On the Completeness of Interpolation Procedures

Raheleh Jalali
(joint work with Stefan Hetzl)

Institute of Computer Science, The Czech Academy of Sciences

CIBD

23 April, 2024

Outline

Motivation

Aim: Gauging the expressive power of interpolation algorithms.

Question

Which interpolants can be obtained from an interpolation algorithm?

Motivated by this question, we initiate the study of the *completeness properties* of interpolation algorithms.

Motivation

Aim: Gauging the expressive power of interpolation algorithms.

Question

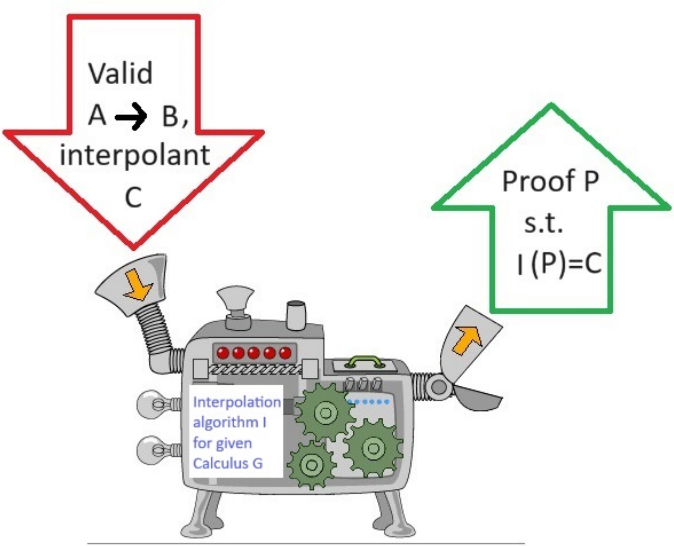
Which interpolants can be obtained from an interpolation algorithm?

Motivated by this question, we initiate the study of the *completeness properties* of interpolation algorithms.

Definition

Fix a calculus and an interpolation algorithm $\mathcal{I}$. We say $\mathcal{I}$ is *complete* if, for every semantically possible interpolant C of an implication $A \to B$, there is a proof P of $A \to B$ such that C is logically equivalent to $\mathcal{I}(P)$.

Completeness of interpolation algorithms



Overview

The practical relevance of a completeness result is that it provides a guarantee that, at least in principle, the algorithm allows us to find the "good" interpolants, whatever that may mean in the concrete application under consideration.

Results: We establish incompleteness and different kinds of completeness results for several standard algorithms for resolution and the sequent calculus for propositional, modal, and first-order logic.

Getting ready

- Propositional language $\mathcal{L}_p = \{\bot, \wedge, \vee, \neg\}$. Define $A \to B := \neg A \vee B$ and $\top := \bot \to \bot$.
- A literal ℓ is either an atom or a negation of an atom.
- A *clause* C is a finite disjunction of literals $C = \ell_1 \vee \cdots \vee \ell_n$, also written as $C = \{\ell_1, \ldots, \ell_n\}$.
- By a *clause set* we mean a set $\mathcal{C} = \{C_1, \ldots, C_n\}$ of clauses $C_i = \{\ell_{i1}, \ldots, \ell_{ik_i}\}$ and the *formula interpretation* of $\mathcal{C}$ is $\bigwedge_{i=1}^{n} \bigvee_{j=1}^{k_i} \ell_{ij}$.
- A formula is in *conjunctive normal form* (CNF) if it is a conjunction of disjunctions of literals.
- For a formula A, the set of its variables is denoted by $V(A)$.

Definition

A logic L has the *Craig Interpolation Property* (*CIP*) if for any formulas A and B if $A \to B \in L$ then there exists a formula C such that $V(C) \subseteq V(A) \cap V(B)$ and $A \to C \in L$ and $C \to B \in L$.

Resolution R

Propositional *resolution*, R, is one of the weakest proof systems.
Resolution operates on clauses.
A *resolution proof*, also called a *resolution refutation*, shows the unsatisfiability of a set of initial clauses by starting with these clauses and deriving new clauses by the *resolution rule*

$$\frac{C \cup \{p\} \qquad D \cup \{\neg p\}}{C \cup D}$$

until the empty clause $\perp$ is derived, where C and D are clauses. We can interpret resolution as a refutation system: instead of proving a formula A is true we prove that $\neg A$ is unsatisfiable.
Resolution with weakening: Add the *weakening rule* to the R:

$$\frac{C}{C \cup D}$$

for arbitrary clauses C and D.

Interpolation algorithm for R

Given: P, a resolution proof of $\bot$ from the clauses $A_i(\bar{p}, \bar{q})$ and $B_j(\bar{p}, \bar{r})$, where $i \in I$, $j \in J$, and $\bar{p}, \bar{q}, \bar{r}$ are disjoint sets of atoms.

Define a ternary connective *sel* as
$sel(A, x, y) = (\neg A \to x) \wedge (A \to y) = (A \vee x) \wedge (\neg A \vee y)$.

Example

$sel(\bot, x, y) = x$, $sel(\top, x, y) = y$, $sel(A, \bot, \top) = A$, and $sel(A, \top, \bot) = \neg A$.

