

Craig Interpolation for Guarded Fragments

Balder ten Cate & Jesse Comer

June 2023

B. ten Cate & J. Comer, *Craig Interpolation for Guarded Fragments*, 2023

Table of Contents

Introduction

- Background

CIP and Applications

- CIP in Database Theory

Decidable Fragments

- Modal Logic as a Fragment of First-Order Logic

- The Two-Variable Fragment

- The Guarded Fragment

- The Guarded-Negation Fragment

- The Landscape of FO-Fragments

CIP Fails for GFO

- Failure of CIP for GFO

Repairing CIP

- What To Do When CIP Fails?

- Expressibility Results via Interpolation

- FO² and CIP

- GFO and CIP

Table of Contents

Introduction

Background

CIP and Applications

CIP in Database Theory

Decidable Fragments

Modal Logic as a Fragment of First-Order Logic

The Two-Variable Fragment

The Guarded Fragment

The Guarded-Negation Fragment

The Landscape of FO-Fragments

CIP Fails for GFO

Failure of CIP for GFO

Repairing CIP

What To Do When CIP Fails?

Expressibility Results via Interpolation

FO² and CIP

GFO and CIP

FO-Fragments

This is a talk about *fragments* of first-order logic (FO): logics L such that, for every formula φ in L , there is an FO-formula equivalent to φ .

Note: we are not worried about formulas as *syntactic objects* – we only care about their *semantics* over the class of first-order structures.

In particular, we are interested in fragments with nice *computational* properties.

Decidability

For example, consider the general satisfiability problem for a logic L :

Definition (The satisfiability problem)

Given a formula φ of L , is there a model in which φ is satisfied?

For FO, it is well-known that this problem is undecidable.

However, there exist *less expressive* proper fragments of FO for which this problem *is* decidable.

“Nice” Properties of FO-Fragments

Decidability of the satisfiability problem is not the only property of a logic we may want for applications in computer science.

Other nice properties include

- Finite model property (FMP):
every satisfiable formula can be satisfied in a finite model,
- Beth definability property (BDP):
implicit definability implies explicit definability, and
- the Craig interpolation property (CIP):
every valid implication has an interpolant in the common language.

We are primarily interested CIP, which implies BDP.

However, it is worth noting that many proper FO-fragments we will discuss also have FMP.

Background

Definition (The Craig Interpolation Property)

A logic L satisfies the Craig interpolation property (CIP) if, for all formulas φ and ψ of L , if $\varphi \models \psi$, then there exists a formula θ such that $\varphi \models \theta$ and $\theta \models \psi$, and such that all non-logical symbols occurring in θ occur both in φ and in ψ .

Example

$$\varphi(x) := \exists y(G(x, y) \wedge R(x, y) \wedge R(y, x))$$

$$\psi(x) := P(x) \rightarrow \exists y \exists u(R(x, y) \wedge R(y, u) \wedge P(u))$$

$$\theta(x) := \exists y(R(x, y) \wedge R(y, x))$$

Clearly $\varphi(x) \models \psi(x)$. Furthermore, $\theta(x)$ is an interpolant for this implication.

Background

Definition (The Craig Interpolation Property)

A logic L satisfies the Craig interpolation property (CIP) if, for all formulas φ and ψ of L , if $\varphi \models \psi$, then there exists a formula θ such that $\varphi \models \theta$ and $\theta \models \psi$, and such that all non-logical symbols occurring in θ occur both in φ and in ψ .

CIP is fragile and fails for many logics, including the two-variable fragment, the guarded fragment, and various description logics.

Our goal is to map out the landscape of decidable FO-fragments with CIP.

We will approach this from an abstract model-theoretic perspective, asking: “given a language L for which CIP fails, if L' extends L and has certain “reasonable closure properties,” what can L' express?”

Table of Contents

Introduction

Background

CIP and Applications

CIP in Database Theory

Decidable Fragments

Modal Logic as a Fragment of First-Order Logic

The Two-Variable Fragment

The Guarded Fragment

The Guarded-Negation Fragment

The Landscape of FO-Fragments

CIP Fails for GFO

Failure of CIP for GFO

Repairing CIP

What To Do When CIP Fails?

