Complexity of Cubical Cofibration Logics I: coNP-Complete Examples

Robert Rose
Dept. of Mathematics & Computer Science, Wesleyan University, Middletown, CT, USA

Daniel R. Licata
Dept. of Mathematics & Computer Science, Wesleyan University, Middletown, CT, USA

Abstract

We provide a comprehensive classification of the cofibration entailment problem, COFENT, for the cofibration logics of various cubical type theories in use today. The problem COFENT arose from the need of cubical proof assistants to automate reasoning about cubical complexes included in an n-dimensional hypercube. Intuitively, it asks: given logical descriptions of two such complexes, is one a subcomplex of the other? We show that the common variants of COFENT are coNP-complete.

2012 ACM Subject Classification Theory of computation → Type theory

Keywords and phrases cubical sets, internal language, intuitionistic logic, dependent type theory, homotopy type theory, decision procedures

Digital Object Identifier 10.4230/LIPIcs.TYPES.2024.9

Funding This material is based upon work supported by the Air Force Office of Scientific Research under Grant No.FA9550-21-0009 (Tristan Nguyen, program manager). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Air Force Office of Scientific Research.
Robert Rose: US Air Force Research Lab, FA-95502110009 and FA-95501510053
Daniel R. Licata: US Air Force Research Lab, FA-95502110009 and FA-95501510053

1 Introduction

A well-known tool in topos theory for reasoning about a given topos is its Mitchell-Bénabou language [5] (or simply, its "language"), a formal abstraction of the topos as a typed (i.e., multi-sorted) first-order intuitionistic language with a fixed interpretation in that topos. Because the syntax of the Mitchell-Bénabou language is defined directly from the topos (e.g., the types of the logic are just the objects of the topos), this interpretation is nearly trivial: terms denote morphisms and are represented by syntax for certain categorical operations which produce morphisms, such as pairing or post-composition with a given morphism; formulas are, idiosyncratically, terms whose type is the subobject classifier of the topos (or a subobject thereof).

Proving or disproving that a topos has a given property expressed in the Mitchell-Bénabou language is greatly facilitated by the Kripke-Joyal semantics (or "sheaf semantics", for a topos of sheaves), which provides a rigorous translation of arguments in this language to ordinary classical mathematics. The semantics has the familiar form of a Tarskian truth definition. (However, because the interpretation is fixed, the forcing relation is defined a priori in terms of this definition, and the semantics is presented as a theorem.)

This semantics affords a convenient point of departure for investigating the computability and complexity theoretic properties of formal first-order reasoning in a topos. Specifically, this area gives rise to generalizations of the *model-checking problem*. It is a rich area which has received relatively little attention.

1.1 Language fragments for cubical homotopy theory

One application of Mitchell-Bénabou language is to the homotopy theory of cubical sets, where by cubical set we mean a contravariant functor on a particular indexing category $\mathcal{C}$ (a "cube category"). The topos is the category of cubical sets which are sheaves for a given Grothendieck topology on $\mathcal{C}$. In this paper, this topology is trivial: the toposes are presheaf toposes.

The language of a topos of cubical sets is expressive enough to define important homotopical structure: notably, to specify which morphisms of cubical sets qualify as maps of spaces (i.e., "fibrations") and which maps of spaces further qualify as equivalences (i.e., "trivial fibrations"). This is usually achieved first by selecting a collection of subfunctors of representable functors (e.g., in the case of simplicial homotopy theory, a standard choice is the "horn inclusions") and hence by taking a fibration to be a morphism f for which any map from a select subfunctor inclusion to f affords an extension to the base representable. Trivial fibrations have this extension property with respect to a larger selection of such subfunctors. These collections of subfunctors form the bases for generating classes of morphisms called, respectively, "trivial cofibrations" and "cofibrations". Extension of maps along trivial cofibrations is the fundamental operation provided by homotopical structure.

While these extension properties are readily expressible in the language of the topos, a more restricted usage of the language may suffice in certain frameworks (e.g., homotopy type theory). For example, the language of the topos may be used just for the purposes of specifying cofibrations into representables and expressing their inclusion relation. Since cofibrations are classified by Ω, the inclusion relation is naturally expressed by entailment in the language. The sacrifice in expressivity brings significant computational benefits. In Section 3, we will define a number of such language fragments.

1.2 Applications for cubical type theory

Our interest in decision problems arising from the language of a topos indeed began with a practical need. A key component of a proof assistant we were designing would automatically decide formula entailment in a fragment of the language of a topos: namely, presheaves on the finite-product theory of distributive lattices, an important definitional variant of the category of cubical sets. Analogous decision procedures are core components in existing implementations of cubical type theories. For example, entailment in a fragment of the language of the topos of presheaves on the finite-product theory of De Morgan algebras is decided in the proof assistant Agda's cubical mode, which is an implementation of the cubical type theory presented in [3]. Entailment in a fragment of the language of the topos of presheaves on the finite-product theory of bipointed sets is decided in the proof assistant `cooltt`, which is an implementation of the cubical type theory presented in [1]. Our original motivation began with investigating the feasibility of implementing the directed cubical type theory presented in [7], which is an adaptation of the directed simplicial type theory presented in [6].

2 Motivating examples

For the sake of intuition, we present some instances and non-instances of the decision problems of interest before launching into the technical development.

2.1 Cube boundary inclusion

Consider a 3-dimensional cube, with 8 corner points $(0,0,0),\ldots,(1,1,1)$. As a cubical set, it is modeled by a representable functor $\mathrm{Hom}(-,\mathrm{X}^3)$. The object X is a formal element which corresponds to a single dimension. The identity morphism on X corresponds to a variable x_1 along that dimension. Hence, the object X^3 corresponds to three dimensions; and x_1, x_2 and x_3 each correspond to a dimension in the given cube. Let us picture x_1 as varying from left to right; x_2 as varying bottom to top; and x_3 as varying from front to back.

By means of equations involving these variables and end points 0 and 1, we can describe certain parts of this cube. For example, the equation $(x_1 = 0)$ describes just the left face, and $(x_1 = 1)$ describes just the right face. We can combine these using $\vee$ to get exactly the left and the right faces: $(x_1 = 0) \vee (x_1 = 1)$. The entire 2-dimensional boundary of the 3-cube is described by the formula

$$(x_1 = 0) \vee (x_1 = 1) \vee (x_2 = 0) \vee (x_2 = 1) \vee (x_3 = 0) \vee (x_3 = 1) \tag{1}$$

We can also describe the 1-dimensional edges of the 3-cube by combining formulas describing faces using $\wedge$. For example, the top-front edge is described by the formula $(x_2 = 1) \wedge (x_3 = 0)$. The 1-dimensional boundary of the 3-cube is described by the formula

$$\begin{array}{cccccc} & (x_1 = 0) \wedge (x_2 = 0) & \vee & (x_1 = 0) \wedge (x_2 = 1) & \vee & (x_1 = 0) \wedge (x_3 = 0) \\ \vee & (x_1 = 0) \wedge (x_3 = 1) & \vee & (x_1 = 1) \wedge (x_2 = 0) & \vee & (x_1 = 1) \wedge (x_2 = 1) \\ \vee & (x_1 = 1) \wedge (x_3 = 0) & \vee & (x_1 = 1) \wedge (x_3 = 1) & \vee & (x_2 = 0) \wedge (x_3 = 0) \\ \vee & (x_2 = 0) \wedge (x_3 = 1) & \vee & (x_2 = 1) \wedge (x_3 = 0) & \vee & (x_2 = 1) \wedge (x_3 = 1) \end{array} \tag{2}$$

(Recall that $\wedge$ has higher precedence than $\vee$.)

Let ϕ_1 be the formula (1), and let ϕ_2 be the formula (2). Here are two queries for a decision problem: does ϕ_1 entail ϕ_2? or does ϕ_2 entail ϕ_1? Geometric intuition tells us no and yes. The internal language of the topos of cubical sets will also tell us this.

2.2 Simplex and pyramid

The last pair of problems are common to all of the cubical type theories mentioned in Section 1.2. We now present an example which, among the three, can be expressed only in the language fragment used in [7].

We just saw how to combine equations using $\wedge$ and $\vee$ to describe parts of a cube. It is also possible to combine variables directly to describe the *minimum* of two variables or the *maximum* of two variables. The notation is the same: $x_1 \wedge x_2$ denotes the min of x_1 and x_2, while $x_1 \vee x_2$ denotes their max.

The 2-simplex below the diagonal from $(0,0)$ to $(1,1)$ of the solid square (2-cube) is described by $(x_1 = x_1 \vee x_2)$ (or alternatively by $(x_2 = x_1 \wedge x_2)$). The same formula can be used to describe part of the 3-cube: it is the wedge containing the right face and the bottom face. Similarly, $(x_2 = x_2 \vee x_3)$ describes the wedge formed between the 2-simplices above the diagonal on the left and right faces. Combining the two wedges with $\wedge$ yields a 3-simplex:

$$(x_1 = x_1 \vee x_2) \wedge (x_2 = x_2 \vee x_3)$$

The pyramid whose apex is the corner point $(0,0,0)$ and whose base is the right face of the 3-cube is described by $(x_1 = x_1 \vee x_2 \vee x_3)$. Thus, we have two more queries for a decision problem. Is this formula valid?

$$(x_1 = x_1 \vee x_2) \wedge (x_2 = x_2 \vee x_3) \;\Rightarrow\; (x_1 = x_1 \vee x_2 \vee x_3) \tag{3}$$

Or this one?

$$(x_1 = x_1 \vee x_2 \vee x_3) \;\Rightarrow\; (x_1 = x_1 \vee x_2) \wedge (x_2 = x_2 \vee x_3) \tag{4}$$

2.3 Example reduction

In what follows, we will show that the instances (2) ⇒ (1), (1) ⇒ (2), (3), and (4) may be decided by an efficient translation into second-order logic, where the problem of checking the truth of the corresponding formula in a suitable model is in the complexity class coNP. For example, the translation of (3) to second-order logic is

$$\begin{aligned}\forall R\,\big(\chi_3 \to \forall x\,(R(x) \to x_1 = x_1 \vee x_2)\ \wedge\ \forall x\,(R(x) \to x_2 = x_2 \vee x_3)\\ \to \forall x\,(R(x) \to x_1 = x_1 \vee x_2 \vee x_3)\big)\end{aligned}$$

where χ_3 is a formula encoding the constraint that ternary relations assigned to R have minimal and maximal elements. The entailment (3) holds if and only if the formula above is true in the model $\mathcal{A} = (\{0,1\}, 0^{\mathcal{A}} = 0, 1^{\mathcal{A}} = 1, \wedge^{\mathcal{A}} = \min, \vee^{\mathcal{A}} = \max)$.