Interpolation algorithm: Assign $\bot$ to clauses A_i for each $i \in I$ and assign $\top$ to clauses B_j for $j \in J$. Then:

Interpolation algorithm for R

For the resolution rule is of the following form for $p_k \in \bar{p}$ define

$$\frac{\Gamma, p_k \qquad \Delta, \neg p_k}{\Gamma, \Delta} \quad \overset{\text{int}}{\rightsquigarrow} \quad \frac{x \qquad y}{sel(p_k, x, y)}$$

or when $q_k \in \bar{q}$

$$\frac{\Gamma, q_k \qquad \Delta, \neg q_k}{\Gamma, \Delta} \quad \overset{\text{int}}{\rightsquigarrow} \quad \frac{x \qquad y}{x \vee y}$$

or when $r_k \in \bar{r}$

$$\frac{\Gamma, r_k \qquad \Delta, \neg r_k}{\Gamma, \Delta} \quad \overset{\text{int}}{\rightsquigarrow} \quad \frac{x \qquad y}{x \wedge y}$$

Interpolation algorithm for R

For the resolution rule is of the following form for $p_k \in \bar{p}$ define

$$\frac{\Gamma, p_k \qquad \Delta, \neg p_k}{\Gamma, \Delta} \quad \overset{\text{int}}{\leadsto} \quad \frac{x \qquad y}{sel(p_k, x, y)}$$

or when $q_k \in \bar{q}$

$$\frac{\Gamma, q_k \qquad \Delta, \neg q_k}{\Gamma, \Delta} \quad \overset{\text{int}}{\leadsto} \quad \frac{x \qquad y}{x \vee y}$$

or when $r_k \in \bar{r}$

$$\frac{\Gamma, r_k \qquad \Delta, \neg r_k}{\Gamma, \Delta} \quad \overset{\text{int}}{\leadsto} \quad \frac{x \qquad y}{x \wedge y}$$

Theorem (Krajíček,Pudlák)

Let π be a resolution refutation of the set of clauses $\{A_i(\bar{p}, \bar{q}) \mid i \in I\}$ and $\{B_j(\bar{p}, \bar{q}) \mid j \in J\}$. Then, the interpolation algorithm outputs an interpolant for the valid formula $\bigwedge_{i \in I} A_i(\bar{p}, \bar{q}) \to \bigvee_{j \in J} \neg B_j(\bar{p}, \bar{q})$.

Example

Example

Consider the unsatisfiable sets of clauses:

$A_1 = \{p, \neg q\}$, $A_2 = \{q\}$, $B_1 = \{\neg p, r\}$, $B_2 = \{\neg r\}$

$$\dfrac{\dfrac{\dfrac{p, \neg q \qquad q}{p} \qquad \neg p, r}{r} \qquad \neg r}{\varnothing} \quad \overset{\text{int}}{\rightsquigarrow} \quad \dfrac{\dfrac{\dfrac{\bot \qquad \bot}{\bot \vee \bot = \bot} \qquad \top}{sel(p, \bot, \top) = p} \qquad \top}{p \wedge \top = p}$$

The algorithm outputs p. The unsatisfiable formula that we started with was $F = A_1 \wedge A_2 \wedge B_1 \wedge B_2 = (p \vee \neg q) \wedge q \wedge (\neg p \vee r) \wedge \neg r$. Thus, $\neg F = (p \wedge q) \rightarrow (p \vee r)$, which is $(A_1 \wedge A_2) \rightarrow (\neg B_1 \vee \neg B_2)$ is valid and p is its interpolant.

Sequent calculus **LK**

Γ, Δ: multisets of formulas. Interpretation of sequent $\Gamma \Rightarrow \Delta$: $\bigwedge \Gamma \to \bigvee \Delta$.

$$p \Rightarrow p \quad (\mathrm{Ax}) \qquad\qquad \bot \Rightarrow \quad (\bot)$$

$$\frac{\Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}\ (Lw) \qquad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, A}\ (Rw)$$

$$\frac{A, A, \Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}\ (Lc) \qquad \frac{\Gamma \Rightarrow \Delta, A, A}{\Gamma \Rightarrow \Delta, A}\ (Rc)$$

$$\frac{A, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta}\ (L\wedge_1) \qquad \frac{B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta}\ (L\wedge_2)$$

$$\frac{\Gamma \Rightarrow \Delta, A \qquad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B}\ (R\wedge) \qquad \frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, A \vee B}\ (R\vee_1)$$

$$\frac{\Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \vee B}\ (R\vee_2) \qquad \frac{A, \Gamma \Rightarrow \Delta \qquad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta}\ (L\vee)$$

$$\frac{\Gamma \Rightarrow \Delta, A}{\neg A, \Gamma \Rightarrow \Delta}\ (L\neg) \qquad \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg A}\ (R\neg)$$

$$\frac{\Gamma \Rightarrow \Delta, A \qquad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ (cut)$$

The cut rule is called **atomic** when the cut formula is an atom or $\bot$ or $\top$.

Denote **LK** with only atomic cuts by $\mathbf{LK}^{\mathrm{at}}$, denote **LK** with cuts only on literals by $\mathbf{LK}^{\mathrm{lit}}$, and denote cut-free **LK** by $\mathbf{LK}^{-}$.

The cut rule is called **atomic** when the cut formula is an atom or $\bot$ or $\top$.

Denote **LK** with only atomic cuts by $\mathbf{LK}^{\mathrm{at}}$, denote **LK** with cuts only on literals by $\mathbf{LK}^{\mathrm{lit}}$, and denote cut-free **LK** by $\mathbf{LK}^{-}$.