Expressibility Results via Interpolation

FO² and CIP

GFO and CIP

The Craig Interpolation Property

Definition (The Craig Interpolation Property)

A logic L satisfies the Craig interpolation property (CIP) if, for all formulas φ and ψ of L , if $\varphi \models \psi$, then there exists a formula θ such that $\varphi \models \theta$ and $\theta \models \psi$, and such that all non-logical symbols occurring in θ occur both in φ and in ψ .

But why do we care about CIP in the first place?

CIP has applications in numerous areas within logic and computer science, including

- Database theory,
- Knowledge representation,
- Formal verification,
- Automated deduction,
- and more...

CIP in Database Theory

CIP implies the BDP, a powerful tool in *view-based query rewriting* (Nash, Segoufin, Vianu, 2010).

View-based query rewriting comes up in the context of *query reformulation*, in which we want to transform a query Q into another query Q' which is “better” in some way – say, faster to execute.

CIP in Database Theory

Suppose we have relational signatures σ, τ and a σ -model M .

Furthermore, suppose we have defined the τ -relations over the domain of M by σ -formulas.

In other words, for each $R \in \tau$, there's some σ -formula $\varphi_R(x_1, \dots, x_k)$ such that

$$R^M(a_1, \dots, a_k) \iff M \models \varphi_R(a_1, \dots, a_k).$$

We write M^τ for the τ -structure with the same domain as M and where R^{M^τ} is defined as above for each $R \in \tau$. This is the “view-based model.”

CIP in Database Theory

In view-based query rewriting, we are given a “query” (formula) $\psi(\bar{x})$ in the σ -signature, and we want to know if we can answer it in the τ -signature.

There are two perspectives on this problem:

1. Syntactic: there's a τ -formula $\chi(\bar{x})$ such that $\llbracket \chi \rrbracket^{M_\tau} = \llbracket \psi \rrbracket^M$.
2. Semantic: whether or not $M \models \psi(\bar{a})$ is an answer is fully determined by the model defined by the τ -relations over M
($\llbracket \varphi_R \rrbracket^M = \llbracket \varphi_R \rrbracket^{M'}$ for all $R \in \tau \implies \llbracket \psi \rrbracket^M = \llbracket \psi \rrbracket^{M'}$)

With BDP (and hence CIP), these perspectives are equivalent. This is because (1) is the same as saying that the relation defined by $\psi(\bar{x})$ is *explicitly definable*, while (2) essentially states that it is *implicitly definable*.

Table of Contents

Introduction

Background

CIP and Applications

CIP in Database Theory

Decidable Fragments

Modal Logic as a Fragment of First-Order Logic

The Two-Variable Fragment

The Guarded Fragment

The Guarded-Negation Fragment

The Landscape of FO-Fragments

CIP Fails for GFO

Failure of CIP for GFO

Repairing CIP

What To Do When CIP Fails?

Expressibility Results via Interpolation

FO² and CIP

GFO and CIP

Modal Logic

Definition

The basic modal language (ML) is given by the following recursive grammar:

$$\varphi := p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \Diamond\varphi.$$

Note: We can obtain the box modality using the fact that

$$\Box\varphi \equiv \neg\Diamond\neg\varphi.$$

Modal Logic as a Fragment of First-Order Logic

Definition

The *standard translation* of ML formulas is given by the following recursive definition.

$$ST_x(p) := P(x)$$

$$ST_x(\varphi \vee \psi) := ST_x(\varphi) \vee ST_x(\psi)$$

$$ST_x(\varphi \wedge \psi) := ST_x(\varphi) \wedge ST_x(\psi)$$

$$ST_x(\neg\varphi) := \neg ST_x(\varphi)$$

$$ST_x(\diamond\varphi) := \exists y(R(x, y) \wedge ST_y(\varphi))$$

Note that formulas of the standard translation have only one free variable.

The Two-Variable Fragment

Definition

The two-variable fragment (FO^2) is the fragment of first-order logic with only two variables.

Note: we restrict the signature to those with only unary and binary relation symbols.

To see that FO^2 extends the standard translation of ML, the only interesting case is for the diamond modality. For this, it suffices to observe that the formula

$$ST_x(\diamond\varphi) := \exists y(R(x,y) \wedge ST_y(\varphi))$$

uses only two free variables in the quantified conjunction. By re-using the same two variables (x and y), FO^2 can express all ML formulas.

FO^2 lacks CIP (Comer, 1969).

