Logical Methods in Computer Science
Volume 22, Issue 1, 2026, pp. 21:1–21:45
https://lmcs.episciences.org/

Submitted Aug. 09, 2024
Published Mar. 10, 2026

ON POLYNOMIAL-TIME DECIDABILITY OF k-NEGATIONS FRAGMENTS OF FIRST-ORDER THEORIES

CHRISTOPH HAASE [a], ALESSIO MANSUTTI [b], AND AMAURY POULY [c]

[a] Department of Computer Science, University of Oxford, UK

[b] IMDEA Software Institute, Spain

[c] CNRS, Université Paris Cité, IRIF, France

Abstract. This paper introduces a generic framework that provides sufficient conditions for guaranteeing polynomial-time decidability of fixed-negation fragments of first-order theories that adhere to certain fixed-parameter tractability requirements. It enables deciding sentences of such theories with arbitrary existential quantification, conjunction and a fixed number of negation symbols in polynomial time.

It was recently shown by Nguyen and Pak *[SIAM J. Comput. 51(2): 1–31 (2022)]* that an even more restricted such fragment of Presburger arithmetic, the first-order theory of the structure $(\mathbb{Z}, 0, 1, +, \leq)$, is NP-hard. In contrast, by application of our framework, we show that the fixed negation fragment of weak Presburger arithmetic, which drops the order relation $\leq$ from Presburger arithmetic in favor of the equality relation $=$, is decidable in polynomial time. We give two further examples of instantiations of our framework, showing polynomial-time decidability of the fixed negation fragments of weak linear real arithmetic (the first-order theory of the structure $(\mathbb{R}, 0, 1, +, =)$) and of the restriction of Presburger arithmetic in which each inequality contains at most one variable.

1. Introduction

It is well-known that even the simplest first-order theories are computationally difficult to decide [Grä91]. In particular, it follows from a result of Stockmeyer that every theory with a non-trivial predicate such as equality is PSPACE-hard to decide [Sto76]. Even when restricting to existential fragments or fragments with a fixed number of quantifier alternations, deciding such fragments is NP-hard at best. There are two further kinds of restrictions that may lead to tractability. First, restricting the Boolean structure of the matrix of formulae in prenex form yields tractable fragments of, e.g., the Boolean satisfiability problem. For instance, the Horn and XOR-fragments of propositional logic are decidable in polynomial time, and this even applies to quantified Boolean Horn formulae, see e.g. [Che09]. Second, restricting the number of variables can also lead to tractable fragments of a first-order theory, especially for structures over infinite domains such as Presburger arithmetic, the first-order theory of the structure $(\mathbb{Z}, 0, 1, +, \leq)$. While the existential fragment of Presburger arithmetic is NP-complete in general [BT76, vzGS78], it becomes polynomial-time decidable

Key words and phrases: First-order theories, Presburger arithmetic, tractability, arithmetic theories, integer lattices, difference normal form.

LOGICAL METHODS IN COMPUTER SCIENCE
DOI:10.46298/LMCS-22(1:21)2026

when additionally fixing the number of variables [Sca84]; this is a consequence of polynomial-time decidability of integer programming in fixed dimension [Len83]. Already when moving to an $\exists\forall$ quantifier prefix, Presburger arithmetic becomes NP-hard [Sch97]. On the first sight, this result seems to preclude any possibility of further restrictions that may lead to tractable fragments of Presburger arithmetic. However, another tractable fragment was identified in the context of investigating the complexity of the classical *Frobenius problem*. Given $a_1, \ldots, a_n \in \mathbb{N}$, this problem asks to determine the largest integer that cannot be obtained as a non-negative linear combination of the a_i, which is called the *Frobenius number*. For $n > 0$ fixed, deciding whether the Frobenius number exceeds a given threshold can be reduced to the so-called *short fragment* of Presburger arithmetic, a highly restricted fragment in which everything, the number of atomic formulae and the number of variables (and *a fortiori* the number of quantifier alternations), is fixed — except for the coefficients of variables appearing in linear terms of atomic inequalities. Kannan [Kan90] showed that the $\forall^k\exists^\ell$-fragment of short Presburger arithmetic is decidable in polynomial time for all fixed k, ℓ, which implies that the decision version of the Frobenius problem is in polynomial time for fixed n. However, in a recent breakthrough, Nguyen and Pak showed that there are fixed k, ℓ, m such that the $\exists^k\forall^\ell\exists^m$-fragment of short Presburger arithmetic is NP-hard, and by adding further (fixed) quantifier alternations the logic climbs the polynomial hierarchy [NP22].

The main contribution of this paper is to develop an algorithmic framework that enables us to show that *fixed negation fragments* of certain first-order theories are decidable in polynomial time. Formulae in this fragment are generated by the following grammar, where Ψ are atomic formulae of the underlying first-order theory, and an arbitrary but a priori fixed number of negation symbols is allowed to occur:

$$\Phi ::= \exists x\, \Phi \mid \neg\Phi \mid \Phi \wedge \Phi \mid \Psi\,.$$

We give sufficient conditions for the fixed negation fragment of a first-order theory to be decidable in polynomial time. We highlight that this fragment is more permissive than the "short fragment" of Kannan, as it allows for an unbounded number of quantified variables and an unbounded number of conjunctions. However, it also implicitly fixes the number of quantifier alternations as well as the number of disjunctions.

Our algorithmic framework is parametric on a concrete representation of the sets definable within the first-order theory $\mathcal{T}$ under consideration and only requires a sensible representation of solution sets for conjunctions of atomic formulae. From this representation, the framework guides us to the definition of a companion structure $\mathcal{R}$ for the theory $\mathcal{T}$ in which function symbols and relations in $\mathcal{R}$ are interpreted as reductions from parametrized complexity theory, such as UXP reductions, see e.g. [DF99, Chapter 15]. By requiring mild conditions on the types of reductions and parameters that the functions and relations in $\mathcal{R}$ must obey, we are able to give a general theorem for the tractability of the fixed negation satisfiability and entailment problems for $\mathcal{T}$. One technical issue we show how to overcome in a general way is how to treat negation, which is especially challenging when the initial representation provided to the framework is not closed under complementation. Our main source of inspiration here is the notion of the so-called *difference normal form* of propositional logic, a rather unorthodox normal form introduced by Hausdorff [Hau14, Ch. 1§5].

