

Craig Interpolation for Decidable First-Order Fragments

Balder ten Cate¹[0000–0002–2538–5846] and Jesse Comer²[0009–0006–9734–3457]

¹ ILLC, University of Amsterdam, Amsterdam 1098 XH, NL

² University of Pennsylvania, Philadelphia, PA 19104, USA

Abstract. We show that the guarded-negation fragment (GNFO) is, in a precise sense, the smallest extension of the guarded fragment (GFO) with Craig interpolation. In contrast, we show that the smallest extension of the two-variable fragment (FO^2), and of the forward fragment (FF) with Craig interpolation, is full first-order logic. Similarly, we also show that all extensions of FO^2 and of the fluted fragment (FL) with Craig interpolation are undecidable.

Keywords: Craig interpolation · Decidability · Abstract model theory.

1 Introduction

The study of decidable fragments of first-order logic (FO) is a topic with a long history, dating back to the early 1900s ([39,51], cf. also [16]), and more actively pursued since the 1990s. Inspired by Vardi [54], who asked “what makes modal logic so robustly decidable?” and Andreka et al. [1], who asked “what makes modal logic tick?” many decidable fragments have been introduced and studied over the last 25 years that take inspiration from modal logic (ML), which itself can be viewed as a fragment of FO that features a restricted form of quantification. These include the following fragments, each of which naturally generalizes modal logic in a different way: the *two-variable fragment* (FO^2) [41], the *guarded fragment* (GFO) [1], and the *unary negation fragment* (UNFO) [21]. Further decidable extensions of these fragments were subsequently identified, including the *two-variable fragment with counting quantifiers* (C^2) [28] and the *guarded negation fragment* (GNFO) [4]. The latter can be viewed as a common generalization of GFO and UNFO. Many decidable logics used in computer science and AI, including various description logics and rule-based languages, can be translated into GNFO and/or C^2 . In this sense, GNFO and C^2 are convenient tools for explaining the decidability of other logics. Extensions of GNFO have been studied that push the decidability frontier even further (for instance with fixed-point operators and using clique-guards), but these fall outside the scope of this paper.

In an earlier line of investigation, Quine identified the decidable *fluted fragment* (FL) [50], the first of several *ordered logics* which have been the subject of recent interest [46,47,48,49,43]. The idea behind ordered logics is to restrict the order in which variables are allowed to occur in atomic formulas and quantifiers.

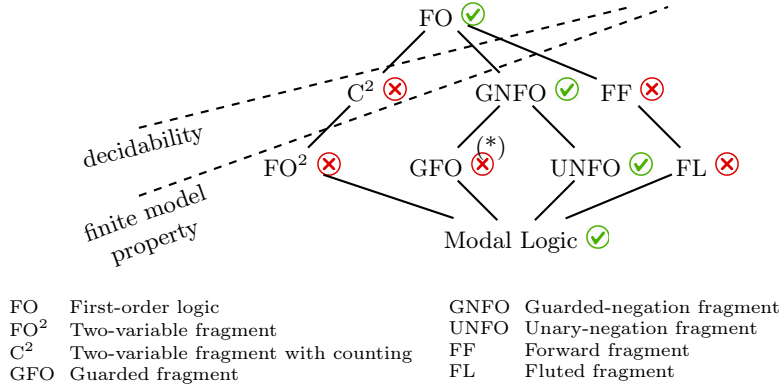


Fig. 1. Landscape of decidable fragments of FO with (✓) and without (✗) CIP. The inclusion marked (*) holds only for sentences and self-guarded formulas.

Another recently introduced decidable fragment that falls in this family is the *forward fragment* (FF), whose syntax strictly generalizes that of FL. Both FL and FF have the finite model property (FMP) [43] and embed ML [33], but are incomparable in expressive power to GFO [44], FO², and UNFO.³

Ideally, an FO-fragment is not only decidable, but also model-theoretically well behaved. A particularly important model-theoretic property of logics is the *Craig Interpolation Property* (CIP). It states that, for all formulas φ, ψ , if $\varphi \models \psi$, then there exists a formula ϑ such that $\varphi \models \vartheta$ and $\vartheta \models \psi$, and such that all non-logical symbols occurring in ϑ occur both in φ and in ψ . Craig [23] proved in 1957 that FO itself has this property (hence the name). Several refinements of Craig’s result have subsequently been obtained (e.g., [42,10]). These have found applications in various areas of computer science and AI, including formal verification, modular hard/software specification and automated deduction [40,18,30], and are emerging as a new prominent technology in databases [52,12] and knowledge representation [38,20,36]. While we have described CIP here as a model theoretic property, it also has a proof-theoretic interpretation. Indeed, it has been argued that CIP is an indicator for the existence of nice proof systems [31].

Turning our attention to the decidable fragments of FO we mentioned earlier, it turns out that, although GFO is in many ways model-theoretically well-behaved [1], it lacks CIP [32]. Likewise, FO² lacks CIP [22] and the same holds for C² ([34, Example 2] yields a counterexample). Both FF and FL lack CIP [7]. On the other hand, UNFO and GNFO have CIP [21,3]. Figure 1 summarizes these known results. Note that we restrict attention to relational signatures without constant symbols and function symbols. Some of the results depend on this restriction. Other known results not reflected in Figure 1 (to avoid clutter) are that

³ Specifically, the FO-sentence $\exists xy(R(x, y) \wedge R(y, x))$ belongs to GFO, FO² and UNFO, but is not expressible in FF, since the structure consisting of two points with symmetric edges and the structure (\mathbb{Z}, S) with S the successor relation, are “infix bisimilar,” as described in [7].

the intersection of GFO and FO^2 (also known as GFO^2) has CIP [32]. Similarly, the intersection of FF with GFO and the intersection of FL with GFO (known as G_{FF} and G_{FL} , respectively) have CIP [7].

When a logic L lacks CIP, the question naturally arises as to whether there exists a more expressive logic L' that has CIP. If such an L' exists, then, in particular, interpolants for valid L -implications can be found in L' . This line of analysis is sometimes referred to as *Repairing Interpolation* [2]. If L' is an FO-fragment, and our aim is to repair interpolation by extension, then there is a trivial solution: FO itself is an extension of L satisfying CIP. We will instead consider the following refinement of the question: can a natural extension L' of L be identified which satisfies CIP while retaining decidability? We will answer this question for three of the fragments depicted in Figure 1 that lack CIP, by identifying the minimal natural extension L' of L satisfying CIP. Our main results can be stated informally as follows:

1. The smallest logic extending GFO that has CIP is GNFO.
2. The smallest logic extending FO^2 that has CIP is FO, and no decidable extension of FO^2 has CIP.
3. The smallest logic extending FF that has CIP is FO, and no decidable extension of FL has CIP.

The precise statements of these results will be given in the respective sections. They involve some natural closure assumptions on the logics in question, and, for the undecidability results, some assumptions regarding the effective computability of the translation between the extension and the logic that it extends.

These results give us a clear sense of where, in the larger landscape of decidable fragments of FO, we may find logics that enjoy CIP. What makes the above results remarkable is that, from the definition of the Craig interpolation property, it doesn't appear to follow that a logic without CIP would have a unique minimal extension with CIP. Note that a valid implication may have many possible interpolants, and the Craig interpolation property merely requires the existence of one such interpolant. Nevertheless, the above results show that, in the case FO^2 , GFO, and FF, such a unique minimal extension indeed exists (assuming suitable closure properties, which will be spelled out in detail in the next sections).

Related Work. Several other approaches have been proposed for dealing with logics that lack CIP. One approach is to weaken CIP. For example, it was shown in [32] that GFO satisfies a weak, “modal” form of Craig interpolation, where, roughly speaking, only the relation symbols that occur in non-guard positions in the interpolant are required to occur both in the premise and the conclusion. As it turns out, this weakening of CIP is strong enough to entail the (non-projective) *Beth Definability Property*, which is one important use case of CIP. See also Section 7 for further discussion of weak forms of CIP.

Another recent approach [34] is to develop algorithms for testing whether an interpolant exists for a given entailment. That is, rather than viewing Craig interpolation as a property of logics, the existence of interpolants is studied as an

algorithmic problem at the level of individual entailments. The interpolant existence problem turns out to be indeed decidable (although of higher complexity than the satisfiability problem) for both GFO and FO^2 [34].

Additional results are known for UNFO and GNFO beyond the fact that they have CIP. In particular, CIP holds for their fixed-point extensions [9,8], interpolants can be constructed effectively, and tight bounds are known on the size of interpolants and the computational complexity of computing them [11].

Our paper can be viewed as an instance of *abstract model theory* for fragments of FO. One large driving force behind the development of abstract model theory was the identification of *extensions* of FO which satisfy desirable model-theoretic properties, such as the compactness theorem, the Löwenheim-Skolem, and Craig interpolation. One takeaway from this line of research is that CIP is scarce among many “reasonable” FO-extensions. An early result of Lindström showed that FO-extensions with finitely many generalized quantifiers and satisfying the downward Löwenheim-Skolem property do not have the Beth property (and hence fail to satisfy CIP) [37]. Similarly, Caicedo [17], generalizing an early result by Friedman [25], established a strong negative CIP result that applies to arbitrary proper FO-extensions with monadic generalized quantifiers. For a survey of negative interpolation results among FO-extensions, see [53]. These negative results not only show that CIP is scarce among extensions of FO, they also provide clues as to where, within the space of all extensions, one may hope to find logics with CIP. Our results can be viewed similarly, except that they pertain to (extensions of) fragments of FO.

Our results can also be appreciated as characterizations of GNFO and of FO. While traditional Lindström-style characterizations are maximality theorems (e.g., FO is a maximal logic having the compactness and Löwenheim-Skolem properties), our results can be viewed as minimality theorems (e.g., GNFO is the minimal logic extending GFO and having CIP).

