Which classes of structures are both pseudo-elementary and definable by an infinitary sentence?

Will Boney, Barbara F. Csima*, Nancy A. Day, Matthew Harrison-Trainor†

May 6, 2021

Abstract

When classes of structures are not first-order definable, we might still try to find a nice description. There are two common ways for doing this. One is to expand the language, leading to notions of pseudo-elementary classes, and the other is to allow infinite conjuncts and disjuncts. In this paper we examine the intersection. Namely, we address the question: Which classes of structures are both pseudo-elementary and $\mathcal{L}_{\omega_1,\omega}$-elementary? We find that these are exactly the classes that can be defined by an infinitary formula that has no infinitary disjunctions.

1 Introduction

It is well-known that many properties of structures are not expressible in elementary first-order logic, even by a theory rather than a single sentence. Common examples are the property (of graphs) of being connected, the property (of abelian groups) of being torsion, and the property (of linear orders) of being well-founded. To capture such properties, one can pass to extensions of elementary first-order logic. This paper is about a characterization of the common expressive power of two such extensions.

The first extension of elementary first-order logic that we consider allows countably infinite conjunctions and disjunctions; this is, morally, similar to allowing quantifiers over the (standard) natural numbers. One can then express properties such as being torsion by saying "for each group element x, there is an n such that $nx = 0$," or formally,

$$(\forall x) \bigvee_{n\in\mathbb{N}} nx = 0.$$

This work grew out of initial discussions with Vakili about the generality of expressing properties not definable in first-order logic in a pseudo-elementary way, and whether such phenomena might be of use for model checking (as the pseudo-elementary definability of graph reachability was used for model checking by Vakili in his thesis [Vak16] and with the third author in [VD14]). We thank one of the referees for pointing us towards some very helpful references.

*Partially supported by Canadian NSERC Discovery Grant 312501.

†Supported by an NSERC Banting Fellowship.

This infinitary logic is known as $\mathcal{L}_{\omega_1,\omega}$. One loses compactness, but gains other powerful tools. For example, every countable structure is characterized, up to isomorphism among countable structures, by a sentence of $\mathcal{L}_{\omega_1,\omega}$ [Sco65].

The second extension of elementary first-order logic allows existential second-order quantifiers. For example, the property of a linear order being non-well-founded can be expressed by the sentence "there is a set with no least element." We work with existential second-order quantifiers using the framework of pseudo-elementary classes (and so replace existential second-order quantifiers with expansions of the language). More formally, we say that a class $\mathbb{K}$ of τ-structures is pseudo-elementary (PC_Δ) if there is an expanded language $\tau^* \supseteq \tau$ and a τ^*-theory T such that $\mathbb{K}$ consists exactly of the τ-structures admitting an τ^*-expansion to a model of T. We will describe both of these extensions of first-order logic in more detail later.

These two extensions of elementary first-order logic have different expressive powers. For example, the class of non-well-founded linear orders is pseudo-elementary but not $\mathcal{L}_{\omega_1,\omega}$-definable. Also, the compliment of a pseudo-elementary class is not necessarily pseudo-elementary, but the compliment of an $\mathcal{L}_{\omega_1,\omega}$-definable class is again $\mathcal{L}_{\omega_1,\omega}$-definable (by the negation of the original defining sentence). Nevertheless, there are classes that are not elementary first-order axiomatizable, but that are both pseudo-elementary and $\mathcal{L}_{\omega_1,\omega}$-definable. The class of disconnected graphs is such an example; we provide a more detailed discussion of various examples in Section 2.3. The main result of this paper is a complete classification of such properties.

Theorem 1.1. *Let $\mathbb{K}$ be a class of structures closed under isomorphism. The following are equivalent:*

1. *$\mathbb{K}$ is both a pseudo-elementary (PC_Δ) class and defined by an $\mathcal{L}_{\omega_1,\omega}$-sentence.*
2. *$\mathbb{K}$ is defined by a $\bigwedge\!\!\!\!\bigwedge$-sentence.*

There is some notation in this theorem that we must explain. The $\bigwedge\!\!\!\!\bigwedge$-sentences in the theorem are the $\mathcal{L}_{\omega_1,\omega}$ sentences which (in normal form) involve infinitary conjunctions, but no infinitary disjunctions (see Definition 2.4). For example, the property of being infinite is definable by the $\bigwedge\!\!\!\!\bigwedge$-sentence

$$\bigwedge\!\!\!\!\bigwedge_{n\in\mathbb{N}} \exists x_1,\ldots,x_n\Big(\bigwedge_{i\neq j} x_i \neq x_j\Big).$$

The negation, the property of being finite, is $\mathcal{L}_{\omega_1,\omega}$-definable by the sentence

$$\bigvee\!\!\!\!\bigvee_{n\in\mathbb{N}} \forall x_1,\ldots,x_n\Big(\bigvee_{i\neq j} x_i = x_j\Big)$$

but this sentence is not a $\bigwedge\!\!\!\!\bigwedge$-sentence because it involves an infinitary disjunct. Although $\bigwedge\!\!\!\!\bigwedge$-formulas cannot have infinite disjunctions, they can have *finite* disjunctions.

The proof of (1)⇒(2) uses an argument inspired by the proof of Craig Interpolation for $\mathcal{L}_{\omega_1,\omega}$. This was originally proved by Lopez-Escobar [LE65] who also gave the following corollary: a class which is both pseudo-elementary and co-pseudo-elementary with respect to $\mathcal{L}_{\omega_1,\omega}$ (i.e., both $\mathbf{\Sigma}^1_1$ and $\mathbf{\Pi}^1_1$) is actually $\mathcal{L}_{\omega_1,\omega}$-definable.

In the direction (2)⇒(1), there are several possible proofs. We give the simplest and shortest argument in Section 4. A second proof is to note that any $\bigwedge\!\!\!\!\bigwedge$-sentence is equivalent

to a closed game formula, and classes defined by such formulas are known to be PC_Δ [Bar75, Kol85]. We describe this in Section 5. A third proof, for which we do not give the details, proceeds by coding computable formulas in models of weak arithmetic. This is an approach that was taken by Craig and Vaught [CV58] to prove:

Theorem 1.2 (Craig and Vaught [CV58])**.** *Every computably axiomatizable class in a finite language is a basic pseudo-elementary class* (PC').

By a basic pseudo-elementary class, we mean the class of reducts of a basic elementary class (one defined by a single sentence) in an expanded language. (See Definition 2.9 for the precise definition of PC'.) The latter two proofs of our main Theorem 1.1 give a strengthening of this result of Craig and Vaught:

Theorem 1.3. *Let $\mathbb{K}$ be a class of structures in a finite language that is axiomatized by a computable $\Game$-sentence. Then $\mathbb{K}$ is a basic pseudo-elementary class* (PC').

Unfortunately, we do not know how to reverse Theorem 1.3. We conjecture:

Conjecture 1.4. *A PC' class which is also $\mathcal{L}_{\omega_1,\omega}$-axiomatizable is axiomatizable by a computable $\Game$-sentence.*

The argument in Section 4 for (2)⇒(1) of Theorem 1.1 goes through for $\Game$-sentences of $\mathcal{L}_{\kappa,\omega}$ for any κ. However, we do not know if (1)⇒(2) holds for $\mathcal{L}_{\kappa,\omega}$ for $\kappa > \omega_1$.

