Propositional Quantifiers and Uniform Interpolation

Sam van Gool

GT SCALP, 16 February 2023

IRIF, Université Paris Cité

Interpolation

Interpolation is the problem that asks, given a deduction

$$A \vdash B$$

to find C such that

$$A \vdash C \vdash B$$

and C only uses symbols that are in both A and B.

Interpolation

Interpolation is the problem that asks, given a deduction

$$A \vdash B$$

to find C such that

$$A \vdash C \vdash B$$

and C only uses symbols that are in both A and B.

- What are A, B, C? Which symbols? What is $\vdash$?

Interpolation

Interpolation is the problem that asks, given a deduction

$$A \vdash B$$

to find C such that

$$A \vdash C \vdash B$$

and C only uses symbols that are in both A and B.

- What are A, B, C? Which symbols? What is $\vdash$?

We will look at propositional logics, and take symbols to mean propositional variables.

The Classical Case

Suppose that

$$A(p, q) \vdash B(p, r)$$

for propositional formulas A and B.

The Classical Case

Suppose that

$$A(p, q) \vdash B(p, r)$$

for propositional formulas A and B.

If $\vdash$ is classical entailment, then the formula

$$C(p) := A(p, \bot) \vee A(p, \top)$$

is an interpolant:

$$A(p, q) \vdash C(p) \vdash B(p, r).$$

The Classical Case

Suppose that

$$A(p, q) \vdash B(p, r)$$

for propositional formulas A and B.

If $\vdash$ is classical entailment, then the formula

$$C(p) := A(p, \bot) \vee A(p, \top)$$

is an interpolant:

$$A(p, q) \vdash C(p) \vdash B(p, r).$$

So is

$$C'(p) := B(p, \bot) \wedge B(p, \top).$$

Uniform Interpolants

Note that each of the interpolants

$$C(p) := A(p, \bot) \vee A(p, \top) \quad \text{and} \quad C'(p) := B(p, \bot) \wedge B(p, \top)$$

only depends on one of the formulas in the entailment $A \vdash B$.

Uniform Interpolants

Note that each of the interpolants

$$C(p) := A(p, \bot) \vee A(p, \top) \quad \text{and} \quad C'(p) := B(p, \bot) \wedge B(p, \top)$$

only depends on one of the formulas in the entailment $A \vdash B$.

These uniform interpolants encode propositional quantifiers:

$$C(p) \equiv \exists q.\ A(p, q) \quad \text{and } C'(p) \equiv \forall q.\ B(p, q)\ .$$

Uniform Interpolants

Note that each of the interpolants

$$C(p) := A(p, \bot) \vee A(p, \top) \quad \text{and} \quad C'(p) := B(p, \bot) \wedge B(p, \top)$$

only depends on one of the formulas in the entailment $A \vdash B$.

These uniform interpolants encode propositional quantifiers:

$$C(p) \equiv \exists q.\ A(p, q) \quad \text{and}\ C'(p) \equiv \forall q.\ B(p, q)\ .$$

The simple encoding works because classical logic is locally finite.

The Intuitionistic Case

Intuitionistic Propositional Logic is not locally finite.

$$p, \neg p, \neg\neg p, \neg\neg p \to p, \ldots$$

The Intuitionistic Case

Intuitionistic Propositional Logic is not locally finite.

$$p, \neg p, \neg\neg p, \neg\neg p \to p, \ldots$$

Surprisingly, we still have:

Theorem. (Pitts 1992) There is an effective interpretation of propositional quantifiers in intuitionistic propositional logic.

Detailed Statement of Pitts' Theorem

For every propositional formula $\phi(\bar{p}, q)$, there effectively exist propositional formalas

$$E_q(\phi) \quad \text{and} \quad A_q(\phi)$$

with variables in $\bar{p}$, and such that, for any formula $\psi(\bar{p})$,

$$\text{if } \phi \vdash \psi \text{ then } \phi \vdash E_q\phi \vdash \psi \ ,$$

and

$$\text{if } \psi \vdash \phi \text{ then } \psi \vdash A_q\phi \vdash \phi \ ,$$

where $\phi \vdash \psi$ means intuitionistic entailment (provability).

Intuitionistic Propositional Quantifiers

We no longer have the simple encoding of $\exists q.\ A$.