Some prior work exists that studies abstract model theory for (extensions of) fragments of FO. Most closely related is [19], which studies modal logics and hybrid logics. Among other things, it was shown in [19] that the smallest extension of modal logic with the difference operator (ML(D)) which satisfies CIP is full first-order logic. Additionally, in [27], the authors identified minimal extensions of various fragments of propositional linear temporal logic (PLTL) with CIP. Furthermore, it was shown in [19] that every abstract logic extending GFO with CIP can express all FO sentences and formulas with one free variable, and is thus undecidable. A crucial difference between this result and ours is that [19] assumes signatures with constant symbols and concerns a stronger version of CIP, interpolating not only over relation symbols but also over constant symbols. In contrast, we only consider purely relational signatures without constant symbols. Other prior work on abstract model theory for fragments of FO are [13,15,26]. Repairing interpolation has also been pursued in the context of quantified modal logics, which typically lack CIP; in [2], the authors showed that CIP can be repaired for such logics by adding nominals, @-operators and the \downarrow -binder.

Outline. Section 2 introduces the abstract model-theoretic framework. In Sections 3, 4, and 5, we repair interpolation for FO^2 , GFO, and FF, respectively. In Section 6, we provide results showing that, even with weak expressive assumptions, extensions of FO^2 and FL with CIP are undecidable. In Section 7, we discuss the implications and limitations of our results, and future directions.

2 Preliminaries

We assume familiarity with the syntax and semantics of FO. Signatures are denoted by σ and τ , and are assumed to be relational and finite. If φ contains only relation symbols occurring in σ , then we write $M, g \models \varphi$ to denote that a σ -structure M satisfies φ under the variable assignment g . We write x_i, y_i, z_i, u_i to denote variables, and $\bar{x}, \bar{y}, \bar{z}, \bar{u}$ to denote tuples of variables. We write a_i, b_i, c_i to denote elements of structures and $\bar{a}, \bar{b}, \bar{c}$ to denote tuples of such elements. Given a tuple of elements $\bar{a} = a_1, \dots, a_n$ in a structure M , a tuple of variables $\bar{x} = x_1, \dots, x_n$, and a variable assignment g , we write $g[\bar{x}/\bar{a}]$ to denote the variable assignment which is the same as g except that $g(x_i) = a_i$ for each $i \leq n$. In order to state our main results precisely, we must formally define what we mean by *extensions* L' of L (where L is some fragment of FO that lacks CIP). One option is to let L' range over fragments of FO that syntactically include L . However, as it turns out, our main results apply even to extensions that are not themselves contained in FO. We therefore opt, instead, to work with an abstract definition of logics, as typically used in abstract model theory.

Abstract Logics. An *abstract logic* (or *logic*) is a pair (L, \models_L) , where L is a map from relational signatures σ to collections of *formulas*, and \models_L is a ternary *satisfaction relation*. A *formula* of an abstract logic (L, \models_L) is an element of $L(\sigma)$ for some finite relational signature σ . L must be monotone: if $\sigma \subseteq \tau$, then $L(\sigma) \subseteq L(\tau)$. Each formula φ has an associated finite set of free variables $\text{free}(\varphi)$, and we write $\varphi(\bar{x})$ or $\varphi(x_1, \dots, x_k)$ to denote that the free variables of φ are exactly those in the tuple $\bar{x} = x_1, \dots, x_k$. As in the case of FO, a formula φ is a *sentence* if $\text{free}(\varphi) = \emptyset$. We write $\text{sig}(\varphi)$ to denote the least signature σ such that $\varphi \in L(\sigma)$. The ternary *satisfaction relation* \models_L is defined over triples (M, g, φ) , where φ is an L -formula, M is a τ -structure such that $\text{sig}(\varphi) \subseteq \tau$, and g is a variable assignment with $\text{free}(\varphi) \subseteq \text{dom}(g)$; we write $M, g \models_L \varphi$ if this relation holds between these objects. The notions of logical consequence and logical equivalence for abstract logics are defined completely analogously to FO. In later sections, we will prefer to suppress the subscript L in the notation for the satisfaction relation and write L to denote an abstract logic (L, \models_L) . Furthermore, we often write $\varphi \in L$ rather than $\varphi \in L(\sigma)$, leaving the signature implicit.

All abstract logics L are assumed to satisfy the *reduct property* and the *renaming property*. The *reduct property* states that if $\sigma \subseteq \tau$, then for all $\varphi \in L(\sigma)$, all τ -structures M , and all assignments g , if $M, g \models_L \varphi$, then $M \upharpoonright \sigma, g \models_L \varphi$. In other words, the truth of a formula of an abstract logic L in a structure depends only on the interpretations of the symbols in the signature of that formula. The

renaming property states that if $\rho : \sigma \rightarrow \tau$ is an injective map preserving the arity of relation symbols, then for each formula $\varphi \in L(\sigma)$, there is a formula $\psi \in L(\tau)$ such that for all τ -structures M , we have that $M, g \models_L \psi$ if and only if $\rho^{-1}[M], g \models_L \varphi$, where $\rho^{-1}[M]$ is the σ -structure with the same domain as M where, for each $R \in \sigma$, we have that $R^{\rho^{-1}[M]} = \rho(R)^M$. Intuitively, the renaming property states that if a formula over a signature σ can be expressed in a logic L , then the formula obtained by renaming all of its relation symbols can also be expressed in L .

For arbitrary abstract logics L , the Craig interpolation property states that if $\varphi \models_L \psi$ for L -formulas φ and ψ , then there exists a formula $\vartheta \in L(\text{sig}(\varphi) \cap \text{sig}(\psi))$ with $\text{free}(\vartheta) = \text{free}(\varphi) \cap \text{free}(\psi)$ such that $\varphi \models_L \vartheta$ and $\vartheta \models_L \psi$.

We say a formula φ of a logic L expresses a formula ψ of a logic L' if $\text{free}(\varphi) = \text{free}(\psi)$, $\text{sig}(\varphi) = \text{sig}(\psi)$, and for all structures M and assignments g , we have that $M, g \models_L \varphi$ if and only if $M, g \models_{L'} \psi$. We say that a logic L' is an *extension* of a logic L (notation: $L \preceq L'$) if L' can express all formulas of L . An FO-fragment can then be precisely defined, without reference to syntax, as a logic of which FO is an extension. We say that L' is a *sentential extension* of L (notation: $L \preceq_{\text{sent}} L'$) if L' can express all sentences of L .

Let L be a logic and $\psi(x_1, \dots, x_n)$ be an L -formula. We write $\llbracket \psi \rrbracket^M$ for the collection of tuples $(a_1, \dots, a_n) \in M^n$ such that there exists an assignment g where $M, g \models \psi$ and $g(x_i) = a_i$ for each $i \leq n$. Given formulas $\psi_1, \dots, \psi_k \in L(\sigma)$, a σ -structure M , and relation symbols $R_1, \dots, R_k \in \sigma$ with $|\text{free}(\psi_i)| = \text{arity}(R_i)$ for each $i \leq k$, we define $M[R_1/\psi_1, \dots, R_k/\psi_k]$ to be the σ -structure with the same domain as M and such that $R_i^{M[R_1/\psi_1, \dots, R_k/\psi_k]} = \llbracket \psi_i \rrbracket^M$ for each $i \leq k$. We now describe a syntax-free notion of uniform substitution for formulas of an abstract logic.

Definition 2.1. *Let L be a logic and $\varphi \in L(\sigma)$ with $R_1, \dots, R_k \in \text{sig}(\varphi)$, where for each $i \leq k$, we have that R_i is a k_i -ary relation symbol. Furthermore, let $\psi_1, \dots, \psi_k \in L(\sigma)$ be formulas with $|\text{free}(\psi_i)| = k_i$ for each $i \leq k$. We say that L expresses the substitution of ψ_1, \dots, ψ_k for R_1, \dots, R_k in φ if there exists a formula $\chi \in L(\sigma)$ such that, for every σ -structure M ,*

$$M, g \models \chi \iff M[R_1/\psi_1, \dots, R_k/\psi_k], g \models \varphi.$$

Most studies in abstract logic assume that the logics under study are *regular*, roughly meaning that they can express atomic formulas, Boolean connectives, and existential quantification. In other words, to study regular logics is to study extensions of FO. Since we are interested in a more fine-grained view of logics including FO-fragments, these assumptions are too strong. As a result, the first step of studying extensions of FO-fragments from the perspective of abstract logic is to identify natural expressive assumptions for those extensions which are strictly weaker than regularity. We do this in the respective sections.

Some of our proofs will use second-order quantification (for expository reasons only), and we recall the semantics of these quantifiers here. Given a formula $\varphi \in L(\sigma \cup \{P\})$ of some abstract logic L , we can form new formulas $\exists P\varphi$ and

$\forall P\varphi$ with signature σ and the same free variables as φ . Given a σ -structure M and an assignment g , the semantics of these formulas are defined as follows:

$$\begin{aligned} M, g \models \exists P\varphi & \text{ if there is a } \sigma \cup \{P\}\text{-expansion } M' \text{ of } M \\ & \text{ such that } M', g \models \varphi, \text{ and} \\ M, g \models \forall P\varphi & \text{ if for all } \sigma \cup \{P\}\text{-expansions } M' \text{ of } M, \\ & \text{ we have that } M', g \models \varphi. \end{aligned}$$

If L itself does not allow second-order quantification, we can view $\exists P\varphi$ and $\forall P\varphi$ as elements of $L'(\sigma)$ for a suitable extension L' of L . In particular, if φ is an FO-formula, then $\exists P\varphi$ and $\forall P\varphi$ are formulas of second-order logic (SO).