Split sequent: $\Gamma_1; \Gamma_2 \Rightarrow \Delta_1; \Delta_2$ such that $\Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2$ is a sequent. Let π be a proof of $\Gamma_1; \Gamma_2 \Rightarrow \Delta_1; \Delta_2$ in $\mathbf{LK}^{at}$. Define the Maehara *interpolant* $\mathcal{M}(\pi) = C$, recursively s.t.

$$\mathbf{LK}^{\mathrm{at}} \vdash \Gamma_1 \Rightarrow \Delta_1, C \quad \text{and} \quad \mathbf{LK}^{\mathrm{at}} \vdash C, \Gamma_2 \Rightarrow \Delta_2$$

and $V(C) \subseteq V(\Gamma_1 \cup \Delta_1) \cap V(\Gamma_2 \cup \Delta_2)$. Denote $\Gamma_1; \Gamma_2 \overset{C}{\Longrightarrow} \Delta_1; \Delta_2$.

Maehara interpolation algorithm $\mathcal{M}$

• If π is an axiom: $p; \overset{\bot}{\Longrightarrow} p;$ $\quad ; p \overset{\top}{\Longrightarrow} ; p$ $\quad \bot; \overset{\bot}{\Longrightarrow} ;$

$p; \overset{p}{\Longrightarrow} ; p$ $\quad ; p \overset{\neg p}{\Longrightarrow} p;$ $\quad ; \bot \overset{\top}{\Longrightarrow} ;$

• If the last rule applied in π is one of the one-premise rules, then the interpolant of the premise works as the interpolant for the conclusion.

• If the last rule in π is $(R\wedge)$, then:

$$\frac{\Gamma_1;\Gamma_2 \overset{C}{\Longrightarrow} \Delta_1, A; \Delta_2 \qquad \Gamma_1;\Gamma_2 \overset{D}{\Longrightarrow} \Delta_1, B; \Delta_2}{\Gamma_1;\Gamma_2 \overset{C\vee D}{\Longrightarrow} \Delta_1, A \wedge B; \Delta_2}$$

or

$$\frac{\Gamma_1;\Gamma_2 \overset{C}{\Longrightarrow} \Delta_1; A, \Delta_2 \qquad \Gamma_1;\Gamma_2 \overset{D}{\Longrightarrow} \Delta_1; B, \Delta_2}{\Gamma_1;\Gamma_2 \overset{C\wedge D}{\Longrightarrow} \Delta_1; A \wedge B, \Delta_2}$$

Maehara interpolation algorithm $\mathcal{M}$, cont.

- If the last rule in π is $(L\vee)$, then:

$$\frac{\Gamma_1, A; \Gamma_2 \overset{C}{\Longrightarrow} \Delta_1; \Delta_2 \qquad \Gamma_1, B; \Gamma_2 \overset{D}{\Longrightarrow} \Delta_1; \Delta_2}{\Gamma_1, A \vee B; \Gamma_2 \overset{C \vee D}{\Longrightarrow} \Delta_1; \Delta_2}$$

or

$$\frac{\Gamma_1; A, \Gamma_2 \overset{C}{\Longrightarrow} \Delta_1; \Delta_2 \qquad \Gamma_1; B, \Gamma_2 \overset{D}{\Longrightarrow} \Delta_1; \Delta_2}{\Gamma_1; A \vee B, \Gamma_2 \overset{C \wedge D}{\Longrightarrow} \Delta_1; \Delta_2}$$

- Let the last rule in π be an instance of a cut rule and A the cut formula. Then, $V(A) \subseteq V(\Gamma_1 \cup \Delta_1)$ or $V(A) \subseteq V(\Gamma_2 \cup \Delta_2)$. In former case, define

$$\frac{\Gamma_1; \Gamma_2 \overset{C}{\Longrightarrow} \Delta_1, A; \Delta_2 \qquad \Gamma_1, A; \Gamma_2 \overset{D}{\Longrightarrow} \Delta_1; \Delta_2}{\Gamma_1; \Gamma_2 \overset{C \vee D}{\Longrightarrow} \Delta_1; \Delta_2}$$

In the latter case, define

$$\frac{\Gamma_1; \Gamma_2 \overset{E}{\Longrightarrow} \Delta_1; A, \Delta_2 \qquad \Gamma_1; \Gamma_2, A \overset{F}{\Longrightarrow} \Delta_1; \Delta_2}{\Gamma_1; \Gamma_2 \overset{E \wedge F}{\Longrightarrow} \Delta_1; \Delta_2}$$

Theorem

Let π be a proof of $A; \Rightarrow ; B$ in $\mathbf{LK}^{at}$. $\mathcal{M}(\pi)$ outputs an interpolant of $A \rightarrow B$.

Example

$$\dfrac{\dfrac{p; \overset{p}{\Longrightarrow}; p}{p \wedge q; \overset{p}{\Longrightarrow}; p}}{p \wedge q; \overset{p}{\Longrightarrow}; p \vee q}$$

$$\dfrac{\dfrac{p; \overset{p}{\Longrightarrow}; p}{p \wedge q; \overset{p}{\Longrightarrow}; \color{red}{p}} \qquad \dfrac{; p \overset{\top}{\Longrightarrow}; p}{; \color{red}{p} \overset{\top}{\Longrightarrow}; p \vee q}}{p \wedge q; \overset{p \wedge \top}{\Longrightarrow}; p \vee q}\ \text{cut}$$

Theorem

Let π be a proof of $A; \Rightarrow ; B$ in $\mathbf{LK}^{at}$. $\mathcal{M}(\pi)$ outputs an interpolant of $A \rightarrow B$.