As one of the main application of our framework, we show that the fixed negation fragment of weak Presburger arithmetic (*weak PA*) is polynomial-time decidable. Weak PA is the first-order theory of the structure $(\mathbb{Z}, 0, 1, +, =)$, which is strictly less expressive than standard Presburger arithmetic. It was recently shown that unrestricted weak PA has the same

complexity as standard Presburger arithmetic [CHHM22]. In contrast, Bodirsky et al. showed that the weak PA fragment of existential linear Horn equations $\bigwedge_{i\in I}(A_i\cdot\mathbf{x} = b_i) \to (C_i\cdot\mathbf{x} = d_i)$ over $\mathbb{Z}$ with $|I|$ unbounded can be decided in PTIME [BMMM18]. It follows from the generic results in this paper that the quantified versions of those formulae with the number of quantifier alternations and $|I|$ fixed is also polynomial-time decidable. In fact, our framework not only allows for deciding satisfiability and validity of fixed negation formulae of Weak PA in PTIME, but also to compute a representation of the set of solutions of a given formula. This is the best possible such result, since we can show that, for I unbounded, the $\exists\forall$ fragment of linear Horn equations in two variables is NP-hard.

Proposition 1.1. *Deciding two-variables $\exists\forall$ weak PA Horn sentences is NP-hard.*

Proof. The $\exists\forall$ Horn sentences of weak PA are of the form $\exists x\forall y \bigwedge_{i=1}^{k} \psi_i(x, y)$, where each ψ_i is a Horn clause, i.e., a disjunction of literals in which at most one literal occurs positively. NP-hardness for deciding these sentences follows by a straightforward reduction from the problem of deciding a univariate system of non-congruences $\bigwedge_{i=1}^{k} x \not\equiv r_i \pmod{m_i}$, where $m_i \geq 2$ and $r_i \in [0, m_i-1]$ for every $i \in [1, k]$. This problem is NP-hard [BS96, Theorem 5.5.7]. For the reduction, simply apply the following equivalence: for every $x \in \mathbb{Z}$,

$$\bigwedge\nolimits_{i=1}^{k} x \not\equiv r_i \pmod{m_i} \quad\Longleftrightarrow\quad \forall y : \bigwedge\nolimits_{i=1}^{k} \neg\big(x - r_i = m_i \cdot y\big)\,. \qquad \square$$

An extended abstract of this paper appeared in the proceedings of the 48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023) [HMP23].

1.1. **Structure of this paper.** The goals of this paper are twofold, and consequently the paper consists of two parts. In the first part, after recalling and introducing some basic definitions concerning first-order logic in Section 2 as well as concepts underlying representations of objects and concepts from parametrized complexity in Section 3, we present our general algorithmic framework in Section 4. This framework is parametrized by a first-order theory $\mathcal{T}$, and we develop sufficient conditions on $\mathcal{T}$ for its k-negations fragment to be decidable in polynomial time, for every fixed $k \geq 0$. Section 4 is intended to enable a reader to easily apply the framework to determine whether a given first-order theory has a polynomial-time decidable k-negations fragment. For that reason, all proofs are relegated to Sections 5 and 6, and the section's concepts are illustrated using a simple fragment of Presburger arithmetic with one variable per inequality as a running example.

In the second part of the paper, Sections 7 and 8, we instantiate the framework to weak linear real arithmetic and weak Presburger arithmetic, proving that their k-negations fragments are decidable in polynomial-time.

1.2. **Related work.** There is a long history of research on identifying syntactic fragments of first-order logic with the goal of obtaining decidable fragments, possibly of low complexity. Specifically, restrictions on the use of negation have been widely studied. For instance, Kozen [Koz81] showed that deciding positive sentences in which no negation symbol occurs is NP-complete. Voronkov [Vor99] generalized this fragment and showed that the ground-negative fragment of first-order logic, where any negated atomic formula is required to be a ground term, is Π_2^{P}-complete. Bárány, Ten Cate and Segoufin [BCS15] identified *guarded-negation first-order logic* which requires all occurrences of negation to be of the

form $\alpha \wedge \neg\varphi$ such that α is an atomic formula containing all free variables of φ. Guarded-negation first-order logic is 2EXPTIME-complete and unified various previously identified decidable fragments of first-order logic such as the *unary negation* [TCS13] and *guarded fragment* [ANVB98] of first-order logic. More recent work explores restrictions on features other than negation. Jonsson, Lagerkvist and Osipov [JLO24], investigated constraint satisfaction problems (CSPs) over two signatures, $\mathcal{A}$ and $\mathcal{B}$, where at most k constraints from $\mathcal{B}$ are allowed. Under further assumptions on the CSPs over $\mathcal{A}$, and on the definability of $\mathcal{B}$ within $\mathcal{A}$, they showed that these CSPs can be solved in polynomial time for fixed k.

2. Preliminaries

This section focuses on notation and simple definitions that might be non-standard to some readers. We assume familiarity with basic concepts from logic and abstract algebra.

2.1. **Sets and functions.** We write $\mathrm{seq}(A)$ for the set of all finite tuples over a set A, and denote by $()$ the empty tuple. This definition corresponds to the standard notion of Kleene star A^* of a set A. The discrepancy in notation is introduced to avoid writing $(\Sigma^*)^*$ for the domain of all tuples of finite words over an alphabet Σ, as in formal language theory the Kleene star comes equipped with the axiom $(\Sigma^*)^* = \Sigma^*$. We denote this domain by $\mathrm{seq}(\Sigma^*)$.