3 Repairing Interpolation for FO²

The two-variable fragment (FO²) consists of all FO-formulas containing only two variables, say, x and y , where we allow for nested quantifiers that reuse the same variable (as in $\exists xy(R(x, y) \wedge \exists x(R(y, x)))$), expressing the existence of a path of length 2). In this context, as is customary, we restrict attention to relations of arity at most 2. It is known that FO² is decidable [41] but does not have CIP [22].

3.1 Natural Extensions of FO²

While FO² is restricted to only two variables and predicates of arity at most 2, it has no restriction on its connectives: it is fully closed under Boolean connectives and existential and universal quantification. Because of this fact, we will consider in this section those abstract logics which are *strong extensions* of FO².

Definition 3.1. *We say that a logic L' strongly extends a logic L if L' extends L and, for each formula $\varphi \in L'$ with $R_1, \dots, R_k \in \text{sig}(\varphi)$, where φ expresses some $\psi \in L$, and all formulas $\psi_1, \dots, \psi_k \in L'$, we have that L' expresses the substitution of ψ_1, \dots, ψ_k for R_1, \dots, R_k in φ (cf. Definition 2.1).*

Intuitively, Definition 3.1 means that L' can express uniform substitutions of its formulas into formulas of L . In other words, the notion of a strong extension is a syntax-free way to say that L' extends L and is closed under the connectives of L . In particular, if L strongly extends FO², then L can express all of the usual first-order connectives: for ψ_0 and ψ_1 expressible in L , it must also be the case that $\neg\psi_0$, $\psi_0 \wedge \psi_1$, and $\exists x\psi_0$ are expressible in L , under the usual semantics of these connectives. Clearly FO² is the smallest strong extension of itself.

3.2 Finding the Minimal Extension of FO² with CIP

Recall that we write $L \preceq_{\text{sent}} L'$ if every sentence of L is expressible in L' . Our main result in this section is the following.

Theorem 3.1. *If L is a strong extension of FO² with CIP, then $\text{FO} \preceq_{\text{sent}} L$.*

Proof. We will show by induction on the complexity of formulas 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 $n = 0$ (i.e., the case that φ is a sentence), we then have that ψ is equivalent to φ , which shows that $FO \preceq_{sent} L$.

The base case of the induction is straightforward (recall that we restrict attention to relations of arity at most 2). The induction step for the Boolean connectives is straightforward as well (using the fact that L is a strong extension of FO^2 , and thus can express all connectives of FO^2). In fact, the only non-trivial part of the argument is the induction step for the existential quantifier. Let $\varphi(x_1, \dots, x_n)$ be of the form $\exists x_{n+1} \varphi'(x_1 \dots, x_n, x_{n+1})$. By the inductive hypothesis, there is an L -sentence ψ with $\text{sig}(\psi) = \text{sig}(\varphi') \cup \{P_1, \dots, P_{n+1}\}$, where P_1, \dots, P_{n+1} are unary predicates not in $\text{sig}(\varphi')$, which is equivalent to

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

Now, let ψ' be obtained from ψ by replacing every occurrence of P_{n+1} by P' for some fresh unary predicate P' ; this is expressible in L by the renaming property. Furthermore, let

$$\begin{aligned} \gamma(x) &:= \psi \wedge P_{n+1}(x), \text{ and} \\ \chi(x) &:= (P'(x) \wedge \forall y (P'(y) \rightarrow y = x)) \rightarrow \psi'. \end{aligned}$$

(where x is either of the two variables we have at our disposal; it does not matter which). Since L strongly extends FO^2 , both can be written as an L -formula. Then

$$\gamma(x) \models \chi(x).$$

Let $\vartheta(x) \in L$ be an interpolant. Observe that since P_{n+1} occurs only in $\gamma(x)$ and P' only in $\chi(x)$, the following second-order entailment is also valid:

$$\exists P_{n+1} \gamma(x) \models \vartheta(x) \models \forall P' \chi(x).$$

It is not hard to see that $\exists P_{n+1} \gamma(x)$ and $\forall P' \chi(x)$ are equivalent. Indeed, both are satisfied in a structure M under an assignment g precisely if $M', g \models \varphi$, where M' is the expansion of M in which P_{n+1} denotes the singleton set $\{g(x_{n+1})\}$. It then follows that $\vartheta(x)$, being sandwiched between the two, is also equivalent to $\exists P_{n+1} \gamma(x)$. This implies that $\vartheta(x)$ is the unique interpolant (up to logical equivalence) of the entailment $\gamma(x) \models \chi(x)$, and so it is expressible in L . Then since L strongly extends FO^2 , it can express $\exists x \vartheta(x)$. We claim that this sentence satisfies the requirement of our claim. To see this, observe that $\exists x \vartheta(x)$ is equivalent to $\exists x \exists P_{n+1} \gamma(x)$, which is equivalent to $\exists P_{n+1} \psi$, which clearly satisfies the requirement of our claim. \square

4 Repairing Interpolation for GFO

The guarded fragment (GFO) [1] allows formulas in which all quantifiers are “guarded.” Formally, a *guard* for a formula φ is an atomic formula α whose free variables include all free variables of φ . Following [29], we allow α to be an equality. More generally, by an \exists -*guard* for φ , we will mean a possibly-existentially-quantified atomic formula $\exists \bar{x}\beta$ whose free variables include all free variables of φ . The formulas of GFO are generated by the following grammar:

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

where, in the last clause, α is a guard for φ . Note again that we do not allow constants and function symbols.

In the guarded-negation fragment (GNFO) [4], arbitrary existential quantification is allowed, but every negation is required to be guarded. More precisely, the formulas of GNFO are generated by the following grammar:

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

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

As is customary, the above definitions are phrased in terms of ordinary guards α . However, it is easy to see that if we allow for \exists -guards, this would not affect the expressive power (or computational complexity) of these logics in any way. This is because, when the variables in the tuple \bar{x} do not occur free in φ , as is the case when $\exists \bar{x}\beta$ is an \exists -guard for φ , then we can write $\exists \bar{x}\beta \wedge \varphi$ equivalently as $\exists \bar{x}(\beta \wedge \varphi)$. In other words, an \exists -guard is as good as an ordinary guard. We call an FO-formula *self-guarded* if it is either a sentence or it is of the form $\alpha \wedge \varphi$ where α is an \exists -guard for φ .

In this section, we will require the notions of *conjunctive queries* (CQs) and *unions of conjunctive queries* (UCQs). A CQ is an FO-formula of the form

$$\varphi(x_1, \dots, x_n) := \exists y_1 \dots \exists y_m \left(\bigwedge_{i \in I} \alpha_i \right),$$

where each α_i is an atomic relation, possibly an equality, whose free variables are among $\{x_1, \dots, x_n, y_1, \dots, y_m\}$. The collection of all CQs is expressively equivalent to the fragment $\text{FO}_{\exists, \wedge}$ of first-order logic, which is generated by the following grammar:

$$\varphi := R(x_1, \dots, x_k) \mid x = y \mid \varphi \wedge \varphi \mid \exists x\varphi.$$

A UCQ is a finite disjunction of CQs. Importantly, GNFO can be alternatively characterized as the smallest logic which can express every UCQ and is closed under guarded negation [4]. This is made explicit in the following expressively equivalent grammar for GNFO:

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

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.

4.1 Natural Extensions of GFO

Unlike FO², guarded fragments are peculiar in that they are not closed under substitution. For example, $\exists xy(R(x, y) \wedge \neg S(x, y))$ belongs to GFO, but if we substitute $x = x \wedge y = y$ for $R(x, y)$, we obtain $\exists xy(x = x \wedge y = y \wedge \neg S(x, y))$, which does not belong to GFO (and is not even expressible in GNFO). GFO and GNFO are, however, closed under *self-guarded substitution*: we can uniformly substitute self-guarded formulas for atomic relations. We generalize the notion of a self-guarded formula to abstract logics L as follows: a formula $\varphi(\bar{x}) \in L(\sigma)$ with $\text{free}(\varphi) = \{x_1, \dots, x_k\}$ is *self-guarded* if there is a n -ary relation symbol $G \in \sigma$, where $n \geq k$, and a tuple of variables \bar{y} containing exactly the variables $\text{free}(\varphi) \cup \{z_1, \dots, z_m\}$, such that for all σ -structures M and assignments g ,

$$M, g \models \varphi \implies M, g \models \exists z_1 \dots \exists z_m G(\bar{y}).$$

Intuitively, we can think of a self-guarded L -formula as a conjunction of the form $\alpha \wedge \psi$, where α is an \exists -guard for ψ . We can then capture the notion of self-guarded substitution for abstract logics by the following definition.

Definition 4.1. *We say that an abstract logic L expresses self-guarded substitutions if, for each formula $\varphi \in L$ with $R_1, \dots, R_k \in \text{sig}(\varphi)$, and all self-guarded formulas $\psi_1, \dots, \psi_k \in L$, we have that L can express the substitution of ψ_1, \dots, ψ_k for R_1, \dots, R_k in φ (cf. Definition 2.1).*

It was shown in [4] that every self-guarded GFO-formula is expressible in GNFO. In particular, this applies to all GFO-sentences and GFO-formulas with at most one free variable (since all such formulas can be equivalently written as $x = x \wedge \varphi$). It is therefore common to treat GNFO as an extension of GFO. To make this precise, we say that L' is a *self-guarded extension* of L if L' can express all *self-guarded* formulas of L (notation: $L \preceq_{sg} L'$). In Figure 1, the line marked (*) indicates that GNFO extends GFO in this weaker sense. Furthermore, it is worth noting that GNFO is also not closed under implication, while GFO is. If it were, then GNFO would be able to express full negation (using formulas of the form $\varphi \rightarrow \perp$). However, GFO and GNFO both have disjunction and conjunction in common. We formalize all of these considerations into the following notion.