Example

$$\dfrac{\dfrac{p; \overset{p}{\Longrightarrow}; p}{p \wedge q; \overset{p}{\Longrightarrow}; p}}{p \wedge q; \overset{p}{\Longrightarrow}; p \vee q}$$

$$\dfrac{\dfrac{p; \overset{p}{\Longrightarrow}; p}{p \wedge q; \overset{p}{\Longrightarrow}; \color{red}{p}} \qquad \dfrac{; p \overset{\top}{\Longrightarrow}; p}{; \color{red}{p} \overset{\top}{\Longrightarrow}; p \vee q}}{p \wedge q; \overset{p \wedge \top}{\Longrightarrow}; p \vee q} \text{ cut}$$

A formula is in *negation normal form (NNF)* when the negation is only allowed on atoms and the other connectives in the formula are $\wedge$ and $\vee$.

Observation

The interpolants constructed via the Maehara algorithm are in NNF.

Outline

Simple incompleteness results

Definition

Interpolation algorithm $\mathcal{I}$ is *syntactically complete* if for any valid $A \to B$ and any interpolant C of $A \to B$ there is a proof π s.t. $C = \mathcal{I}(\pi)$.

Simple incompleteness results

Definition

Interpolation algorithm $\mathcal{I}$ is *syntactically complete* if for any valid $A \to B$ and any interpolant C of $A \to B$ there is a proof π s.t. $C = \mathcal{I}(\pi)$.

Observation

$\mathcal{M}$ is syntactically incomplete.

Proof.

$\neg\neg p$ is an interpolant of $p \to p$ and not in NNF. So there is no π s.t. $\mathcal{M}(\pi) = \neg\neg p$. □

Simple incompleteness results

Definition

Interpolation algorithm $\mathcal{I}$ is *syntactically complete* if for any valid $A \to B$ and any interpolant C of $A \to B$ there is a proof π s.t. $C = \mathcal{I}(\pi)$.

Observation

$\mathcal{M}$ is syntactically incomplete.

Proof.

$\neg\neg p$ is an interpolant of $p \to p$ and not in NNF. So there is no π s.t. $\mathcal{M}(\pi) = \neg\neg p$. □

Definition

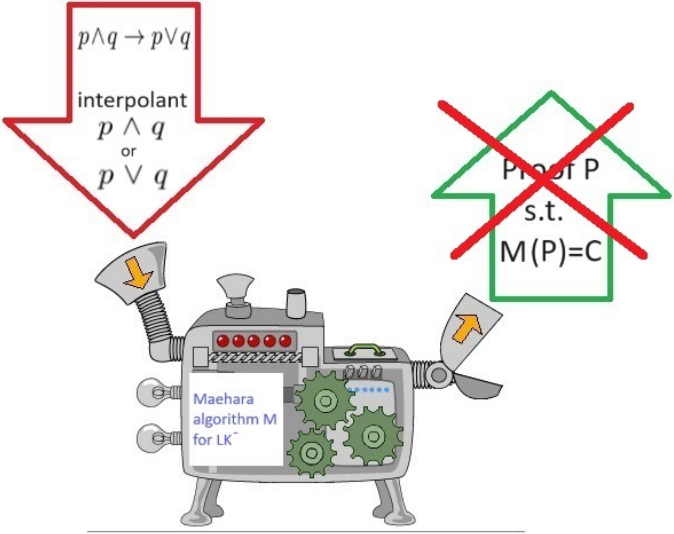
Interpolation algorithm $\mathcal{I}$ is *(semantically) complete* if for any valid $A \to B$ and any interpolant C of $A \to B$ there is a proof π s.t. C is logically equivalent to $\mathcal{I}(\pi)$, denoted by $C \equiv \mathcal{I}(\pi)$.

Incompleteness result for $\mathbf{LK}^-$

The implication $p \wedge q \rightarrow p \vee q$ has the four interpolants $p, q, p \wedge q, p \vee q$.

Proposition

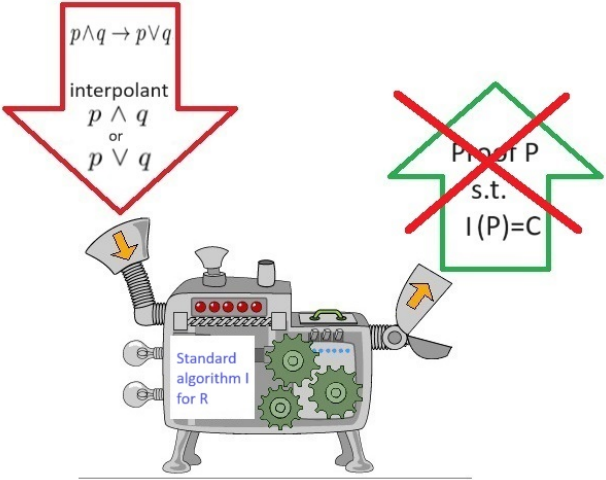
Maehara interpolation in $\mathbf{LK}^-$ (i.e., **LK** without cut) is not complete.


$p \wedge q \to p \vee q$
interpolant
$p \wedge q$
or
$p \vee q$
Proof P
s.t.
M(P)=C
Maehara
algorithm M
for LK⁻

Incompleteness result for R

Proposition

Standard interpolation in propositional resolution is not complete.



Question

Are the standard interpolation algorithms in **resolution with weakening** and in algebraic proof systems, such as **cutting planes** complete?

Outline

1. Motivation
2. Incompleteness results
3. Completeness of Maehara algorithm for $\mathbf{LK}^{at}$
4. Completeness up to pruning and subsumption

Completeness of Maehara algorithm for $\mathbf{LK}^{at}$

Moving from $\mathbf{LK}^{-}$ to the slightly stronger $\mathbf{LK}^{\mathrm{at}}$ we get a full completeness result. Let us first prove the completeness for $\mathbf{LK}^{\mathrm{lit}}$.