We write $f :\subseteq X \to Y$ (resp. $f\colon X \to Y$) to denote a *partial (resp. total) function* from X to Y. The domain of f is denoted by $\mathrm{dom}(f)$. We write $\mathrm{id}_A\colon A \to A$ for the identity function on A. Given $f :\subseteq A \to B$, $g :\subseteq B \to C$ and $h :\subseteq D \to E$, we denote by $(g \circ f) :\subseteq A \to C$ and $(f \times h) :\subseteq A \times D \to B \times E$ the *composition* and the *Cartesian product* of functions.

2.2. **Structures with indexed families of functions.** We consider a generalization of the traditional definition of structure from universal algebra that accommodates for a potentially infinite number of functions. As usual, a *structure* $\mathcal{A} = (A, \sigma, I)$ consists of a *domain* A (a set), a *signature* σ, and an *interpretation function* I. In this paper, the signature is a quadruple $\sigma = (\mathcal{F}, \mathcal{G}, \mathcal{R}, ar)$ containing not only a set of *function symbols* $\mathcal{F}$, a set of *relation symbols* $\mathcal{R}$, and the *arity function* $ar\colon \mathcal{F} \uplus \mathcal{R} \uplus \mathcal{G} \to \mathbb{N}$, but also a set of *(indexed) families of function symbols* $\mathcal{G}$. Each element in the finite set $\mathcal{G}$ is a pair (g, X) where g is a function symbol and X is a countable set of indices. The interpretation function I associates to every $f \in \mathcal{F}$ a map $f^{\mathcal{A}}\colon A^{ar(f)} \to A$, to every $(g, X) \in \mathcal{G}$ a map $g^{\mathcal{A}}\colon X \times A^{ar((g,X))} \to A$, and to every $R \in \mathcal{R}$ a relation $R^{\mathcal{A}} \subseteq A^{ar(R)}$ which we often view as a function $R^{\mathcal{A}}\colon A^{ar(R)} \to \{\top, \bot\}$.

Example 2.1. Consider the structure $\mathcal{A} = (\mathbb{Z}, \sigma, I)$ in which the signature σ contains a single family of functions $(\mathrm{mul}, \mathbb{N})$ of arity one, and the interpretation function I associates to mul the map $\mathrm{mul}^{\mathcal{A}}(n, x) = n \cdot x$ for all $n \in \mathbb{N}$ and $x \in \mathbb{Z}$. Here, the family of functions $(\mathrm{mul}, \mathbb{N})$ uniformly defines multiplication by a non-negative integer constant n. ◇

The standard notions (see, e.g., [BS81]) of *homomorphism, embedding and isomorphism of structures*, as well as the notions of *congruence for a structure* and *quotient structure* extend naturally to structures having families of functions. For instance, a *homomorphism* from $\mathcal{A} = (A, \sigma, I)$ into $\mathcal{B} = (B, \sigma, J)$ is a map $h\colon A \to B$ that *preserves* all functions, families of functions and relations; so in particular given $(g, X) \in \mathcal{G}$, the map h satisfies $g^{\mathcal{B}}(x, h(a_1), \ldots, h(a_{ar(g)})) = h(g^{\mathcal{A}}(x, a_1, \ldots, a_{ar(g)}))$ for every $x \in X$ and $a_1, \ldots, a_{ar(g)} \in A$.

We denote structures in calligraphic letters $\mathcal{A}, \mathcal{B}, \ldots$ and their domains in capital letters $A, B, \ldots$. When the arity function ar and the interpretation I are clear from the

context, we write $(A, f_1^{\mathcal{A}}, ..., f_j^{\mathcal{A}}, (g_1^{\mathcal{A}}, X_1), ..., (g_\ell^{\mathcal{A}}, X_\ell), R_1^{\mathcal{A}}, ..., R_k^{\mathcal{A}})$ for $\mathcal{A} = (A, \sigma, I)$ with $\sigma = (\{f_1, ..., f_j\}, \{(g_1, X_1), ..., (g_\ell, X_\ell)\}, \{R_1, ..., R_k\}, ar)$, and often drop the superscript $\mathcal{A}$. For instance, the structure from Example 2.1 can be denoted as $(\mathbb{Z}, (\text{mul}, \mathbb{N}))$.

2.3. **First-order theories (finite tuples semantics).** The first-order (FO) language of the signature $\sigma = (\mathcal{F}, \mathcal{G}, \mathcal{R}, ar)$ is the set of all formulae $\Phi, \Psi, \ldots$ built from the grammar

$$\begin{aligned} \Phi, \Psi &::= r(t_1, \ldots, t_{ar(r)}) \mid \neg\Phi \mid \Phi \wedge \Psi \mid \exists x.\Phi \\ t &::= x \mid f(t_1, \ldots, t_{ar(f)}) \mid g(i, t_1, \ldots, t_{ar(g)}), \end{aligned} \tag{2.1}$$

where $x \in \mathbb{V}$ is a first-order variable, $r \in \mathcal{R}$, $f \in \mathcal{F}$, $(g, X) \in \mathcal{G}$ and $i \in X$ (more precisely, i belongs to a representation of X, more details are given in Section 3 below). Lexemes of the form $r(t_1, \ldots, t_{ar(r)})$ are the *atomic formulae* of the language. Throughout this paper, we implicitly assume an order on the variables in $\mathbb{V}$, and write x_j for the j-th variable (indexed from 1). We write $\text{CQ}(\sigma)$ for the set of all *conjunctive queries* of the first-order language of σ, that is the set of all (quantifier-free) conjunctions of atomic formulae in the language.