The translation alone provides the core of a practical decision procedure. What remains is boilerplate and optimizations. The model-checking instance above may in turn be efficiently encoded in propositional logic and checked with a state-of-the-art SAT-solver. Alternatively, the model-checking instance may be handed almost as-is (after some lightweight wrangling) to an SMT-solver like `z3` or to an answer-set programming system like `clingo`.

3 Decision problems

Because terms in the language of a topos are built directly from its morphisms, this language is not in general recursively enumerable, and is often a proper class. Thus, it may not be that the entire language of a topos can be encoded as a countable set of strings over a finite alphabet. In computability and complexity theory, however, decision problems are ordinarily framed as such. For example, in a sheaf topos, any set at all induces a sheaf, and hence a type in the language. But for the purposes of defining a conventional decision problem, we can reasonably expect to encode no more than a countable number of types.

Furthermore, because terms in the language of a topos are formal combinations of morphisms, morphism equality is reducible to term equality: in particular, if morphism equality is undecidable, so is term equality. In contrast, equality of instances of a conventional decision problem is efficiently reducible to equality of strings over a finite alphabet.

Hence, one straightforward strategy is simply to limit one's focus to a suitable fragment of the language of the topos, and to define our decision problems relative to this fragment. To this end, we will next consider fragments which are "finitely presented" by formula languages.

3.1 Formula languages

Given a set A, we will denote a generic element of A^m by $a \in A^m$ where $a = (a_1, \ldots, a_m)$. We will extend this notational convention to variables, parameters, and terms: x will denote an m-tuple of variables $x = (x_1, \ldots, x_m)$; and likewise for $\alpha = (\alpha_1, \ldots, \alpha_m)$ and for $s = (s_1, \ldots, s_m)$. Annotations or constructors may be "broadcast" over an m-tuple: for example, $\forall x\, \psi$ abbreviates $\forall x_1 \cdots \forall x_m\, \psi$. The concatenation of a k-tuple a and an m-tuple b will be denoted by $a\,;b$. We will freely omit parentheses when they may be inferred.

3.1.1 Types and terms

We will define a term language T prior to the formula language in the conventional way. Fix a finite set F of morphisms in $\mathcal{E}$. The term types of the fragment are generated by the domains and codomains appearing in F. The terms of the formula language will be the fragment of the language of the topos generated by the identity morphisms on term types under the term-forming operations corresponding to post-composition by morphisms $f \in F$, post-composition by projections π_i, and pairing [5, Section VI.5]. Thus, terms can be viewed as composable chains of tuples of morphisms, generated from a finite set of morphisms and projections. For any given domain type $S_1 \times \cdots \times S_m$, we fix a variable set $\{x_1, \ldots, x_m\}$ where each x_i is interpreted by the corresponding projection. We will refer to such a term language as *finitely presented.* Equality of terms ("syntactic equality") is clearly efficient to decide.

3.1.2 Parameterized first-order formulas

We consider formulas with parameters, or metavariables. From the point of view of the language of the topos, parameters are just variables defined on a subobject of Ω. In addition to term types, the definition of the fragment includes a finite collection of subobjects $\Phi_i \subseteq \Omega$. For convenience, we assume that formulas are given with domain in the form $S_1 \times \cdots \times S_m \times \Phi_1 \times \cdots \times \Phi_k$. We fix a variable set $\{\alpha_1, \alpha_2, \ldots\}$ where each α_i is interpreted by the corresponding projection from Φ_i.

▶ **Definition 1.** *Let T be a finitely presented term language. The corresponding language of formulas $L_{T,S\times\Phi}$ is inductively defined by:*

$$\begin{array}{rcll}
\bot & \in & L_{T,S\times\Phi} & \\
\top & \in & L_{T,S\times\Phi} & \\
\alpha_i & \in & L_{T,S\times\Phi} & \textit{where } 1 \le i \le k \\
(f = g) & \in & L_{T,S\times\Phi} & \textit{where } f, g : S \to R \in T \textit{ for some } R \in \mathcal{E} \\
\phi_1 \wedge \phi_2 & \in & L_{T,S\times\Phi} & \textit{for } \phi_1, \phi_2 \in L_{T,S\times\Phi} \\
\phi_2 \vee \phi_2 & \in & L_{T,S\times\Phi} & \textit{for } \phi_1, \phi_2 \in L_{T,S\times\Phi} \\
\phi_2 \Rightarrow \phi_2 & \in & L_{T,S\times\Phi} & \textit{for } \phi_1, \phi_2 \in L_{T,S\times\Phi} \\
\exists x_{m+1}\, \psi & \in & L_{T,S\times\Phi} & \textit{for } \psi \in L_{T,S\times S_{m+1}\times\Phi} \\
\forall x_{m+1}\, \psi & \in & L_{T,S\times\Phi} & \textit{for } \psi \in L_{T,S\times S_{m+1}\times\Phi}
\end{array}$$

We define the language $L_T = \bigcup_{S\times\Phi} L_{T,S\times\Phi}$. For a given $\phi \in L_T$ denotes the set of subformulas of ϕ.

We will usually abbreviate $L_{T,S\times\Phi}$ to L_S when T and Φ are inferable from context or are arbitrary. When the term type language is generated by a single type, we will index L by the length of S instead.

3.2 Semantics

Our focus in this paper is on the problem of deciding when a formula in the language of the topos holds when interpreted "locally". One way to make this concept precise is in terms of a forcing relation, which is the basis the sheaf semantics.

3.2.1 Forcing relation

For an object C in a category $\mathcal{C}$, $\mathrm{t}(C)$ will denote the sieve $\bigcup_{D\in\mathcal{C}} \mathrm{Hom}(D, C)$. We first recall the definition of forcing in toposes of sheaves from [5, Section VI.7]:

▶ **Definition 2.** *Let $\mathcal{E}$ be a topos of sheaves with indexing category $\mathcal{C}$, and let $S \in \mathcal{E}$. Let ϕ be a formula defined on S in the language of $\mathcal{E}$. Let $X \in \mathcal{C}$, and let $s \in SX$. The* forcing relation *is defined by*

$$X \Vdash \phi\,(s) \quad \textit{iff} \quad s \in \{x \mid \phi(x)\}X$$

where

$$\begin{tikzcd}
\{x \mid \phi(x)\} \arrow[r] \arrow[d, phantom, "\subseteq" rotate=90] & 1 \arrow[d, "\mathrm{t}"] \\
S \arrow[r, "\phi"] & \Omega
\end{tikzcd}$$

is a pullback square.

We will refer to the element $s \in SX$ in Definition 2 as a *local assignment*. When we say "X forces ϕ with local assignment s" we mean that $X \Vdash \phi\,(s)$.

The forcing relation models a form of "truth definition", which is attributed to Kripke and Joyal, and which we also call the sheaf semantics. Furthermore, the truth definition may be usefully refined in light of specific features of the topos of sheaves in question.

3.2.2 Presheaf semantics

Next we present the *presheaf semantics* for a language fragment L (Definition 1). The following theorem is adapted from [5, Theorem VI.7.1]. It will be our main tool for proving the correctness of mapping reductions. We recall that for an element $s \in SX$ and a morphism $f : Y \to X$, the *restriction of s along f*, denoted by $s \cdot f$, is defined as $S(f)(s)$.

▶ **Theorem 3.** *Let $\mathcal{C}$ be a category, and let $\mathcal{E}$ be the topos of presheaves on $\mathcal{C}$. Let $S = S_1 \times \cdots \times S_m, S_{m+1} \in \mathcal{E}$. Let $\Phi = \Phi_1 \times \cdots \times \Phi_k \in \mathcal{E}$ with each $\Phi_i \subseteq \Omega$. Let $\phi_1, \phi_2 : S \times \Phi \to \Omega$ and $\psi : S \times S_{m+1} \times \Phi \to \Omega$ be formulas. Let $X \in \mathcal{C}$, let $s \in SX$, and let $h \in \Phi X$.*

$$\begin{array}{ll}
X \Vdash \top\,(s\,;h) & \\
X \nVdash \bot\,(s\,;h) & \\
X \Vdash (f = g)\,(s\,;h) & \textit{iff } f_X(s) = g_X(s) \\
X \Vdash \alpha_i\,(s\,;h) & \textit{iff } h_i = \mathrm{t}(X) \\
X \Vdash \phi_1 \wedge \phi_2\,(s\,;h) & \textit{iff } X \Vdash \phi_1\,(s\,;h) \textit{ and } X \Vdash \phi_2\,(s\,;h) \\
X \Vdash \phi_1 \vee \phi_2\,(s\,;h) & \textit{iff } X \Vdash \phi_1\,(s\,;h) \textit{ or } X \Vdash \phi_2\,(s\,;h) \\
X \Vdash \phi_1 \Rightarrow \phi_2\,(s\,;h) & \textit{iff for all } Y \in \mathcal{C} \textit{ and } t \in \mathrm{Hom}(Y, X), \\
& \quad Y \Vdash \phi_1\,(s \cdot t\,; h \cdot t) \textit{ implies } Y \Vdash \phi_2\,(s \cdot t\,; h \cdot t) \\
X \Vdash \exists x_{m+1}\,\psi\,(s\,;h) & \textit{iff there exists } s_{m+1} \in S_{m+1}X \textit{ such that} \\
& \quad X \Vdash \psi\,(s\,; s_{m+1}\,; h) \\
X \Vdash \forall x_{m+1}\,\psi\,(s\,;h) & \textit{iff for all } Y \in \mathcal{C}, t \in \mathrm{Hom}(Y, X), \textit{and } s_{m+1} \in S_{m+1}Y, \\
& \quad Y \Vdash \psi\,(s \cdot t\,; s_{m+1}\,; h \cdot t)
\end{array}$$

