A uniform interpolant calculator with Coq

Iris van der Giessen (UvA)

j.w.w.

Hugo Férée (Université Paris Cité, CNRS, IRIF)
Sam van Gool (Université Paris Cité, CNRS, IRIF)
Ian Shillito (University of Birmingham)

LLAMA Seminar – Amsterdam – 5 February 2025

Uniform interpolation in many forms

Knowledge representation

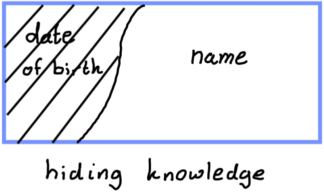
Automated deduction

Proof Theory

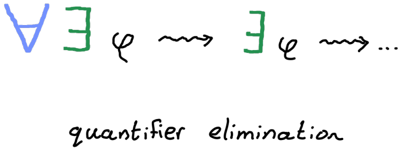
Algebra

Uniform interpolation in many forms

Knowledge representation



Automated deduction

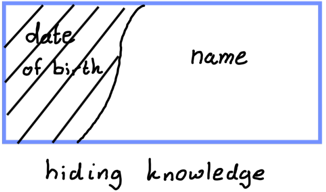


Proof Theory

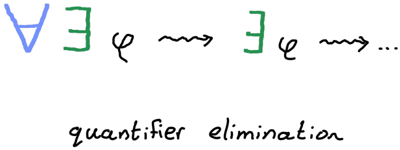
Algebra

Uniform interpolation in many forms

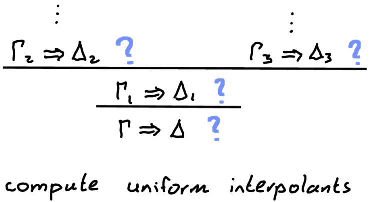
Knowledge representation



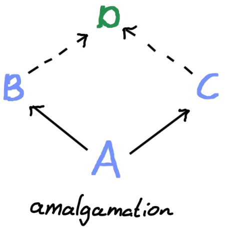
Automated deduction



Proof Theory



Algebra



Our work in short

What?
Mechanised uniform interpolation (for modal logics) in Coq ( )

Why?
Our formalisation provides the following benefits:

Our work in short

What?

Mechanised uniform interpolation (for modal logics) in Coq ( )

- Proof-theoretic construction of uniform interpolants in Coq
- Correctness of construction verified in Coq
- Online Uniform Interpolant Calculator extracted from Coq

Why?

Our formalisation provides the following benefits:

Our work in short

What?

Mechanised uniform interpolation (for modal logics) in Coq ()

- Proof-theoretic construction of uniform interpolants in Coq
- Correctness of construction verified in Coq
- Online Uniform Interpolant Calculator extracted from Coq

Why?

Our formalisation provides the following benefits:

- Uniform interpolants for new logics $\rightarrow$ iSL
- Fix bugs in paper-and-pen proofs $\rightarrow$ GL
- Use calculator to derive theoretical results [Kocsis '23]

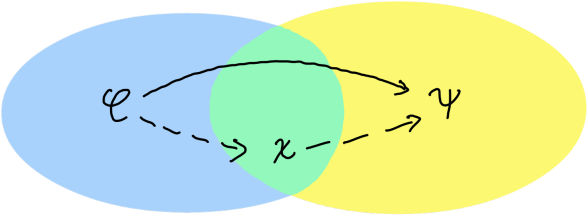
Craig interpolation

Language: $\varphi ::= p \in Var \mid \bot \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \Box\varphi$

Definition

Let L be a (modal) logic. L has the *Craig interpolation property* if for any $\vdash_{\mathsf{L}} \varphi \to \psi$ there exists formula χ such that both:

1. $Var(\chi) \subseteq Var(\varphi) \cap Var(\psi)$
2. $\vdash_{\mathsf{L}} \varphi \to \chi$ and $\vdash_{\mathsf{L}} \chi \to \psi$



Uniform interpolation

Slogan

A Craig interpolant for $\varphi \to \psi$ that only depends on φ is *uniform*.