Theorem

Maehara interpolation in $\mathbf{LK}^{\mathrm{lit}}$ is complete.

Proof.

Let $\mathcal{C} = \{C_1, \ldots, C_n\}$ be an interpolant of an implication $A \to B$, where $C_i = \{\ell_{i,1}, \ldots, \ell_{i,k_i}\}$, for $i = 1, \ldots, n$.

Completeness of Maehara algorithm for $\mathbf{LK}^{at}$

Moving from $\mathbf{LK}^{-}$ to the slightly stronger $\mathbf{LK}^{\mathrm{at}}$ we get a full completeness result. Let us first prove the completeness for $\mathbf{LK}^{\mathrm{lit}}$.

Theorem

Maehara interpolation in $\mathbf{LK}^{\mathrm{lit}}$ is complete.

Proof.

Let $\mathcal{C} = \{C_1, \ldots, C_n\}$ be an interpolant of an implication $A \to B$, where $C_i = \{\ell_{i,1}, \ldots, \ell_{i,k_i}\}$, for $i = 1, \ldots, n$. Our strategy contains two parts:

1. Constructing proofs $\pi_i : A; \Rightarrow ; \ell_{i,1}, \ldots, \ell_{i,k_i}$ s.t. $\mathcal{M}(\pi_i) \equiv C_i$.
2. Constructing proofs $\sigma_{\bar{j}} : ; \ell_{1,j_1}, \ldots, \ell_{n,j_n} \Rightarrow ; B$, for $\bar{j} = (j_1, \cdots, j_n) \in \{1, \cdots, k_1\} \times \{1, \cdots, k_n\}$.

Proof cont.

Proof.

Step 1. As $\mathcal{C}$ is an interpolant of $A \to B$, we have $\mathbf{LK} \vdash A \Rightarrow \bigwedge_{i=1}^{n} C_i$. Thus, $\mathbf{LK} \vdash A \Rightarrow C_i$ and $\mathbf{LK} \vdash A \Rightarrow \ell_{i1}, \ldots, \ell_{ik_i}$.

Proof cont.

Proof.

Step 1. As $\mathcal{C}$ is an interpolant of $A \to B$, we have $\mathbf{LK} \vdash A \Rightarrow \bigwedge_{i=1}^{n} C_i$. Thus, $\mathbf{LK} \vdash A \Rightarrow C_i$ and $\mathbf{LK} \vdash A \Rightarrow \ell_{i1}, \ldots, \ell_{ik_i}$. Let α_i be a cut-free proof of $A; \Rightarrow \ell_{i1}, \ldots, \ell_{ik_i};$. Easy: $\mathcal{M}(\alpha_i) \equiv \bot$.

Proof cont.

Proof.

Step 1. As $\mathcal{C}$ is an interpolant of $A \to B$, we have $\mathbf{LK} \vdash A \Rightarrow \bigwedge_{i=1}^{n} C_i$. Thus, $\mathbf{LK} \vdash A \Rightarrow C_i$ and $\mathbf{LK} \vdash A \Rightarrow \ell_{i1}, \dots, \ell_{ik_i}$. Let α_i be a cut-free proof of $A; \Rightarrow \ell_{i1}, \dots, \ell_{ik_i};$. Easy: $\mathcal{M}(\alpha_i) \equiv \bot$. Define π_i as:

$$\dfrac{\dfrac{\dfrac{\alpha_i}{A; \overset{\bot}{\Longrightarrow} \ell_{i1}, \ell_{i2}, \dots, \ell_{ik_i};}}{A; \Rightarrow \ell_{i1}, \ell_{i2}, \dots, \ell_{ik_i}; \ell_{i1}} \qquad \dfrac{\ell_{i1}; \overset{\ell_{i1}}{\Longrightarrow}; \ell_{i1}}{A, \ell_{i1}; \Rightarrow \ell_{i2}, \dots, \ell_{ik_i}; \ell_{i1}}\,(w)}{A; \overset{\bot \vee \ell_{i1}}{\Longrightarrow} \ell_{i2}, \dots, \ell_{ik_i}; \ell_{i1}}\,\text{cut}$$

$$\vdots$$

$$\dfrac{\dfrac{A; \Rightarrow \ell_{ik_i}; \ell_{i1}, \dots, \ell_{ik_i-1}}{A; \Rightarrow \ell_{ik_i}; \ell_{i1}, \dots, \ell_{ik_i-1}, \ell_{ik_i}} \qquad \dfrac{\ell_{ik_i}; \overset{\ell_{ik_i}}{\Longrightarrow}; \ell_{ik_i}}{A, \ell_{ik_i}; \Rightarrow; \ell_{i1}, \dots \ell_{ik_i}}\,(w)}{A; \overset{\bot \vee \ell_{i1} \vee \dots \vee \ell_{ik_i}}{\Longrightarrow}; \ell_{i1}, \dots \ell_{ik_i}}\,\text{cut}$$

We get $\mathcal{M}(\pi_i) \equiv C_i$.

Proof cont.

Proof.

Step 2. As $\mathcal{C}$ is an interpolant, $\mathbf{LK} \vdash \mathcal{C} \Rightarrow B$. Thus $\mathbf{LK} \vdash \ell_{1,j_1}, \ldots, \ell_{n,j_n} \Rightarrow B$, for $\bar{j} = (j_1, \cdots, j_n) \in \{1, \cdots, k_1\} \times \{1, \cdots, k_n\}$. Take $\sigma_{\bar{j}}$ as a cut-free proof of $; \ell_{1,j_1}, \ldots, \ell_{n,j_n} \Rightarrow ; B$. Clearly, $\mathcal{M}(\sigma_{\bar{j}}) \equiv \top$.