Question 1.5. For $\kappa > \omega_1$, is every PC_Δ class defined by an $\mathcal{L}_{\kappa,\omega}$ sentence actually defined by a $\Game$-sentence?

We note that interpolation fails in $\mathcal{L}_{\omega_2,\omega}$ [Mal71, Theorem 4.2]. Intriguingly, Malitz goes on to give a proof system for $\mathcal{L}_{\kappa,\omega}$ that goes through $\mathcal{L}_{(2^{<\kappa})^+,\kappa}$ that gives rise to an interpolation theorem [Mal71, Section 5]. Shelah [She12] uses this to define a logic $\mathcal{L}^1_\kappa$ that is intermediate between $\mathcal{L}_{\kappa,\omega}$ and $\mathcal{L}_{\kappa,\kappa}$ that has interpolation and other nice properties (when $\kappa = \beth_\kappa$). This suggests the right answer to Question 1.5 goes through $\mathcal{L}^1_\kappa$ instead of $\mathcal{L}_{\kappa,\omega}$. However, this logic lacks any syntax in the normal sense (formulas are defined by the existence of winning strategies in a delayed Ehrenfeucht-Fraisse game), which causes additional problems, e.g., it is not clear what a $\Game$-sentence should mean, or what Skolem functions should look like.

2 Notation and Definitions

2.1 Infinitary Logic

For the most part, we follow Marker's new book [Mar16]. Elementary first-order logic has a number of properties which, while useful, make it hard to completely characterize structures. For example, the Ryll-Nardzewski theorem says that any countably categorical structure is relatively simple: for each n, there are only finitely many automorphism orbits of n-types. The infinitary logic $\mathcal{L}_{\omega_1,\omega}$ adds more expressive power and hence allows us to characterize every countable structure up to isomorphism among countable structures [Sco65].

The infinitary logic $\mathcal{L}_{\omega_1,\omega}$ is defined recursively in the same way as finitary first-order logic, except that for $\mathcal{L}_{\omega_1,\omega}$ we can take countable conjunctions and disjunctions. Throughout the paper, let τ be a countable language.

Definition 2.1. The $\mathcal{L}_{\omega_1,\omega}(\tau)$-formulas are defined inductively as follows:

1. every atomic τ-formula is an $\mathcal{L}_{\omega_1,\omega}(\tau)$-formula,
2. if φ is an $\mathcal{L}_{\omega_1,\omega}(\tau)$-formula, then so are $\neg\varphi$, $(\exists x)\varphi$ and $(\forall x)\varphi$,
3. if $(\varphi_i)_{i\in\omega}$ are $\mathcal{L}_{\omega_1,\omega}(\tau)$-formulas with finitely many free variables, then so are $\bigwedge_{i\in\omega}\varphi_i$ and $\bigvee_{i\in\omega}\varphi_i$.

In general, we will drop the reference to τ when it is clear what we mean.

Definition 2.2. An $\mathcal{L}_{\omega_1,\omega}$ formula is in $\mathcal{L}_{\omega_1,\omega}$ *normal form* if the $\neg$ only occurs applied to atomic formulas.

Every $\mathcal{L}_{\omega_1,\omega}$ can be placed into a normal form. The negation $\neg\varphi$ of a sentence φ in normal form is not immediately in normal form itself; this gives rise to the formal negation $\sim\varphi$, which is logically equivalent to $\neg\varphi$ but is in normal form.

Definition 2.3. For any $\mathcal{L}_{\omega_1,\omega}$-formula φ, the formula $\sim\varphi$ is defined inductively as follows:

1. if φ is atomic, $\sim\varphi$ is $\neg\varphi$,
2. $\sim\neg\varphi$ is φ, $\sim(\exists x)\varphi$ is $(\forall x)\sim\varphi$ and $\sim(\forall x)\varphi$ is $(\exists x)\sim\varphi$,
3. $\sim\bigwedge_{i\in\omega}\varphi_i$ is $\bigvee_{i\in\omega}\sim\varphi_i$ and $\sim\bigvee_{i\in\omega}\varphi_i$ is $\bigwedge_{i\in\omega}\sim\varphi_i$.

Definition 2.4. An $\mathcal{L}_{\omega_1,\omega}$-sentence φ is a $\bigwedge$*-formula* if it can be written in normal form without any infinite disjunctions. More concretely, the $\bigwedge$-formulas are formed by the following inductive process:

1. every finitary quantifier-free sentence is a $\bigwedge$-formula,
2. if φ is a $\bigwedge$-formula, then so are $(\exists x)\varphi$ and $(\forall x)\varphi$,
3. if φ and ψ are $\bigwedge$-formulas, then so is $\varphi\vee\psi$,
4. if $(\varphi_i)_{i\in\omega}$ are $\bigwedge$-formulas with finitely many free variables, then so is $\bigwedge_{i\in\omega}\varphi_i$.

Remark 2.5. The third condition allowing one to take the disjunction of finitely many formulas is in some sense unnecessary; any $\bigwedge$-formula is equivalent to one in which all of the disjunctions occur on the inside. For example,

$$\left(\bigwedge_{i\in\omega}\varphi_i\right)\vee\left(\bigwedge_{i\in\omega}\psi_i\right)$$

is equivalent to

$$\bigwedge_{i,j\in\omega}\varphi_i\vee\psi_j.$$

An $\mathcal{L}_{\omega_1,\omega}$ (or $\bigwedge$-) formula is computable if, essentially, there is a computable syntactic representation of the formula (see [AK00]).

2.2 Pseudo-elementary Classes

In this section, we follow the book by Hodges [Hod08]. Many classes of structures can be described by the existence of some feature that can be added to them; for example, a linear ordering is non-well-founded if it has a subset with no least element, and a group is orderable if there exists an ordering. Such classes of structures may not be elementary, but by thinking of them as pseudoelementary classes we can still apply the tools of model theory to them. The main notion of pseudo-elementary class in infinitary model theory is the following:

Definition 2.6. We say that a class $\mathbb{K}$ of $\mathcal{L}$-structures is a *pseudoelementary class* (PC_Δ-class) if there is a language $\tau^* \supseteq \tau$ and an elementary first-order τ^* theory T such that

$$\mathbb{K} = \{\mathcal{M} \mid \text{there is a } \tau^*\text{-structure } \mathcal{M}^* \text{ expanding } \mathcal{M} \text{ with } \mathcal{M}^* \vDash T\}.$$

Pseudoelementary classes have some nice properties such as being closed under ultraproducts. (On the other hand, $\mathcal{L}_{\omega_1,\omega}$-definable classes may not be closed under ultraproducts.)

Just as there is a distinction in model theory between elementary classes and basic elementary classes, the former being axiomatized by a theory and the latter by a single sentence, there is a distinction between pseudoelementary classes and basic pseudoelementary classes.