When $S_{m+1} = \mathrm{Hom}_{\mathcal{C}}(-, U)$ for some $U \in \mathcal{C}$ and $\mathcal{C}$ is Cartesian, the semantics of the universal quantifier simplifies to

$$X \Vdash \forall x_{m+1}\, \psi\,(s\,;h) \qquad \textit{iff } X \times U \Vdash \psi\,(s \cdot \pi_1\,; \pi_2\,; h \cdot \pi_1)$$

▶ Remark 4. We may extend Theorem 3 to formulas in the language of the topos of the form $\forall \alpha_1 \cdots \forall \alpha_k\, \phi$ where $\phi \in L$ and k is the maximal parameter index in ϕ:

$$\begin{aligned} X \Vdash \forall \alpha_1 \cdots \forall \alpha_k\, \phi\,(s) \quad & \text{iff for all } Y \in \mathcal{C}, t \in \mathrm{Hom}(Y, X), \text{ and } h \in \Phi_1 \times \cdots \times \Phi_k Y, \\ & \quad Y \Vdash \phi\,(s \cdot t\,; h) \end{aligned}$$

▶ **Corollary 5.** *In the context of Theorem 3, the semantics of the universal quantifier when $S_{m+1} = \mathrm{Hom}_{\mathcal{C}}(-, U)$ for some $U \in \mathcal{C}$ and $\mathcal{C}$ is Cartesian validates the following equivalence:*

$$X \Vdash \forall x_{m+1}\,(\psi_1 \vee \psi_2) \Leftrightarrow \forall x_{m+1}\,\psi_1 \vee \forall x_{m+1}\,\psi_2\,(x)$$

3.3 Languages of representable types

One natural way to obtain a finitely presented formula language is to restrict attention to sheaves of the form $S = \mathrm{Hom}(-, U)$ for a selection of $U \in \mathcal{C}$. Elements of S would just be morphisms in $\mathcal{C}$ from X to U. Thus, we may induce a formula language from a collection of morphisms from $\mathcal{C}$.

Coincidentally, elements of $SX = \mathrm{Hom}(X, U)$ then have a compositional structure analogous to that induced by a formal term system. Below, this coincidence will be taken further: the indexing category $\mathcal{C}$ will be presented as a quotiented term language, and the term language defining $\mathcal{C}$ will take the place of the term language used to define the formula language. Conveniently, restrictions will then be computed by term substitution.

3.4 Indexing category from a finite algebra

The indexing categories for cubical sets – "cube categories" – are readily and usefully definable as quotients of formal term languages.

Beyond encodability, another requirement for bringing a decision problem under the purview of complexity theory is that it be decidable. But an indexing category $\mathcal{C}$ may fail to have decidable equality of morphisms; and so even validity of the atomic formulas of the language may fail to be decidable. The following method of construction however ensures that the morphisms of $\mathcal{C}$ at least have decidable equality.[1]

3.4.1 Generation by a finite algebra

Fix a finite algebraic signature σ. Let $\mathcal{T}_{\sigma,n}$ be the term algebra of type σ over the variable set $\{x_1, \ldots, x_n\}$. We will abbreviate $\mathcal{T}_{\sigma,n}$ to $\mathcal{T}_n$. For $m \geq 0$, the m-fold (left-associated) product algebra $\mathcal{T}_n \times \cdots \times \mathcal{T}_n$ will be denoted by $\mathcal{T}_n^m$. The term algebra $\mathcal{T}$ is defined by $\mathcal{T} = \bigcup_n \mathcal{T}_n$.

For $s \in \mathcal{T}_m$ and $t \in \mathcal{T}_n^m$, the simultaneous substitution of the terms t_i for variables x_i in s will be denoted by $s[t_1/x_1, \ldots, t_m/x_m]$, by $s[t/x]$, or simply by st. Let $\mathcal{A}$ be a finite σ-algebra. The equational theory of $\mathcal{A}$ contains the universally quantified equations of terms satisfied by $\mathcal{A}$. The variety of this theory, denoted by $\mathrm{V}(\mathcal{A})$, contains the σ-algebras which satisfy this theory (i.e., its models). Examples of varieties generated by a finite algebra $\mathcal{A}$

[1] We are grateful to Alex Kruckman for directing our attention to the central role played by finite algebras in this area. See also [2].

include Boolean algebras, De Morgan algebras, (bounded) distributive lattices, (bounded) meet (join) semi-lattices, bi-pointed sets, pointed sets, and sets. All save Boolean algebras and sets have been used in the construction of indexing categories of cubical sets. Bounded distributive lattices have also been used to construct the indexing category in a variant definition of simplicial sets.

We may use the term algebra $\mathcal{T}_n$ to give a formal definition of a free algebra in $\mathrm{V}(\mathcal{A})$ generated by variables $x_1, \ldots, x_n$. We will denote these algebras by $\mathcal{F}_{\mathcal{A},n}$, or more briefly by $\mathcal{F}_n$. Given terms $r, s \in \mathcal{T}_n$, we say that r and s denote the same elements in $\mathcal{F}_n$ if and only if $r^{\mathcal{A}} = s^{\mathcal{A}}$ (i.e., their interpretation as term functions on $\mathcal{A}^n$ are equal). Recall that this relation respects term substitution: for $s \in \mathcal{T}_m$ and $t \in \mathcal{T}_n^m$, $s[t/x]^{\mathcal{A}} = s^{\mathcal{A}} \circ t^{\mathcal{A}}$.

3.4.2 Finite product theories

From here, we may use the free algebras $\mathcal{F}_n$ to give a formal definition of an indexing category $\mathcal{C}$. We will denote these categories by $\mathcal{C}_{\mathcal{A}}$, or more briefly by $\mathcal{C}$. The object set of $\mathcal{C}_{\mathcal{A}}$ contains a formal object X, and its remaining objects are formal n-fold (left-associated) products of X for $n \geq 0$. The set of morphisms from X^n to X^m is simply the (left-associated) product algebra $\mathcal{F}_n^m$. Thus, morphisms are denoted by m-tuples of terms in n-variables, and two morphisms denoted by tuples of terms r and s are equal if and only if their interpretations as term functions on $\mathcal{A}^n$ are equal. For $n \geq 0$, the identity morphism on X^n is $(x_1, \ldots, x_n)$. Composition of morphisms is given by composition of denoted term functions; which factors through the substitution of the representing tuples of terms.

The category $\mathcal{C}_{\mathcal{A}}$ is an example of a finite-product theory (or Lawvere theory). A well-known alternative view of this construction is as the category opposite to a chosen skeleton of the category of free algebras on a finite number of generators and homomorphisms. Note however that the language of a topos is interpreted by contravariant functors on $\mathcal{C}_{\mathcal{A}}$. This usage is different than the usage for which the concept of a finite-product theory was developed by Lawvere: the categorification of universal algebra. Models of this finite-product theory would be product-preserving functors on $\mathcal{C}_{\mathcal{A}}$, not arbitrary functors on $\mathcal{C}_{\mathcal{A}}^{op}$ satisfying sheaf conditions.

3.5 Cofibration formulas

From now on, we will assume that the formula language L is defined over a term algebra $\mathcal{T}$ corresponding to a finite algebra $\mathcal{A}$ which induces the indexing category $\mathcal{C}$, and that the term translation is the natural one. The letter c will denote a generic constant term in the signature of $\mathcal{A}$.

Standard cofibration formulas (i.e., those used in cubical type theories) constitute only a small fragment of L: specifically, the entailment connective $\Rightarrow$ and existential quantifier $\exists$ do not appear in cofibration formulas. Another restriction which appears concerns allowable atomic equations: in some cubical type theories, an equation may compare a given non-constant term f only to a constant.

▶ **Definition 6.** *We define several fragments of L by inductive generation:*

- *L_{cof} is generated by $\bot$, $\top$, $(f = g)$, $\wedge$, $\vee$, and $\forall$*
- *L_{coftc} is generated by $\bot$, $\top$, $(f = c)$, $\wedge$, $\vee$, and $\forall$*
- *L_{cofvc} is generated by $\bot$, $\top$, $(x_i = c)$, $\wedge$, $\vee$, and $\forall$*

Each fragment above also has a variant – L_{pcof}, L_{pcoftc}, L_{pcofvc} – whose generators also contain α_i.

A formula in L_{cof} is called a *cofibration formula*. The variants L_{coftc} and L_{cofvc} are called *term-constant* and *variable-constant*. The variants containing α_i are called *parameterized*. Although the focus of our study is on these fragments, it is natural to investigate related decision problems for the full language. In several cases, the methods we use below do apply more generally. In the future, the provisional definition above of what is a cofibration formula may merit revisiting.

3.6 Entailment problems

We are at last in position to define the decision problems we aim to classify in this paper.

▶ **Definition 7.** *Let $\mathcal{A}$ be a finite algebra with signature σ. Let $\mathcal{T}$ be the term algebra of type σ over variable set $\{x_1, x_2, \dots\}$, let $\mathcal{C}$ be the finite-product theory induced by $\mathcal{A}$, and let* X *be the generating object of $\mathcal{C}$. The* cofibration entailment problem for $\mathcal{A}$ *is the subset of $L_{cof} \times L_{cof}$ defined by*

$$\mathrm{COFENT} = \big\{(\phi_1, \phi_2) \mid \exists m \ \mathrm{X}^m \Vdash \phi_1 \Rightarrow \phi_2\,(x)\big\}$$

The parameterized cofibration entailment problem for $\mathcal{A}$ and Φ *is the subset of $L_{pcof} \times L_{pcof}$ defined by*

$$\mathrm{PCOFENT} = \big\{(\phi_1, \phi_2) \mid \exists m \ \mathrm{X}^m \Vdash \forall\alpha\,\big(\phi_1 \Rightarrow \phi_2\big)\,(x)\big\}$$

Lastly, for algebras $\mathcal{A}$ with at least one constant, we denote the term-constant *and* variable-constant *variants of each problem with the suffixes TC and VC.*

3.7 Problem characterization in terms of derivability

Given the proof-theoretic character of many presentations of cubical type theories, it is reasonable to wonder if there is an alternative, logical characterization of COFENT. Is there a suitable notion of derivability such that $\mathrm{COFENT} = \{(\phi_1, \phi_2) \mid \phi_1 \vdash \phi_2\}$?