Claim: using cuts, weakening, and contraction on the proofs π_i and $\sigma_{\bar{j}}$ we get an $\mathbf{LK}^{\mathrm{lit}}$ proof π for $A; \Rightarrow ; B$ where the cut formula is on the right-hand side of the semicolon everywhere. Hence, the interpolant of the conclusion of each cut rule will be the conjunction of the interpolants of the premises. Thus we get $\mathcal{M}(\pi) \equiv \bigwedge_{i=1}^{n} C_i \wedge \top \cdots \wedge \top$. □

Completeness for $\mathbf{LK}^{\mathrm{at}}$

$\mathcal{M}$ is just as complete in $\mathbf{LK}^{\mathrm{at}}$ as it is in $\mathbf{LK}^{\mathrm{lit}}$. Function CNF maps formulas in NNF to clause sets: $\mathrm{CNF}(\top) = \varnothing$, $\mathrm{CNF}(\bot) = \{\varnothing\}$, $\mathrm{CNF}(\ell) = \{\ell\}$, $\mathrm{CNF}(A \wedge B) = \mathrm{CNF}(A) \cup \mathrm{CNF}(B)$, $\mathrm{CNF}(A \vee B) = \mathrm{CNF}(A) \times \mathrm{CNF}(B)$, where ℓ is a literal, A and B are formulas, and define $\mathcal{C} \times \mathcal{D} := \{C \cup D \mid C \in \mathcal{C} \text{ and } D \in \mathcal{D}\}$.

Completeness for $\mathbf{LK}^{\mathrm{at}}$

$\mathcal{M}$ is just as complete in $\mathbf{LK}^{\mathrm{at}}$ as it is in $\mathbf{LK}^{\mathrm{lit}}$. Function CNF maps formulas in NNF to clause sets: $\mathrm{CNF}(\top) = \varnothing$, $\mathrm{CNF}(\bot) = \{\varnothing\}$, $\mathrm{CNF}(\ell) = \{\ell\}$, $\mathrm{CNF}(A \wedge B) = \mathrm{CNF}(A) \cup \mathrm{CNF}(B)$, $\mathrm{CNF}(A \vee B) = \mathrm{CNF}(A) \times \mathrm{CNF}(B)$, where ℓ is a literal, A and B are formulas, and define $\mathcal{C} \times \mathcal{D} := \{C \cup D \mid C \in \mathcal{C} \text{ and } D \in \mathcal{D}\}$.

Lemma

If π is an $\mathbf{LK}^{\mathrm{lit}}$ proof of $\Gamma_1; \Gamma_2 \Rightarrow \Delta_1; \Delta_2$ then there is an $\mathbf{LK}^{\mathrm{at}}$ proof π' of $\Gamma_1; \Gamma_2 \Rightarrow \Delta_1; \Delta_2$ with $\mathrm{CNF}(\mathcal{M}(\pi')) = \mathrm{CNF}(\mathcal{M}(\pi))$.

Proof.

By a version of inversion lemma for negation that preserves the interpolant. □

Completeness for $\mathbf{LK}^{\mathrm{at}}$

$\mathcal{M}$ is just as complete in $\mathbf{LK}^{\mathrm{at}}$ as it is in $\mathbf{LK}^{\mathrm{lit}}$. Function CNF maps formulas in NNF to clause sets: $\mathrm{CNF}(\top) = \varnothing$, $\mathrm{CNF}(\bot) = \{\varnothing\}$, $\mathrm{CNF}(\ell) = \{\ell\}$, $\mathrm{CNF}(A \wedge B) = \mathrm{CNF}(A) \cup \mathrm{CNF}(B)$, $\mathrm{CNF}(A \vee B) = \mathrm{CNF}(A) \times \mathrm{CNF}(B)$, where ℓ is a literal, A and B are formulas, and define $\mathcal{C} \times \mathcal{D} := \{C \cup D \mid C \in \mathcal{C} \text{ and } D \in \mathcal{D}\}$.

Lemma

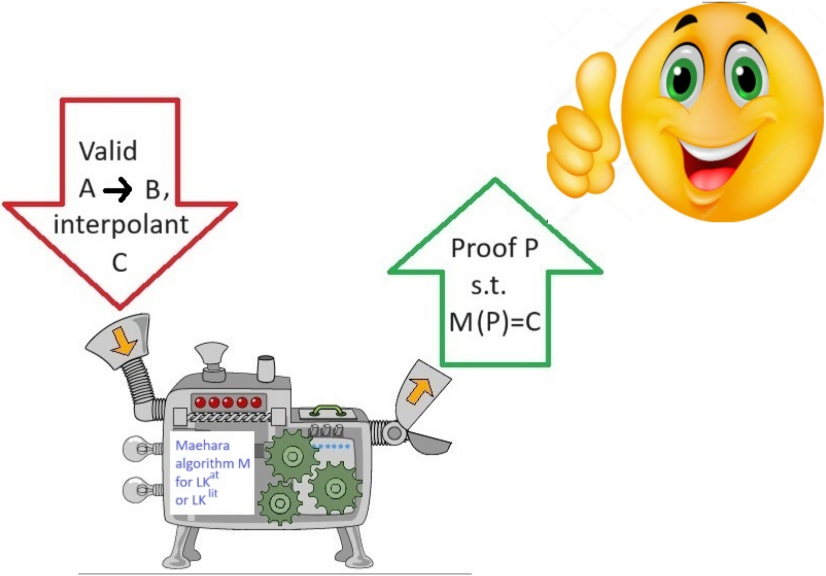
If π is an $\mathbf{LK}^{\mathrm{lit}}$ proof of $\Gamma_1; \Gamma_2 \Rightarrow \Delta_1; \Delta_2$ then there is an $\mathbf{LK}^{\mathrm{at}}$ proof π' of $\Gamma_1; \Gamma_2 \Rightarrow \Delta_1; \Delta_2$ with $\mathrm{CNF}(\mathcal{M}(\pi')) = \mathrm{CNF}(\mathcal{M}(\pi))$.