Definition 2.7. We say that a class $\mathbb{K}$ of $\mathcal{L}$-structures is a *basic pseudoelementary class* (PC-class) if there is a language $\tau^* \supseteq \tau$ and an elementary first-order τ^* sentence φ such that

$$\mathbb{K} = \{\mathcal{M} \mid \text{there is a } \tau^*\text{-structure } \mathcal{M}^* \text{ expanding } \mathcal{M} \text{ with } \mathcal{M}^* \vDash \varphi\}.$$

In finite model theory, it is basic elementary classes that play the more important role, and indeed in finite model theory the term Δ-elementary class is often used for what we call elementary classes, while the term elementary class is reserved for what we call basic elementary classes. Similarly, the main notion of pseudoelementary class in finite model theory is that of basic pseudoelementary classes. Basic pseudo-elementary classes seem to have a connection with computability, e.g., Theorems 1.2 and 1.3.

Some classes seem like they should be pseudo-elementary but do not immediately fit under the above definitions. For example, consider the class of multiplicative groups of fields, i.e., a group G is in this class if there is a field F such that $G = F^\times$. The field F is not going to be a subset of the field G; rather, G will be a subset of F. We can expand our definitions as follows to allow these types of classes, which we call PC' and PC'_Δ. The classes PC' and PC'_Δ differ from PC and PC_Δ respectively in that in addition to expanding the language, one is allowed to add additional elements.

Definition 2.8. Let $\tau \subseteq \tau^*$ be a pair of languages, with a unary predicate $P \in \tau^* \setminus \tau$. Given a τ^*-structure $\mathcal{A}$, we denote by $\mathcal{A}_P$ the substructure of $\mathcal{A} \mid \tau$ whose domain is $P^{\mathcal{A}}$ (if this is a τ-structure; otherwise $\mathcal{A}_P$ is not defined).

Definition 2.9. We say that a class $\mathbb{K}$ of τ-structures is a *basic pseudoelementary class* (PC'-class) if there is a language $\tau^* \supseteq \tau$, with a unary relation $P \in \tau^* \setminus \tau$, and a τ^*-formula ϕ, such that

$$\mathbb{K} = \{\mathcal{A}_P \mid \mathcal{A} \vDash \phi \text{ and } \mathcal{A}_P \text{ is defined}\}.$$

We say that $\mathbb{K}$ is a *pseudoelementary class* (PC'_Δ-class) if ϕ is a first-order theory.

We will always clarify whether a pseudoelementary class is PC_Δ or PC'_Δ, and whether a basic pseudoelementary class is PC or PC′.

Note that in Definition 2.9, if the language is finite (or we are dealing with a PC'_Δ-class), it suffices to ask that

$$\mathbb{K} = \{\mathcal{A}_P \mid \mathcal{A} \vDash \phi\}$$

as ϕ can say that $\mathcal{A}_P$ is defined.

We have defined four different types of pseudo-elementary classes. However, it turns out that PC_Δ and PC'_Δ classes are actually the same; so for example the class of multiplicative groups of fields, which is easily seen to be PC'_Δ, is PC_Δ.

Theorem 2.10 (Theorem 5.2.1 of [Hod08])**.** *Let $\mathbb{K}$ be a class of structures.*

1. *$\mathbb{K}$ is a PC_Δ-class if and only if it is a PC'_Δ-class.*
2. *If all the structures in $\mathbb{K}$ are infinite, then $\mathbb{K}$ is a PC-class if and only if it is a PC′-class.*

In Example 2.15 we exhibit a class which is PC′ but not PC.

The proof of the first point in [Hod08] is not obvious and quite interesting. For the second, essentially the only reason that PC and PC′ are different is that the model might be finite; if a model is infinite, one could just have the elements of the model "wear two hats," on the one hand being the domain of the expansion of the original model, and on the other hand playing the role of the elements of the new sort P.

2.3 Examples

In this section we will give a few examples of classes of various types, separating some of the notions defined in the previous two sections, and including some applications of the theorems of this paper.

The motivating example for this paper was the class of connected graphs. It is easier to think of the compliment, the class of non-connected graphs. This class is both PC and definable by a computable $\bigwedge\!\!\backslash$-sentence. Thus the class of connected graphs is both co-PC and definable by a computable $\bigvee\!\!\!\bigvee$-sentence (the definition of which should be clear). These classes are not elementary classes.

Example 2.11. Let $\tau = \{R\}$ the language of graphs. The class $\mathbb{K}$ of non-connected graphs is a PC-class. Indeed, an undirected graph $\mathcal{G} = (G, R)$ is disconnected if and only if there is a binary relation C of connectedness such that

1. $(\forall x)(\forall y)\,[R(x,y) \to C(x,y)]$,
2. $(\forall x)(\forall y)(\forall z)\,[C(x,y) \wedge C(y,z) \to C(x,z)]$, and
3. $\neg(\forall x)(\forall y)\,C(x,y)$.

An undirected graph $\mathcal{G}$ is also disconnected if and only if

$$(\exists x \neq y) \bigwedge\!\!\!\backslash_{n\in\omega} (\forall u_0, \ldots, u_n)[x \neq u_0 \vee \neg R(u_0,u_1) \vee \neg R(u_1,u_2) \vee \cdots \vee \neg R(u_{n-1},u_n) \vee u_n \neq y].$$

So $\mathbb{K}$ is also defined by a computable $\bigwedge\!\!\backslash$-sentence.

The prototypical example of a PC-class which is not $\mathcal{L}_{\omega_1,\omega}$-definable is class of non-well-founded linear orders.

Example 2.12. Let $\tau = \{<\}$ the language of linear orders. The class $\mathbb{K}$ of non-well-founded linear orders is a PC-class as a linear order $(S, <)$ is non-well-founded if and only if there is a unary relation U such that

$$(\forall x)\big[x \in U \to (\exists y)[y \in U \ \wedge \ y < x]\big].$$

$\mathbb{K}$ is not definable by any $\mathcal{L}_{\omega_1,\omega}$ formula.

A simple example where one can apply Theorem 1.3 is the class of infinite models.

Example 2.13. Let τ be any language and ϕ a finitary τ-sentence. The class $\mathbb{K}$ of infinite models of ϕ is easily seen to be defined by the conjuction of ϕ and the computable $\bigwedge\!\!\!\bigwedge$-sentence

$$\bigwedge\!\!\!\bigwedge_{n\in\omega}(\exists x_0, \ldots, x_n)\left[\bigwedge_{i\neq j} x_i \neq x_j\right].$$

By Theorem 1.3, $\mathbb{K}$ is a PC$'$-class, and by Theorem 2.10 it is a PC-class. Being slightly clever, we can also see that $\mathbb{K}$ is a PC-class by noting that $\mathcal{A} \vDash \phi$ is infinite if and only if there is a linear order $<$ on $\mathcal{A}$ such that $(\forall x)(\exists y)[x < y]$.

We have already mentioned the class of orderable groups.

Example 2.14. Orderable groups are a PC-class. By compactness, they are also universally axiomatizable (in elementary first-order logic) by saying that every finite subset can be ordered in a way that is compatible with the group operation.

Example 2.14 is a particular instance of a more general phenomena: if we take a PC-class that such that (a) the expanded vocabulary only adds relations and (b) the added relations are only universally quantified over, then the resulting class is actually elementary (though it may require infinitely many axioms). This is very particular case in which we can answer Conjecture 1.4.

As an application of Theorem 1.2, let us give an example of a PC$'$-class which is not a PC-class.