For this discussion, let us consider the cube category corresponding to the theory of bounded distributive lattices. We know that the language of a topos can be viewed as an intuitionistic first-order logic. A natural first pass is intuitionistic derivability from this theory. This notion of derivability is sound with respect to the Kripke-Joyal semantics. Unsurprisingly, it is not complete. For example,

$$(x_1 \wedge x_2 = 0) \Rightarrow (x_1 = 0) \vee (x_2 = 0)$$

is valid in the language of the topos; but a corresponding derivation does not exist (not even classically: consider the smallest congruence on the free distributive lattice on two generators which identifies the meet of the generators and 0). Another example is Corollary 5, which is certainly not derivable. The notion of derivability above, it seems, would need to be significantly augmented with axioms to accommodate the structural peculiarities of categories of cubical sets as well as semantic properties specific to the choice of generating algebra and of language fragment.

However, even given a complete axiomatization, we do not expect a notion of derivability to yield an efficient-enough decision procedure to prove our main results. In practice, where this approach has been made to work, it remains the case that the smallest-known derivability proofs may be exponential in the length of the given instance. So instead, we ask, what form might a "short witness" of $\phi_1 \nvdash \phi_2$ take? The answer in logic traditionally has been a countermodel.

4 Results

To classify COFENT and related problems for various cube categories, we will first show uniformly that they are all coNP-hard by a reduction from the canonical coNP-hard problem UNSAT. Next we will show more-or-less on a case-by-case basis that they belong to coNP via efficient translations to the fragment of second-order logic ∀SO. Here we rely on Fagin's theorem, the foundational result in descriptive complexity theory (cf. [4, Section 9.2]). Note that the Cook-Levin theorem is an easy corollary of Fagin's theorem, and its typical proof contains an efficient translation to UNSAT (ibid.); so these translations may also be viewed as indirect reductions to UNSAT.

4.1 Reduction from UNSAT

▶ **Proposition 8.** *If σ_0 has constants 0 and 1 denoting distinct elements of $\mathcal{A}$, then the decision problem* COFENTVC *is* coNP*-hard.*

Proof. Let F be a formula of propositional logic in disjunctive normal form. Assume wlog that $\mathrm{Var}(F) \subseteq \{x_1, \dots, x_m\}$. Let $F' \in L_{cof,m}$ be the formula induced by mapping positive literals x_i to equations $(x_i = 1)$ and negative literals to equations $(x_i = 0)$. The edge cases: empty clauses are mapped to $\top$ while empty formulas are mapped to $\bot$. We claim that

$$F \in \mathrm{UNSAT} \quad \text{iff} \quad \mathrm{X}^m \Vdash F' \Rightarrow \bot\,(x)$$

First, we prove the case where F is a single clause C. Suppose that C is unsatisfiable. Then C contains literals x_i and $\neg x_i$ for some i. So $(x_i = 1), (x_i = 0) \in \mathrm{SF}(F')$. Let $s \in \mathcal{T}_n^m$. Note that $\mathrm{X}^n \Vdash C\,(s)$ implies both $s_i = 0$ and $s_i = 1$, which is a contradiction. So $\mathrm{X}^n \nVdash C\,(s)$, as desired. Conversely, suppose that C is satisfiable. Let $b \in \{0,1\}^m$ be a satisfying assignment (whereby $x_i \mapsto b_i$). Note that for $x_i \in \mathrm{Var}(C)$, $b_i = 0$ if and only if C contains $\neg x_i$. Hence, for $x_i \in \mathrm{Var}(C)$, $b_i = 0$ if and only if $(x_i = 0) \in \mathrm{SF}(C')$ and $b_i = 1$ if and only if $(x_i = 1) \in \mathrm{SF}(C')$. Therefore, $\mathrm{X}^0 \Vdash C'\,(b)$.

We finish with an induction on F. If F is the empty formula, then F is unsatisfiable while $\mathrm{X}^m \Vdash \bot \Rightarrow \bot\,(x)$. Now suppose $F = F_1 \vee F_2$. On the one hand, observe that $F_1 \vee F_2$ is unsatisfiable if and only if F_1 and F_2 are unsatisfiable. On the other hand, $\mathrm{X}^m \Vdash F_1' \vee F_2' \Rightarrow \bot\,(x)$ if and only if $\mathrm{X}^m \Vdash F_1' \Rightarrow \bot\,(x)$ and $\mathrm{X}^m \Vdash F_2' \Rightarrow \bot\,(x)$. The conclusion follows by induction. ◀

▶ **Corollary 9.** *If σ_0 has constants 0 and 1 denoting distinct elements of $\mathcal{A}$, then any of the decision problems defined in Definition 7 is* coNP*-hard.*

4.2 Reduction to ∀SO

4.2.1 Translation of L_{cof} to second-order logic

Consider the equation $(f = g) \in L$. We would like to transform this equation into the sentence $\forall x\,\big(R(x) \to f(x) = g(x)\big)$. Informally, this sentence holds if the equation holds for all m-tuples satisfying R. Given $\phi \in L_{cof}$, the purpose of the following transformation is to replace equations $(f = g) \in \mathrm{SF}(\phi)$ with such formulas.

Below, $\vDash$ denotes the conventional formula satisfaction relation from mathematical logic for second-order logic ([4, Section 7.1]). The term function interpreting a tuple of terms $s \in \mathcal{T}_n^m$ will be denoted by $s^{\mathcal{A}} : \mathcal{A}^n \to \mathcal{A}^m$.

▶ **Definition 10.** *Let $m, n \geq 0$, and let $\phi \in L_{cof,m}$. We define the second-order formula ϕ'_n with a free second-order variable R of arity n as follows:*

$$\phi'_n = \begin{cases} \bot & \phi = \bot \\ \top & \phi = \top \\ \forall x \left(R(x) \to f(x) = g(x)\right) & \phi = (f = g) \\ \phi'_{1,n} \wedge \phi'_{2,n} & \phi = \phi_1 \wedge \phi_2 \\ \phi'_{1,n} \vee \phi'_{2,n} & \phi = \phi_1 \vee \phi_2 \\ \forall x_{m+1}\, \psi'_n & \phi = \forall x_{m+1}\, \psi \end{cases}$$

When $n = m$, we will abbreviate ϕ'_n to ϕ'. Note that ϕ' has no free first-order variables.

▶ Remark 11. The transformation in Definition 10 admits a reduction of the universal quantifier:

$$\mathcal{A} \vDash \forall x_{m+1}\, \psi'_m\,(M) \quad \text{iff} \quad \mathcal{A} \vDash \psi'_{m+1}\,(M \times \mathcal{A})$$

4.2.2 First schema

As it turns out, if a formula in the fragment L_{cof} is forced by a given index with local assignment s, it is forced by any other index and local assignment provided it has the same term-functional image as s. We now present a reduction schema which takes advantage of this fact to obtain a bound on the number of tuples of terms which must be considered to force an entailment.

▶ **Lemma 12** (Term image invariance). *Let $\phi \in L_{cof,m}$. Let $s \in \mathcal{T}^m_n$ and let $t \in \mathcal{T}^m_p$. If $\mathrm{img}(s^{\mathcal{A}}) = \mathrm{img}(t^{\mathcal{A}})$ and $\mathrm{X}^n \Vdash \phi\,(s)$, then $\mathrm{X}^p \Vdash \phi\,(t)$.*

Proof. Induction on ϕ. The only interesting case is when $\phi = (f = g)$. Then it suffices to prove that if $fs^{\mathcal{A}} = gs^{\mathcal{A}}$, then $ft^{\mathcal{A}} = gt^{\mathcal{A}}$. Let $a \in \mathcal{A}^p$. By assumption, there exists $b \in \mathcal{A}^n$ such that $s^{\mathcal{A}}(b) = t^{\mathcal{A}}(a)$. But then

$$ft^{\mathcal{A}}(a) \quad = \quad fs^{\mathcal{A}}(b) \quad = \quad gs^{\mathcal{A}}(b) \quad = \quad gt^{\mathcal{A}}(a)$$ ◀

As a consequence, forcing of a cofibration formula ϕ with a local assignment s is reducible to checking that ϕ' holds with respect to the term-functional image of s given an assignment to the second-order variable R.

▶ **Lemma 13.** *Let $m, n \geq 0$, let $\phi \in L_{cof,m}$, and let $s \in \mathcal{T}^m_n$. Let $M = \mathrm{img}(s^{\mathcal{A}})$. Then*

$$X^n \Vdash \phi\,(s) \quad \textit{iff} \quad \mathcal{A} \vDash \phi'(M)$$

Proof. Induction on ϕ. We omit the cases which follow directly by induction.

- Suppose $\phi = \bot$ or $\phi = \top$. Immediate.
- Suppose $\phi = (f = g)$. Fix $a \in \mathcal{A}^m$, and suppose that $a \in M$. Let $b \in \mathcal{A}^n$ satisfy $s^{\mathcal{A}}(b) = a$. By assumption, $X^n \Vdash (f = g)\,(s)$, and so $fs^{\mathcal{A}} = gs^{\mathcal{A}}$. Hence $f^{\mathcal{A}}(a) = fs^{\mathcal{A}}(b) = gs^{\mathcal{A}}(b) = g^{\mathcal{A}}(a)$.
  For the converse, we need to show that $fs^{\mathcal{A}} = gs^{\mathcal{A}}$. Fix $b \in \mathcal{A}^n$. Because $s^{\mathcal{A}}(b) \in M$, $fs^{\mathcal{A}}(b) = gs^{\mathcal{A}}(b)$ by assumption.
- Suppose $\phi = \forall x_{m+1}\, \psi$. Recall that $\mathrm{X}^n \Vdash \forall x_{m+1}\, \psi\,(s)$ if and only if $\mathrm{X}^{n+1} \Vdash \psi\,(s, x_{n+1})$; while the induction hypothesis is that

  $$X^{n+1} \Vdash \psi\,(s, x_{n+1}) \quad \text{iff} \quad \mathcal{A} \vDash \psi'_{m+1}(M \times \mathcal{A})$$

  The conclusion follows by Remark 11. ◀

In the event that the cofibration ϕ is a conjunction of atomic formulas, we can dispense with the translation (Definition 10) altogether:

▶ **Corollary 14.** *Let $m, n \geq 0$, and let $\phi \in L_{cof,m}$ be a conjunction of atomic formulas. Let $s \in \mathcal{T}_n^m$, and let $M = \operatorname{img}(s^{\mathcal{A}})$. Then*

$$X^n \Vdash \phi\,(s) \quad \textit{iff} \quad \textit{for all } a \in M, \mathcal{A} \vDash \phi(a)$$

To extend the translation (Definition 10) of formulas to entailment instances, we must reduce the quantification over all m-tuples of terms. Term image invariance allows us to reduce this to quantification over all term-functional images included by $\mathcal{A}^m$. Whence stems the decidability of the problem as well as its inefficiency. Notice that while belonging to a term-functional image of a given $s \in \mathcal{T}^m$ defines an m-ary relation over $\mathcal{A}$, being a term-functional image defines a property of m-ary relations over $\mathcal{A}$, one which we may try to capture by a second-order formula:

▶ **Definition 15.** *Let $m \geq 0$. Let χ_m be a formula in the second-order language over the signature of $\mathcal{A}$ whose only free variable is the second-order variable R with arity m. We say that χ_m* characterizes images for $\mathcal{A}$-term functions *whenever the following conditions are equivalent for all $M \subseteq \mathcal{A}^m$:*
1. *$\mathcal{A} \vDash \chi_m\,(M)$*
2. *there exist $n \geq 0$ and $s \in \mathcal{T}_n^m$ such that $M = \operatorname{img}(s^{\mathcal{A}})$*

Indeed, a second-order characterization of term-functional images is all that is needed for this reduction schema.