The Guarded Fragment

The guarded fragment (Andréka, van Benthem, Németi, 1998) is an extension of modal logic intended to embody all of the desirable properties mentioned previously. However, it lacks CIP (Marx, Hoogland, 2002).

A *guard* for a formula φ is a (possibly existentially-quantified) atomic formula α whose free variables include all free variables of φ .

A *self-guarded formula* is a conjunction in which one of the conjuncts is a guard for the entire formula.

Example

$$\exists y \exists z (\underbrace{G(x, y, z) \wedge R(x, y) \wedge R(y, z) \wedge R(z, x)}_{\text{self-guarded formula}})$$

The Guarded Fragment

Definition

The guarded fragment (GFO) is the fragment of first-order logic given by the following recursive grammar:

$$\varphi := R(\bar{x}) \mid x = y \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \exists\bar{x}(\alpha \wedge \varphi),$$

where, in the last clause, α is a guard for φ .

To see that GFO is an extension of ML, we again need only consider the case of the diamond modality:

$$ST_x(\diamond\varphi) := \exists y(\underbrace{R(x, y) \wedge ST_y(\varphi)}_{\text{self-guarded}})$$

Observe that $R(x, y)$ is a valid guard, since y is the only free variable of $ST_y(\varphi)$.

The Guarded-Negation Fragment

Definition

The guarded-negation fragment (GNFO) is the fragment of first-order logic given by the following recursive grammar:

$$\varphi := R(\bar{x}) \mid x = y \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x \varphi \mid \alpha \wedge \neg \varphi,$$

where, in the last clause, α is a guard for φ .

GNFO is a *self-guarded extension* of GFO, meaning that it can express all self-guarded GFO-formulas. In particular, GNFO can express all sentences of GFO. Furthermore, GNFO has CIP (Bárány, ten Cate, Segoufin, 2013).

An alternative syntax for GNFO

An important alternative characterization for GNFO is that it is the logic which can express every union of conjunctive queries (UCQ) and is closed under guarded negation.

Definition

A *conjunctive query* (CQ) is a first-order formula of the form

$$\varphi(x_1, \dots, x_m) := \exists y_1 \dots \exists y_n \left(\bigwedge_{i \leq k} \alpha_i \right),$$

where each α_i is an atomic formula with free variables among $x_1, \dots, x_m, y_1, \dots, y_n$.

Definition

A *union of conjunctive queries* (UCQ) is a disjunction of CQs. In other words, a formula of the form

$$\varphi_1 \vee \dots \vee \varphi_n,$$

where each φ_i is a CQ.

An alternative syntax for GNFO

This characterization is made explicit by the following alternative grammar for GNFO, which will come into play in our later proofs (Bárány, ten Cate, Segoufin, 2015).

Proposition

GNFO is the fragment of first-order logic given by the following recursive grammar:

$$\varphi := R(\bar{x}) \mid x = y \mid \alpha \wedge \neg\varphi \mid q[\varphi_1/R_1, \dots, \varphi_n/R_n],$$

where q is a UCQ with relation symbols R_1, \dots, R_n and $\varphi_1, \dots, \varphi_n$ are self-guarded formulas with the appropriate number of free variables and generated by the same recursive grammar.

We refer to this as the UCQ syntax for GNFO.

The Landscape of FO-Fragments

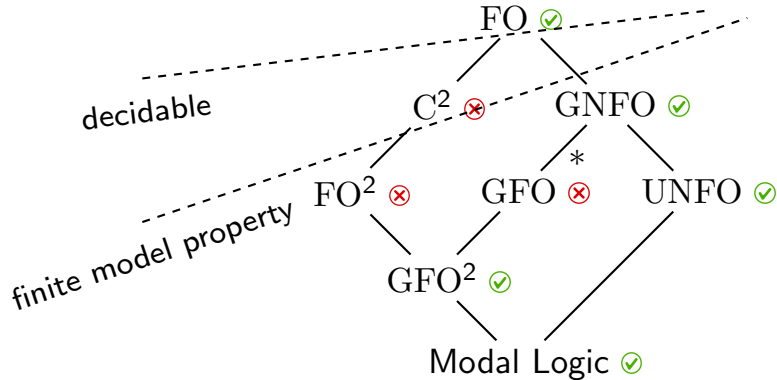


Figure: FO-fragments with (✓) and without (✗) CIP. The inclusion marked with * holds only for sentences and self-guarded formulas.