Example 2.15. Define an elementary first-order theory T as follows. The language of T will be the language of graphs. Fix an enumeration of the sentences ϕ_n in finite languages $\mathcal{L}_n$ expanding the language of graphs. Note that for every finite graph G, we can decide effectively whether there is an expansion of G to a model of ϕ_n. For each n, let C_n be cycle of length n. Then, let T be the theory that says that there is no cycle of length n for exactly those n where C_n does not have an expansion to a model of ϕ_n.

Note that T is c.e. and universal. By diagonalization, the models of T are not a PC-class, though by Theorem 1.2 they are a PC$'$-class.

As suggested by Theorem 2.10, this example uses finite structures in an integral way.

3 An Application of Craig Interpolation

To prove the direction (1) implies (2) of Theorem 1.1, we will adapt a proof of the Craig Interpolation Theorem for $\mathcal{L}_{\omega_1,\omega}$. We state the standard Craig Interpolation Theorem here for completeness:

Theorem 3.1 (Craig Interpolation Theorem [LE65])**.** *Suppose ϕ_1 and ϕ_2 are $\mathcal{L}_{\omega_1,\omega}$-sentences with $\phi_1 \vDash \phi_2$. There is an $\mathcal{L}_{\omega_1,\omega}$-sentence θ such that $\phi_1 \vDash \theta$, $\theta \vDash \phi_2$, and every relation, function and constant symbol occurring in θ occurs in both ϕ_1 and ϕ_2.*

The proof we adapt is not the original proof by Lopez-Escobar, but one that appears in the book by Marker [Mar16]. The proof of Craig Interpolation makes use of consistency properties. Consistency properties are the infinitary equivalent of Henkin-style constructions in finitary logic. Consistency properties were first introduced by Makkai [Mak69]; the exact definition we use seems to be due to Keisler [Kei71]. See also Definition 4.1 of [Mar16].

Definition 3.2. Let C be a countable collection of new constants. A consistency property Σ is a collection of countable sets σ of $\mathcal{L}_{\omega_1,\omega}$-sentences with the following properties. For $\sigma \in \Sigma$:

(C1) if $\phi \in \sigma$, then $\neg\phi \notin \sigma$;

(C2) if $\neg\phi \in \sigma$, then $\sigma \cup \{\sim \phi\} \in \Sigma$;

(C3) if $\bigwedge\!\!\!\bigwedge_{\phi\in X} \phi \in \sigma$, then for all $\phi \in X$, $\sigma \cup \{\phi\} \in \Sigma$;

(C4) if $\bigvee\!\!\!\bigvee_{\phi\in X} \phi \in \sigma$, then there is $\phi \in X$ such that $\sigma \cup \{\phi\} \in \Sigma$;

(C5) if $(\forall v)\phi(v) \in \sigma$, then for all $c \in C$, $\sigma \cup \{\phi(c)\} \in \Sigma$;

(C6) if $(\exists v)\phi(v) \in \sigma$, then there is $c \in C$ such that $\sigma \cup \{\phi(c)\} \in \Sigma$;

(C7) let t be a term with no variables and let $c, d \in C$,

- (a) if $c = d \in \sigma$, then $\sigma \cup \{d = c\} \in \Sigma$;
- (b) if $c = t \in \sigma$ and $\phi(t) \in \sigma$, then $\sigma \cup \{\phi(c)\} \in \Sigma$;
- (c) there is $e \in C$ such that $\sigma \cup \{e = t\} \in \Sigma$.

Marker [Mar16] includes another condition, that a consistency property be closed under subsets. However he shows in Exercise 4.1.4 that this is unnecessary. Keisler [Kei71] states his definition in the same way as ours, and proves that the closure of a consistency property under subsets is against a consistency property.

A consistency property is in some sense a recipe for building a model.

Theorem 3.3 (Model Existence Theorem)**.** *If Σ is a consistency property and $\sigma \in \Sigma$, there is $\mathcal{M} \vDash \sigma$.*

We are now ready to prove our variant of the Craig Interpolation Theorem. We strengthen the hypotheses to assume that one of the sentences is a $\bigwedge\!\!\!\bigwedge$-sentence, and in return, we get that the interpolant is also a $\bigwedge\!\!\!\bigwedge$-sentence. The proof follows the same structure as that of the Craig Interpolation Theorem in [Mar16] (Theorem 4.3.1).

Theorem 3.4. *Suppose ϕ_1 is a $\bigwedge\!\!\!\bigwedge$-sentence and ϕ_2 is an $\mathcal{L}_{\omega_1,\omega}$-sentence with $\phi_1 \vDash \phi_2$. There is a $\bigwedge\!\!\!\bigwedge$-sentence θ such that $\phi_1 \vDash \theta$, $\theta \vDash \phi_2$, and every relation, function and constant symbol occurring in θ occurs in both ϕ_1 and ϕ_2.*

Proof. Let C be a countable collection of new constants. Let τ_i be the smallest language containing ϕ_i and C, and let $\tau = \tau_1 \cap \tau_2$.

Let Σ be the collection of finite sets of sentences σ containing only finitely many new constants that can be written as $\sigma = \sigma_1 \cup \sigma_2$, where σ_1 is a finite set of $\bigwedge\!\!\!\bigwedge$-τ_1-sentences and σ_2 is a finite set of τ_2-sentences, and such that for all τ-sentences ψ_1 and ψ_2, with ψ_1 a $\bigwedge\!\!\!\bigwedge$-sentence, if $\sigma_1 \vDash \psi_1$ and $\sigma_2 \vDash \psi_2$ then $\psi_1 \wedge \psi_2$ is satisfiable.

In the rest of the proof, we make the convention that if $\sigma \in \Sigma$ and we write $\sigma = \sigma_1 \cup \sigma_2$, then σ_1 and σ_2 are the witnesses that $\sigma \in \Sigma$, i.e., σ_1 consists of $\bigwedge\!\!\!\bigwedge$-τ_1-sentences, σ_2 consists of τ_2-sentences, and they satisfy the satisfiability condition above.

We claim that Σ is a consistency property. The following claim will verify many of the conditions.

Claim. *Fix $\sigma \in \Sigma$ and write $\sigma = \sigma_1 \cup \sigma_2$. If ϕ is a τ_i-sentence (and a $\bigwedge\!\!\!\bigwedge$-sentence if $i = 1$) with $\sigma_i \vDash \phi$, then $\sigma \cup \{\phi\} \in \Sigma$.*

Proof. We will show the case $i = 1$. We can write $\sigma \cup \{\phi\} = (\sigma_1 \cup \{\phi\}) \cup \sigma_2$. If $\sigma_1 \cup \{\phi\} \vDash \psi_1$ and $\sigma_2 \vDash \psi_2$, with ψ_1 a $\bigwedge\!\!\!\bigwedge$-sentence, then since $\sigma_1 \vDash \phi$, $\sigma_1 \vDash \psi_1$. Hence $\psi_1 \wedge \psi_2$ is satisfiable. □

We now check the conditions of a consistency property.