▶ **Theorem 16.** *Let $m \geq 0$, let $\phi_1, \phi_2 \in L_{cof,m}$. Let χ_m be a formula which characterizes images for $\mathcal{A}$-term functions. Then*

$$\mathrm{X}^m \Vdash \phi_1 \Rightarrow \phi_2\,(x) \quad \textit{iff} \quad \mathcal{A} \vDash \forall R\left(\chi_m \to \phi_1' \to \phi_2'\right)$$

Proof. Fix $M \subseteq \mathcal{A}^m$ and $a \in \mathcal{A}^m$. Suppose that $\mathcal{A} \vDash \chi_m\,(M)$. Fix $n \geq 0$ and $s \in \mathcal{T}_n^m$ according to Definition 15. Specializing the assumption, we have that $\mathrm{X}^n \Vdash \phi_1\,(s)$ implies $\mathrm{X}^n \Vdash \phi_2\,(s)$. Hence, by Lemma 13, we have that $\mathcal{A} \vDash \phi_1'\,(M)$ implies $\mathcal{A} \vDash \phi_2'\,(M)$.

For the converse, fix $n \geq 0$ and $s \in \mathcal{T}_n^m$. Let $M = \operatorname{img}(s^{\mathcal{A}})$. According to Definition 15, we have that $\mathcal{A} \vDash \chi_m\,(M)$. And by assumption, $\mathcal{A} \vDash \phi_1'\,(M)$ implies $\mathcal{A} \vDash \phi_2'\,(M)$. So, by Lemma 13, $\mathrm{X}^n \Vdash \phi_1'\,(s)$ implies $\mathrm{X}^n \Vdash \phi_2'\,(s)$. ◀

4.2.3 Second schema

We will now present a reduction schema which applies to generating algebras with language fragments which have the property of *local factorization* (Definition 17). To abstract over the choice of language fragment, we let the variable γ denote a generic element of $\{cof, coftc, cofvc\}$.

▶ **Definition 17.** *Let $m \geq 0$. A formula $\rho \in L_{\gamma,m}$ is* locally prime *if ρ does not entail $\bot$, and for all $\phi_1, \phi_2 \in L_{\gamma,m}$, if ρ entails $\phi_1 \vee \phi_2$, then either ρ entails ϕ_1 or ρ entails ϕ_2. An algebra $\mathcal{A}$ paired with a language fragment L_γ has* local factorization *if for all m and formulas $\phi \in L_{\gamma,m}$, ϕ is equivalent to the disjunction (in $L_{\gamma,m}$) of locally prime formulas which entail ϕ.*

We assume for the remainder of this section that $(\mathcal{A}, L_\gamma)$ has local factorization.

▶ **Lemma 18.** *Let $m \geq 0$, and let $\phi, \rho \in L_{\gamma,m}$. Let $M = \{a \in \mathcal{A}^m \mid \mathcal{A} \vDash \rho\,(a)\}$. Assume that for all terms $f, g \in \mathcal{T}_m$,*

$$\mathrm{X}^m \Vdash \rho \Rightarrow (f = g)\,(x) \quad \textit{iff} \quad \mathcal{A} \vDash (f = g)'\,(M)$$

If ρ is locally prime, then

$$\mathrm{X}^m \Vdash \rho \Rightarrow \phi\,(x) \quad \textit{iff} \quad \mathcal{A} \vDash \phi'\,(M)$$

Proof. Induction on ϕ.

- Suppose $\phi = \bot$ or $\phi = \top$. Trivial.
- Suppose $\phi = (f = g)$. By assumption.
- Suppose $\phi = \phi_1 \wedge \phi_2$. Directly by induction.
- Suppose $\phi = \phi_1 \vee \phi_2$. Suppose that $\mathrm{X}^m \Vdash \rho \Rightarrow \phi_1 \vee \phi_2\,(x)$. Because ρ is $\vee$-prime, either $\mathrm{X}^m \Vdash \rho \Rightarrow \phi_1\,(x)$ or $\mathrm{X}^m \Vdash \rho \Rightarrow \phi_2\,(x)$. The conclusion follows by induction. The converse also follows directly by induction.
- Suppose $\phi = \forall x_{m+1}\,\psi$. Recall that $\mathrm{X}^m \Vdash \rho \Rightarrow \forall x_{m+1}\,\psi\,(x)$ if and only if $\mathrm{X}^{m+1} \Vdash \rho \Rightarrow \psi\,(x)$; while the induction hypothesis is that

  $$X^{n+1} \Vdash \rho \Rightarrow \psi\,(x) \quad \text{iff} \quad \mathcal{A} \vDash \psi'_{m+1}(M \times \mathcal{A})$$

  The conclusion follows by Remark 11. ◀

▶ **Lemma 19.** *Let $m \geq 0$, and let $\phi_1, \phi_2 \in L_{\gamma,m}$. Let P be the set of locally prime formulas in $L_{\gamma,m}$. Then*

$$\begin{aligned}&\mathrm{X}^m \Vdash \phi_1 \Rightarrow \phi_2\,(x) \quad \textit{iff} \\ &\textit{for all } \rho \in P,\ \mathrm{X}^m \Vdash \rho \Rightarrow \phi_1\,(x) \textit{ implies } \mathrm{X}^m \Vdash \rho \Rightarrow \phi_2\,(x)\end{aligned}$$

Proof. The forward direction follows directly by the semantics. For the converse, let $s \in \mathcal{T}^m_n$ and suppose that $\mathrm{X}^n \Vdash \phi_1\,(s)$. Let $P' = \{\rho \in P \mid \mathrm{X}^m \Vdash \rho \Rightarrow \phi_1\,(x)\}$. From local factorization, we have

$$\mathrm{X}^m \Vdash \phi_1 \Leftrightarrow \bigvee_{\rho \in P'} \rho\,(x)$$

Fix $\rho \in P'$ such that $\mathrm{X}^n \Vdash \rho\,(s)$. By definition of P' and hypothesis, $\mathrm{X}^m \Vdash \rho \Rightarrow \phi_2\,(x)$. In particular, $\mathrm{X}^m \Vdash \phi_2\,(s)$. ◀

▶ **Definition 20.** *Let $m \geq 0$. Let P be the set of locally prime formulas in $L_{\gamma,m}$. Let χ_m be a formula in the second-order language over the signature of $\mathcal{A}$ whose only free variable is the second-order variable R with arity m. We say that χ_m* defines P (the locally prime formulas in $L_{\gamma,m}$) *when the following are equivalent for all $M \subseteq \mathcal{A}^m$:*

- $\mathcal{A} \vDash \chi_m\,(M)$
- *there exists $\rho \in P$ such that $M = \{a \in \mathcal{A}^m \mid \mathcal{A} \vDash \rho\,(a)\}$*

▶ **Theorem 21.** *Let $m \geq 0$, and let $\phi_1, \phi_2 \in L_{\gamma,m}$. Let P be the set of locally prime formulas in $L_{\gamma,m}$. Let the formula χ_m define P (Definition 20). If for all $\rho \in P$, and for all terms $f, g \in \mathcal{T}_m$,*

$$\mathrm{X}^m \Vdash \rho \Rightarrow (f = g)\,(x) \quad \textit{iff} \quad \mathcal{A} \vDash (f = g)'\,(M)$$

then

$$\mathrm{X}^m \Vdash \phi_1 \Rightarrow \phi_2\,(x) \quad \textit{iff} \quad \mathcal{A} \vDash \forall R\,(\chi_m \to \phi'_1 \to \phi'_2)$$

Proof. By Lemma 19, it suffices to show that

$$\begin{array}{l} \text{for all } \rho \in P,\ \mathrm{X}^m \Vdash \rho \Rightarrow \phi_1\,(x) \text{ implies } \mathrm{X}^m \Vdash \rho \Rightarrow \phi_2\,(x) \\ \text{iff} \quad \mathcal{A} \vDash \forall R\,(\chi_m \to \phi'_1 \to \phi'_2) \end{array}$$

Let $M \subseteq \mathcal{A}^m$, and suppose that $\mathcal{A} \vDash \chi_m\,(M)$ and $\mathcal{A} \vDash \phi'_1\,(M)$. Then, by Lemma 18, there exists $\rho \in P$ such that $\mathrm{X}^m \Vdash \rho \Rightarrow \phi_1\,(x)$; and by assumption, $\mathcal{A} \vDash \phi'_2\,(M)$.

For the converse, suppose $\rho \in P$, and that $\mathrm{X}^m \Vdash \rho \Rightarrow \phi_1\,(x)$. Let $M = \{a \in \mathcal{A}^m \mid \mathcal{A} \vDash \rho\,(a)\}$. Then $\mathcal{A} \vDash \chi_m\,(M)$, and by Lemma 18, $\mathcal{A} \vDash \phi'_1\,(M)$; and by assumption, $\mathrm{X}^m \Vdash \rho \Rightarrow \phi_2\,(x)$. ◀

4.3 Parameter elimination

In this section we will prove the following proposition, which will allow us to transport a classification of COFENT to PCOFENT:

▶ **Proposition 22.** *There exist efficient mapping reductions from* PCOFENT *to* COFENT*, from* PCOFENTTC *to* COFENTTC*, and from* PCOFENTVC *to* COFENTVC.