Consider a structure $\mathcal{A} = (A, \sigma, I)$. Given an atomic formula $r(t_1, \ldots, t_{ar(r)})$ from the grammar in Equation (2.1) having x_n as the largest appearing variable, we write $[\![r(t_1, ..., t_{ar(r)})]\!]_{\mathcal{A}} \subseteq A^n$ for the set of n-tuples, corresponding to values of the first n variables, that makes the formula $r(t_1, ..., t_{ar(r)})$ true under the given interpretation I. Furthermore, let us define $\mathbf{I} \coloneqq \{(i_1, ..., i_k) \in \text{seq}(\mathbb{N}) : i_1, ..., i_k \text{ all distinct}\}$. We denote by $\text{FO}(\mathcal{A})$ the structure all *first-order sets definable in* $\mathcal{A}$, which is the structure

$$\text{FO}(\mathcal{A}) \coloneqq ([\![\mathcal{A}]\!]_{\text{FO}}, \bot, \top, \vee, \wedge, -, (\pi, \mathbf{I}), (\pi^\forall, \mathbf{I}), \leq), \text{ where}$$

(1) $[\![\mathcal{A}]\!]_{\text{FO}}$ is the least set containing $[\![r(t_1, ..., t_{ar(r)})]\!]_{\mathcal{A}}$, for each atomic formula $r(t_1, ..., t_{ar(r)})$, and that is closed under the functions $\bot, \top, \vee, \wedge, -, (\pi, \mathbf{I})$ and $(\pi^\forall, \mathbf{I})$, defined below.
(2) The functions $\bot$ and $\top$ are interpreted as $\emptyset$ and $\{()\}$, respectively.
(3) Given $X \subseteq A^n$, $Y \subseteq A^m$ and $\mathbf{i} = (i_1, \ldots, i_k) \in \mathbf{I}$, and defining $M \coloneqq \max(n, m)$,

$$\begin{aligned} X \vee Y &\coloneqq \{(a_1, ..., a_M) : (a_1, ..., a_n) \in X \text{ or } (a_1, ..., a_m) \in Y\}, \\ X \wedge Y &\coloneqq \{(a_1, ..., a_M) : (a_1, ..., a_n) \in X \text{ and } (a_1, \ldots, a_m) \in Y\}, \\ X - Y &\coloneqq \{(a_1, ..., a_M) : (a_1, ..., a_n) \in X \text{ and } (a_1, ..., a_m) \notin Y\}, \\ \pi(\mathbf{i}, X) &\coloneqq \{\gamma \in A^n : \text{there is } \mathbf{a} \in A^k \text{ s.t. } \gamma[\mathbf{i} \leftarrow \mathbf{a}] \in X\}, \\ \pi^\forall(\mathbf{i}, X) &\coloneqq \{\gamma \in A^n : \text{for every } \mathbf{a} \in A^k, \gamma[\mathbf{i} \leftarrow \mathbf{a}] \in X\}, \\ X \leq Y &\text{ if and only if } X \times A^m \subseteq Y \times A^n, \end{aligned}$$

where $\gamma[\mathbf{i} \leftarrow \mathbf{a}]$ is the tuple obtained from $\gamma \in A^n$ by replacing its i_j-th component with the j-th component of $\mathbf{a}$, for every $j \in [1, \min(k, n)]$.

The semantics $[\![.]\!]_{\mathcal{A}}$ of the FO language of σ is extended to non-atomic formulae via $\text{FO}(\mathcal{A})$. As usual, $[\![\neg\Phi]\!]_{\mathcal{A}} \coloneqq \top - [\![\Phi]\!]_{\mathcal{A}}$, $[\![\Phi \wedge \Psi]\!]_{\mathcal{A}} \coloneqq [\![\Phi]\!]_{\mathcal{A}} \wedge [\![\Phi]\!]_{\mathcal{A}}$, and $[\![\exists x_i.\Phi]\!]_{\mathcal{A}} \coloneqq \pi((i), [\![\Phi]\!]_{\mathcal{A}})$. We omit the subscript $\mathcal{A}$ from $[\![\cdot]\!]_{\mathcal{A}}$ when it is clear from context, writing simply $[\![\cdot]\!]$. We remark that $\text{FO}(\mathcal{A})$ contains operators whose syntactic counterpart is absent from the FO language of σ, such as the universal projection $\pi^\forall$. This is done for algorithmic purposes, as the framework we introduce in Section 4 treats these operators as first-class citizens.

2.4. **Fixed negations fragments.** Let $k \in \mathbb{N}$ be fixed. The *k-negations fragment of the FO language* of a signature σ is the set of all formulae having at most k negations $\neg$. Note that, following the grammar provided in Equation (2.1), this restriction also bounds the number of disjunctions and alternations between existential and universal quantifiers that formulae can have. Given a structure $\mathcal{A} = (A, \sigma, I)$, we study the following problem:

k negations satisfiability problem: Given a formula Φ with at most k negations, decide whether $[\![\Phi]\!] \neq \emptyset$.

3. Representations and parametrised complexity of signatures

Per se, a structure $\mathcal{A}$ cannot be analysed algorithmically, in particular because the elements of A do not have a notion of size. A standard way to resolve this issue is defining computability via the notion of representations (as it is done for instance in computable analysis [Wei00]).

3.1. **Representations.** A *representation* for a set A is a surjective partial map $\rho :\subseteq \Sigma^* \to A$, where Σ is a finite alphabet. Words $w \in \Sigma^*$ are naturally equipped with a notion of *size*, that is their length, denoted by $|w|$. Observe that not all words are valid representations for elements of A (ρ is partial) and each element from A may be represented in several ways (ρ is not assumed to be injective). We write $(\approx_\rho) \subseteq \Sigma^* \times \Sigma^*$ for the equivalence relation $\{(w_1, w_2) : w_1, w_2 \in \mathrm{dom}(\rho) \text{ and } \rho(w_1) = \rho(w_2)\}$ and define $h_\rho \colon \mathrm{dom}(\rho)/_{\approx_\rho} \to A$ to be the bijection satisfying $h_\rho([w]_{\approx_\rho}) = \rho(w)$, for every $w \in \mathrm{dom}(\rho)$. Here, $\mathrm{dom}(\rho)/_{\approx_\rho}$ is the set of all equivalence classes $[w]_{\approx_\rho}$ of words $w \in \mathrm{dom}(\rho)$.