Uniform interpolation

Slogan

A Craig interpolant for $\varphi \to \psi$ that only depends on φ is *uniform*.

Definition

Logic L has the *(right) uniform interpolation property* if for any formula φ and atom p there exists formula χ such that:

1. $Var(\chi) \subseteq Var(\varphi) \setminus \{p\}$
2. For all p-free formulas ψ:

$$\text{if } \vdash_{\mathsf{L}} \varphi \to \psi, \text{ then } \vdash_{\mathsf{L}} \varphi \to \chi \text{ and } \vdash_{\mathsf{L}} \chi \to \psi$$

Uniform interpolation

Slogan

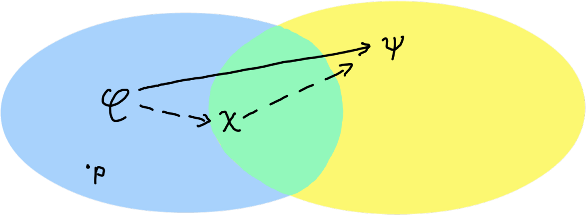
A Craig interpolant for $\varphi \to \psi$ that only depends on φ is *uniform*.

Definition

Logic L has the *(right) uniform interpolation property* if for any formula φ and atom *p* there exists formula χ such that:

1. $Var(\chi) \subseteq Var(\varphi) \setminus \{p\}$
2. For all *p*-free formulas ψ:

$$\text{if } \vdash_L \varphi \to \psi, \text{ then } \vdash_L \varphi \to \chi \text{ and } \vdash_L \chi \to \psi$$



Uniform interpolation

Slogan

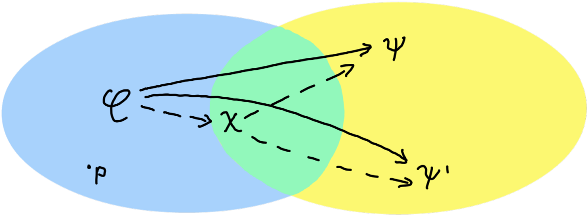
A Craig interpolant for $\varphi \to \psi$ that only depends on φ is *uniform*.

Definition

Logic L has the *(right) uniform interpolation property* if for any formula φ and atom p there exists formula χ such that:

1. $Var(\chi) \subseteq Var(\varphi) \setminus \{p\}$
2. For all p-free formulas ψ:

$$\text{if } \vdash_{\mathsf{L}} \varphi \to \psi, \text{ then } \vdash_{\mathsf{L}} \varphi \to \chi \text{ and } \vdash_{\mathsf{L}} \chi \to \psi$$



Uniform interpolation

Slogan

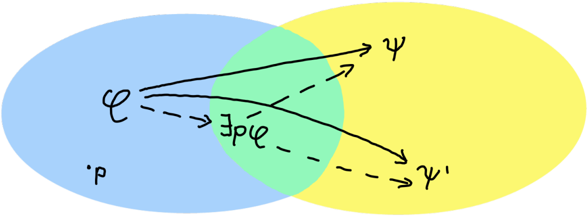
A Craig interpolant for $\varphi \to \psi$ that only depends on φ is *uniform*.

Definition

Logic L has the *(right) uniform interpolation property* if for any formula φ and atom p there exists formula $\exists p\varphi$ such that:

1. $Var(\exists p\varphi) \subseteq Var(\varphi) \setminus \{p\}$
2. For all p-free formulas ψ:

$$\text{if } \vdash_{\mathsf{L}} \varphi \to \psi, \text{ then } \vdash_{\mathsf{L}} \varphi \to \exists p\varphi \text{ and } \vdash_{\mathsf{L}} \exists p\varphi \to \psi$$



Uniform interpolation

Slogan

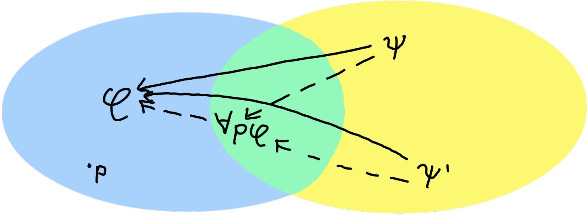
A Craig interpolant for $\psi \to \varphi$ that only depends on φ is *uniform*.