Definition 4.2. *A guarded logic is a logic L such that*

1. $\text{GFO} \preceq_{sg} L$,
2. L expresses self-guarded substitutions, and
3. L expresses conjunction and disjunction.

Clearly, GFO and GNFO are both guarded logics. Furthermore, observe that the *smallest* guarded logic consists of all conjunctions and disjunctions of self-guarded formulas of GFO.

4.2 Finding the Minimal Extension of GFO with CIP

Our main result in this section is the following.

Theorem 4.1. *Let L be a guarded logic with CIP. Then $\text{GNFO} \preceq L$.*

In other words, loosely speaking, GNFO is the smallest extension of GFO with CIP. It is based on similar ideas as the proof of Theorem 3.1, but the argument is more intricate. The main thrust of the argument will be to show that our abstract logic L can express all positive existential formulas, from which it will follow easily that L is able to express all formulas in the UCQ syntax for GNFO. Toward this end, the main technical result is the following proposition.

Proposition 4.1. *Let L be a logic with CIP that can express atomic formulas, guarded quantification, conjunction, and unary implication. Then $\text{FO}_{\exists, \wedge} \preceq L$.*

Here, we say that a logic L can *express guarded quantification* if, whenever $\varphi \in L$ and α is a guard for φ , L can express $\exists \bar{x}(\alpha \wedge \varphi)$; we say that L can *express unary implications* if, whenever $\varphi \in L$ and α is an atomic formula with only one free variable, L can express $\alpha \rightarrow \varphi$.

The following definition is used in the proof of Proposition 4.1.

Definition 4.3. *Let φ be a formula in $\text{FO}_{\exists, \wedge}$, let $\bar{y} = y_1, \dots, y_n$ be a tuple of distinct variables, and let $\bar{P} = P_1, \dots, P_n$ be a tuple of unary predicates of the same length. Then $\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi)$ is defined recursively as follows:*

$$\begin{aligned} \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\alpha) &= \exists \bar{y}(\alpha \wedge \bigwedge_{1 \leq i \leq n} P_i(y_i)) \\ \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\phi \wedge \psi) &= \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\phi) \wedge \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi) \\ \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\exists z \psi) &= \exists z(\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)), \end{aligned}$$

where α is an atomic formula (possibly an equality).

The $\text{BIND}_{\bar{y} \rightarrow \bar{P}}$ operation applied to a formula $\varphi \in \text{FO}_{\exists, \wedge}$ wraps each atomic subformula of φ with quantifiers for the variables in \bar{y} , and adds additional unary predicates for these variables. Thus, the free variables of $\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi)$, for $\bar{y} = y_1, \dots, y_n$, are exactly $\text{free}(\varphi) \setminus \{y_1, \dots, y_n\}$, which justifies our use of the word ‘‘BIND’’. The utility of this definition is due to the following fact: for any $\varphi \in \text{FO}_{\exists, \wedge}$, whenever $M, g \models \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi)$, and the interpretation in M of each unary predicate P_i in \bar{P} is a singleton, then $M, g' \models \varphi$, where g' is the extension of g which maps each y_i to the unique element satisfying P_i (cf. Propositions 4.3, 4.4). The following proposition is a simple consequence of the definition of BIND.

Proposition 4.2. *For all $\text{FO}_{\exists, \wedge}$ -formulas φ and for all \bar{x}, \bar{y} and \bar{P}, \bar{Q} , if \bar{x} and \bar{y} are disjoint, then $\text{BIND}_{\bar{x}\bar{y} \rightarrow \bar{P}\bar{Q}}(\varphi) \equiv \text{BIND}_{\bar{x} \rightarrow \bar{P}}(\text{BIND}_{\bar{y} \rightarrow \bar{Q}}(\varphi))$.*

A formula φ is *clean* if no free variable of φ also occurs bound in φ , and φ does not contain two quantifiers for the same variable. Every FO-formula is equivalent to a clean FO-formula, and all subformulas of a clean formula are also clean. We now state two technical propositions, whose proofs can be found in the appendix.

Proposition 4.3. *For every clean $\text{FO}_{\exists,\wedge}$ -formula φ , for every tuple of distinct variables $\bar{y} = y_1, \dots, y_n$ (with each $y_i \in \text{free}(\varphi)$), and for every tuple of unary predicates $\bar{P} = P_1, \dots, P_n$, we have that*

$$\left(\bigwedge_{i=1, \dots, n} P_i(y_i) \right) \models \varphi \rightarrow \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi).$$

Proposition 4.4. *For every clean $\text{FO}_{\exists,\wedge}$ -formula $\varphi(x, \bar{y})$ with $\bar{y} = y_1, \dots, y_n$ distinct from x , and for every n -tuple of unary predicates $\bar{P} = P_1, \dots, P_n$ not occurring in φ , we have that*

$$\exists x \varphi(x, \bar{y}) \equiv \forall \bar{P} \left(\left(\bigwedge_{i=1 \dots n} P_i(y_i) \right) \rightarrow \exists x \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi(x, \bar{y})) \right).$$

The following lemma enables the proof of Proposition 4.1.

Lemma 4.1. *Let L be an FO-fragment which can express atomic formulas and is closed under guarded quantification, conjunction, and unary implication. If L can express a formula $\varphi \in \text{FO}_{\exists,\wedge}$ and all of its subformulas, then for all tuples \bar{y} of variables, we have that L can express $\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi)$.*

Proof. We show by strong induction on the complexity of clean $\text{FO}_{\exists,\wedge}$ -formulas φ that this proposition holds.

Base Case

Suppose φ is an atomic formula. Fix an arbitrary tuple $\bar{y} = y_1 \dots, y_n$. Then

$$\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi) \equiv \exists \bar{y} (\varphi \wedge \bigwedge_{1 \leq i \leq n} P_i(y_i)),$$

which L can express by closure under conjunction and guarded quantification.

Inductive Step

Suppose inductively that, for all formulas ψ of lesser complexity than φ , and all tuples \bar{z} of variables, we have that L can express $\text{BIND}_{\bar{z} \rightarrow \bar{P}}(\psi)$. Fix an arbitrary tuple \bar{y} of variables.

Suppose that $\varphi = \psi_1 \wedge \psi_2$. Since L can express φ and all of its subformulas, it can also express ψ_1 , ψ_2 , and all of their subformulas. Then by the inductive hypothesis, L can express $\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi_1)$ and $\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi_2)$. Then by closure under conjunctions, L can express $\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi_1) \wedge \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi_2)$, which is the same as $\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi)$ (cf. Definition 4.3).

Now suppose that $\varphi(\bar{x}, \bar{y}) = \exists z \psi(\bar{x}, \bar{y}, z)$, where the (possibly empty) tuple \bar{x} consists of all free variables of φ not in the tuple \bar{y} . We need to show that L can express $\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi(\bar{x}, \bar{y}))$, which is the same as $\exists z (\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi(\bar{x}, \bar{y}, z)))$ (cf. Definition 4.3). Since L can express φ and all of its subformulas, it can also

express ψ and all of its subformulas. Then, by the inductive hypothesis, L can express $\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)$, whose free variables are those in the tuple $\bar{x}z$, as well as $\text{BIND}_{\bar{x}\bar{y} \rightarrow \overline{QP}}(\psi)$, whose only free variable is z . Since L is closed under conjunction and guarded quantification, it follows that L can express

$$\gamma(\bar{x}) := \exists z(G(\bar{x}, z) \wedge \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)) \quad \text{and} \quad \exists z(z = z \wedge \text{BIND}_{\bar{x}\bar{y} \rightarrow \overline{QP}}(\psi)),$$

where G is a fresh relation symbol not occurring in ψ . Then by closure under unary implications, we have that L can also express

$$\chi(\bar{x}) := \left(\bigwedge_i Q_i(x_i) \right) \rightarrow \exists z(z = z \wedge \text{BIND}_{\bar{x}\bar{y} \rightarrow \overline{QP}}(\psi)).$$

Claim: $\gamma(\bar{x}) \models \chi(\bar{x})$

Proof of claim: By Proposition 4.2,

$$\text{BIND}_{\bar{x}\bar{y} \rightarrow \overline{QP}}(\psi) \equiv \text{BIND}_{\bar{x} \rightarrow \overline{Q}}(\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)). \quad (1)$$

Then by applying Proposition 4.3 and inverting the hypotheses, we have

$$\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi) \models \left(\bigwedge_i Q_i(x_i) \right) \rightarrow \text{BIND}_{\bar{x}\bar{y} \rightarrow \overline{QP}}(\psi).$$

From this, it follows (because z is distinct from x_i variables) that

$$\exists z(\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)) \models \left(\bigwedge_i Q_i(x_i) \right) \rightarrow \exists z \text{BIND}_{\bar{x}\bar{y} \rightarrow \overline{QP}}(\psi),$$

and therefore $\gamma(\bar{x}) \models \chi(\bar{x})$. This concludes the proof of the claim.

Since L can express both $\gamma(\bar{x})$ and $\chi(\bar{x})$, we have by the Craig interpolation property that L can express some Craig interpolant $\vartheta(\bar{x})$. Since G and the Q_i predicates do not occur in φ , they do not occur in $\vartheta(\bar{x})$, and therefore, the following second-order implication is valid:

$$\exists G \gamma(\bar{x}) \models \vartheta(\bar{x}) \models \forall \overline{Q} \chi(\bar{x}).$$

It is easy to see that $\exists G \gamma(\bar{x}) \equiv \exists z \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)$. Similarly, it follows from Proposition 4.4 and equation (1) that $\forall \overline{Q} \chi(\bar{x}) \equiv \exists z \text{BIND}_{\bar{x}\bar{y} \rightarrow \overline{QP}}(\psi)$. Therefore, $\vartheta(\bar{x}) \equiv \exists z \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)$. In particular, $\exists z \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)$ is expressible in L . \square

We are now ready to prove Proposition 4.1, restated below.