Proof. Corollaries 24 and 29 ◀

For the remainder of this subsection, let $\mathrm{B} \subseteq \Omega$ be defined by $\mathrm{B}\,C = \{\emptyset, \mathrm{t}(C)\}$ for all $C \in \mathcal{C}$. Let Φ be an arbitrary subobject satisfying the constraint $\mathrm{B} \subseteq \Phi \subseteq \Omega$.

4.3.1 Bivalent parameters

▶ **Lemma 23.** *Let $\phi \in L_{pcof,m}$. Let k be the maximal parameter index of ϕ. Let $n \geq 0$, let $s \in \mathrm{Hom}_{\mathcal{C}}(\mathrm{X}^n, \mathrm{X}^m)$, and let $h \in \Phi\,\mathrm{X}^n$. Define $b_h \in \mathrm{B}^k\,\mathrm{X}^n$ by*

$$b_{h,i} = \begin{cases} h_i & h_i = \mathrm{t}(\mathrm{X}^n) \\ \emptyset & \textit{otherwise} \end{cases}$$

Then $\mathrm{X}^n \Vdash \phi\,(s\,;b_h)$ if and only if $\mathrm{X}^n \Vdash \phi\,(s\,;h)$.

Proof. Induction on ϕ. When $\phi = \alpha_i$, we have

$$\begin{array}{lll} \mathrm{X}^n \Vdash \alpha_i\,(s\,;b_h) & \text{iff} & b_{h,i} = \mathrm{t}(\mathrm{X}^n) \\ & \text{iff} & h_i = \mathrm{t}(\mathrm{X}^n) \\ & \text{iff} & \mathrm{X}^n \Vdash \alpha_i\,(s\,;h) \end{array}$$

When ϕ is otherwise atomic, then $\mathrm{X}^n \Vdash \phi\,(s\,;b_h)$ if and only if $\mathrm{X}^n \Vdash \phi\,(s)$; and likewise for $\mathrm{X}^n \Vdash \phi\,(s\,;h)$. The induction cases are straightforward. ◀

▶ **Corollary 24.** *Let $\phi_1, \phi_2 \in L_{pcof,m}$. Let k be the maximal parameter index of $\phi_1 \Rightarrow \phi_2$. Then*

$$\mathrm{X}^m \Vdash \forall\alpha^\Phi\,\big(\phi_1 \Rightarrow \phi_2\big)\,(x) \quad \textit{iff} \quad \mathrm{X}^m \Vdash \forall\alpha^B\,\big(\phi_1 \Rightarrow \phi_2\big)\,(x)$$

Proof. The forward direction is trivial. For the converse, fix $n \geq 0$, $s \in \mathrm{Hom}_{\mathcal{C}}(\mathrm{X}^n, \mathrm{X}^m)$, and $h \in \Phi^k\,\mathrm{X}^n$. Our goal is to show that $\mathrm{X}^n \Vdash \phi_1 \Rightarrow \phi_2\,(s\,;h)$. Hence, fix $p \geq 0$ and $t \in \mathrm{Hom}_{\mathcal{C}}(\mathrm{X}^p, \mathrm{X}^n)$, and suppose that $\mathrm{X}^p \Vdash \phi_1\,(st\,;f \cdot t)$. We must show that $\mathrm{X}^p \Vdash \phi_2\,(st\,;f \cdot t)$. By Lemma 23, it suffices to check that $\mathrm{X}^p \Vdash \phi_2\,(st\,;b_{f \cdot t})$; and by assumption, that $\mathrm{X}^p \Vdash \phi_1\,(st\,;b_{f \cdot t})$; which follows again from Lemma 23. ◀

4.3.2 Formula lifting

Next we introduce a piece of technical machinery to weaken a formula in L_m: that is, add to the list of available free variables, irrespective of their non-occurrence. Because variables are bound in order of decreasing index, dummy free variables should be inserted before existing variables.

▶ **Definition 25.** *Let $l, m \geq 0$, and let $\phi \in L_m$. We define $\mathrm{lift}_{l,m} : L_m \to L_{l+m}$ by*

$$\mathrm{lift}_{l,m}(\phi) = \begin{cases} \bot & \phi = \bot \\ \top & \phi = \top \\ (f = g)[x_{l+1}/x_1, \ldots, x_{l+m}/x_m] & \phi = (f = g) \\ \alpha_i & \phi = \alpha_i \\ \mathrm{lift}_{l,m}(\phi_1) \wedge \mathrm{lift}_{l,m}(\phi_2) & \phi = \phi_1 \wedge \phi_2 \\ \mathrm{lift}_{l,m}(\phi_1) \vee \mathrm{lift}_{l,m}(\phi_2) & \phi = \phi_1 \vee \phi_2 \\ \mathrm{lift}_{l,m}(\phi_1) \Rightarrow \mathrm{lift}_{l,m}(\phi_2) & \phi = \phi_1 \Rightarrow \phi_2 \\ \exists x_{l+m+1}\ \mathrm{lift}_{l,m+1}(\psi) & \phi = \exists x_{m+1}\, \psi \\ \forall x_{l+m+1}\ \mathrm{lift}_{l,m+1}(\psi) & \phi = \forall x_{m+1}\, \psi \end{cases}$$

▶ **Proposition 26.** *Let $k, l, m \geq 0$. Let $\phi \in L_m$ with maximal index k. Let $\Phi = \Phi_1 \times \cdots \times \Phi_k$, where $\Phi_i \subseteq \Omega$ for all i. Let $R = R_1 \times \cdots \times R_l$ and let $S = S_1 \times \cdots \times S_m$. Let $X \in \mathcal{C}$, let $h \in \Phi X$, let $r \in RX$, and let $s \in SX$. Then $X \Vdash \phi\,(s\,;h)$ if and only if $X \Vdash \mathrm{lift}(\phi)\,(r\,;s\,;h)$.*

4.3.3 Grounding

In Lemmas 27 and 28 and Corollary 29 below, given $k \geq 0$, let $\epsilon \in L_{2k}^k$ be the k-tuple of equations defined so that $\epsilon_i = (x_i = x_{k+i})$.

▶ **Lemma 27.** *Let $\phi \in L_{pcof,m}$. Let k be the maximal parameter index of ϕ. Let $n, p \geq 0$, let $s \in \mathrm{Hom}_{\mathcal{C}}(\mathrm{X}^n, \mathrm{X}^m)$, and let $t \in \mathrm{Hom}_{\mathcal{C}}(\mathrm{X}^p, \mathrm{X}^{k+n})$. Define $b_t \in \mathrm{B}^k\,\mathrm{X}^p$ by*

$$b_{t,i} = \begin{cases} \mathrm{t}(\mathrm{X}^p) & t_i = 1 \\ \emptyset & \textit{otherwise} \end{cases}$$

The following are equivalent:

- $\mathrm{X}^p \Vdash \phi\,(s(t_{k+1}, \ldots, t_{k+n})\,; b_t)$
- $\mathrm{X}^p \Vdash \mathrm{lift}(\phi)\,\big[\epsilon/\alpha\big]\,(t_1, \ldots, t_k\,; 1, \ldots, 1\,; s(t_{k+1}, \ldots, t_{k+n}))$

Proof. Induction on ϕ. When $\phi = \alpha_i$, we have

$$\begin{aligned} &\mathrm{X}^p \Vdash \alpha_i\,(s(t_{k+1}, \ldots, t_{k+n})\,; b_t) \\ &\text{iff } b_{t,i} = \mathrm{t}(\mathrm{X}^n) \\ &\text{iff } t_i = 1 \\ &\text{iff } \mathrm{X}^p \Vdash (x_i = x_{k+i})\,(t_1, \ldots, t_k\,; 1, \ldots, 1\,; s(t_{k+1}, \ldots, t_{k+n})) \end{aligned}$$

When ϕ is otherwise atomic, then

$$\mathrm{X}^p \Vdash \phi\,(s(t_{k+1}, \ldots, t_{k+n})\,; b_t) \quad \text{iff} \quad \mathrm{X}^p \Vdash \phi\,(s(t_{k+1}, \ldots, t_{k+n}))$$

This holds, by Proposition 26, if and only if

$$\mathrm{X}^p \Vdash \mathrm{lift}(\phi)\,\big[\epsilon/\alpha\big]\,(t_1, \ldots, t_k\,; 1, \ldots, 1\,; s(t_{k+1}, \ldots, t_{k+n}))$$

The induction cases are straightforward. ◀

▶ **Lemma 28.** *Let $\phi \in L_{pcof,m}$. Let k be the maximal parameter index of ϕ. Let $n \geq 0$, let $s \in \mathrm{Hom}_{\mathcal{C}}(\mathrm{X}^n, \mathrm{X}^m)$, and let $b \in \mathrm{B}^k\,\mathrm{X}^n$. Define $a_b \in \mathrm{B}^k\,\mathrm{X}^n$ by*

$$a_{b,i} = \begin{cases} 1 & b_i = \mathrm{t}(\mathrm{X}^n) \\ 0 & \textit{otherwise} \end{cases}$$

Then

$$\mathrm{X}^n \Vdash \phi\,(s\,;b) \quad \textit{iff} \quad \mathrm{X}^n \Vdash \mathrm{lift}(\phi)\big[\epsilon/\alpha\big]\,(a_b\,;1,\ldots,1\,;s)$$

Proof. Induction on ϕ. When $\phi = \alpha_i$, we have

$$\begin{aligned} \mathrm{X}^n \Vdash \alpha_i\,(s\,;b) \quad &\text{iff} \quad b_i = \mathrm{t}(\mathrm{X}^n) \\ &\text{iff} \quad a_{b,i} = 1 \\ &\text{iff} \quad \mathrm{X}^n \Vdash (x_i = x_{k+i})\,(a_b\,;1,\ldots,1\,;s) \end{aligned}$$

