau & Dowek'07] [Assaf et al.'23]
- Supports high-level encodings of many proof systems
  - Pure type systems, including universes
  - HOL Light
- Many contributors; more in other talks

Dedukti

- Dedukti is a logical framework (including implementation) based on $\lambda\Pi/\equiv$ [Cousineau & Dowek'07] [Assaf et al.'23]
- Supports high-level encodings of many proof systems
  - Pure type systems, including universes
  - HOL Light
- Many contributors; more in other talks
- Solves the problem of fixed equality with LF

Summary

- Logical frameworks are more than
  - a formal metalanguage
  - with a methodology for representation

Summary

- Logical frameworks are more than
  - a formal metalanguage
  - with a methodology for representation
- They embody fundamental principles underlying deduction
  - Hypothetical judgments
  - Linear hypothetical judgments
  - Algorithmic reading of inference systems
  - Proof irrelevance
  - Rewriting and computation

Summary

- Logical frameworks are more than
  - a formal metalanguage
  - with a methodology for representation
- They embody fundamental principles underlying deduction
  - Hypothetical judgments
  - Linear hypothetical judgments
  - Algorithmic reading of inference systems
  - Proof irrelevance
  - Rewriting and computation
- Gilles has been a major driver of logical frameworks research

Summary

- Logical frameworks are more than
  - a formal metalanguage
  - with a methodology for representation
- They embody fundamental principles underlying deduction
  - Hypothetical judgments
  - Linear hypothetical judgments
  - Algorithmic reading of inference systems
  - Proof irrelevance
  - Rewriting and computation
- Gilles has been a major driver of logical frameworks research
- I miss Gilles, both personally and professionally

A Dedication

- Gilles has been a major contributor (RTA 6, TLCA 5, FSCD 3)

A Dedication

- Gilles has been a major contributor (RTA 6, TLCA 5, FSCD 3)
- Served as PC Chair of RTA-TLCA 2014

A Dedication

- Gilles has been a major contributor (RTA 6, TLCA 5, FSCD 3)
- Served as PC Chair of RTA-TLCA 2014
- I dedicated this year's FSCD proceedings to the memory of Gilles Dowek