Proposition 4.1. *Let L be a logic with CIP that can express atomic formulas, guarded quantification, conjunction, and unary implication. Then $\text{FO}_{\exists, \wedge} \preceq L$.*

Proof. By strong induction on the complexity of $\text{FO}_{\exists, \wedge}$ -formulas. The base case is immediate, since L can express all atomic formulas. For the inductive step, if $\varphi := \psi_1 \wedge \psi_2$, then by the inductive hypothesis, L can express ψ_1 and ψ_2 , and so

by closure under conjunction, L can express φ . Now suppose $\varphi(\bar{y}) := \exists x(\psi(x, \bar{y}))$. By the inductive hypothesis, together with closure under guarded quantification, L can express

$$\gamma(\bar{y}) := \exists x(G(x, \bar{y}) \wedge \psi).$$

Furthermore, by Lemma 4.1, L can express $\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)$, and therefore, by closure under guarded quantification and unary implications, L can express

$$\chi(\bar{y}) := \left(\bigwedge_i P_i(y_i) \right) \rightarrow \exists x(x = x \wedge \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)).$$

Claim: $\gamma(\bar{y}) \models \chi(\bar{y})$.

Proof of claim: It is clear that $\gamma(\bar{y}) \models \exists x\psi$. Furthermore, by Proposition 4.3, $\psi \models (\bigwedge_i P_i(y_i)) \mapsto \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)$, from which it follows that $\exists x\psi \models \chi(\bar{y})$ (since the variable x is distinct from y_1, \dots, y_n). Therefore, $\gamma(\bar{y}) \models \chi(\bar{y})$.

Let $\vartheta(\bar{y})$ be an interpolant for $\gamma(\bar{y}) \models \chi(\bar{y})$ in L . Since G and the predicates in \bar{P} do not occur in ψ , the following second-order entailments are valid:

$$\exists G \exists x(G(x, \bar{y}) \wedge \psi) \models \vartheta(\bar{y}) \models \forall \bar{P} \left(\left(\bigwedge_i P_i(y_i) \right) \rightarrow \exists x \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi) \right).$$

It is easy to see that

$$\exists G \exists x(G(x, \bar{y}) \wedge \psi) \equiv \exists x\psi.$$

Furthermore, by Proposition 4.4,

$$\psi \equiv \forall \bar{P} \left(\left(\bigwedge_i P_i(y_i) \right) \rightarrow \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi) \right).$$

from which it follows (since x is distinct from y_1, \dots, y_n) that

$$\exists x\psi \equiv \forall \bar{P} \left(\left(\bigwedge_i P_i(y_i) \right) \rightarrow \exists x \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi) \right).$$

Therefore, $\vartheta(\bar{y}) \equiv \varphi(\bar{y})$, and so we are done. \square

Our main result follows easily from Proposition 4.1, the closure properties of guarded logics, and the UCQ characterization of GNFO.

Theorem 4.1. *Let L be a guarded logic with CIP. Then $\text{GNFO} \preceq L$.*

Proof. L can express self-guarded GFO-formulas, so it can express formulas of the form $\exists \bar{x}\beta$, where β is an atomic formula. Then since L can express self-guarded substitution, L can express guarded quantification. Furthermore, L can express all self-guarded formulas of the form $\alpha \wedge \neg\beta$, where α and β are atomic formulas such that $\text{free}(\alpha) = \text{free}(\beta)$. Furthermore, for every formula φ expressible in L with $\text{free}(\varphi) \subseteq \text{free}(\alpha)$, $\alpha \wedge \varphi$ is a self-guarded formula. Thus by expressibility of self-guarded substitution, L can also express $\alpha \wedge \neg(\alpha \wedge \varphi)$, which

is equivalent to $\alpha \wedge \neg\varphi$; hence L can express guarded negation. If L can express φ , then by expressibility of guarded negation and disjunction, it can also express the formula $(x = x \wedge \neg P(x)) \vee \varphi$, which is equivalent to $P(x) \rightarrow \varphi$. Hence L can express unary implications. Therefore, by Proposition 4.1, L can express all formulas in $\text{FO}_{\exists, \wedge}$. Then by expressibility of disjunction, L can express all unions of conjunctive queries. The result then follows immediately from the UCQ-syntax for GNFO, by closure under self-guarded substitution. \square

5 Repairing Interpolation for FF

The fluted fragment (FL) [50] is an ordered logic, in which all occurrences of variables in atomic formulas and quantifiers must follow a fixed order. In the context of ordered logics, we assume a fixed infinite sequence of variables $X = \langle x_i \rangle_{i \in \mathbb{Z}^+}$. A *suffix n -atom* is an atomic formula of the form $R(x_j, \dots, x_n)$, where x_j, \dots, x_n is a finite contiguous subsequence of X . FL is defined by the following recursion.

Definition 5.1. For each $n \in \mathbb{N}$, define collections of formulas FL^n as follows:

1. FL^n contains all suffix n -atoms,
2. FL^n is closed under Boolean combinations, and
3. If φ is in FL^{n+1} , then $\exists x_{n+1}\varphi$ and $\forall x_{n+1}\varphi$ are in FL^n .

We set $\text{FL} = \bigcup_{n \in \mathbb{N}} \text{FL}^n$.

The forward fragment (FF), introduced in [6], is a syntactic generalization of FL. We say that $R(x_j, \dots, x_k)$ is an *infix n -atom* if x_j, \dots, x_n is a finite contiguous subsequence of X and $k \leq n$. FF is defined by the following recursion.

Definition 5.2. For each $n \in \mathbb{N}$, define collections of formulas FF^n as follows:

1. FF^n contains all infix n -atoms,
2. FF^n is closed under Boolean combinations, and
3. If φ is in FF^{n+1} , then $\exists x_{n+1}\varphi$ and $\forall x_{n+1}\varphi$ are in FF^n .

We set $\text{FF} = \bigcup_{n \in \mathbb{N}} \text{FF}^n$.

In contrast to the other logics we have seen, FL and FF do not allow the primitive equality symbol. It can be seen by a simple formula induction that every formula in FF^k can be expressed by a formula in FF^n for every $n > k$; it follows easily that FF can express arbitrary Boolean combinations of its formulas. However, FL cannot: $P(x_1)$ and $P(x_2)$ are in FL, but $P(x_1) \wedge P(x_2)$ is not expressible in FL. Although FF contains formulas which are not in FL, it is known that FF and FL are expressively equivalent at the level of sentences [7]. Furthermore, the satisfiability problems for FL and FF are decidable [47,7].

5.1 Natural Extensions of FF

Given a formula φ , we write $gfv(\varphi)$ to denote the greatest $n \in \mathbb{Z}^+$ such that x_n occurs free in φ ; if φ is a sentence, then we set $gfv(\varphi) = 0$. We define *forward logics* to capture the notion of a natural extension of FF.

Definition 5.3. A forward logic is an abstract logic L such that

1. L can express all infix n -atoms for every $n \in \mathbb{Z}^+$,
2. L can express all Boolean combinations of its formulas, and
3. L can express $\exists x_n \varphi$ and $\forall x_n \varphi$ whenever L can express φ and $n = gfv(\varphi)$.

We refer to the last property of a forward logic as *expressibility of ordered quantification*. Clearly FF is a forward logic, and every forward logic extends FF.

5.2 Finding the Minimal Extension of FF with CIP

Unlike the other fragments we have seen, one peculiar property of FF is that the logic is not closed under variable substitutions. This can be seen simply by considering relational atoms: for a 3-ary relational symbol R , the formula $R(x_1, x_2, x_3)$ is in FF, but the formula $R(x_3, x_1, x_2)$ is not. Before proving our main theorem, we prove the following lemma asserting that whenever a formula is expressible in a forward logic L satisfying CIP, the result of making arbitrary substitutions for the free variables of the formula is also expressible in L .

Lemma 5.1. Let L be a forward logic satisfying CIP, and let $\varphi(x_{i_1}, \dots, x_{i_k})$ be a formula of first-order logic expressible in L , where x_{i_1}, \dots, x_{i_k} is not necessarily a contiguous subsequence of variables. Then for every map

$$\pi : \{i_1, \dots, i_k\} \rightarrow \mathbb{Z}^+,$$

we have that L can also express $\varphi(x_{\pi(i_1)}, \dots, x_{\pi(i_k)})$. In other words, L is closed under renamings of free variables.

Proof. For brevity, let $\bar{x} = x_{i_1}, \dots, x_{i_k}$, and let $\pi(\bar{x}) = x_{\pi(i_1)}, \dots, x_{\pi(i_k)}$. Without loss of generality, assume that $i_1 \leq \dots \leq i_k$ (we can do this since the notation $\varphi(x_{i_1}, \dots, x_{i_k})$ only indicates that the variables occur free, but says nothing about where or in what order they occur in the formula). Since L can express $\varphi(\bar{x})$, it can evidently express the following formulas, by the definition of a forward logic:

$$\begin{aligned} \gamma(\bar{x}) &:= \bigwedge_{m \leq k} G_m(x_{i_m}) \wedge \forall x_{i_1} \dots \forall x_{i_k} \left(\bigwedge_{m \leq k} G_m(x_{\pi(i_m)}) \rightarrow \varphi(\bar{x}) \right) \\ \chi(\bar{x}) &:= \bigwedge_{m \leq k} P_m(x_{i_m}) \rightarrow \exists x_{i_1} \dots \exists x_{i_k} \left(\varphi(\bar{x}) \wedge \bigwedge_{m \leq k} P_m(x_{\pi(i_m)}) \right) \end{aligned}$$

Clearly $\gamma \models \chi$, and so there exists an interpolant ϑ . Hence

$$\exists G_1 \dots G_k \gamma \models \vartheta \models \forall P_1 \dots P_k \chi$$

is a valid second-order entailment. Furthermore, it is easy to see that

$$\exists G_1 \dots G_k \gamma \equiv \forall P_1 \dots P_k \chi \equiv \varphi.$$

Therefore, $\varphi(x_{\pi(i_1)}, \dots, x_{\pi(i_k)})$ is expressible in L . □

We now prove our main theorem, which follows easily from Lemma 5.1.