Example 3.1. The two's complement least significant digit first representation of $\mathbb{Z}$ is given by the map $\rho :\subseteq \{0,1\}^* \to \mathbb{Z}$ mapping every non-empty word of binary digits $d_0 \dots d_m \in \{0,1\}^{m+1}$ to the integer $-d_m \cdot 2^m + \sum_{i=0}^{m-1} d_i \cdot 2^i$. This representation is not defined on the empty word. A property of this representation is that padding each word to the right by repeating its most significant digit does not change the encoded number, that is, $(d_0 \dots d_m) \approx_\rho (d_0 \dots d_m d_m \dots d_m)$, where d_m is repeated an arbitrary number of times. ◇

It is often more practical to represent elements of A by objects that are more sophisticated than words in Σ^*, such as tuples, automata, graphs, etc. Taking these representations does not change the notion of computability or complexity, because they can be easily encoded as words (over a bigger alphabet, if necessary). In our setting, of particular interest are representations as tuples of words. The notion of size for words trivially extends to tuples: $|(w_1, \dots, w_n)| \coloneqq n + \sum_{i=1}^{n} |w_i|$. Given representations $\rho :\subseteq \Sigma^* \to A$ and $\rho' :\subseteq \Pi^* \to A'$, we rely on the following *operations on representations*:

- The Cartesian product $\rho \times \rho'$ of representations, defined as in Section 2.
- The representation $\mathrm{seq}(\rho) :\subseteq \mathrm{seq}(\Sigma^*) \to \mathrm{seq}(A)$ that, for every $n \in \mathbb{N}$, given a tuple $(w_1, \dots, w_n) \in \mathrm{dom}(\rho)^n$ returns $(\rho(w_1), \dots, \rho(w_n))$.

We also require representations for basic objects such as $\mathbb{N}$, $\mathbb{Z}$ and so on. Specifically, we assume to have *canonical representations* ν_X for the following countable domains X:

- $X = \mathbb{N}$ or $X = \mathbb{Z}$, so that ν_X is a representation of $\mathbb{N}$ or $\mathbb{Z}$, respectively. We assume this representation to be any standard binary encoding of natural numbers or integers that allows arithmetic operations such as addition, multiplication and integer division to be implemented in polynomial time, as for instance the representation in Example 3.1.

- X is any finite set, e.g., we assume to have a representation $\nu_{\mathbb{B}}$ for the Booleans $\mathbb{B} = \{\top, \bot\}$. Note that, since X is finite, operations on this set are constant time.
- $X = \Sigma^*$ where Σ is any finite alphabet. In this case, $\nu_{\Sigma^*} \coloneqq \mathrm{id}_{\Sigma^*}$.

For a canonical representation ν_X and $n \in \mathbb{N}$, we write ν_{X^n} for the Cartesian product $(\nu_X)^n$.

3.2. **Implementations and computability.** Let $\rho :\subseteq \Pi^* \to A$ and $\rho_1, \ldots, \rho_n$ be representations, with $\rho_i :\subseteq \Sigma_i^* \to A_i$. A function $f\colon A_1 \times \cdots \times A_n \to A$ is said to be $(\rho_1 \times \cdots \times \rho_n, \rho)$-*computable* if there is a function $F\colon \mathrm{dom}(\rho_1) \times \cdots \times \mathrm{dom}(\rho_n) \to \mathrm{dom}(\rho)$ that is computable (by a Turing machine) and satisfies $\rho(F(w_1, \ldots, w_n)) = f(\rho_1(w_1), \ldots, \rho_n(w_n))$ for all $w_i \in \mathrm{dom}(\rho_i)$, $i \in [1, n]$. The function F is said to be a $(\rho_1 \times \ldots \times \rho_n, \rho)$-*implementation* of f. For simplicity, we do not mention the representations of a computable function when it operates on canonical types: for sets $A, A_1, \ldots, A_n$ admitting canonical representations, a function $f\colon A_1 \times \cdots \times A_n \to A$ is said to be *computable* whenever it is $(\nu_{A_1} \times \ldots \times \nu_{A_n}, \nu_A)$-*computable* (the ν_{A_i} and ν_A are the canonical representations of A_i and A).

Example 3.2. The addition function $+\colon \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ is $(\nu_{\mathbb{Z}} \times \nu_{\mathbb{Z}}, \nu_{\mathbb{Z}})$-computable. Since $\nu_{\mathbb{Z}}$ is a canonical representation, we simply say that $+$ is *computable*. This is the standard notion of computability over $\mathbb{Z}$, with respect to a binary encoding of integers. ⋄

Let $\mathcal{A} = (A, \sigma, I)$ be a structure and $\rho :\subseteq \Sigma^* \to A$ be a representation. Let $\mathcal{M} \coloneqq (\mathrm{dom}(\rho), \sigma, J)$ be a structure where the interpretation J associates computable functions to each function, family of functions and relations in σ, and makes $\approx_\rho$ a congruence for $\mathcal{M}$. Then, $\mathcal{M}$ is said to be a ρ-*implementation* of $\mathcal{A}$ whenever ρ is a homomorphism between $\mathcal{M}$ and $\mathcal{A}$. We highlight the fact that, compared to a standard homomorphism between structures, an implementation is always surjective (since ρ is surjective) and forces J to give an interpretation to functions and relations in σ in terms of computable functions.