Proof.

By a version of inversion lemma for negation that preserves the interpolant. □

Corollary

Maehara interpolation in $\mathbf{LK}^{\mathrm{at}}$ is complete.


Valid
A → B,
interpolant
C
Proof P
s.t.
M(P)=C
Maehara
algorithm M
for LK^at
or LK^lit

Example

Find a proof $\pi : p \wedge q; \Rightarrow; p \vee q$ in $\mathbf{LK}^{\mathrm{at}}$ s.t. $\mathcal{M}(\pi) = p \wedge q$. Denote $C_1 = p$ and $C_2 = q$.

$$\pi_1 : \quad \frac{p; \Rightarrow; p}{p \wedge q; \Rightarrow; p} \qquad \pi_2 : \quad \frac{q; \Rightarrow; q}{p \wedge q; \Rightarrow; q}$$

and $\mathcal{M}(\pi_1) = p$ and $\mathcal{M}(\pi_2) = q$. Take the following proof tree $\pi : p \wedge q; \Rightarrow; p \vee q$ in $\mathbf{LK}^{\mathrm{at}}$ where $\mathcal{M}(\pi)$ is logically equivalent to $p \wedge q$.

$$\dfrac{\dfrac{\pi_2}{p \wedge q; \overset{q}{\Longrightarrow}; q} \qquad \dfrac{\dfrac{\pi_1}{p \wedge q; \overset{p}{\Longrightarrow}; p} \qquad \dfrac{\dfrac{; p \overset{\top}{\Longrightarrow}; p}{; p, q \overset{\top}{\Longrightarrow}; p}}{; p, q \overset{\top}{\Longrightarrow}; p \vee q}}{p \wedge q; q \overset{p \wedge \top}{\Longrightarrow}; p \vee q}}{p \wedge q; \overset{q \wedge p \wedge \top}{\Longrightarrow}; p \vee q}$$

Outline

Although Maehara interpolation in $\mathbf{LK}^-$ is incomplete, it is still possible to obtain positive results for $\mathbf{LK}^-$: if we restrict our attention to *pruned interpolants*, then Maehara interpolation is complete up to subsumption.

Subsumption

Definition

A clause set $\mathcal{A}$ *subsumes* a clause set $\mathcal{B}$, in symbols $\mathcal{A} \leqslant_{\mathrm{ss}} \mathcal{B}$, if for all $B \in \mathcal{B}$ there is an $A \in \mathcal{A}$ s.t. $A \subseteq B$.

For instance, $\{\{p\}\}$ subsumes $\{\{p, q\}, \{p\}\}$.

Subsumption is one of the most useful and one of the most thoroughly studied mechanisms for the detection and elimination of redundancy in automated deduction. Note that, if $\mathcal{A} \leqslant_{\mathrm{ss}} \mathcal{B}$ then $\mathcal{A} \models \mathcal{B}$. In this sense, subsumption is a restricted form of implication.

Pruned interpolant

Definition

A clause set $\mathcal{A}$ is called *pruned* if no atom occurs both positively and negatively in $\mathcal{A}$ and $\mathcal{A}$ does not contain the literal $\top$.

For instance, none of the following clause sets are pruned:

$$\{\{p\}, \{r, \neg p\}\} \qquad \{\{\top, p\}\} \qquad \{\{p, \neg p\}, \{r\}\}$$

Pruned interpolant

Definition

A clause set $\mathcal{A}$ is called *pruned* if no atom occurs both positively and negatively in $\mathcal{A}$ and $\mathcal{A}$ does not contain the literal $\top$.

For instance, none of the following clause sets are pruned:

$$\{\{p\}, \{r, \neg p\}\} \qquad \{\{\top, p\}\} \qquad \{\{p, \neg p\}, \{r\}\}$$

Definition

A pruned clause set $\mathcal{C}$ is called *pruned interpolant* of a formula $A \to B$ if it is an interpolant of $A \to B$ and there are no $C' \subset C \in \mathcal{C}$ with $A \models C'$.

So a pruned interpolant, in addition to being a pruned clause set, must not contain redundant literals in the sense of the above definition.