Theorem 5.1. *Let L be a forward logic satisfying CIP. Then $\text{FO} \preceq L$.*

Proof. We proceed by formula induction on FO-formulas φ . For the base case, clearly L can express all atomic FO-formulas by applying Lemma 5.1 to an appropriate infix atom. For the inductive step, the Boolean cases are immediate since L can express all Boolean combinations. Hence the only interesting case is when $\varphi := \exists x_k \psi$ for some formula ψ . By the inductive hypothesis, L can express ψ . Applying Lemma 5.1, L can also express φ' , the result of substituting x_{n+1} for all free occurrences of x_k , where $n = \text{gfv}(\varphi)$, and leaving all other free variables the same. Then by expressibility of ordered quantification, L can express $\exists x_{n+1} \varphi'$, which is equivalent to φ . □

6 Undecidability of Extensions of FO^2 and FL with CIP

In Section 3, we showed that every strong extension of FO^2 with CIP can express all sentences of FO, and in Section 5, we showed that every forward logic with CIP can express all formulas of FO. These results suggest the undecidability of the satisfiability problems for such logics. In this section, we formalize this idea, showing that extensions of FO^2 and FL with CIP and satisfying very limited expressive assumptions are undecidable. These results rely primarily on known results on the undecidability of FO^2 and FL with additional transitive relations.

Proposition 6.1. *Every abstract logic L with CIP extending FO^2 or FL can express the following formulas:*

$$\begin{aligned} \psi_0(x_1) &:= \forall x_2 \forall x_3 (R(x_1, x_2) \wedge R(x_2, x_3) \rightarrow R(x_1, x_3)), \text{ and} \\ \psi_1 &:= \neg \forall x_1 \forall x_2 \forall x_3 (R(x_1, x_2) \wedge R(x_2, x_3) \rightarrow R(x_1, x_3)). \end{aligned}$$

The proof of Proposition 6.1 can be found in the appendix. We also need two additional definitions. First, an *effective translation* from a logic L to a logic L' is a computable function which takes formula of $\varphi \in L$ as input and outputs an equivalent formula $\varphi' \in L'$. Second, we say that a logic L has *effective conjunction* if there is a computable function taking formulas $\varphi, \psi \in L$ as input and outputting a formula $\chi \in L$ which is equivalent to $\varphi \wedge \psi$.

Theorem 6.1. *Let L be an extension of FL which satisfies CIP. Suppose further that there is an effective translation from FL to L , and L has effective conjunction. The satisfiability problem for L is undecidable if either*

1. *L can express ordered quantification, or*
2. *L can express negation.*

Proof. Let χ be the sentence asserting the transitivity of the relation R . Since L has CIP and extends FL, it can express both $\psi_0(x_1)$ and ψ_1 by Proposition 6.1. If L can express ordered quantification, it can express $\forall x_1 \psi_0(x_1)$, which is equivalent to χ . If L can express negation, then it can express $\neg \psi_1$, which is also equivalent to χ . Since L , as an abstract logic, can express χ and is closed under predicate renamings, it can express that any number of binary relations are transitive. Let χ_1 , χ_2 , and χ_3 be sentences expressing transitivity of binary relation symbols R_1 , R_2 , and R_3 , respectively. Let tr be an effective translation from FL to L . Then a formula φ of FL with three designated transitive relations is satisfiable if and only if $tr(\varphi) \wedge \chi_1 \wedge \chi_2 \wedge \chi_3$ is satisfiable. Since tr is computable and L is effectively closed under conjunction, this reduction is computable. Since the satisfiability problem for FL with three transitive relations is undecidable [45], the satisfiability problem for L is undecidable. \square

It is also known that satisfiability is undecidable for FO^2 -formulas with two transitive relations [35]. Using this fact, along with Proposition 6.1, we obtain the following theorem, by a similar proof to that of Theorem 6.1.

Theorem 6.2. *Let L be an extension of FO^2 which satisfies CIP. Suppose further that there is an effective translation from FO^2 to L , and L has effective conjunction. The satisfiability problem for L is undecidable if either*

1. *L can express universal quantification, or*
2. *L can express negation.*

We remark that all forward logics and strong extensions of FO^2 with CIP, assuming appropriate effective translations and effective conjunction, meet the requirements of Theorems 6.1 and 6.2, and hence are undecidable.

7 Discussion

In the introduction, we mentioned several results indicating the failure of CIP among many natural proper extension of FO. In [14], van Benthem points out that there is a similar scarcity among FO-fragments as well. Our results in Sections 3 and 5 may be interpreted as additional confirmation of this observation. Furthermore, one tends to study proper fragments of FO for their desirable computational properties, and so our broader undecidability results show that CIP fails for large swaths of *decidable* FO-fragments. However, there are a few notable fragments for which the determination of a minimal extension satisfying CIP is still open, such as FL and the quantifier prefix fragments.

One limitation of our methodology and results is their dependence on a definition of Craig interpolation which mandates the existence of interpolants between proper *formulas*, while many practical applications only require CIP for *sentences*. Throughout this paper, we have established expressibility of a formula ϑ in a logic L by induction (and by constructing two formulas φ and ψ such that $\varphi \models \psi$ and arguing that every interpolant is equivalent to ϑ). In general, this method is difficult to apply unless free variables are allowed; it is not clear how to apply this type of inductive argument if we were only concerned with the existence of interpolants for sentences of the logic.

There are several well-studied properties strictly weaker than CIP. The Δ -interpolation property (also known as Suslin-Kleene interpolation) holds for a logic L if, whenever $\varphi \models \psi$, and (intuitively speaking) there is only one possible interpolant ϑ up to logical equivalence for this entailment, then L contains a formula equivalent to ϑ [5]. It is not hard to see that, unlike the Craig interpolation property, every logic L has a unique extension, denoted $\Delta(L)$, satisfying the Δ -interpolation property. In fact, in our proofs we only rely on Δ -interpolation; every application of the assumption that some abstract logic L satisfies CIP yields a provably unique interpolant, up to logical equivalence. Therefore, all of our results hold also when CIP is replaced by Δ -interpolation.

Two additional weakenings of CIP are the projective and non-projective Beth definability properties. The projective Beth property states, roughly, that whenever a $\sigma \cup \tau \cup \{R\}$ -theory Σ implicitly defines a relation R in terms of the relations in σ , then Σ entails an explicit definition of R in terms of σ (the non-projective Beth property being the special case for $\tau = \emptyset$). Many practical applications of CIP in database theory and knowledge representation require only the projective Beth property. It is not immediately clear how to extend our methodology to a systematic study of the (projective) Beth property among decidable FO-fragments. Indeed, GFO already satisfies the non-projective Beth property [32]. Given their applications, an interesting avenue of future work is to map the landscape of FO-fragments satisfying these properties. In the other direction, minimal extensions of logics with *uniform* interpolation (a strengthening of CIP) were studied in [24], although with limited results so far (cf. [24, Thm. 14]). Some of the minimal extensions of PLTL fragments with CIP identified in [27], however, do satisfy uniform interpolation.

Acknowledgements. We thank Jean Jung, Frank Wolter, and Malvin Gattlinger for feedback on an earlier draft, and we thank Ian Pratt-Hartmann and Michael Benedikt for helpful remarks during a related workshop presentation. Balder ten Cate is supported by EU Horizon 2020 grant MSCA-101031081.

References

1. Andr eka, H., N emeti, I., van Benthem, J.: Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic* **27** (06 1998). <https://doi.org/10.1023/A:1004275029985>