(C1) Suppose for a contradiction that $\phi, \neg\phi \in \sigma = \sigma_1 \cup \sigma_2$. If $\phi \in \sigma_i$ while $\neg\phi \in \sigma_j$ for $i \neq j$, then ϕ is a τ-sentence such that $\sigma_i \vDash \phi$ and $\sigma_j \vDash \neg\phi$, so since $\phi \wedge \neg\phi$ is not satisfiable, this witnesses that $\sigma \notin \Sigma$. If both $\phi, \neg\phi \in \sigma_i$, then $\sigma_i \vDash \phi \wedge \neg\phi$. Now since $\phi \wedge \neg\phi$ is unsatisfiable, letting ψ_1 be any unsatisfiable τ-sentence, we also have that $\sigma_i \vDash \psi_1$. Letting ψ_2 be any τ-sentence such that $\sigma_j \vDash \psi_2$, we see that $\psi_1 \wedge \psi_2$ is unsatisfiable and provides a witness to the fact that $\sigma \notin \Sigma$.

(C2) This follows from the claim.

(C3) This follows from the claim.

(C4) Write $\sigma = \sigma_1 \cup \sigma_2$. We have two cases which are different, depending on whether $\bigvee\!\!\!\bigvee_{\phi \in X} \phi \in \sigma_1$ or $\bigvee\!\!\!\bigvee_{\phi \in X} \phi \in \sigma_2$.

First suppose that $\bigvee\!\!\!\bigvee_{\phi \in X} \phi \in \sigma_2$. Let $\sigma_{2,\phi} = \sigma_2 \cup \{\phi\}$. We claim that for some $\phi \in X$, $\sigma_{2,\phi} \cup \sigma_1 \in \Sigma$. If not, then for each $\phi \in X$ there are τ-sentences $\psi_{2,\phi}$ and $\psi_{1,\phi}$, with $\psi_{1,\phi}$ a $\bigwedge\!\!\!\bigwedge$-sentence, such that $\sigma_{2,\phi} \vDash \psi_{2,\phi}$ and $\sigma_1 \vDash \psi_{1,\phi}$, and such that $\psi_{2,\phi} \wedge \psi_{1,\phi}$ is unsatisfiable. So $\psi_{2,\phi} \vDash \neg\psi_{1,\phi}$. Since

$$\sigma_2 \vDash \bigvee\!\!\!\bigvee_{\phi \in X} \phi$$

we have that

$$\sigma_2 \vDash \bigvee\!\!\!\bigvee_{\phi \in X} \psi_{2,\phi}.$$

On the other hand,

$$\sigma_1 \vDash \bigwedge\!\!\!\bigwedge_{\phi \in X} \psi_{1,\phi}.$$

This formula is a $\bigwedge\!\!\!\bigwedge$-sentence as each $\psi_{1,\phi}$ is. Finally,

$$\bigvee\!\!\!\bigvee_{\phi \in X} \psi_{2,\phi} \vDash \neg \bigwedge\!\!\!\bigwedge_{\phi \in X} \psi_{1,\phi}$$

which contradicts that $\sigma \in \Sigma$.

Now suppose that $\bigvee\!\!\!\bigvee_{\phi \in X} \phi \in \sigma_1$; then X is finite. We begin in a similar way as before. Let $\sigma_{1,\phi} = \sigma_1 \cup \{\phi\}$. We claim that for some $\phi \in X$, $\sigma_{1,\phi} \cup \sigma_2 \in \Sigma$. If not, then for each $\phi \in X$ there are τ-sentences $\psi_{1,\phi}$ and $\psi_{2,\phi}$, with $\psi_{1,\phi}$ a $\bigwedge\!\!\!\bigwedge$-sentence, such that $\sigma_{1,\phi} \vDash \psi_{1,\phi}$ and $\sigma_2 \vDash \psi_{2,\phi}$, and such that $\psi_{1,\phi} \wedge \psi_{2,\phi}$ is unsatisfiable. So $\psi_{1,\phi} \vDash \neg\psi_{2,\phi}$. Since

$$\sigma_1 \vDash \bigvee\!\!\!\bigvee_{\phi \in X} \phi$$

we have that

$$\sigma_1 \vDash \bigvee\!\!\!\bigvee_{\phi \in X} \psi_{1,\phi}.$$

As X is finite this a $\bigwedge\!\!\!\bigwedge$-sentence. On the other hand,

$$\sigma_2 \vDash \bigwedge\!\!\!\bigwedge_{\phi \in X} \psi_{2,\phi}$$

and

$$\bigvee\!\!\!\bigvee_{\phi \in X} \psi_{1,\phi} \vDash \neg \bigwedge\!\!\!\bigwedge_{\phi \in X} \psi_{2,\phi}$$

which contradicts that $\sigma \in \Sigma$.

(C5) This follows from the claim as $(\forall x)\phi(x) \vDash \phi(c)$ for all $c \in C$.

(C6) If $(\exists x)\phi(x) \in \sigma$, then choose $c \in C$ which does not appear in σ. Suppose that $(\exists x)\phi(x) \in \sigma_1$; the case where $(\exists x)\phi(x) \in \sigma_2$ is similar. We claim that $\sigma \cup \{\phi(c)\} \in \Sigma$. Since $(\exists x)\phi(x) \in \sigma_1$, $\phi(x)$ is a $\bigwedge\!\!\!\bigwedge$-formula, and thus so is $\phi(c)$.

Suppose that $\sigma_1 \cup \{\phi(c)\} \vDash \psi_1$ and $\sigma_2 \vDash \psi_2$, where ψ_1 is a $\bigwedge\!\!\!\bigwedge$-sentence. Write $\psi_1 = \theta_1(c)$ and $\psi_2 = \theta_2(c)$. We have $\sigma_1 \vDash \phi(c) \to \theta_1(c)$, and so since c does not appear in σ_1, $\sigma_1 \vDash (\forall x)[\phi(x) \to \theta_1(x)]$. Similarly, $\sigma_2 \vDash (\forall x)\theta_2(x)$. Also, $\sigma_1 \vDash (\exists x)\phi(x)$ and so $\sigma_1 \vDash (\exists x)\theta_1(x)$. So $(\exists x)\theta_1(x) \wedge (\forall x)\theta_2(x)$ is satisfiable, say in a model $\mathcal{M}$. Note that the constant c does not appear in the formula $(\exists x)\theta_1(x) \wedge (\forall x)\theta_2(x)$, so we may choose the interpretation of c in $\mathcal{M}$ such that $\mathcal{M} \vDash \theta_1(c)$. Then $\mathcal{M} \vDash \theta_1(c) \wedge \theta_2(c)$. So $\psi_1 \wedge \psi_2$ is satisfiable, and $\sigma \cup \{\phi(c)\} \in \Sigma$.

(C7) let t be a term with no variables and let $c, d \in C$,

(a) This follows from the claim.

(b) Suppose $c = t \in \sigma$ and $\phi(t) \in \sigma$. Write $\sigma = \sigma_1 \cup \sigma_2$. Consider $\mu = \sigma \cup \{\phi(c)\} = \sigma_1 \cup \sigma_2 \cup \{\phi(c)\}$. Suppose $c = t \in \sigma_i$ and $\phi(t) \in \sigma_j$. The case $i = j$ follows from the claim, so we consider the case $i \neq j$. Suppose that $\sigma_i \vDash \psi_i$ and $\sigma_j \cup \{\phi(c)\} \vDash \psi_j$. Then $\sigma_i \vDash c = t \wedge \psi_i$ and $\sigma_j \vDash c = t \to \psi_j$, so $c = t \wedge \psi_i \wedge (c = t \to \psi_i)$ is satisfiable. So $\psi_i \wedge \psi_j$ is satisfiable.