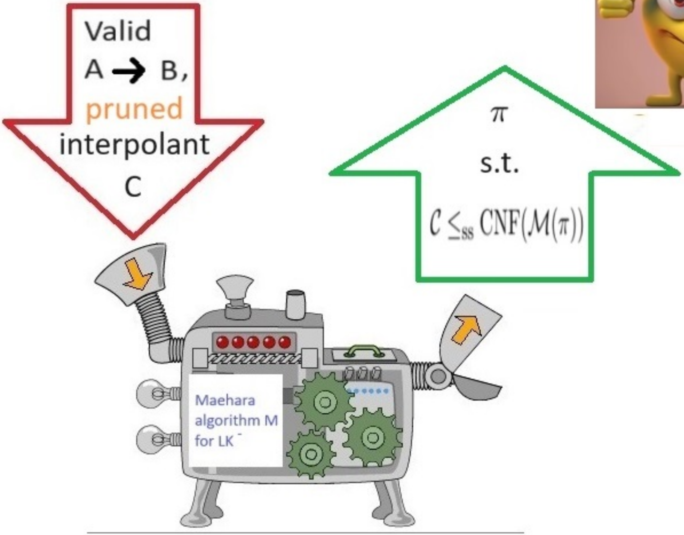
Theorem

Let $\mathcal{C}$ be a pruned interpolant of an implication $A \to B$. Then there is an $\mathbf{LK}^-$ *proof* π *of* $A; \Rightarrow ; B$ *with* $\mathcal{C} \leqslant_{\mathrm{ss}} \mathrm{CNF}(\mathcal{M}(\pi))$.

Proof.

The proof strategy consists of carrying out a cut elimination argument on a carefully chosen class of proofs. This class of proofs, called "tame" proofs, is a new invariant for cut-elimination. This class on the one hand is large enough to permit an embedding of all pruned interpolants, but on the other hand small enough to exhibit a very nice behavior during cut-elimination: the interpolant of the reduced proof is subsumed by the interpolant of the original proof. □

Although interpolation in $\mathbf{LK}^-$ is not complete, we still recover a desired interpolant I in a restricted sense: after transforming I into a pruned interpolant $\mathcal{C}$ we obtain a proof whose interpolant is subsumed by $\mathcal{C}$.


Valid
A → B,
pruned
interpolant
C
π
s.t.
$\mathcal{C} \leq_{ss} \mathrm{CNF}(\mathcal{M}(\pi))$
Maehara
algorithm M
for LK⁻

Example

The formula $p \wedge q \rightarrow p \vee q$ has the four interpolants $p \wedge q, p, q, p \vee q$. We know that the only interpolants obtainable from $\mathbf{LK}^-$ proofs are p and q. The clause set $\{\{p, q\}\}$, representing the formula $p \vee q$, is not a pruned interpolant. The clause set $\{\{p\}, \{q\}\}$, representing the formula $p \wedge q$, subsumes both $\{\{p\}\}$ and $\{\{q\}\}$.

Example

The formula $p \wedge q \to p \vee q$ has the four interpolants $p \wedge q, p, q, p \vee q$. We know that the only interpolants obtainable from $\mathbf{LK}^-$ proofs are p and q. The clause set $\{\{p, q\}\}$, representing the formula $p \vee q$, is not a pruned interpolant. The clause set $\{\{p\}, \{q\}\}$, representing the formula $p \wedge q$, subsumes both $\{\{p\}\}$ and $\{\{q\}\}$.

Question

- Is standard interpolation in resolution complete up to subsumption for pruned interpolants?
- Can we extend these results to the calculus **LJ** for the intuitionistic logic? How about other super intuitionistic or substructural logics?

Conclusion

Initiated the study of completeness properties of interpolation algorithms:

- Incompleteness of the standard algorithms for:
  - Resolution and $\mathbf{LK}^-$.
  - Cut-free sequent calculus for propositional modal logics $\mathbf{K}, \mathbf{KD}, \mathbf{KT}, \mathbf{K4}, \mathbf{KD4}, \mathbf{S4}$.
  - Sequent calculus without cut or with atomic cuts for first-order logic.

Conclusion

Initiated the study of completeness properties of interpolation algorithms:

- Incompleteness of the standard algorithms for:
  - Resolution and $\mathbf{LK}^{-}$.
  - Cut-free sequent calculus for propositional modal logics $\mathbf{K}, \mathbf{KD}, \mathbf{KT}, \mathbf{K4}, \mathbf{KD4}, \mathbf{S4}$.
  - Sequent calculus without cut or with atomic cuts for first-order logic.
- Completeness of the Maehara interpolation in:
  - $\mathbf{LK}^{\mathrm{at}}$, $\mathbf{LK}^{\mathrm{lit}}$.
  - $\mathbf{LK}^{-}$: completeness of pruned interpolants up to subsumption.
  - $\mathbf{K}, \mathbf{KD}, \mathbf{KT}, \mathbf{K4}, \mathbf{KD4}, \mathbf{S4}$ with cuts on atoms and boxed formulas.

Completeness properties of interpolation algorithms $\xrightarrow{\text{corresponds to}}$

Completeness properties of Beth's definability theorem

Conclusion

Initiated the study of completeness properties of interpolation algorithms:

- Incompleteness of the standard algorithms for:
  - Resolution and $\mathbf{LK}^-$.
  - Cut-free sequent calculus for propositional modal logics $\mathbf{K}, \mathbf{KD}, \mathbf{KT}, \mathbf{K4}, \mathbf{KD4}, \mathbf{S4}$.
  - Sequent calculus without cut or with atomic cuts for first-order logic.
- Completeness of the Maehara interpolation in:
  - $\mathbf{LK}^{\mathrm{at}}$, $\mathbf{LK}^{\mathrm{lit}}$.
  - $\mathbf{LK}^-$: completeness of pruned interpolants up to subsumption.
  - $\mathbf{K}, \mathbf{KD}, \mathbf{KT}, \mathbf{K4}, \mathbf{KD4}, \mathbf{S4}$ with cuts on atoms and boxed formulas.

Completeness properties of interpolation algorithms $\xRightarrow{\text{corresponds to}}$

Completeness properties of Beth's definability theorem

Thank you for your attention.