Intuitionistic Propositional Quantifiers

We no longer have the simple encoding of $\exists q.\ A$.

For example, when

$$A = (\neg p \to q) \wedge (q \to r)$$

we have

$$A[\bot/q] \equiv \neg\neg p, \quad A[\top/q] \equiv r$$

but

$$A \nvdash \neg\neg p \vee r\ .$$

Intuitionistic Propositional Quantifiers

We no longer have the simple encoding of $\exists q.\ A$.

For example, when

$$A = (\neg p \to q) \wedge (q \to r)$$

we have

$$A[\bot/q] \equiv \neg\neg p, \quad A[\top/q] \equiv r$$

but

$$A \nvdash \neg\neg p \vee r\ .$$

In this example, it turns out that $\exists q.\ A$ can be encoded as

$$\neg p \to r\ ,$$

which is equivalent to $A[\neg p/q]$.

A Glimpse At Pitts' Proof

Given a formula $A(\bar{p}, q)$, we have

$$A(\bar{p}, q) \vdash \bigwedge \{B(\bar{p}) \ \mid \ A \vdash B\}.$$

A Glimpse At Pitts' Proof

Given a formula $A(\bar{p}, q)$, we have

$$A(\bar{p}, q) \vdash \bigwedge \{B(\bar{p}) \ \mid \ A \vdash B\}.$$

The expression on the right is q-free, but not a formula.

A Glimpse At Pitts' Proof

Given a formula $A(\bar{p}, q)$, we have

$$A(\bar{p}, q) \vdash \bigwedge \{B(\bar{p}) \mid A \vdash B\}.$$

The expression on the right is q-free, but not a formula.

The idea is to replace it by

$$\bigwedge \mathcal{E}_q(A)$$

where $\mathcal{E}_q(A)$ is a finite set of formulas that is a basis.

A Glimpse At Pitts' Proof

Given a formula $A(\bar{p}, q)$, we have

$$A(\bar{p}, q) \vdash \bigwedge \{B(\bar{p}) \mid A \vdash B\}.$$

The expression on the right is q-free, but not a formula.

The idea is to replace it by

$$\bigwedge \mathcal{E}_q(A)$$

where $\mathcal{E}_q(A)$ is a finite set of formulas that is a basis.

The encoding of $\forall q.\ A$ is similar, using a disjunction of $\mathcal{A}_q(A)$.

Pitts' definition recurses on the shape of the formula A, using already computed sets $\mathcal{E}_q(A')$ and $\mathcal{A}_q(A')$ for smaller formulas A'.

A Glimpse At Pitts' Proof

	Δ matches:	$\mathcal{E}(\Delta)$ contains:
E_1	$\Delta' \bullet q$	$E(\Delta') \wedge q$
E_4	$\Delta' \bullet (q \to \delta)$	$q \to E(\Delta' \bullet \delta)$
E_5	$\Delta'' \bullet p \bullet (p \to \delta)$	$E(\Delta'' \bullet p \bullet \delta)$
E_6	$\Delta' \bullet (\delta_1 \wedge \delta_2) \to \delta_3$	$E(\Delta' \bullet (\delta_1 \to (\delta_2 \to \delta_3)))$
E_8	$\Delta' \bullet ((\delta_1 \to \delta_2) \to \delta_3)$	$(E(\Delta' \bullet (\delta_2 \to \delta_3)) \to A(\Delta' \bullet (\delta_2 \to \delta_3), \delta_1 \to \delta_2)) \to E(\Delta' \bullet \delta_3)$

	Δ, ϕ matches:	$\mathcal{A}(\Delta, \phi)$ contains:
A_3	$\Delta' \bullet \delta_1 \vee \delta_2, \phi$	$(E(\Delta' \bullet \delta_1) \to A(\Delta' \bullet \delta_1, \phi)) \wedge (E(\Delta' \bullet \delta_2) \to A(\Delta' \bullet \delta_2, \phi))$
A_7	$\Delta' \bullet (\delta_1 \vee \delta_2) \to \delta_3, \phi$	$A(\Delta' \bullet (\delta_1 \to \delta_3) \bullet (\delta_2 \to \delta_3), \phi)$
A_8	$\Delta' \bullet ((\delta_1 \to \delta_2) \to \delta_3), \phi$	$(E(\Delta' \bullet (\delta_2 \to \delta_3)) \to A(\Delta' \bullet (\delta_2 \to \delta_3), (\delta_1 \to \delta_2))) \wedge A(\Delta' \bullet \delta_3, \phi)$
A_{11}	$\Delta, \phi_1 \wedge \phi_2$	$A(\Delta, \phi_1) \wedge A(\Delta, \phi_2)$
A_{12}	$\Delta, \phi_1 \vee \phi_2$	$A(\Delta, \phi_1) \vee A(\Delta, \phi_2)$
A_{13}	$\Delta, \phi_1 \to \phi_2$	$E(\Delta \bullet \phi_1, \phi_2) \to A(\Delta \bullet \phi_1, \phi_2)$