(c) Pick $e \in C$ which does not appear in $\sigma = \sigma_1 \cup \sigma_2$. Then if $\sigma_1 \cup \{e = t\} \vDash \psi_1$ and $\sigma_2 \cup \{e = t\} \vDash \psi_2$, write $\psi_1 = \theta_1(e)$ and $\psi_2 = \theta_2(e)$. Then since e does not appear in σ_1 or σ_2, $\sigma_1 \vDash \theta_1(t)$ and $\sigma_2 \vDash \theta_2(t)$. Thus $\theta_1(t) \wedge \theta_2(t)$ is satisfiable. Given a model of $\theta_1(t) \wedge \theta_2(t)$, setting the interpretation of c to t, we get a model of $\psi_1 \wedge \psi_2$. So $\psi_1 \wedge \psi_2$ is satisfiable.

Since $\phi_1 \vDash \phi_2$, $\{\phi_1, \neg\phi_2\} \notin \Sigma$ as otherwise by the Model Existence Theorem there would be a model of $\phi_1 \wedge \neg\phi_2$. By definition of Σ, there are τ-sentences ψ_1 and ψ_2, with ψ_1 a $\bigwedge\!\!\!\bigwedge$-sentence, such that $\phi_1 \vDash \psi_1$, $\neg\phi_2 \vDash \psi_2$, and $\psi_1 \wedge \psi_2$ is not satisfiable. So we have that $\phi_1 \vDash \psi_1$, $\psi_1 \vDash \neg\psi_2$, and $\neg\psi_2 \vDash \phi_2$. Hence $\phi_1 \vDash \psi_1$ and $\psi_1 \vDash \phi_2$.

Thus ψ_1 is the desired interpolant, except that it may contain constants from C. Write $\psi_1 = \theta(\bar{c})$, where θ is an τ-formula with no constants from $\bar{c}$. Neither ϕ_1 nor ϕ_2 contains constants from C, and so $\phi_1 \vDash (\forall \bar{x})\theta(\bar{x})$ and $(\exists \bar{x})\theta(\bar{x}) \vDash \phi_2$. Since $(\forall \bar{x})\theta(\bar{x}) \vDash (\exists \bar{x})\theta(\bar{x})$, we can take $(\forall \bar{x})\theta(\bar{x})$ as the interpolant. □

We get the following corollary, which is (1) implies (2) of Theorem 1.1. Interestingly, when we apply the interpolation theorem in the proof, one of the languages contains the other (i.e., we have $\tau_1 \supseteq \tau_2$ so that $\tau = \tau_1 \cap \tau_2 = \tau_2$). If it were not for our added assumptions on the form of the formulas involved, finding an interpolant would be trivial as we could just take the sentence in the smaller language.

Corollary 3.5. *Let $\mathbb{K}$ be a class of τ-structures closed under isomorphism. If $\mathbb{K}$ is both a PC_Δ-class and $\mathcal{L}_{\omega_1,\omega}$-elementary, then it is defined by a $\bigwedge\!\!\!\bigwedge$-sentence.*

Proof. Let $\tau^* \supseteq \tau$ be an expanded language and let X be a set of first-order sentences such that $\mathbb{K}$ is the class of reducts to τ of models of $\psi_1 = \bigwedge\!\!\!\bigwedge_{\phi \in X} \phi$. Note that ψ_1 is a $\bigwedge\!\!\!\bigwedge$-sentence.

Let ψ_2 be an $\mathcal{L}_{\omega_1,\omega}(\tau)$-sentence defining $\mathbb{K}$. We have that $\psi_1 \vDash \psi_2$, so by the Interpolation Theorem, there is a $\bigwedge\!\!\!\bigwedge$-τ-sentence θ such that $\psi_1 \vDash \theta$ and $\theta \vDash \psi_2$.

Every $\mathcal{M} \in \mathbb{K}$ has an expansion which is a model of ψ_1 and hence is itself a model of θ; and every model of θ is a model of ψ_2, and hence in the class $\mathbb{K}$. So θ defines $\mathbb{K}$. □

4 The Skolem Argument

For the direction (2)⇒(1) of Theorem 1.1, we must prove the following theorem. The proof works for sentences from $\mathcal{L}_{\kappa,\omega}$ for any κ, though the reader should feel free to take $\kappa = \omega_1$. (The logic $\mathcal{L}_{\kappa,\omega}$ is defined in the same way as $\mathcal{L}_{\omega_1,\omega}$ except that we allow conjunctions and disjunctions of size $< \kappa$.)

Theorem 4.1. *Let $\mathbb{K}$ be a class of structures closed under isomorphism. If $\mathbb{K}$ is defined by a $\bigwedge\!\!\!\bigwedge$-sentence of $\mathcal{L}_{\kappa,\omega}$, then it is a pseudo-elementary (PC_Δ) class.*

We have extended the notion of a $\bigwedge\!\!\!\bigwedge$-formula from $\mathcal{L}_{\omega_1,\omega}$ to $\mathcal{L}_{\kappa,\omega}$ using the same definition (see Definition 2.4).

Remark 4.2. One can extend this theorem to $\bigwedge\!\!\!\bigwedge$-theories (sets of $\bigwedge\!\!\!\bigwedge$-sentences) because every $\bigwedge\!\!\!\bigwedge$-theory can be turned into a $\bigwedge\!\!\!\bigwedge$-sentence by taking the conjunction, but this might change the logic. For instance, any uncountable first-order theory T is a $\bigwedge\!\!\!\bigwedge$-theory in $\mathcal{L}_{\omega_1,\omega}$, but not a $\bigwedge\!\!\!\bigwedge$-sentence in $\mathcal{L}_{\omega_1,\omega}$ (this can be proved by noting the lack of countable models). Of course, T is a $\bigwedge\!\!\!\bigwedge$-sentence in $\mathcal{L}_{|T|^+,\omega}$.

Morally, the idea of the proof is to Skolemize the language to be left with a universal $\bigwedge\!\!\!\bigwedge$-theory in an expanded language, and then the infinitary conjunctions can be dropped. The main construction is the following lemma.

Lemma 4.3. *Let $\varphi(\bar{x})$ be a $\bigwedge\!\!\!\bigwedge$-formula in $\mathcal{L}_{\kappa,\omega}(\tau)$. There is an expanded language $\tau_\varphi \supset \tau$ and a set $\Phi(\varphi)$ of first-order τ_φ-formulas with the same free variables that* verifies *φ in the following sense:*

1. *Given any τ_φ-structure $\mathcal{A}^+$ and $\bar{a} \in \mathcal{A}^+$,*

$$\forall \theta \in \Phi(\varphi), \mathcal{A}^+ \vDash \theta(\bar{a}) \Longrightarrow \mathcal{A}^+ \vDash \varphi(\bar{a}).$$

2. *Given any τ-structure $\mathcal{A}$, there is an expansion $\mathcal{A}^+_\varphi$ such that for all $\bar{a} \in \mathcal{A}$,*

$$\mathcal{A} \vDash \varphi(\bar{a}) \iff \forall \theta \in \Phi(\varphi), \mathcal{A}^+_\varphi \vDash \theta(\bar{a}).$$

Proof. **Construction:** We work by induction on the formula $\varphi(\bar{x})$. Although there is no prenex normal form for formulas of $\mathcal{L}_{\kappa,\omega}$, formulas are defined inductively. In particular, we follow the definition given for $\bigwedge\!\!\!\bigwedge$-formulas from Definition 2.4, using Remark 2.5 to assume that any finite disjunctions occur only as part of finitary, quantifier-free formulas.