Definition

Logic L has the *(left) uniform interpolation property* if for any formula φ and atom *p* there exists formula $\forall p \varphi$ such that:

1. $Var(\forall p\varphi) \subseteq Var(\varphi) \setminus \{p\}$
2. For all *p*-free formulas ψ:

$$\text{if } \vdash_L \psi \to \varphi, \text{ then } \vdash_L \psi \to \forall p\varphi \text{ and } \vdash_L \forall p\varphi \to \varphi$$



Example: classical logic

Uniform interpolants form strongest ($\exists$) and weakest ($\forall$) interpolants.

Example

Let

$$\varphi = p \wedge q \wedge r$$
$$\psi = p \vee q \vee s$$

We have $\vdash_{\mathsf{CPC}} \varphi \to \psi$.

Example: classical logic

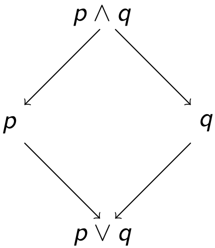
Uniform interpolants form strongest ($\exists$) and weakest ($\forall$) interpolants.

Example

Let

$$\varphi = p \wedge q \wedge r$$
$$\psi = p \vee q \vee s$$

We have $\vdash_{\mathsf{CPC}} \varphi \to \psi$.
Its Craig interpolants are:



Example: classical logic

Uniform interpolants form strongest ($\exists$) and weakest ($\forall$) interpolants.

Example

Let

$$\varphi = p \wedge q \wedge r$$
$$\psi = p \vee q \vee s$$

We have $\vdash_{\mathsf{CPC}} \varphi \to \psi$.
Its Craig interpolants are:

$$\exists r \varphi = (p \wedge q \wedge \top) \vee (p \wedge q \wedge \bot) = p \wedge q$$

$$p \qquad\qquad q$$

$$\forall s \psi = (p \vee q \vee \top) \wedge (p \vee q \vee \bot) = p \vee q$$

Difficulties in constructing uniform interpolants

Given a logic with Craig interpolation, we define

$$\exists p.\varphi(p,\overline{q}) := \bigwedge\{\psi(\overline{q}) \ \mid\ \vdash \varphi \to \psi\}$$
$$\forall p.\varphi(p,\overline{q}) := \bigvee\{\psi(\overline{q}) \ \mid\ \vdash \psi \to \varphi\}$$

Difficulties in constructing uniform interpolants

Given a logic with Craig interpolation, we define

$$\exists p.\varphi(p, \overline{q}) := \bigwedge\{\psi(\overline{q}) \mid \vdash \varphi \to \psi\}$$
$$\forall p.\varphi(p, \overline{q}) := \bigvee\{\psi(\overline{q}) \mid \vdash \psi \to \varphi\}$$

Problem: These expressions may NOT be a formula! (IPC, K, ...)
NB: It works for locally tabular logics (CPC, S5)

Difficulties in constructing uniform interpolants

Given a logic with Craig interpolation, we define

$$\exists p.\varphi(p,\overline{q}) := \bigwedge\{\psi(\overline{q}) \ \mid \ \vdash \varphi \to \psi\}$$
$$\forall p.\varphi(p,\overline{q}) := \bigvee\{\psi(\overline{q}) \ \mid \ \vdash \psi \to \varphi\}$$

Problem: These expressions may NOT be a formula! (IPC, K, ...)
NB: It works for locally tabular logics (CPC, S5)