2. Areces, C., Blackburn, P., Marx, M.: Repairing the interpolation theorem in quantified modal logic. *Annals of Pure and Applied Logic* **124**(1), 287–299 (2003). [https://doi.org/10.1016/S0168-0072\(03\)00059-9](https://doi.org/10.1016/S0168-0072(03)00059-9)
3. Bárány, V., Benedikt, M., ten Cate, B.: Rewriting guarded negation queries. In: *Proceedings of MFCS 2013*. pp. 98–110. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
4. Barany, V., ten Cate, B., Segoufin, L.: Guarded negation. *Journal of the ACM* **62**(3), 22.1–22:26 (2015)
5. Barwise, J., Feferman, S. (eds.): *Model-Theoretic Logics, Perspectives in Logic*, vol. 8. Cambridge University Press (2017)
6. Bednarczyk, B.: Exploiting forwardness: Satisfiability and query-entailment in forward guarded fragment. In: *Logics in Artificial Intelligence: 17th European Conference, JELIA 2021, Virtual Event, May 17–20, 2021, Proceedings*. p. 179–193. Springer-Verlag, Berlin, Heidelberg (2021). https://doi.org/10.1007/978-3-030-75775-5_13
7. Bednarczyk, B., Jaakkola, R.: Towards a Model Theory of Ordered Logics: Expressivity and Interpolation. In: Szeider, S., Ganian, R., Silva, A. (eds.) *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*. *Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 241, pp. 15:1–15:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022). <https://doi.org/10.4230/LIPIcs.MFCS.2022.15>
8. Benedikt, M., Bourhis, P., Boom, M.V.: Definability and Interpolation within Decidable Fixpoint Logics. *Logical Methods in Computer Science* **Volume 15, Issue 3** (Sep 2019). [https://doi.org/10.23638/LMCS-15\(3:29\)2019](https://doi.org/10.23638/LMCS-15(3:29)2019)
9. Benedikt, M., ten Cate, B., Boom, M.V.: Interpolation with decidable fixpoint logics. In: *LICS*. pp. 378–389 (2015). <https://doi.org/10.1109/LICS.2015.43>
10. Benedikt, M., ten Cate, B., Tsamoura, E.: Generating plans from proofs. *ACM Trans. Database Syst.* **40**(4), 22:1–22:45 (2016). <https://doi.org/10.1145/2847523>
11. Benedikt, M., Cate, B.ten., Boom, M.V.: Effective interpolation and preservation in guarded logics. *ACM Trans. Comput. Logic* **17**(2) (2015). <https://doi.org/10.1145/2814570>
12. Benedikt, M., Leblay, J., ten Cate, B., Tsamoura, E.: Generating plans from proofs : the interpolation-based approach to query reformulation. *Synthesis Lectures on Data Management*, Morgan & Claypool (2016)
13. van Benthem, J.: A new modal lindström theorem. *Logica Universalis* **1**(1), 125–138 (2007). <https://doi.org/10.1007/s11787-006-0006-3>
14. van Benthem, J.: The many faces of interpolation. *Synthese* **164**(3), 451–460 (2008), <http://www.jstor.org/stable/40271083>
15. van Benthem, J., ten Cate, B., Väänänen, J.A.: Lindström theorems for fragments of first-order logic. *Log. Methods Comput. Sci.* **5**(3) (2009), <http://arxiv.org/abs/0905.3668>
16. Börger, E., Grädel, E., Gurevich, Y.: *The Classical Decision Problem*. *Perspectives in Mathematical Logic*, Springer (1997)
17. Caicedo, X.: Failure of interpolation for quantifiers of monadic type. In: Di Prisco, C.A. (ed.) *Methods in Mathematical Logic*. pp. 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg (1985)
18. Calvanese, D., Ghilardi, S., Gianola, A., Montali, M., Rivkin, A.: Combined covers and beth definability. In: *Proceedings of the 10th International Joint Conference on Automated Reasoning, Part I, IJCAR 2020*. pp. 181–200. Springer (2020). https://doi.org/10.1007/978-3-030-51074-9_11

19. ten Cate, B.: Interpolation for extended modal languages. *The Journal of Symbolic Logic* **70**(1), 223–234 (2005), <http://www.jstor.org/stable/27588355>
20. ten Cate, B., Franconi, E., Seylan, I.: Beth definability in expressive description logics. *J. Artif. Int. Res.* **48**(1), 347–414 (oct 2013)
21. ten Cate, B., Segoufin, L.: Unary negation. *Logical Methods in Computer Science* **Volume 9, Issue 3** (Sep 2013). [https://doi.org/10.2168/LMCS-9\(3:25\)2013](https://doi.org/10.2168/LMCS-9(3:25)2013)
22. Comer, S.D.: Classes without the amalgamation property. *Pacific Journal of Mathematics* **28**, 309–318 (1969)
23. Craig, W.: Three uses of the herbrand-gentzen theorem in relating model theory and proof theory. *Journal of Symbolic Logic* **22**(3), 269–285 (1957). <https://doi.org/10.2307/2963594>
24. D’Agostino, G., Lenzi, G., French, T.: μ -programs, uniform interpolation and bisimulation quantifiers for modal logics. *Journal of Applied Non-Classical Logics* **16**(3-4), 297–309 (2006). <https://doi.org/10.3166/janc1.16.297-309>
25. Friedman, H.: Beth’s theorem in cardinality logics. *Israel Journal of Mathematics* **14**(2), 205–212 (1973)
26. Garcá-Matos, M.: Abstract model theory without negation. Ph.D. thesis, University of Helsinki (2005)
27. Gheerbrant, A., ten Cate, B.: Craig interpolation for linear temporal languages. In: Grädel, E., Kahle, R. (eds.) *Computer Science Logic*. pp. 287–301. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
28. Graedel, E., Otto, M., Rosen, E.: Two-variable logic with counting is decidable. In: *Proceedings of LICS 1997*. p. 306 (1997)
29. Grädel, E.: On the restraining power of guards. *The Journal of Symbolic Logic* **64**(4), 1719–1742 (1999), <http://www.jstor.org/stable/2586808>
30. Hoder, K., Holzer, A., Kovács, L., Voronkov, A.: Vinter: A Vampire-based tool for interpolation. In: Jhala, R., Igarashi, A. (eds.) *Programming Languages and Systems - 10th Asian Symposium, APLAS 2012, Kyoto, Japan, December 11-13, 2012*. *Proceedings. Lecture Notes in Computer Science*, vol. 7705, pp. 148–156. Springer (2012). https://doi.org/10.1007/978-3-642-35182-2_11
31. Hoogland, E.: Definability and interpolation: model-theoretic investigations. Ph.D. thesis, University of Amsterdam (2000)
32. Hoogland, E., Marx, M.: Interpolation and definability in guarded fragments. *Studia Logica* **70**(3), 373–409 (2002), <http://www.jstor.org/stable/20016403>
33. Hustadt, U., Schmidt, R., Georgieva, L.: A survey of decidable first-order fragments and description logics. *Journal on Relational Methods in Computer Science* **1**, 251–276 (01 2004)
34. Jung, J.C., Wolter, F.: Living without beth and craig: Definitions and interpolants in the guarded and two-variable fragments. In: *Proceedings of LICS 2021*. pp. 1–14. IEEE Computer Society (jul 2021). <https://doi.org/10.1109/LICS52264.2021.9470585>
35. Kieroński, E.: Results on the guarded fragment with equivalence or transitive relations. In: *Computer Science Logic. Lecture Notes in Computer Science*, vol. 3634, pp. 309–324. Springer Verlag (2005)
36. Koopmann, P., Schmidt, R.A.: Uniform interpolation and forgetting for \mathcal{ALC} ontologies with ABoxes. In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI 2015*. pp. 175–181. AAAI Press (2015), <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9981>
37. Lindström, P.: On extensions of elementary logic. *Theoria* **35**(1) (1969)

38. Lutz, C., Wolter, F.: Foundations for uniform interpolation and forgetting in expressive description logics. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI 2011. pp. 989–995. IJCAI/AAAI (2011). <https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-170>
39. Löwenheim, L.: Über möglichkeiten im relativkalkül. *Mathematische Annalen* **76**, 447–470 (1915), <http://eudml.org/doc/158703>
40. McMillan, K.L.: Interpolation and model checking. In: Clarke, E.M., Henzinger, T.A., Veith, H., Bloem, R. (eds.) *Handbook of Model Checking*, pp. 421–446. Springer (2018). https://doi.org/10.1007/978-3-319-10575-8_14
41. Mortimer, M.: On languages with two variables. *Math. Log. Q.* **21**, 135–140 (1975)
42. Otto, M.: An interpolation theorem. *The Bulletin of Symbolic Logic* **6**(4), 447–462 (2000), <http://www.jstor.org/stable/420966>
43. Pratt-Hartman, I., Szwast, W., Tendera, L.: The fluted fragment revisited. *The Journal of Symbolic Logic* **84**(3), 1020–1048 (2019). <https://doi.org/10.1017/jsl.2019.33>
44. Pratt-Hartmann, I., Szwast, W., Tendera, L.: Quine’s fluted fragment is non-elementary. In: Regnier, L., Talbot, J. (eds.) *25th EACSL Annual Conference on Computer Science Logic. 25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (Jun 2016). <https://doi.org/10.4230/LIPIcs.CSL.2016.39>
45. Pratt-Hartmann, I., Tendera, L.: The fluted fragment with transitive relations. *Annals of Pure and Applied Logic* **173**(1), 103042 (2022). <https://doi.org/10.1016/j.apal.2021.103042>
46. Purdy, W.C.: Decidability of Fluted Logic with Identity. *Notre Dame Journal of Formal Logic* **37**(1), 84 – 104 (1996). <https://doi.org/10.1305/ndjfl/1040067318>
47. Purdy, W.C.: Fluted formulas and the limits of decidability. *The Journal of Symbolic Logic* **61**(2), 608–620 (1996). <https://doi.org/10.2307/2275678>
48. Purdy, W.C.: Quine’s ‘limits of decision’. *The Journal of Symbolic Logic* **64**(4), 1439–1466 (1999). <https://doi.org/10.2307/2586789>
49. Purdy, W.C.: Complexity and nicety of fluted logic. *Studia Logica* **71**, 177–198 (2002)
50. Quine, W.V.: On the limits of decision. *14th International Congress for Philosophy* **3**, 57–62 (1969)
51. Skolem, T.: Logisch-Kombinatorische Untersuchungen über die Erfüllbarkeit oder Bewiesbarkeit mathematischer Sätze nebst einem Theorem über dichte Mengen. I. *Matematisk-naturvidenskabelig Klasse 4*, 1-36, Videnskapsselskapet Skrifter (1920)
52. Toman, D., Weddell, G.E.: *Fundamentals of Physical Design and Query Compilation. Synthesis Lectures on Data Management*, Morgan & Claypool Publishers (2011)
53. Väänänen, J.: The craig interpolation theorem in abstract model theory. *Synthese* **164**(3), 401–420 (2008)
54. Vardi, M.Y.: Why is modal logic so robustly decidable? In: Immerman, N., Kolaitis, P.G. (eds.) *Descriptive Complexity and Finite Models. DIMACS*, vol. 31, pp. 149–183 (1996). <https://doi.org/10.1090/dimacs/031/05>