When ϕ is otherwise atomic, then $\mathrm{X}^n \Vdash \phi\,(s\,;b)$ if and only if $\mathrm{X}^n \Vdash \phi\,(s)$; which holds, by Proposition 26, if and only if $\mathrm{X}^n \Vdash \mathrm{lift}(\phi)[\epsilon/\alpha]\,(a_b\,;1,\ldots,1\,;s)$. The induction cases are straightforward. ◀

▶ **Corollary 29.** *Let $\phi_1, \phi_2 \in L_{pcof,m}$. Let k be the maximal parameter index of $\phi_1 \Rightarrow \phi_2$. The following are equivalent:*

- $\mathrm{X}^m \Vdash \forall\alpha^B\,\big(\phi_1 \Rightarrow \phi_2\big)\,(x)$
- $\mathrm{X}^{k+m} \Vdash \mathrm{lift}(\phi_1 \Rightarrow \phi_2)\big[\epsilon/\alpha\big]\,(x_1,\ldots,x_k\,;1,\ldots,1\,;x_{k+1},\ldots,x_{k+m})$

Proof. Fix $n \geq 0$ and $s \in \mathrm{Hom}_{\mathcal{C}}(\mathrm{X}^n, \mathrm{X}^{k+m})$, and suppose that

$$\mathrm{X}^n \Vdash \mathrm{lift}(\phi_1)\big[\epsilon/\alpha\big]\,(s_1,\ldots,s_k\,;1,\ldots,1\,;s_{k+1},\ldots,s_{k+m})$$

By Lemma 27 and by assumption, $\mathrm{X}^n \Vdash \phi_2\,(s_{k+1},\ldots,s_{k+m}\,;b_t)$. And again by Lemma 27,

$$\mathrm{X}^n \Vdash \mathrm{lift}(\phi_2)\big[\epsilon/\alpha\big]\,(s_1,\ldots,s_k\,;1,\ldots,1\,;s_{k+1},\ldots,s_{k+m})$$

For the converse, fix $n \geq 0$, $s \in \mathrm{Hom}_{\mathcal{C}}(\mathrm{X}^n, \mathrm{X}^m)$, and $b \in \mathrm{B}^k\,\mathrm{X}^n$. Our goal is to check that $\mathrm{X}^n \Vdash \phi_1 \Rightarrow \phi_2\,(s\,;b)$. So fix $p \geq 0$ and $t \in \mathrm{Hom}_{\mathcal{C}}(\mathrm{X}^p, \mathrm{X}^n)$, and suppose that $\mathrm{X}^p \Vdash \phi_1\,(st\,;b\cdot t)$. By Lemma 28 and by assumption,

$$\mathrm{X}^p \Vdash \mathrm{lift}(\phi_2)\big[\epsilon/\alpha\big]\,(a_{b\cdot t}\,;1,\ldots,1\,;st)$$

So by Lemma 28, $\mathrm{X}^p \Vdash \phi_2\,(st\,;b\cdot t)$. ◀

4.4 Applications

To apply the first reduction schema to a concrete algebra $\mathcal{A}$, it suffices to supply a formula χ (Definition 15) characterizing local images for $\mathcal{A}$. To apply the second reduction schema to a concrete algebra $\mathcal{A}$ and language L_γ, it suffices to supply a formula χ (Definition 20) which defines the locally prime formulas of L_γ.

For either schema, if the prenex normal form of χ contains no universal second-order quantifiers, and it is efficiently computable from m, then the sentence checked in Theorem 16 is in ∀SO; and by Fagin's theorem, the corresponding model-checking problem is in coNP (cf. [4, Section 9.2]). This is the case for all of the following applications.

Below, $\leq$ will denote the standard ordering on the set $\{0,1\}$, as well as this ordering extended component-wise to $\{0,1\}^n$. We will denote the minimal and maximal elements of the poset $\{0,1\}^n$ also by 0 and 1.

4.4.1 Bipointed sets

For a theory of bipointed sets, the signature will be $\sigma_0 = \{0,1\}$. We take the generating algebra $\mathcal{A}$ to be $\{0,1\}$ with the constants interpreted as $0^{\mathcal{A}} = 0$ and $1^{\mathcal{A}} = 1$. When needed, we will annotate constructions associated to this algebra with *bp*.

We apply the first schema. The following second order formula encodes each term function $s_i^{\mathcal{A}} : \mathcal{A}^m \to \mathcal{A}$ as a second-order variable T^{m+1}: the first conjunct ensures that T encodes either a projection or a constant, and the second conjunct ensures that R is indeed the image of functions encoded by the T_i.

▶ **Definition 30.** *Let $m \geq 0$. Define χ_m (for bp terms) to be the formula*

$$\begin{aligned}\exists T_1^{m+1} \ldots \exists T_m^{m+1} \Big(\bigwedge_i \big(\bigvee_j \forall y \forall x \left(T_i(y\,;x) \ \leftrightarrow \ x = y_j\right) \\ \vee \ \forall y \forall x \left(T_i(y\,;x) \ \leftrightarrow \ x = 0\right) \\ \vee \ \forall y \forall x \left(T_i(y\,;x) \ \leftrightarrow \ x = 1\right)\big) \\ \wedge \forall x \left(R(x) \ \leftrightarrow \ \exists y \left(\bigwedge_i T_i(y\,;x_i)\right)\right)\Big)\end{aligned}$$

▶ Remark 31. Let $s \in \mathcal{T}_n^m$ be an m-tuple of bipointed-set terms. There exist indices $i_1 < i_2 < \cdots < i_m$ in $\{1,\ldots,n\}$ and $s' \in \mathcal{T}_m^m$ such that $s = s'[x_{i_1}/x_1,\ldots,x_{i_m}/x_m]$. In particular, for all $n \geq 0$ and $s \in \mathcal{T}_n^m$, there exists $s' \in \mathcal{T}_m^m$ such that $\mathrm{img}(s'^{\mathcal{A}}) = \mathrm{img}(s^{\mathcal{A}})$.

▶ **Proposition 32.** *Let $m \geq 0$. The formula χ_m in Definition 30 characterizes images of bp term functions.*

▶ **Corollary 33.** *The decision problem* $\mathrm{PCOFENT}_{bp}$ *is* coNP*-complete.*

4.4.2 Distributive lattices

For the theory of bounded distributive lattices, the signature will be $\sigma_0 = \{0,1\}$ and $\sigma_2 = \{\wedge,\vee\}$. We take the generating algebra $\mathcal{A}$ to be $\{0,1\}$ with the constants interpreted as $0^{\mathcal{A}} = 0$ and $1^{\mathcal{A}} = 1$, and with the basic operations defined as $\wedge^{\mathcal{A}} = \min$ and $\vee^{\mathcal{A}} = \max$. When needed, we will annotate constructions associated to this algebra with *dl*.

▶ **Lemma 34.**

$$\{s^{\mathcal{A}} : \mathcal{A}^n \to \mathcal{A} \mid s \in \mathcal{T}_n\} \quad \cong \quad \{f : \mathcal{A}^n \to \mathcal{A} \mid f \textit{ is monotone } \}$$

Proof. Let $s \in \mathcal{T}_n$. That $s^{\mathcal{A}}$ is monotone follows from the fact that the constant functions, the projections, min and max are all monotone.

Conversely, let f be a monotone function. There are many ways to define $s \in \mathcal{T}_n$ so that $s^{\mathcal{A}} = f$, and here is one way. Define s by

$$s = \bigvee_{f(a)=1} \bigwedge_{a_k=1} x_k$$

Suppose $f(a) = 1$. Since $\left(\bigwedge_{a_k=1} x_k\right)^{\mathcal{A}}(a) = 1$, $s^{\mathcal{A}}(a) = 1$. Conversely, suppose that $s^{\mathcal{A}}(a) = 1$. Then there exists a' such that $f(a') = 1$ and $(\bigwedge_{a'_k=1} x_k)^{\mathcal{A}}(a) = 1$. Hence, $a' \leq a$. But f is monotone, so $f(a) = 1$. ◀

▶ **Lemma 35.** *Let $B \subseteq \mathcal{A}^m$. Then B contains* $\min(B)$ *and* $\max(B)$ *if and only if there exist $n \geq 0$ and $s \in \mathcal{T}_n^m$ such that* $\mathrm{img}(s^{\mathcal{A}}) = B$.

Proof. Let $\{b_1, \dots, b_n\}$ be an enumeration of B. Define the monotone function $f : \mathcal{A}^n \to \mathcal{A}^m$ by

$$f(a) = \begin{cases} \min(B) & a = 0 \\ b_j & a_k = 1 \text{ iff } j = k \\ \max(B) & \text{otherwise} \end{cases}$$

By Lemma 34, there exists an m-tuple of terms $s \in \mathcal{T}_n^m$ such that $s^{\mathcal{A}} = f$.

For the converse, observe that $\min(\mathrm{img}(s^{\mathcal{A}})) = s^{\mathcal{A}}(0)$ and $\max(\mathrm{img}(s^{\mathcal{A}})) = s^{\mathcal{A}}(1)$. ◀

▶ **Definition 36.** *Let $m \geq 0$. Define χ_m (for dl terms) to be the formula*

$$\exists x \left(R(x) \wedge \forall y \left(R(y) \to \textstyle\bigwedge_i (x_i = x_i \wedge y_i)\right)\right)$$
$$\wedge\ \exists x \left(R(x) \wedge \forall y \left(R(y) \to \textstyle\bigwedge_i (y_i = x_i \wedge y_i)\right)\right)$$

▶ **Proposition 37.** *Let $m \geq 0$. The formula χ_m in Definition 36 characterizes images for dl term functions.*

▶ **Corollary 38.** *The decision problem* $\mathrm{PCOFENT}_{dl}$ *is* coNP*-complete.*

4.4.3 Variable-constant fragments

We will now apply the second schema to show that irrespective of the choice of algebra $\mathcal{A}$, PCOFENTVC belongs to coNP.

▶ **Lemma 39.** *Let $m \geq 0$. On the one hand, if $\mathcal{A} \subseteq \{c^{\mathcal{A}}\}$, then $\forall x_{m+1}\,(x_{m+1} = c)$ is valid; and if not, $\forall x_{m+1}\,(x_{m+1} = c)$ is equivalent to $\bot$. On the other hand, for $(x_i = c) \in L_{cof,m}$, $\forall x_{m+1}\,(x_i = c)$ is equivalent to $(x_i = c)$.*

▶ Remark 40. Let $m \geq 0$, and let $\phi \in L_{cofvc,m}$. By Corollary 5 and Lemma 39, we may assume wlog that ϕ is, up to equivalence, either $\bot$, $\top$, or a disjunction of conjunctions of equations.