Idea:
Replace it by

$$E_p(\varphi) := \bigwedge \mathcal{E}_p(\varphi) \quad \text{and} \quad A_p(\varphi) := \bigvee \mathcal{A}_p(\varphi)$$

where $\mathcal{E}_p(\varphi)$ is a finite basis for the set of consequences of φ (dually for $\mathcal{A}_p(\varphi)$)

Pitts's solution: a finite base via proof theory

Idea of Pitts (1992)
Define

$$E_p(\varphi) := \bigwedge \mathcal{E}_p(\varphi) \quad \text{and} \quad A_p(\varphi) := \bigvee \mathcal{A}_p(\varphi)$$

where $\mathcal{E}_p(\varphi)$ represents all possible consequences of φ in a finite proof search in a suitable sequent calculus (dually for $\mathcal{A}_p(\varphi)$)

Pitts's solution: a finite base via proof theory

Idea of Pitts (1992)
Define

$$E_p(\varphi) := \bigwedge \mathcal{E}_p(\varphi) \quad \text{and} \quad A_p(\varphi) := \bigvee \mathcal{A}_p(\varphi)$$

where $\mathcal{E}_p(\varphi)$ represents all possible consequences of φ in a finite proof search in a suitable sequent calculus (dually for $\mathcal{A}_p(\varphi)$)

Examples

- IPC [Pitts '92]
- Classical modal logics K, T, GL [Bílková '06]
- Intuitionistic modal logics iK, iKD [Iemhoff '19]
- Our paper: K, GL and iSL

Example in modal logic K

Consider formula $\varphi = \Box p \vee \Box\neg p$. We compute $A_p(\varphi)$ as follows:

Example in modal logic K

Consider formula $\varphi = \Box p \vee \Box\neg p$. We compute $A_p(\varphi)$ as follows:

- Finite proof search

$$\dfrac{\dfrac{\Rightarrow p \qquad \dfrac{p \Rightarrow}{\Rightarrow \neg p}\,R\neg}{\Rightarrow \Box p, \Box\neg p}\,\text{Proof search on } \Box\text{-rule}}{\Rightarrow \Box p \vee \Box\neg p}\,R\vee$$

Example in modal logic K

Consider formula $\varphi = \Box p \vee \Box \neg p$. We compute $A_p(\varphi)$ as follows:

- Finite proof search

$$\dfrac{\dfrac{\Rightarrow p \quad (\bot) \qquad \dfrac{p \Rightarrow \quad (\bot)}{\Rightarrow \neg p \quad (\bot)}\ R\neg}{\Rightarrow \Box p, \Box \neg p \quad (\Box\bot \vee \Box\bot)}\ \text{Proof search on } \Box\text{-rule}}{\Rightarrow \Box p \vee \Box \neg p \quad (\Box\bot \vee \Box\bot)}\ R\vee$$

- Compute uniform interpolants. Result: $A_p(\varphi) = \Box\bot$
-

Example in modal logic K

Consider formula $\varphi = \Box p \vee \Box\neg p$. We compute $A_p(\varphi)$ as follows:

- Finite proof search

$$
\dfrac{\dfrac{\Rightarrow p \quad (\bot) \qquad \dfrac{p \Rightarrow \quad (\bot)}{\Rightarrow \neg p \quad (\bot)}\, R\neg}{\Rightarrow \Box p, \Box\neg p \quad (\Box\bot \vee \Box\bot)}\text{ Proof search on }\Box\text{-rule}}{\Rightarrow \Box p \vee \Box\neg p \quad (\Box\bot \vee \Box\bot)}\, R\vee
$$

- Compute uniform interpolants. Result: $A_p(\varphi) = \Box\bot$
- Check correctness, e.g., $A_p(\varphi)$ is p-free, ...