Main Results

Theorem

Any language extending the two-variable fragment which is closed under substitution and satisfies CIP can express all sentences of first-order logic.

Theorem

GNFO is, in a precise sense, the smallest extension of GFO satisfying CIP.

Undecidability of FO^2 Extensions with CIP

Theorem

Any language extending the two-variable fragment which is closed under substitution and satisfies CIP can express all sentences of first-order logic.

It follows that any such logic L lacks the finite model property and, if the translation from FO to L is computable, then L must be undecidable.

Corollary

Let L be any FO -fragment that extends FO^2 , is closed under substitution, and has CIP. Furthermore, assume that L is effectively closed under conjunction. Then the satisfiability problem for L is undecidable.

Table of Contents

Introduction

Background

CIP and Applications

CIP in Database Theory

Decidable Fragments

Modal Logic as a Fragment of First-Order Logic

The Two-Variable Fragment

The Guarded Fragment

The Guarded-Negation Fragment

The Landscape of FO-Fragments

CIP Fails for GFO

Failure of CIP for GFO

Repairing CIP

What To Do When CIP Fails?

Expressibility Results via Interpolation

FO² and CIP

GFO and CIP

Interlude: Second-Order Logic

Our proofs will use second-order logic.

In second-order logic, we can quantify over predicate symbols. The semantics of these quantifiers are interpreted using *expansions* of models.

$M \models \forall P\varphi$ if $M' \models \varphi$ for all expansions M' of M with an interpretation for P .

$M \models \exists P\varphi$ if $M' \models \varphi$ for some expansion M' of M with an interpretation for P .

Example

The following second-order formula expresses the induction axiom of arithmetic:

$$\forall P(P(0) \wedge \forall x(P(x) \rightarrow P(S(x)))) \rightarrow \forall xP(x).$$

Failure of CIP for GFO

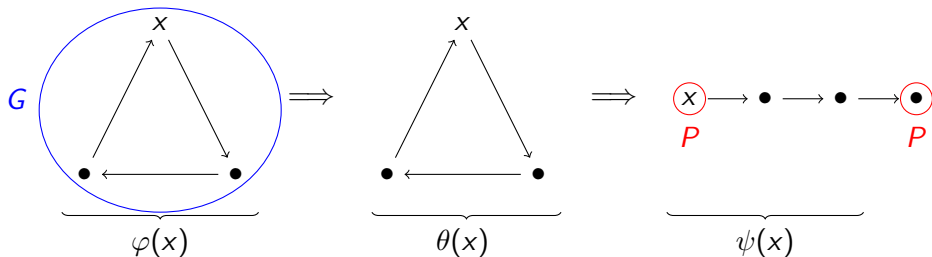
Consider the following formulas:

$$\varphi(x) := \exists y \exists z (G(x, y, z) \wedge R(x, y) \wedge R(y, z) \wedge R(z, x))$$

$$\psi(x) := P(x) \rightarrow \exists y (R(x, y) \wedge \exists z (R(y, z) \wedge \exists u (R(z, u) \wedge P(u))))$$

Then $\varphi(x) \models \psi(x)$ is a valid implication, and the following is an interpolant:

$$\theta(x) := \exists y \exists z (R(x, y) \wedge R(y, z) \wedge R(z, x)).$$



Failure of CIP for GFO

$$\varphi(x) := \exists y \exists z (G(x, y, z) \wedge R(x, y) \wedge R(y, z) \wedge R(z, x))$$

$$\psi(x) := P(x) \rightarrow \exists y (R(x, y) \wedge \exists z (R(y, z) \wedge \exists u (R(z, u) \wedge P(u))))$$

$$\theta(x) := \exists y \exists z (R(x, y) \wedge R(y, z) \wedge R(z, x))$$

In fact, $\theta(x)$ is the *unique* (up to logical equivalence) interpolant of $\varphi(x)$ and $\psi(x)$. To see this, suppose $\alpha(x)$ is some arbitrary interpolant of $\varphi(x)$ and $\psi(x)$. Thus

$$\varphi(x) \models \alpha(x) \models \psi(x),$$

and the predicates G and P must not occur in $\alpha(x)$. This implies that following second-order implication is valid:

$$\exists G \varphi(x) \models \alpha(x) \models \forall P \psi(x).$$

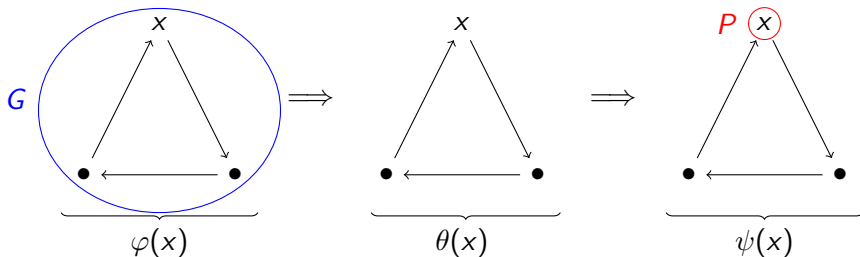
I claim that $\exists G \varphi(x)$ and $\forall P \psi(x)$ are equivalent to $\theta(x)$.

Failure of CIP for GFO

$$\varphi(x) := \exists G \exists y \exists z (G(x, y, z) \wedge R(x, y) \wedge R(y, z) \wedge R(z, x))$$

$$\psi(x) := P(x) \rightarrow \exists y (R(x, y) \wedge \exists z (R(y, z) \wedge \exists u (R(z, u) \wedge P(u))))$$

$$\theta(x) := \exists y \exists z (R(x, y) \wedge R(y, z) \wedge R(z, x))$$



Failure of CIP for GFO

$$\varphi(x) := \exists y \exists z (G(x, y, z) \wedge R(x, y) \wedge R(y, z) \wedge R(z, x))$$

$$\psi(x) := P(x) \rightarrow \exists y (R(x, y) \wedge \exists z (R(y, z) \wedge \exists u (R(z, u) \wedge P(u))))$$

$$\theta(x) := \exists y \exists z (R(x, y) \wedge R(y, z) \wedge R(z, x))$$

Recall, we have an arbitrary interpolant $\alpha(x)$ such that

$$\varphi(x) \models \alpha(x) \models \psi(x).$$

Furthermore, we have that

$$\theta(x) \equiv \exists G \varphi(x) \models \alpha(x) \models \forall P \psi(x) \equiv \theta(x).$$

Therefore,

$$\alpha(x) \equiv \theta(x).$$

Failure of CIP for GFO

Is $\theta(x)$ expressible in GFO?

No: GFO-formulas are preserved by *guarded bisimulations*, and there exist models (M, a) and (N, b) which are guarded bisimilar, but $M \models \theta(a)$ and $N \not\models \theta(b)$.

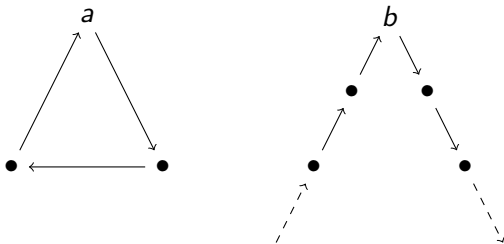


Table of Contents

Introduction

Background

CIP and Applications

CIP in Database Theory

Decidable Fragments

Modal Logic as a Fragment of First-Order Logic

The Two-Variable Fragment

The Guarded Fragment

The Guarded-Negation Fragment

The Landscape of FO-Fragments

CIP Fails for GFO

Failure of CIP for GFO

Repairing CIP

What To Do When CIP Fails?

Expressibility Results via Interpolation

FO² and CIP

GFO and CIP

What to do when CIP fails?

1. We could weaken CIP by requiring that only the relation symbols *not* occurring in a guard position in the interpolant must be in the common signature.
2. Another approach is to develop algorithms for testing whether an interpolant exists for a given entailment. As it turns out, this problem is decidable for both FO^2 and GFO.
3. Given a language L for which CIP fails, we could look for minimal extensions of L satisfying CIP.

Approaches 1 (Hoogland, Marx, 2002) and 2 (Jung, Wolter, 2021) have been taken in the literature.

In our case, we focus on (3).

Expressibility Results via Interpolation

We framed our preceding argument as a proof that GFO does not have CIP.

Another perspective: any extension of GFO with CIP should be able to express $\theta(x)$.

This suggests a general argument:

Suppose L_1 does not have CIP. Let L_2 be some language with CIP that extends L_1 . To show that L_2 can express some formula θ , it suffices to find formulas φ and ψ in L_2 such that $\varphi \models \psi$, and such that θ is the unique interpolant of this implication. Then L_2 must be able to express θ .