Table 1. Excerpt of Pitts' definitions of $\mathcal{E}(\Delta)$ and $\mathcal{A}(\Delta, \phi)$, with respect to a fixed variable p.

Computing Intuitionistic Propositional Quantifiers

Pitts proves correctness by an induction on proofs of $A \vdash B$.

- What proof calculus to use?

Computing Intuitionistic Propositional Quantifiers

Pitts proves correctness by an induction on proofs of $A \vdash B$.

- What proof calculus to use?

Gentzen calculus **LJ** has contraction, and the rule:

$$\frac{\Gamma, \phi_1 \to \phi_2 \vdash \phi_1 \qquad \Gamma, \phi_2 \vdash \psi}{\Gamma, \phi_1 \to \phi_2 \vdash \psi}$$

which make proof search not obviously terminating.

Computing Intuitionistic Propositional Quantifiers

Pitts proves correctness by an induction on proofs of $A \vdash B$.

- What proof calculus to use?

Gentzen calculus **LJ** has contraction, and the rule:

$$\frac{\Gamma, \phi_1 \to \phi_2 \vdash \phi_1 \qquad \Gamma, \phi_2 \vdash \psi}{\Gamma, \phi_1 \to \phi_2 \vdash \psi}$$

which make proof search not obviously terminating.

Using multisets as sequents, and replacing this rule by a finer case analysis on ϕ_1, one obtains the calculus **G4ip** (aka **LJT**).

Theorem (Vorob'ev, Hudelmaier, Dyckhoff)

The sequent calculus **G4ip** *is sound and complete for intuitionistic propositional logic.*

Pitts Verified

In recent joint work with H. Férée, we formalized the proof in Coq, yielding a correct-by-construction program that computes the encoding of the propositional quantifiers.

https://ipqcoq.github.io

Pitts Verified

In recent joint work with H. Férée, we formalized the proof in Coq, yielding a correct-by-construction program that computes the encoding of the propositional quantifiers.

`https://ipqcoq.github.io`

Some take-aways of that work:

- Intricate properties of the proof calculus play a big role
- A usable program (more experimentation to be done)
- Not obviously modular: how to generalize to other logics? (Linear, modal, . . .)
- A question: what does Pitts' theorem mean, computationally?

Detailed Statement of Pitts' Theorem, through Curry-Howard

For every type $\phi(\bar{p}, q)$ we can compute types

$$E_q(\phi) \quad \text{and} \quad A_q(\phi)$$

with variables in $\bar{p}$, and, for any type $\psi(\bar{p})$, functions

$$(\phi \vdash \psi) \longrightarrow (\phi \vdash E_q\phi) \times (E_q\phi \vdash \psi) \ ,$$

and

$$(\psi \vdash \phi) \longrightarrow (\psi \vdash A_q\phi) \times (A_q\phi \vdash \phi) \ ,$$

where types are built from variables and $\bot$ with $\vee, \wedge, \rightarrow$, and $\phi \vdash \psi$ means the type of **G4ip**-proofs of ψ in context ϕ.

The Semantic Approach

Intuitionistic propositional logic is canonically interpreted by Heyting algebras: structures $(H, \vee, \wedge, \bot, \top, \rightarrow)$ satisfying the axioms of a bounded distributive lattice and, for all $a, b, c \in H$,

$$a \wedge b \leq c \iff a \leq b \rightarrow c \ .$$

The Semantic Approach

Intuitionistic propositional logic is canonically interpreted by Heyting algebras: structures $(H, \vee, \wedge, \bot, \top, \to)$ satisfying the axioms of a bounded distributive lattice and, for all $a, b, c \in H$,

$$a \wedge b \leq c \iff a \leq b \to c \ .$$

(For the categorically minded: a Heyting algebra is a cartesian closed partial order with finite sums.)