Example in modal logic K

Consider formula $\varphi = \Box p \vee \Box\neg p$. We compute $A_p(\varphi)$ as follows:

- Finite proof search

$$
\dfrac{\Rightarrow p \quad (\bot) \qquad \dfrac{p \Rightarrow \quad (\bot)}{\Rightarrow \neg p \quad (\bot)}\, R\neg}{\dfrac{\Rightarrow \Box p, \Box\neg p \quad (\Box\bot \vee \Box\bot)}{\Rightarrow \Box p \vee \Box\neg p \quad (\Box\bot \vee \Box\bot)}\, R\vee}\ \text{Proof search on } \Box\text{-rule}
$$

- Compute uniform interpolants. Result: $A_p(\varphi) = \Box\bot$
- Check correctness, e.g., $A_p(\varphi)$ is p-free, . . .

Disclaimer

This is an easy example. In general, the uniform interpolant is computed via recursive calls on *multiple* proof searches.

Terminating sequent calculi

Disclaimer

A terminating calculus may NOT be *sufficient* or *necessary* to prove uniform interpolation. It is important to design a good calculus.

Terminating sequent calculi

Disclaimer

A terminating calculus may NOT be *sufficient* or *necessary* to prove uniform interpolation. It is important to design a good calculus.

For intuitionistic logic IL [Dyckhoff '92],[Hudelmaier '88],[Vorob'ev '52]

- Replace non-terminating rule

$$\frac{\Gamma, \varphi \to \psi \Rightarrow \varphi \qquad \Gamma, \psi \Rightarrow \gamma}{\Gamma, \varphi \to \psi \Rightarrow \gamma}\ \mathrm{L}{\to}$$

Terminating sequent calculi

Disclaimer

A terminating calculus may NOT be *sufficient* or *necessary* to prove uniform interpolation. It is important to design a good calculus.

For intuitionistic logic IL [Dyckhoff '92],[Hudelmaier '88],[Vorob'ev '52]

- Replace non-terminating rule

$$\frac{\Gamma, \varphi \to \psi \Rightarrow \varphi \qquad \Gamma, \psi \Rightarrow \gamma}{\Gamma, \varphi \to \psi \Rightarrow \gamma}\ \text{L}\!\to$$

  by four other rules based on the main connective in φ, e.g.,

$$\frac{\Gamma, \varphi_1 \to \psi \Rightarrow \varphi_0 \to \varphi_1 \qquad \Gamma, \psi \Rightarrow \gamma}{\Gamma, (\varphi_0 \to \varphi_1) \to \psi \Rightarrow \gamma}\ \text{L}\!\to\to$$

Terminating sequent calculi

Disclaimer

A terminating calculus may NOT be *sufficient* or *necessary* to prove uniform interpolation. It is important to design a good calculus.

For intuitionistic logic IL [Dyckhoff '92],[Hudelmaier '88],[Vorob'ev '52]

- Replace non-terminating rule

$$\frac{\Gamma, \varphi \to \psi \Rightarrow \varphi \qquad \Gamma, \psi \Rightarrow \gamma}{\Gamma, \varphi \to \psi \Rightarrow \gamma} \; \mathsf{L}{\to}$$

  by four other rules based on the main connective in φ, e.g.,

$$\frac{\Gamma, \varphi_1 \to \psi \Rightarrow \varphi_0 \to \varphi_1 \qquad \Gamma, \psi \Rightarrow \gamma}{\Gamma, (\varphi_0 \to \varphi_1) \to \psi \Rightarrow \gamma} \; \mathsf{L}{\to}{\to}$$

- Strong termination in multiset ordering
  → We use a *numerical* ordering assigning weights to sequents

Terminating sequent calculi

For classical Gödel-Löb logic GL: [Bílková 06]

- Modal rule

$$\frac{\Box\Gamma, \Gamma, \Box\varphi \Rightarrow \varphi}{\Box\Gamma \Rightarrow \Box\varphi} \text{ GL-rule}$$

- Strong termination using *local loop-check*
- Unusual termination ordering:
  'more boxed formulas on the left yield smaller sequent'

Terminating sequent calculi

For classical Gödel-Löb logic GL: [Bílková 06]

- Modal rule

$$\frac{\Box\Gamma, \Gamma, \Box\varphi \Rightarrow \varphi}{\Box\Gamma \Rightarrow \Box\varphi} \text{ GL-rule}$$

- Strong termination using *local loop-check*
- Unusual termination ordering:
  'more boxed formulas on the left yield smaller sequent'

The quest for a suitable calculus for iSL:

- Terminating with local loop-check [vdG & Iemhoff '20] ✗
- Strongly terminating [Shillito, vdG, Goré & Iemhoff '23] ✓
- Terminating and subformula property [Fiorentini & Ferrari '24] ?