To obtain our results, we apply this kind of argument inductively.

FO² + CIP is at least FO

Theorem

Any language extending the two-variable fragment which is closed under substitution and satisfies CIP can express all sentences of first-order logic.

Proof.

Let L be any language with CIP extending FO² and closed under substitution. We show by formula induction that, for every FO-formula $\varphi(x_1, \dots, x_n)$ there is a sentence $\psi \in L$ over an extended signature containing additional unary predicates P_1, \dots, P_n , that is equivalent to

$$\exists x_1 \dots x_n \left(\left(\bigwedge_{i=1, \dots, n} P_i(x_i) \wedge \forall y (P_i(y) \rightarrow y = x_i) \right) \wedge \varphi(x_1, \dots, x_n) \right).$$

In other words, ψ is a sentence expressing that φ holds under an assignment of its free variables to some tuple of elements which uniquely satisfy the P_i predicates. In the case that φ is a sentence, we then have that $\psi \equiv \varphi$.

GNFO as the minimal GFO-extension with CIP

Theorem

GNFO is, in a precise sense, the smallest extension of GFO satisfying CIP.

We can now clarify what we mean by “in a precise sense.”

We let L be any language with the following properties:

1. L can express all self-guarded formulas of GFO,
2. L is closed under self-guarded substitution,
3. L is closed under conjunction and disjunction, and
4. L has CIP.

It's worth noting that GFO satisfies (1)-(3), while GNFO satisfies (1)-(4).

GNFO as the minimal GFO-extension with CIP

The following is the formal statement of the main theorem:

Theorem

Let L be any FO-fragment such that

- 1. L can express all self-guarded formulas of GFO,*
- 2. L is closed under self-guarded substitution,*
- 3. L is closed under conjunction and disjunction, and*
- 4. L has CIP.*

Then $\text{GNFO} \preceq L$.

Then since GNFO satisfies (1)-(4), the description of GNFO as the “smallest” such extension of GFO is justified.

GNFO as the minimal GFO-extension with CIP

Some easy consequences of our assumptions about L :

1. L includes all atomic formulas.
2. L can express guarded quantification,
3. L can express guarded negation, and
4. L can express “unary implication.”

By *unary implication*, we mean that, whenever L can express a formula φ , it can also express $P(x) \rightarrow \varphi$ for any unary predicate P .

GNFO as the minimal GFO-extension with CIP

The main thrust of proof is a nested induction to show the following technical result:

Proposition

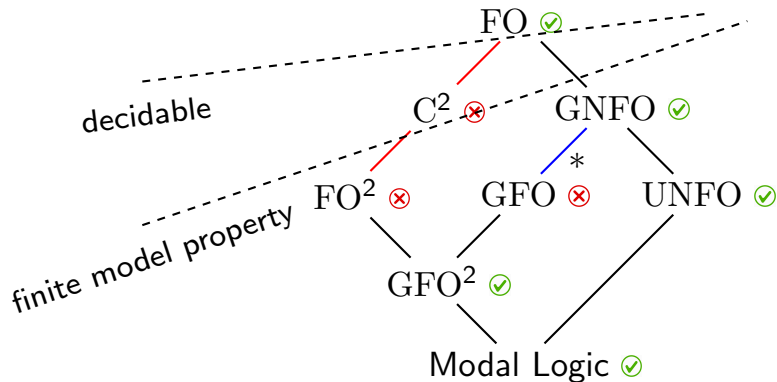
Let L be any FO-fragment with CIP that includes all atomic formulas and is closed under guarded quantification, conjunction, and unary implication. Then $\text{FO}_{\exists, \wedge} \preceq L$.

Once we have that L can express all $\text{FO}_{\exists, \wedge}$ -formulas, it follows quickly from closure under disjunction and self-guarded substitution that L can express all UCQs. Since L can also express guarded negation and is closed under self-guarded substitution, it can express the entire UCQ syntax of GNFO:

$$\varphi := R(\bar{x}) \mid x = y \mid \alpha \wedge \neg\varphi \mid q[\varphi_1/R_1, \dots, \varphi_n/R_n].$$

From this, the theorem follows immediately.

The Landscape of FO-Fragments



Nothing along the **red** line with substitution (except FO itself) can have CIP.

Nothing along the **blue** line with conjunction, disjunction, and self-guarded substitution can have CIP.