Pitts' Theorem, Semantically

Pitts' theorem can be reformulated using Heyting algebras as:

Theorem. Any homomorphism between finitely generated free Heyting algebras has both an upper and a lower adjoint.

Pitts' Theorem, Semantically

Pitts' theorem can be reformulated using Heyting algebras as:

Theorem. Any homomorphism between finitely generated free Heyting algebras has both an upper and a lower adjoint.

Using moreover Heyting categories, another formulation is:

Theorem. The opposite of the category $\mathbf{HA}_{\mathrm{fp}}$ of finitely presented Heyting algebras is a Heyting category.

Aside: Why Pitts Proved His Theorem

"Some ten or so years ago I tried to prove the negation of [the theorem] in connection with (...) the question of whether any Heyting algebra can appear as the algebra of truth-values of an elementary topos. I established that the free Heyting algebra on a countable infinity of generators does not so appear provided [the theorem] does not hold. It seemed likely to me (and to others to whom I posed the question) that a [formula] ϕ could be found for which $A_p\phi$ does not exist (although I could not find one!), thus settling the original question about toposes and Heyting algebras in the negative. That [the theorem] is true is quite a surprise to me. (...) It remains an open question whether every Heyting algebra can be the Lindenbaum algebra of a theory in intuitionistic higher order logic." (Pitts 1992)

A Sheaf Representation

S. Ghilardi and M. Zawadowski (1995) gave a different proof, starting from the observation that every finitely presented Heyting algebra H can be faithfully represented by a covariant presheaf

$$\Phi_H \colon \mathbf{HA}_{\mathrm{fin}} \longrightarrow \mathbf{Set}$$

defined as the restriction of $\mathrm{Hom}(H, -)$ to finite algebras.

A Sheaf Representation

S. Ghilardi and M. Zawadowski (1995) gave a different proof, starting from the observation that every finitely presented Heyting algebra H can be faithfully represented by a covariant presheaf

$$\Phi_H \colon \mathbf{HA}_{\mathrm{fin}} \longrightarrow \mathbf{Set}$$

defined as the restriction of $\mathrm{Hom}(H, -)$ to finite algebras.

GZ noticed that Φ_H can also be seen as a contravariant sheaf on the category $\mathbf{Pos}_{\mathrm{fin}}$ of finite posets, giving a functor

$$\Phi \colon \mathbf{HA}_{\mathrm{fp}} \longrightarrow \mathrm{Sh}(\mathbf{Pos}_{\mathrm{fin}}),$$

A Sheaf Representation

S. Ghilardi and M. Zawadowski (1995) gave a different proof, starting from the observation that every finitely presented Heyting algebra H can be faithfully represented by a covariant presheaf

$$\Phi_H \colon \mathbf{HA}_{\mathrm{fin}} \longrightarrow \mathbf{Set}$$

defined as the restriction of $\mathrm{Hom}(H, -)$ to finite algebras.

GZ noticed that Φ_H can also be seen as a contravariant sheaf on the category $\mathbf{Pos}_{\mathrm{fin}}$ of finite posets, giving a functor

$$\Phi \colon \mathbf{HA}_{\mathrm{fp}} \longrightarrow \mathrm{Sh}(\mathbf{Pos}_{\mathrm{fin}}),$$

and characterized the image of Φ via a combinatorial condition $(*)$.

A Sheaf Representation

S. Ghilardi and M. Zawadowski (1995) gave a different proof, starting from the observation that every finitely presented Heyting algebra H can be faithfully represented by a covariant presheaf

$$\Phi_H\colon \mathbf{HA}_{\mathrm{fin}} \longrightarrow \mathbf{Set}$$

defined as the restriction of $\mathrm{Hom}(H, -)$ to finite algebras.

GZ noticed that Φ_H can also be seen as a contravariant sheaf on the category $\mathbf{Pos}_{\mathrm{fin}}$ of finite posets, giving a functor

$$\Phi\colon \mathbf{HA}_{\mathrm{fp}} \longrightarrow \mathrm{Sh}(\mathbf{Pos}_{\mathrm{fin}}),$$

and characterized the image of Φ via a combinatorial condition $(*)$.