1. $\varphi(\bar{x})$ is a finitary, quantifier-free formula.

   Set $\tau_\varphi = \tau$ and $\Phi(\varphi) = \{\varphi(\bar{x})\}$ (in fact, this works for any finitary formula).

2. $\varphi(\bar{x})$ is $(\exists y)\psi(\bar{x}, y)$.

   Set $\tau_\varphi = \tau_\psi \cup \{f_\theta(\bar{x}) \mid \theta \in \Phi(\psi)\}$ where each f_θ is a new function symbol, and set

$$\Phi(\varphi) = \{\theta(\bar{x}, f_\theta(\bar{x})) \mid \theta(\bar{x}, y) \in \Phi(\psi)\}.$$

3. $\varphi(\bar{x})$ is $(\forall y)\psi(\bar{x}, y)$.

   Set $\tau_\varphi = \tau_\psi$ and

$$\Phi(\varphi) = \{(\forall y)\theta(\bar{x}, y) \mid \theta(\bar{x}, y) \in \Phi(\psi)\}.$$

4. $\varphi(\bar{x})$ is $\bigwedge\!\!\!\bigwedge_{i\in I} \psi_i(\bar{x})$.

   Set $\tau_\varphi = \cup_{i\in I} \tau_{\psi_i}$ where the union is disjoint over τ; that is, new functions in τ_{ψ_i} and τ_{ψ_j} are distinct in τ_φ. Then set

$$\Phi(\varphi) = \bigcup_{i\in I} \Phi(\psi_i).$$

This works: We verify the construction inductively using the same cases. Is is easy to argue inductively that given any τ_φ-structure $\mathcal{A}^+$ and $\bar{a} \in \mathcal{A}^+$,

$$\forall \theta \in \Phi(\varphi), \mathcal{A}^+ \vDash \theta(\bar{a}) \Longrightarrow \mathcal{A}^+ \vDash \varphi(\bar{a}).$$

1. Immediate.

2. Suppose that for all $\theta \in \Phi(\varphi)$, $\mathcal{A}^+ \vDash \theta(\bar{a})$. Then, for each $\theta(\bar{x}, y) \in \Phi(\psi)$, $\mathcal{A}^+ \vDash \theta(\bar{a}, f_\theta(\bar{a}))$. By the induction hypothesis, $\mathcal{A}^+ \vDash \psi(\bar{a}, f_\theta(\bar{a}))$. So $\mathcal{A}^+ \vDash (\exists y)\psi(\bar{a}, y)$, i.e., $\mathcal{A}^+ \vDash \varphi(\bar{a})$.

3. Suppose that for all $\theta \in \Phi(\varphi)$, $\mathcal{A}^+ \vDash \theta(\bar{a})$. Then, for each $\theta(\bar{x}, y) \in \Phi(\psi)$, $\mathcal{A}^+ \vDash (\forall y)\theta(\bar{a}, y)$, and so for each $b \in \mathcal{A}^+$, $\mathcal{A}^+ \vDash \theta(\bar{a}, b)$. By the induction hypothesis, $\mathcal{A}^+ \vDash \psi(\bar{a}, b)$ for each $b \in \mathcal{A}^+$. So $\mathcal{A}^+ \vDash (\forall y)\psi(\bar{a}y)$, i.e., $\mathcal{A}^+ \vDash \varphi(\bar{a})$.

4. Suppose that for all $\theta \in \Phi(\varphi)$, $\mathcal{A}^+ \vDash \theta(\bar{a})$. Then, for each ψ_i and each $\theta(\bar{x}) \in \Phi(\psi_i)$, $\mathcal{A}^+ \vDash \theta(\bar{a})$. By the induction hypothesis, $\mathcal{A}^+ \vDash \psi_i(\bar{a})$ for each i, and so $\mathcal{A}^+ \vDash \varphi(\bar{a})$.

Now we will show inductively how to define $\mathcal{A}^+_\varphi$ and verify that

$$\mathcal{A} \vDash \varphi(\bar{a}) \iff \forall \theta \in \Phi(\varphi), \mathcal{A}^+_\varphi \vDash \theta(\bar{a}).$$

1. Immediate.

2. Fix $\mathcal{A}$. By induction, we have an expansion $\mathcal{A}^+_\psi$. Expand further to form $\mathcal{A}^+_\varphi$ by picking each f_θ to be a Skolem function for θ; that is, ensure

$$\mathcal{A}^+_\varphi \vDash \forall \bar{x} \left((\exists y)\theta(\bar{x}, y) \leftrightarrow \theta(\bar{x}, f_\theta(\bar{x})) \right).$$

   Then fix $\bar{a} \in \mathcal{A}$.

$$\begin{aligned}
\mathcal{A} \vDash \varphi(\bar{a}) &\iff \exists b \in \mathcal{A},\ \mathcal{A} \vDash \psi(\bar{a}, b) \\
&\iff \exists b \in \mathcal{A},\ \forall \theta \in \Phi(\psi),\ \mathcal{A}^+_\psi \vDash \theta(\bar{a}, b) \\
&\iff \exists b \in \mathcal{A},\ \forall \theta \in \Phi(\psi),\ \mathcal{A}^+_\varphi \vDash \theta(\bar{a}, b) \\
&\iff \forall \theta \in \Phi(\psi),\ \mathcal{A}^+_\varphi \vDash \theta(\bar{a}, f_\theta(\bar{a})).
\end{aligned}$$

3. Fix $\mathcal{A}$ and set $\mathcal{A}^+_\varphi = \mathcal{A}^+_\psi$. Fix $\bar{a} \in \mathcal{A}$.

$$\begin{aligned}
\mathcal{A} \vDash \varphi(\bar{a}) &\iff \forall b \in \mathcal{A},\ \mathcal{A} \vDash \psi(\bar{a}, b) \\
&\iff \forall b \in \mathcal{A},\ \forall \theta \in \Phi(\psi),\ \mathcal{A}^+_\psi \vDash \theta(\bar{a}, b) \\
&\iff \forall \theta \in \Phi(\psi),\ \mathcal{A}^+_\varphi \vDash (\forall y)\theta(\bar{a}, y).
\end{aligned}$$

4. Fix $\mathcal{A}$ and set $\mathcal{A}^+_\varphi$ to be the joint expansion of all of the $\mathcal{A}^+_{\psi_i}$'s; here we crucially use that the new functions in the different languages are distinct. Fix $\bar{a} \in \mathcal{A}$.

$$\begin{aligned}
\mathcal{A} \vDash \varphi(\bar{a}) &\iff \forall i \in I,\ \mathcal{A} \vDash \psi_i(\bar{a}) \\
&\iff \forall i \in I,\ \forall \theta \in \Phi(\psi_i),\ \mathcal{A}^+_{\psi_i} \vDash \theta(\bar{a}) \\
&\iff \forall i \in I,\ \forall \theta \in \Phi(\psi_i),\ \mathcal{A}^+_\varphi \vDash \theta(\bar{a}) \\
&\iff \forall \theta \in \bigcup_{i \in I} \Phi(\psi_i),\ \mathcal{A}^+_\varphi \vDash \theta(\bar{a}).
\end{aligned}$$

This completes the proof. □

From this lemma, the proof of the theorem is immediate.