Example 3.3. The structure $(\mathrm{dom}(\nu_{\mathbb{Z}}), +)$ is a $\nu_{\mathbb{Z}}$-implementation of $(\mathbb{Z}, +)$. For a further example, consider the structure $\mathcal{A} = (L, \cup, \cap, (\cdot)^c)$ where L is the set of all regular languages over a fixed finite alphabet Σ, and $\cup$, $\cap$, and $(\cdot)^c$ are the canonical operations of union, intersection and complementation of languages, respectively. As a representation, one can consider the map ρ taking as input a deterministic finite automaton (DFA) over Σ, and returning the language the automaton accepts. We obtain the structure $\mathcal{M} = (\mathrm{dom}(\rho), \cup, \cap, (\cdot)^c)$ in which the functions $\cup$, $\cap$, and $(\cdot)^c$ can be implemented by Turing machines manipulating DFAs; and $\mathcal{M}$ is a ρ-implementation of $\mathcal{A}$. ⋄

3.3. **Parametrised complexity of signatures.** The framework we define in the next section requires the introduction of a notion of parametrised complexity for the signature of a structure (which we call a *UXP signature*) which we now formulate. First, let us recall the standard notion of UXP reduction from parametrised complexity theory [DF99, Chapter 15]. Let Γ and Π be two finite alphabets, and $D \subseteq \Gamma^*$. A *parameter function* is a map $\eta\colon \Gamma^* \to \mathbb{N}$ such that $\eta(w) \geq 1$ for every $w \in \Gamma^*$. A computable function $F\colon D \to \Pi^*$ is said to be a *uniform slicewise polynomial reduction* for two parameter functions η and θ, or (η, θ)-UXP reduction for short, whenever there is an increasing map $G\colon \mathbb{N} \to \mathbb{N}$ such that for every $w \in D$, $F(w)$ runs in time $|w|^{G(\eta(w))}$ (w.l.o.g. assume $|w| \geq 2$), and $\theta(F(w)) \leq G(\eta(w))$.

As usual in computability theory, functions F with multiple arguments are handled by introducing a special symbol to the alphabet Γ, say $\#$, to separate the arguments, thus

viewing F as a function in one input. For instance, an operator $\oplus\colon \Sigma_1^* \times \Sigma_2^* \to \Sigma^*$ can be interpreted by a computable function taking as inputs words $w_1 \# w_2$ with $(w_1, w_2) \in \Sigma_1^* \times \Sigma_2^*$. The product $(\eta_1 \cdot \eta_2)(w_1 \# w_2) \coloneqq \eta_1(w_1) \cdot \eta_2(w_2)$ of parameter functions $\eta_1\colon \Sigma_1^* \to \mathbb{N}$ and $\eta_2\colon \Sigma_2^* \to \mathbb{N}$ can be used to refine the complexity analysis of $\oplus$ to each of its two arguments. We write $\mathbf{1}$ for the trivial parameter function defined as $\mathbf{1}(w) \coloneqq 1$ for all $w \in \Sigma^*$.

Let $\mathcal{A} = (A, \sigma, I)$ be a structure, $\sigma = (\mathcal{F}, \mathcal{G}, \mathcal{R}, ar)$, $\rho :\subseteq \Sigma^* \to A$ be a representation, and $\eta\colon \Sigma^* \to \mathbb{N}$ be a parameter function. We say that $\mathcal{A}$ has a (ρ, η)-*UXP signature* whenever there is an interpretation function J such that *(i)* $(\mathrm{dom}(\rho), \sigma, J)$ is a ρ-implementation of $\mathcal{A}$ and *(ii)* J associates a $(\eta^{ar(f)}, \eta)$-UXP reduction to every $f \in \mathcal{F}$, a $(\mathbf{1} \cdot \eta^{ar(g)}, \eta)$-UXP reduction to every $(g, X) \in \mathcal{G}$, and a $(\eta^{ar(R)}, \mathbf{1})$-UXP reduction to every $R \in \mathcal{R}$. Note that for $\eta = \mathbf{1}$, all those reductions become polynomial time functions. In this case we say that $\mathcal{A}$ has a (ρ-)tractable signature.

Example 3.4. Consider the structures of regular languages $\mathcal{A}$ and of deterministic finite automata $\mathcal{M}$ in Example 3.3. The functions $\cup$, $\cap$, and $(\cdot)^c$ can be implemented in polynomial time on DFAs, therefore $\mathcal{A}$ has a ρ-tractable signature. However, $\mathcal{A}$ does not have a tractable signature for the representation of regular languages as non-deterministic finite automata (NFAs), because computing $(\cdot)^c$ on NFAs requires first to determinise the automaton. $\diamond$

As in the case of representations, it is often more practical to have parameter functions from objects other than words. Given a parameter function $\theta\colon \Sigma^* \to \mathbb{N}$, we consider the operations $\mathrm{len}(\theta)\colon \mathrm{seq}(\Sigma^*) \to \mathbb{N}$, $\max(\theta)\colon \mathrm{seq}(\Sigma^*) \to \mathbb{N}$ and $\mathrm{dep}(\theta)\colon \mathrm{seq}(\mathrm{seq}(\Sigma^*)) \to \mathbb{N}$ on parameter functions. For $\mathbf{w} = (w_1, \ldots, w_n)$, they are defined as

$$\mathrm{len}(\theta)(\mathbf{w}) \coloneqq \textstyle\sum_{i=1}^{n} \theta(w_i); \quad \max(\theta)(\mathbf{w}) \coloneqq \max_{i=1}^{n} \theta(w_i); \quad \mathrm{dep}(\theta) \coloneqq \mathrm{len}(\mathrm{len}(\theta)).$$

4. A framework for the fixed negation fragment of first-order theories

Fix a structure $\mathcal{A} = (A, \sigma, I)$ and consider the structure $\mathrm{FO}(\mathcal{A})$ from Section 2.3:

$$\mathrm{FO}(\mathcal{A}) \coloneqq (\llbracket \mathcal{A} \rrbracket_{\mathrm{FO}}, \bot, \top, \vee, \wedge, -, (\pi, \mathbf{I}), (\pi^{\forall}, \mathbf{I}), \leq).$$

In this section, we describe a framework that can be employed to show that the k negation satisfiability problem for $\mathrm{FO}(\mathcal{A})$ is in PTIME. Part of our framework is generic, i.e., it applies to any first-order theory, while other parts are specific to the theory under consideration. To keep the presentation of the framework concise, we postpone detailed proofs of the formal statements to the subsequent Sections 5 and 6. The current section is best regarded as a blueprint intended to guide the instantiation of the framework.