Pitts' Theorem is then proved by showing that the direct image $(\exists)$ and universal image $(\forall)$ operations on sheaves preserve $(*)$.

An Open Mapping Theorem

A different interpretation of the GZ sheaf theoretic proof.

Any bounded distributive lattice H can be described as a lattice of compact-open subsets of a topological space X, based on the set

$$\mathbf{DL}(H, 2)$$

of homomorphisms to the two-element lattice (Stone 1937).

An Open Mapping Theorem

A different interpretation of the GZ sheaf theoretic proof.

Any bounded distributive lattice H can be described as a lattice of compact-open subsets of a topological space X, based on the set

$$\mathbf{DL}(H, 2)$$

of homomorphisms to the two-element lattice (Stone 1937).

Esakia (1974) derived from this a dual equivalence between Heyting algebras and certain *ordered compact spaces*, now called Esakia spaces. The finite part is Kripke semantics.

An Open Mapping Theorem

A different interpretation of the GZ sheaf theoretic proof.

Any bounded distributive lattice H can be described as a lattice of compact-open subsets of a topological space X, based on the set

$$\mathbf{DL}(H, 2)$$

of homomorphisms to the two-element lattice (Stone 1937).

Esakia (1974) derived from this a dual equivalence between Heyting algebras and certain *ordered compact spaces*, now called Esakia spaces. The finite part is Kripke semantics.

We can prove Pitts' theorem in the dual category $\mathbf{Esakia} \simeq \mathbf{HA}^{\mathrm{op}}$:

Theorem. (vG. & Reggio 2018) Every continuous monotone map between co-finitely presented Esakia spaces is open.

Logical Connections

Ghilardi and Zawadowski use Pitts' theorem to prove:

Theorem. The theory of Heyting algebras has a model completion.

Logical Connections

Ghilardi and Zawadowski use Pitts' theorem to prove:

Theorem. The theory of Heyting algebras has a model completion.

Here, a model completion of a first order theory is an elementary axiomatization of its existentially closed models.

The idea is that propositional quantifiers allow one to encode any Heyting algebra equation in an elementary way.

Logical Connections

Ghilardi and Zawadowski use Pitts' theorem to prove:

Theorem. The theory of Heyting algebras has a model completion.

Here, a model completion of a first order theory is an elementary axiomatization of its existentially closed models.

The idea is that propositional quantifiers allow one to encode any Heyting algebra equation in an elementary way.

One may identify the algebraic conditions needed for this, giving a modular approach to model completions (Ghilardi & Zawadowski 2002; vG., Tsinakis, Metcalfe 2017; Metcalfe & Reggio 2023).

More Logical Connections

Linear temporal logic and Computation tree logic do not have interpolation, but they do have propositional quantifiers.

Theorem. (Ghilardi & vG. 2016) There are finitely axiomatized algebraic theories for LTL and fair CTL with model companions.

More Logical Connections

Linear temporal logic and Computation tree logic do not have interpolation, but they do have propositional quantifiers.

Theorem. (Ghilardi & vG. 2016) There are finitely axiomatized algebraic theories for LTL and fair CTL with model companions.

The proof technique makes essential use of the correspondence between monadic second order logic and automata.

More Logical Connections

Linear temporal logic and Computation tree logic do not have interpolation, but they do have propositional quantifiers.

Theorem. (Ghilardi & vG. 2016) There are finitely axiomatized algebraic theories for LTL and fair CTL with model companions.

The proof technique makes essential use of the correspondence between monadic second order logic and automata.

- Is there a syntactic proof?

More Logical Connections

Linear temporal logic and Computation tree logic do not have interpolation, but they do have propositional quantifiers.

Theorem. (Ghilardi & vG. 2016) There are finitely axiomatized algebraic theories for LTL and fair CTL with model companions.

The proof technique makes essential use of the correspondence between monadic second order logic and automata.

- Is there a syntactic proof?
- What is the computational meaning of these semantic methods?

More Logical Connections

Linear temporal logic and Computation tree logic do not have interpolation, but they do have propositional quantifiers.

Theorem. (Ghilardi & vG. 2016) There are finitely axiomatized algebraic theories for LTL and fair CTL with model companions.

The proof technique makes essential use of the correspondence between monadic second order logic and automata.

- Is there a syntactic proof?
- What is the computational meaning of these semantic methods?

Thank you!