8 Appendix

Proof of Proposition 4.3

Proposition 4.3. *For every clean $\text{FO}_{\exists, \wedge}$ -formula φ , for every tuple of distinct variables $\bar{y} = y_1, \dots, y_n$ (with each $y_i \in \text{free}(\varphi)$), and for every tuple of unary predicates $\bar{P} = P_1, \dots, P_n$, we have that*

$$\left(\bigwedge_{i=1, \dots, n} P_i(y_i) \right) \models \varphi \rightarrow \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi).$$

Proof. The proof is by induction on φ . For the base case, suppose that φ is an atomic formula. Then if $M, g \models \bigwedge_{i=1, \dots, n} P_i(y_i)$ and $M, g \models \varphi$, then it is immediate from the definition of BIND that $M, g \models \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi)$. For the inductive step, suppose that, for every model M and variable assignment g , if $M, g \models \bigwedge_{i=1, \dots, n} P_i(y_i)$ and $M, g \models \varphi$ then $M, g \models \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi)$. The case of conjunction is trivial by the inductive hypothesis and the fact that BIND commutes with conjunction. Finally, suppose that $\varphi := \exists z \psi$. By the inductive hypothesis, we have that

$$\left(\bigwedge_{i=1, \dots, n} P_i(y_i) \right) \models \psi \rightarrow \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi).$$

Suppose that $M, g \models \bigwedge_{i=1, \dots, n} P_i(y_i)$ and $M, g \models \exists z \psi$. Then there exists some $a \in M$ such that $M, g[z/a] \models \psi$, and so by the inductive assumption, $M, g[z/a] \models \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)$. Then because φ is clean, we have that $z \notin \{y_1, \dots, y_n\}$, and so $M, g \models \exists z \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\psi)$.

Proof of Proposition 4.4

Proposition 4.4. *For every clean $\text{FO}_{\exists, \wedge}$ -formula $\varphi(x, \bar{y})$ with $\bar{y} = y_1, \dots, y_n$ distinct from x , and for every n -tuple of unary predicates $\bar{P} = P_1, \dots, P_n$ not occurring in φ , we have that*

$$\exists x \varphi(x, \bar{y}) \equiv \forall \bar{P} \left(\left(\bigwedge_{i=1, \dots, n} P_i(y_i) \right) \rightarrow \exists x \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi(x, \bar{y})) \right).$$

Proof. For the left-to-right direction, suppose that $M, g \models \exists x \varphi(x, \bar{y})$ and M' is an expansion of M such that $M' \models \bigwedge_{i=1, \dots, n} P_i(y_i)$. Then $M', g[x/b] \models \varphi(x, \bar{y}) \wedge \bigwedge_{i=1, \dots, n} P_i(y_i)$ for some $b \in M$. Then, by Proposition 4.3, we have

$$M', g[x/b] \models \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi(x, \bar{y})),$$

and hence $M', g \models \exists x \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi(x, \bar{y}))$.

For the reverse direction, suppose that

$$M, g \models \forall \bar{P} \left(\bigwedge_i P_i(y_i) \rightarrow \exists x \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi(x, \bar{y})) \right).$$

Let M' be the expansion of the structure M in which each unary predicate symbol P_i is interpreted as $\{g(y_i)\}$. Then, by the semantics of second-order quantifiers, we have that $M', g \models \exists x \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi(x, \bar{y}))$, and hence $M', g[x/b] \models \text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi(x, \bar{y}))$ for some $b \in M$. To complete the proof, it suffices to show that $M', g[x/b] \models \varphi(x, \bar{y})$ (since this implies that also $M, g[x/b] \models \varphi(x, \bar{y})$).

For every subformula containing a bound occurrence of a variable $y_i \in \bar{y}$, we have that every witness for that variable y_i must also be in P_i (by construction of $\text{BIND}_{\bar{y} \rightarrow \bar{P}}(\varphi(x, \bar{y}))$ and the assumption that $\varphi(x, \bar{y})$ is clean). Since each P_i is a singleton, this implies that each witness for y_i in each subformula is $g(y_i)$. It follows that $M, g[\bar{y}'/\bar{a}'] \models \alpha$ for each atomic formula α occurring in $\varphi(x, \bar{y})$, where \bar{y}' is the tuple of variables of \bar{y} occurring in α . By a simple subformula induction, we then obtain that $M \models \varphi(b, \bar{a})$, completing the proof. \square

Proof of Proposition 6.1

Proposition 6.1. *Every abstract logic L with CIP extending FO^2 or FL can express the following formulas:*

$$\begin{aligned} \psi_0(x_1) &:= \forall x_2 \forall x_3 (R(x_1, x_2) \wedge R(x_2, x_3) \rightarrow R(x_1, x_3)), \text{ and} \\ \psi_1 &:= \neg \forall x_1 \forall x_2 \forall x_3 (R(x_1, x_2) \wedge R(x_2, x_3) \rightarrow R(x_1, x_3)). \end{aligned}$$

Proof. Consider the following formulas of FL:

$$\begin{aligned} \gamma_0(x_1) &:= \forall x_2 (R(x_1, x_2) \rightarrow \\ &\quad \forall x_3 (R(x_2, x_3) \rightarrow G(x_3)) \wedge \forall x_2 (G(x_2) \rightarrow R(x_1, x_2))), \\ \gamma_1(x_1) &:= \exists x_2 (R(x_1, x_2) \wedge \exists x_3 (R(x_2, x_3) \wedge P(x_3))) \rightarrow \exists x_2 (R(x_1, x_2) \wedge P(x_2)). \end{aligned}$$

We also define the following sentences of FL:

$$\begin{aligned} \delta_0 &:= \forall x_1 (G_1(x_1) \rightarrow \forall x_2 (G_2(x_2) \rightarrow \neg R(x_1, x_2))) \\ &\quad \wedge \exists x_1 (G_1(x_1) \wedge \exists x_2 (R(x_1, x_2) \wedge \exists x_3 (R(x_2, x_3) \wedge G_2(x_3))))), \\ \delta_1 &:= \exists x_1 (\forall x_2 (P(x_2) \leftrightarrow \neg R(x_1, x_2)) \rightarrow \\ &\quad \exists x_2 (R(x_1, x_2) \wedge \exists x_3 (R(x_2, x_3) \wedge P(x_3)))). \end{aligned}$$

With simple variable substitutions, we can turn γ_0 and γ_1 into formulas of FO^2 :

$$\begin{aligned} \gamma'_0(x) &:= \forall y (R(x, y) \rightarrow \forall x (R(y, x) \rightarrow G(x)) \wedge \forall y (G(y) \rightarrow R(x, y))), \\ \gamma'_1(x) &:= \exists y (R(x, y) \wedge \exists x (R(y, x) \wedge P(x))) \rightarrow \exists y (R(x, y) \wedge P(y)). \end{aligned}$$

Similarly, we can turn δ_0 and δ_1 into sentences of FO^2 :

$$\begin{aligned} \delta'_0 &:= \forall x (G_1(x) \rightarrow \forall y (G_2(y) \rightarrow \neg R(x, y))) \\ &\quad \wedge \exists x (G_1(x) \wedge \exists y (R(x, y) \wedge \exists x (R(y, x) \wedge G_2(x)))), \\ \delta'_1 &:= \exists x (\forall y (P(y) \leftrightarrow \neg R(x, y)) \rightarrow \\ &\quad \exists y (R(x, y) \wedge \exists x (R(y, x) \wedge P(x)))). \end{aligned}$$

It is a simple exercise in FO-semantics to see that $\gamma_0 \models \gamma_1$ and $\delta_0 \models \delta_1$. Since we assumed that L has CIP, the following claims complete the argument.

Claim. ψ_0 is the unique interpolant of γ_0 and γ_1 .

Proof. Let $\vartheta(x_1)$ be an L -interpolant, so that $\gamma_0 \models \vartheta \models \gamma_1$; then $\exists G\gamma_0 \models \vartheta \models \forall P\gamma_1$ is also a valid second-order entailment. To complete the proof, we need to show that $\exists S\gamma_0 \equiv \psi_0$ and $\forall P\gamma_1 \equiv \psi_0$. To see that $\psi_0 \models \exists S\gamma_0$, observe that if $M, g \models \psi_0$, then $M', g \models \gamma_0$, where M' is the expansion of M in which S is interpreted as the singleton set $\{g(x_0)\}$. If $M, g \models \forall P\gamma_1$, then $M', g \models \gamma_1$, where M' is the expansion of M in which S is interpreted as the singleton set $\{g(x_0)\}$; hence $M', g \models \psi_0$, and so $M, g \models \psi_0$. Thus we have that $\psi_0 \models \exists S\gamma_0 \models \vartheta \models \forall P\gamma_1 \models \psi_0$; in particular, we have that $\vartheta \equiv \psi_0$, which completes the proof.

Claim. ψ_1 is the unique interpolant of δ_0 and δ_1 .

Proof. Let $\vartheta(x_1)$ be an L -interpolant, so that $\delta_0 \models \vartheta \models \delta_1$; then $\exists G_1 G_2 \delta_0 \models \vartheta \models \forall P\delta_1$ is also a valid second-order entailment. To complete the proof, we need to show that $\exists G_1 \exists G_2 \delta_0 \equiv \psi_1$ and $\forall P\delta_1 \equiv \psi_0$. To see that $\psi_1 \models \exists G_1 \exists G_2 \delta_0$, observe that if that $M, g \models \psi_1$, then $M', g \models \delta_0$, where M' is the expansion of M in which G_1 and G_2 are interpreted as singleton sets containing the witnesses for the first-order existential quantifiers in ψ_1 . If $M, g \models \forall P\delta_1$, then $M', g \models \delta_1$, where M' is the expansion of M in which P is interpreted as the set of elements in $a \in \text{dom}(M)$ such that $R^M(g(x_1), a)$ does not hold. It then follows easily that $M \models \psi_1$.