Construction for iSL

	Γ matches	$\mathcal{E}_p(\Gamma)$ contains
($\mathsf{E}_p^{\mathsf{IL}}0$)	$\Gamma', \bot$	$\bot$
($\mathsf{E}_p^{\mathsf{IL}}1$)	Γ', q	$\mathsf{E}_p(\Gamma') \wedge q$
($\mathsf{E}_p^{\mathsf{IL}}2$)	$\Gamma', \psi_1 \wedge \psi_2$	$\mathsf{E}_p(\Gamma', \psi_1, \psi_2)$
($\mathsf{E}_p^{\mathsf{IL}}3$)	$\Gamma', \psi_1 \vee \psi_2$	$\mathsf{E}_p(\Gamma', \psi_1) \vee \mathsf{E}_p(\Gamma', \psi_2)$
($\mathsf{E}_p^{\mathsf{IL}}4$)	$\Gamma', (q \to \psi)$	$q \to \mathsf{E}_p(\Gamma', \psi)$
($\mathsf{E}_p^{\mathsf{IL}}5$)	$\Gamma', p, (p \to \psi)$	$\mathsf{E}_p(\Gamma', p, \psi)$
($\mathsf{E}_p^{\mathsf{IL}}6$)	$\Gamma', (\delta_1 \wedge \delta_2) \to \delta_3)$	$\mathsf{E}_p(\Gamma', (\delta_1 \to (\delta_2 \to \delta_3)))$
($\mathsf{E}_p^{\mathsf{IL}}7$)	$\Gamma', (\delta_1 \vee \delta_2) \to \delta_3)$	$\mathsf{E}_p(\Gamma', (\delta_1 \to \delta_3), (\delta_2 \to \delta_3)))$
($\mathsf{E}_p^{\mathsf{IL}}8$)	$\Gamma', (\delta_1 \to \delta_2) \to \delta_3)$	$[\mathsf{E}_p(\Gamma', (\delta_2 \to \delta_3)) \to \mathsf{A}_p(\Gamma', (\delta_2 \to \delta_3) \Rightarrow \delta_1 \to \delta_2)]$ $\to \mathsf{E}_p(\Gamma', \delta_3)$
($\mathsf{E}_p^{\mathsf{iSL}}9$)	$\Gamma', \Box\delta$	$\Box\mathsf{E}_p(\otimes\Gamma', \delta)$
($\mathsf{E}_p^{\mathsf{iSL}}10$)	$\Gamma', (\Box\delta_1 \to \delta_2)$	$\Box[\mathsf{E}_p(\otimes\Gamma', \delta_2, \Box\delta_1) \to \mathsf{A}_p(\otimes\Gamma', \delta_2, \Box\delta_1 \Rightarrow \delta_1)]$ $\to \mathsf{E}_p(\Gamma', \delta_2)$