4.1. Ideas underlying the framework.

To understand the framework, we first discuss how we can exploit the fact that our formulae only have a fixed number of negations. For simplicity, let us focus for the time being on *quantified Boolean formulae* (QBF) in prenex form. These are formulae of the form $\exists q_1 \forall q_2 \ldots \exists q_n \Phi$, where Φ is a formula from propositional logic, and $q_1, \ldots, q_n$ are some Boolean variables occurring in it. A first key question is whether bringing the quantifier-free part Φ of a QBF formula into a particular normal form can be computationally beneficial. Of course, due to Φ having a fixed number of negations, Φ could be translated into DNF in PTIME. However, because of quantifier alternation together with the unbounded number of conjunctions, choosing this normal form comes with several intricacies. Another option we might try is to put Φ into a form where all but a fixed amount of constraints are in Horn form, and then try to rely on the

algorithm to solve quantified Horn Boolean satisfiability in PTIME [KKS87]. This works for the Boolean case, but not for an arbitrary theory. For instance, as shown in Section 1, the quantified Horn satisfiability problem for the FO theory of $\mathcal{Z} = (\mathbb{Z}, 0, 1, +, =)$, i.e. weak PA, is already NP-hard for the alternation prefix $\exists\forall$ and 2 variables (and NEXPTIME-hard in general [CHHM22]). It turns out that a suitable normal form for Φ is given by formulae of the form $\Phi_1 - (\Phi_2 - (\cdots - (\Phi_{k-1} - \Phi_k)))$, where each Φ_i is a negation-free formula in disjunctive normal form (DNF), and $\Psi_1 - \Psi_2$ is the relative complementation $\Psi_1 \wedge \neg\Psi_2$. As we will see in Section 4.3, this atypical normal form (introduced by Hausdorff in [Hau14] and called *difference normal form* in [Jun00]) not only fully makes use of our restriction on the number of negations, but also exhibits nice properties in relation to quantification.

Example 4.1. Consider the propositional formula $\Phi(a, b, c) \coloneqq (a \vee b) \wedge (\neg a \vee c) \wedge (\neg b \vee \neg c)$. This formula is satisfied by the assignments $(a = \top, b = \bot, c = \top)$ and $(a = \bot, b = \top, c = \bot)$, so in particular the QBF formula $\forall a \exists b \exists c\, \Phi$ is valid. The difference normal form of Φ is:

$$\Psi \coloneqq (a \vee b) - \big((a \vee (b \wedge c)) - ((a \wedge c) - (a \wedge b \wedge c))\big).$$

All propositional formulae can be converted into difference normal form, as we will see in Section 5. In Example 4.7, we will discuss how to eliminate the quantifiers from $\forall a \exists b \exists c\, \Psi$. $\diamond$

A second key question is what representation of the domain $[\![\mathcal{A}]\!]_{\mathrm{FO}}$ works best for our purposes, as formulae might not be the right "data structures". Though the difference normal form already sets how to treat disjunctions and negations, we have the flexibility to vary the representation of conjunctions of atomic formulae. Let us be a bit more precise. Consider a domain $\mathrm{D} \subseteq [\![\mathcal{A}]\!]_{\mathrm{FO}}$ containing *at least* all the sets $[\![\Psi]\!]$, for every $\Psi \in \mathrm{CQ}(\sigma)$. We define $\mathrm{un}(\mathrm{D})$ to be the smallest set contaning D and being closed under the disjunction $\vee$, and $\mathrm{dfnf}(\mathrm{D})$ to be the smallest set containing $\mathrm{un}(\mathrm{D})$ and being closed under relative complements $X - Y$, with $X \in \mathrm{un}(\mathrm{D})$ and $Y \in \mathrm{dfnf}(\mathrm{D})$. From the fact that all propositional formulae can be converted into difference normal form, we conclude that $\{[\![\Phi]\!] : \Phi \text{ quantifier free}\} \subseteq \mathrm{dfnf}(\mathrm{D})$. Then, for a representation $\rho :\subseteq \Sigma^* \to \mathrm{D}$, the difference normal form gives a straightforward way of representing $\mathrm{dfnf}(\mathrm{D})$. First, we define the representation $\mathrm{un}(\rho) :\subseteq \mathrm{seq}(\Sigma^*) \to \mathrm{un}(\mathrm{D})$, given by $\mathrm{un}(\rho)(c_1, \ldots, c_n) \coloneqq (\rho(c_1) \vee \cdots \vee \rho(c_n))$, where each c_i belongs to Σ^*. A representation for $\mathrm{dfnf}(\mathrm{D})$ is then given by the map $\mathrm{dfnf}(\rho) :\subseteq \mathrm{seq}(\mathrm{seq}(\Sigma^*)) \to \mathrm{dfnf}(\mathrm{D})$ defined as

$$\mathrm{dfnf}(\rho)(u_1, \ldots, u_m) \coloneqq \mathrm{un}(\rho)(u_1) - \big(\mathrm{un}(\rho)(u_2) - \big(\cdots - \big(\mathrm{un}(\rho)(u_{m-1}) - \mathrm{un}(\rho)(u_m)\big)\big)\big),$$

where each u_i belongs to $\mathrm{seq}(\Sigma^*)$. The key point is that the representation ρ can be selected so that the elements in D are encoded as something other than formulae. For instance, for linear arithmetic theories, alternative representations are given by finite automata [Büc60] or geometric objects [CHM22]. In the instantiations of the framework provided in Sections 7 and 8 we will use the geometric objects. Of course, relying on representations other than formulae requires an efficient way of changing representation. This is stressed in the forthcoming Proposition 4.2. One last observation: above, we defined the domain D to be a superset of $\{[\![\Psi]\!]_{\mathcal{A}} : \Psi \in \mathrm{CQ}(\sigma)\}$. This is because more general sets might be required to make $\mathrm{dfnf}(\mathrm{D})$ closed under (universal) projection. For instance, in weak integer arithmetic, the formula $\exists y : x = 2 \cdot y$, stating that x is even, cannot be expressed with a quantifier-free formula, hence $[\![\exists y : x = 2 \cdot y]\!]_{\mathcal{Z}}$ must be added to D.