▶ **Proposition 41.** *Let $\phi \in L_{cofvc,m}$. Then*

$$\mathrm{X}^m \Vdash \phi \Rightarrow (x_j = c)\,(x) \quad \textit{iff} \quad \mathcal{A} \vDash \forall x \left(\phi \to x_j = c\right)$$

Proof. Induction on ϕ (in disjunctive form).

- Suppose $\phi = \bot$ or $\phi = \top$. Immediate.
- Suppose that $\phi = \bigwedge_{i \in I}(x_i = c_i)$ where $I \subseteq \{1, \dots, m\}$.
  - Suppose that $j \in I$ and $c_j = c$. Immediate.
  - Suppose that $j \notin I$ and $c_j = c$. Suppose that $\mathrm{X}^m \Vdash \bigwedge_{i \in I}(x_i = c_i) \Rightarrow (x_j = c)\,(x)$. In particular, $\mathrm{X}^m \Vdash (x_j = c)\,(s)$ when

    $$s_k = \begin{cases} c & k \in I \\ x_k & \text{otherwise} \end{cases}$$

    Therefore, $\mathcal{A} = \{c^{\mathcal{A}}\}$, and the conclusion follows trivially. The converse is analogous.
  - Suppose that $j \in I$ and $c_j \neq c$. But $\mathrm{X}^m \nVdash \bigwedge_{i \in I}(x_i = c_i) \Rightarrow (x_j = c)\,(x)$ because $\mathrm{X}^0 \Vdash (x_j = c_j)\,(c_j)$ while $\mathrm{X}^0 \nVdash (x_j = c)\,(c_j)$. And likewise, $\mathcal{A} \nvDash \bigwedge_{i \in I}(x_i = c_i) \to (x_j = c)\,(c_j)$.
- Suppose $\phi = \phi_1 \vee \phi_2$. Directly by induction. ◀

▶ **Proposition 42.** *Let $m \geq 0$, and let $S_1, S_2 \in \Omega\,\mathrm{X}^m$. Let E be the sieve denoted by the equation $(x_i = c) \in L_{cofvc,m}$: that is,*

$$E = \{s : \mathrm{X}^n \to \mathrm{X}^m \mid n \geq 0,\ s_i = c\} \in \Omega\,\mathrm{X}^m$$

If $E \subseteq S_1 \cup S_2$, then either $E \subseteq S_1$ or $E \subseteq S_2$.

Proof. Observe that E is principal: i.e., every morphism in E factors through

$$s = (x_1, \ldots, x_{i-1}, c, x_{i+1}, \ldots, x_m)$$

So if $s \in S_1$ ($s \in S_2$), then $E \subseteq S_1$ ($E \subseteq S_2$). ◀

▶ **Corollary 43.** *Equations in L_{cofvc} are locally prime.*

▶ **Lemma 44.** *Let $m \geq 0$. A formula $\rho \in L_{cofvc,m}$ is locally prime if and only it is equivalent to a formula of the form*

$$\rho \;=\; \bigwedge_{i \in I} (x_i = c_i) \quad \textit{where } c_1, \ldots, c_m \in \sigma_0$$

for some non-empty $I \subseteq \{1, \ldots, m\}$.

▶ **Definition 45.** *Let $m \geq 0$. Define χ_m (for variable-constant fragments) to be the formula*

$$\begin{aligned}\exists C_1 \ldots \exists C_m \Big(& \bigwedge_i \bigvee_j (C_i(c_j) \vee (x_i = c_j)) \\ & \wedge \bigwedge_i \bigwedge_j \bigwedge_{j<k} (C_i(c_j) \vee C_i(c_k)) \\ & \wedge \big(\bigvee_{i,j} \neg C_i(c_j)\big)\Big)\end{aligned}$$

where $c_1, \ldots, c_{|\sigma_0|}$ is an enumeration of σ_0.

▶ **Proposition 46.** *Let $m \geq 0$. The formula χ_m in Definition 45 defines the locally prime formulas in L_{cofvc}.*

▶ **Corollary 47.** *If σ_0 has constants 0 and 1 denoting distinct elements of $\mathcal{A}$, the decision problem* PCOFENTVC *is* coNP*-complete.*

Proof. By Propositions 8, 41, and 46 and Theorem 21. ◀

4.4.4 De Morgan algebras

Lastly, we turn to the theory of De Morgan algebras and the term-constant fragment L_{coftc}, which provide the cofibration layer of the cubical type theory in [3]. The strategy below is adapted from [3]: it amounts to reducing the term-constant variant of COFENT to the variable-constant variant.

The signature of De Morgan algebras will be $\sigma_0 = \{0, 1\}$, $\sigma_1 = \{\neg\}$, and $\sigma_2 = \{\wedge, \vee\}$. We take the generating algebra $\mathcal{A}$ to be $\{0,1\}^2$ with the constants interpreted as $0^{\mathcal{A}} = (0,0)$ and $1^{\mathcal{A}} = (1,1)$, and with the basic binary operations defined as $\wedge^{\mathcal{A}} = \min$ and $\vee^{\mathcal{A}} = \max$. The operation of De Morgan negation on $\{0,1\}^2$ is

x	$\neg^{\mathcal{A}}(x)$
$(0,0)$	$(1,1)$
$(1,0)$	$(1,0)$
$(0,1)$	$(0,1)$
$(1,1)$	$(0,0)$

Note that while Boolean negation swaps $(0,1)$ and $(1,0)$ as well as $(0,0)$ and $(1,1)$, De Morgan negation fixes $(0,1)$ and $(1,0)$, and swaps just $(0,0)$ and $(1,1)$. When needed, we will annotate constructions associated to this algebra with *dm*.

The following definition is adapted from [3]:

▶ **Definition 48.** *Let $m \geq 0$, and let $\phi \in L_{coftc,m}$.*

$$\phi^+ = \begin{cases} \bot & \phi = \bot \text{ or } (c' = c) \text{ with } c' \neq c \\ \top & \phi = \top \text{ or } (c = c) \\ (f = 1)^- & \phi = (f = 0) \\ (x_i = 1) & \phi = (x_i = 1) \\ (f = 1)^- & \phi = (\neg f = 1) \\ (f_1 = 1)^+ \wedge (f_2 = 1)^+ & \phi = (f_1 \wedge f_2 = 1) \\ (f_1 = 1)^+ \vee (f_2 = 1)^+ & \phi = (f_1 \vee f_2 = 1) \\ \phi_1^+ \wedge \phi_2^+ & \phi = \phi_1 \wedge \phi_2 \\ \phi_1^+ \vee \phi_2^+ & \phi = \phi_1 \vee \phi_2 \\ \forall x_{m+1}\, \psi^+ & \phi = \forall x_{m+1}\, \psi \end{cases}$$

$$\phi^- = \begin{cases} (f = 1)^+ & \phi = (f = 0) \\ (x_i = 0) & \phi = (x_i = 1) \\ (f = 1)^+ & \phi = (\neg f = 1) \\ (f_1 = 1)^- \vee (f_2 = 1)^- & \phi = (f_1 \wedge f_2 = 1) \\ (f_1 = 1)^- \wedge (f_2 = 1)^- & \phi = (f_1 \vee f_2 = 1) \end{cases}$$

▶ **Proposition 49.** *Let $m \geq 0$, and let $\phi \in L_{coftc,m}$. The formula $\phi^+ \in L_{cofvc,m}$ is equivalent to ϕ and is proportional in length.*

▶ **Corollary 50.** *The decision problem* PCOFENTTC$_{dm}$ *is* coNP*-complete.*

5 Conclusion

In this paper, we have assembled a capable framework for studying the computational character of key fragments of the internal languages of cubical sets. We have also put it to use and obtained theoretical results with practical importance: not only have we determined the complexity of variants of COFENT, we have reasonable decision procedures. That said, this framework is capable of more:

- We plan to extend our results to simplicial sets. The cofibration language in [6] corresponds not to all presheaves on the cube category, but to sheaves for a specific Grothendieck topology which defines a topos equivalent to simplicial sets. Preliminary work indicates that our results, summarized above, on presheaves on cubical categories will extend to simplicial sets via this equivalence. This would provide the means to automate the cofibration language layer of a directed type theory based on simplicial sets. It would also provide the means to automate the dependently-typed internal language of simplicial sets.
- We plan to extend our results to full first-order cofibration languages. For certain finite algebras $\mathcal{A}$, either or both schemas will apply: for the first schema, it needs to be shown that term-invariance holds for the more expressive fragment; and for the second schema, that the more expressive fragment has local factorization. Deciding validity in this context will give rise to fresh examples of PSPACE-complete problems.

- Finally, we conjecture that there are finite algebras for which the corresponding forcing relation (in presheaves) is undecidable. Our expectation is that counter-examples will help to explain the efficacy of the key properties already discovered: image invariance and local factorization.

References

1 Carlo Angiuli, Guillaume Brunerie, Thierry Coquand, Robert Harper, Kuen-Bang Hou (Favonia), and Daniel R. Licata. Syntax and models of cartesian cubical type theory. *Mathematical Structures in Computer Science*, 31(4):424–468, 2021. doi:10.1017/S0960129521000347.

2 Ulrik Buchholtz and Edward Morehouse. Varieties of cubical sets. *Lecture Notes in Computer Science*, pages 77–92, 2017. doi:10.1007/978-3-319-57418-9_5.

3 Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical type theory: a constructive interpretation of the univalence axiom, 2016. arXiv:1611.02108.

4 Leonid Libkin. *Elements of Finite Model Theory*. Springer Publishing Company, Incorporated, 1st edition, 2010.

5 Saunders Mac Lane and Ieke Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer New York, New York, NY, 1994.

6 Emily Riehl and Michael Shulman. A type theory for synthetic ∞-categories, 2017. arXiv:1705.07442.

7 Matthew Z. Weaver and Daniel R. Licata. A constructive model of directed univalence in bicubical sets. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, pages 915–928, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3373718.3394794.