	s matches	$\mathcal{A}_p(s)$ contains
($\mathsf{A}_p^{\mathsf{IL}}1$)	$\Gamma, q \Rightarrow \varphi$	$\mathsf{A}_p(\Gamma \Rightarrow \varphi)$
($\mathsf{A}_p^{\mathsf{IL}}2$)	$\Gamma, \psi_1 \wedge \psi_2 \Rightarrow \varphi$	$\mathsf{A}_p(\Gamma, \psi_1, \psi_2 \Rightarrow \varphi)$
($\mathsf{A}_p^{\mathsf{IL}}3$)	$\Gamma, \psi_1 \vee \psi_2 \Rightarrow \varphi$	$[\mathsf{E}_p(\Gamma, \psi_1) \to \mathsf{A}_p(\Gamma, \psi_1 \Rightarrow \varphi)] \wedge$ $[\mathsf{E}_p(\Gamma, \psi_2) \to \mathsf{A}_p(\Gamma, \psi_2 \to \varphi)]$
($\mathsf{A}_p^{\mathsf{IL}}4$)	$\Gamma, (q \to \psi) \Rightarrow \varphi$	$q \wedge \mathsf{A}_p(\Gamma, \psi \Rightarrow \varphi)$
($\mathsf{A}_p^{\mathsf{IL}}5$)	$\Gamma, p, (p \to \psi) \Rightarrow \varphi$	$\mathsf{A}_p(\Gamma, \psi \Rightarrow \varphi)$

Fixing a bug in the proof of GL

An easily made mistake

The *a priori* assumption that algorithm constructs equivalent uniform interpolants for *similar* sequents.

Fixing a bug in the proof of GL

An easily made mistake

The *a priori* assumption that algorithm constructs equivalent uniform interpolants for *similar* sequents.

Problem for GL

Similar sequents may have different proof search trees. E.g.,

$$\dfrac{\Rightarrow p \qquad q \Rightarrow}{p \to q \Rightarrow}\, {\scriptstyle L\to} \qquad \dfrac{\dfrac{\Rightarrow p, p \qquad q \Rightarrow p}{p \to q \Rightarrow p}\, {\scriptstyle L\to} \qquad \dfrac{q \Rightarrow p \qquad q, q \Rightarrow}{p \to q, q \Rightarrow}\, {\scriptstyle L\to}}{p \to q, p \to q \Rightarrow}\, {\scriptstyle L\to}$$

This results in different constructions of $\mathcal{A}_p$ circumventing the fixpoint argument used in [Bílková '06] for GL.

Fixing a bug in the proof of GL

An easily made mistake

The *a priori* assumption that algorithm constructs equivalent uniform interpolants for *similar* sequents.

Problem for GL

Similar sequents may have different proof search trees. E.g.,

$$\dfrac{\Rightarrow p \qquad q \Rightarrow}{p \to q \Rightarrow}\ {\scriptstyle L\to} \qquad \dfrac{\dfrac{\Rightarrow p, p \qquad q \Rightarrow p}{p \to q \Rightarrow p}\ {\scriptstyle L\to} \qquad \dfrac{q \Rightarrow p \qquad q, q \Rightarrow}{p \to q, q \Rightarrow}\ {\scriptstyle L\to}}{p \to q, p \to q \Rightarrow}\ {\scriptstyle L\to}$$

This results in different constructions of $\mathcal{A}_p$ circumventing the fixpoint argument used in [Bílková '06] for GL.

Our solution

Enforce equality of sequents in the computation by using contraction

Our formalisation in Coq

Why?

- Better understand a 30-year old proof [Férée & v. Gool '23]
- Use the implementation to derive theoretical results
  - Non-definability of connectives [Kocsis '23]
- 'Easily' derive uniform interpolants for new logics → iSL
- Fix bugs in paper-and-pen proofs → GL

Challenges of our formalisation

- Prove termination of the algorithm
- Formalise 'easy' admissibility and inversion of rules [Dyckhoff & Negri '20]
- Work in progress: output readable interpolants

It's time for a

Coq $\Rightarrow$ OCaml program $\Rightarrow$ Javascript program

Coq documentation: `https://hferee.github.io/UIML/toc.html`
Calculator: `https://hferee.github.io/UIML/demo.html`

Conclusion

This paper

- Uniform Interpolation Calculator for modal logics in Coq
- Fix bug for GL in [Bílková '06]
- First proof of uniform interpolation for iSL
  → semantic proof in [Litak & Visser '24]

Work in progress

- Simplify interpolants
- Connect proof-theoretic with semantic methods
- How to tackle difficult cases (like iGL)?

My plan for three years at UvA