Proof of Theorem 4.1. Let φ be a $\bigwedge\!\!\!\bigwedge$-sentence of $\mathcal{L}_{\kappa,\omega}$. Apply Lemma 4.3 to φ. Since φ is a sentence (has no free variables), $\Phi(\varphi)$ is a collection of sentences. Then

$$\text{Mod } \varphi = \{\mathcal{A} \mid \text{ there is } \mathcal{A}^+ \text{ expanding } \mathcal{A} \text{ with } \mathcal{A}^+ \vDash \Phi(\varphi)\}.$$ □

5 Game Formulas

In this section, we show how the direction (2)⇒(1) of Theorem 1.1 follows from known results on game formulas.

Definition 5.1. A closed game formula[1] is an expression of the form

$$\forall y_1 \exists z_1 \forall y_2 \exists z_2 \cdots \bigwedge\!\!\!\bigwedge_n \varphi_n(\bar{x}, y_1, z_1, y_2, z_2, \ldots)$$

where each φ_n is an elementary first-order formula. Such a formula is computable if the sequence φ_n is computable.

Satisfaction for such formulas is defined by a game played between two players, with player I playing the $\forall$ quantifiers and player II playing the $\exists$ quantifiers; player II wins, and the formulas is satisfied, if he can make $\varphi_n(\bar{x}, y_1, z_1, \ldots)$ true for every n. Alternatively, satisfaction can be defined by the existence of Skolem functions (which turn out to be the winning strategies for player II).

Note that each φ_n has finitely many free variables. Also, the 'closed' adjective refers to use of conjunctions in the formula.

Every (computable) $\bigwedge\!\!\!\bigwedge$-formula is equivalent to a (computable) closed game formula by moving all of the quantifiers to the front. In doing this, one must take care to rename bound variables so that each variable is quantified over a single time. This may seem at first to be false by a reader familiar with the fact that one cannot do this and obtain an $\mathcal{L}_{\omega_1,\omega}$ formula, but one *can* do this and obtain a closed game formula. For example,

$$\bigwedge\!\!\!\bigwedge_n \exists \bar{x}_n \phi_n(\bar{x}_n) \iff \exists \bar{x}_1 \exists \bar{x}_2 \cdots \bigwedge\!\!\!\bigwedge_n \phi_n(\bar{x}_n).$$

We can define the game formula inductively; for the inductive step, we have:

$$\bigwedge\!\!\!\bigwedge_i \forall y_i^1 \exists z_i^1 \forall y_i^2 \exists z_i^2 \cdots \bigwedge\!\!\!\bigwedge_n \varphi_{i_n}(\bar{x}_i, y_i^1, z_i^1, y_i^1, z_i^2, \ldots)$$

$$\iff \forall y_1^1 \exists z_1^1 \forall y_1^2 \exists z_1^2 \forall y_2^1 \exists z_2^1 \forall y_1^3 \exists z_1^3 \forall y_2^2 \exists z_2^2 \forall y_3^1 \exists z_3^1 \cdots \bigwedge\!\!\!\bigwedge_{i,n} \varphi_{i_n}(\bar{x}_i, y_i^1, z_i^1, y_i^1, z_i^2, \ldots).$$

[1] An important note is that in general a closed game formula is *not* an element of $\mathcal{L}_{\omega_1,\omega}$ or even $\mathcal{L}_{\omega_1,\omega_1}$. It is not in the first logic because there are *infinitely* many quantifiers in front of infinite conjunction, and it is not in the second logic because the quantifiers are not added in a well-founded way.

Essentially we need to merge ω-many sequences (or quantifiers) of order type ω into a single sequence of order type ω, maintaining the order of each of the individual sequences inside the amalgamated sequence.

So we can get we get the direction (2)⇒(1) of Theorem 1.1 as well as Theorem 1.3 as a corollaries of the following theorem:

Theorem 5.2 (Theorem 2.1.4 of [Kol85], Corollary 6.7 of [Bar75])**.**

1. *Any class of τ-structures defined by a closed game formula is* PC_Δ.
2. *Any class of τ-structures defined by a computable closed game formula is* PC'.

The proof given in the previous section is, however, much simpler. Indeed, the proof in Section 4 gives a proof of the first item above because the Skolem functions for closed game formulas are still finitary functions because each stage of the game has only finitely many plays before it (and because each of the formulas φ_n has finitely many free variables). This proof could be further generalized to consider longer games, showing that any class defined by a higher analogue of closed game formulas is PC in some infinitary logic $\mathcal{L}_{\kappa,\lambda}$.

References

[AK00] C. J. Ash and J. Knight. *Computable structures and the hyperarithmetical hierarchy*, volume 144 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 2000.

[Bar75] Jon Barwise. *Admissible sets and structures*. Springer-Verlag, Berlin-New York, 1975. An approach to definability theory, Perspectives in Mathematical Logic.

[CV58] W. Craig and R. L. Vaught. Finite axiomatizability using additional predicates. *J. Symb. Logic*, 23:289–308, 1958.

[Hod08] W. Hodges. *Model Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2008.

[Kei71] H. Jerome Keisler. *Model theory for infinitary logic. Logic with countable conjunctions and finite quantifiers*. North-Holland Publishing Co., Amsterdam-London, 1971. Studies in Logic and the Foundations of Mathematics, Vol. 62.

[Kol85] Ph. G. Kolaitis. Game quantification. In *Model-theoretic logics*, Perspect. Math. Logic, pages 365–421. Springer, New York, 1985.

[LE65] E. G. K. Lopez-Escobar. An interpolation theorem for denumerably long formulas. *Fund. Math.*, 57:253–272, 1965.

[Mak69] M. Makkai. On the model theory of denumerably long formulas with finite strings of quantifiers. *J. Symbolic Logic*, 34:437–459, 1969.

[Mal71] Jerome Malitz. Infinitary analogs of theorems from first order model theory. *The Journal of Symbolic Logic*, 36(2):216–228, Jun 1971.

[Mar16] D. Marker. *Lectures on infinitary model theory*, volume 46 of *Lecture Notes in Logic*. Association for Symbolic Logic, Chicago, IL; Cambridge University Press, Cambridge, 2016.

[Sco65] D. Scott. Logic with denumerably long formulas and finite strings of quantifiers. In *Theory of Models (Proc. 1963 Internat. Sympos. Berkeley)*, pages 329–341. North-Holland, Amsterdam, 1965.

[She12] Saharon Shelah. Nice infinitary logics. *The Journal of the American Mathematical Society*, 25:395–427, 2012.

[Vak16] A. Vakili. *Temporal Logic Model Checking as Automated Theorem Proving*. PhD thesis, University of Waterloo, 2016.

[VD14] A. Vakili and N. A. Day. Reducing ctl-live model checking to first-order logic validity checking. In *2014 Formal Methods in Computer-Aided Design (FMCAD)*, pages 215–218, Oct 2014.