4.2. **What the framework achieves.** The following proposition formalises the observations in Section 4.1. Recall that an algorithm is in χ-UXP, for a parameter $\chi\colon \Sigma^* \to \mathbb{N}$, if it runs in time $|w|^{G(\chi(w))}$ for every $w \in \Sigma^*$, for some function $G\colon \mathbb{N} \to \mathbb{N}$ not depending on w. A decision problem is in χ-UXP if there is a χ-UXP algorithm solving that problem.

Proposition 4.2. *Fix $k \in \mathbb{N}$. Assume the following objects to be defined:*

(1) *A representation ρ of $\mathrm{D} := \bigcup_{n\in\mathbb{N}} \mathrm{D}_n$, where, for all $n \in \mathbb{N}$, $\mathrm{D}_n \subseteq \mathcal{P}(A^n)$ is s.t. $[\![\Psi]\!]_{\mathcal{A}} \in \mathrm{D}_n$ for every $\Psi \in \mathrm{CQ}(\sigma)$ having maximum variable x_n.*

(2) *A (ξ,θ)-UXP reduction $F\colon \mathrm{CQ}(\sigma) \to \mathrm{dom}(\rho)$ s.t. $(\rho \circ F)(\Psi) = [\![\Psi]\!]_{\mathcal{A}}$ for all $\Psi \in \mathrm{CQ}(\sigma)$.*

If $\mathcal{D} \coloneqq (\mathrm{dfnf}(\mathrm{D}), \bot, \top, \vee, \wedge, -, (\pi, \mathbf{I}), (\pi^{\forall}, \mathbf{I}), \leq)$ has a $(\mathrm{dfnf}(\rho), \mathrm{dep}(\theta))$-UXP signature,

- *the k negations satisfiability problem for $\mathrm{FO}(\mathcal{A})$ is in ξ-UXP (in PTIME, if $\xi = \mathbf{1}$), and*
- *there is a ξ-UXP (polynomial time, if $\xi = \mathbf{1}$) algorithm that, given a formula Φ of $\mathrm{FO}(\mathcal{A})$ having at most k negations, returns X in $\mathrm{dom}(\mathrm{dfnf}(\rho))$ such that $\mathrm{dfnf}(\rho)(X) = [\![\Phi]\!]_{\mathcal{A}}$.*

By virtue of the discussion in Section 4.1, establishing Proposition 4.2 is straightforward: the reduction F enables an efficient conversion from $\mathrm{CQ}(\sigma)$ to elements in $\mathrm{dom}(\rho)$, and, since $\mathcal{D}$ is a structure, $\mathrm{dfnf}(\mathrm{D})$ is closed under all the operations in the signature and thus it is equal to $[\![\mathcal{A}]\!]_{\mathrm{FO}}$. Consequently, $\mathrm{FO}(\mathcal{A})$ has a $(\mathrm{dfnf}(\rho), \mathrm{dep}(\theta))$-UXP signature, and one can efficiently use $\mathrm{dfnf}(\rho)$ as a data structure to carry out the algorithm to decide satisfiability: it suffices invoking the various UXP reductions implementing the functions and relations in $\mathcal{D}$. We remark that the sole purpose of the parameter θ is to factor in the parameter ξ, and one can set $\theta \coloneqq \mathbf{1}$ in the case of $\xi \coloneqq \mathbf{1}$ (i.e., the case yielding PTIME algorithms).

4.3. **The framework.** To apply Proposition 4.2, one has to provide the required representation ρ and the reduction F from Items 1 and 2, and show that the structure $\mathcal{D}$ has the desired UXP signature. Whereas the choice of ρ and F depends on the FO theory at hand, we show that a significant portion of the work required to prove that $\mathcal{D}$ has an UXP signature can be treated in a general way, thanks to the notion of difference normal form. This "automation" is the core of our framework, which provides a minimal set of subproblems that are sufficient to conclude that $\mathcal{D}$ has a $(\mathrm{dfnf}(\rho), \mathrm{dep}(\theta))$-UXP signature. Below, we divide those subproblems into two requirements, one for Boolean connectives and one for quantification. One significant result in this context is that negation can be treated in a general way. As a running example, we sketch the instantiation of our framework on the fragment of Presburger arithmetic having one variable per inequality.

Requirement 4.3 (Boolean connectives)**.** *Establish the following properties:*

(F1) *The structure $(\mathrm{D}, \wedge, \leq)$ has a (ρ,θ)-UXP signature.*

(F2) *The structure $(\mathrm{un}(\mathrm{D}), \leq)$ has a $(\mathrm{un}(\rho), \mathrm{len}(\theta))$-UXP signature.*

Requirement 4.3 asks to provide algorithms for solving typical computational problems that are highly domain-specific: Item (F1) considers the intersection and inclusion problems for elements of D, with respect to the representation ρ, whereas Item (F2) deals with the inclusion problem for unions of elements in D, with respect to the representation $\mathrm{un}(\rho)$. In the case of unions, we highlight the parameter $\mathrm{len}(\theta)$ which fixes the length of the union.

Example 4.4 (PA with one variable per inequality)**.** Consider the fragment of Presburger arithmetic in which formulae follow the grammar in Equation (2.1), and atomic formulae are of the form $x \leq k$ or $x \geq k$, where x is a variable ranging over the integers, and $k \in \mathbb{Z}$. We start by defining the objects in Items 1 and 2 of Proposition 4.2:

- Define $\mathrm{D} \coloneqq \{[\![\Psi]\!] : \Psi \in \mathrm{CQ}(\sigma)\}$, and $\theta \coloneqq \mathbf{1}$.
- We set ρ to be a map taking as inputs $\top$, $\bot$, or systems of constraints Ψ of the form $\bigwedge_{i=1}^{m} x_i \in [\ell_i, u_i]$, where $\ell_i \in \mathbb{Z} \cup \{-\infty\}$ and $u_i \in \mathbb{Z} \cup \{+\infty\}$, for every $i \in [1, m]$. The definition of ρ is given by $\rho(\bot) \coloneqq [\![\bot]\!]$, $\rho(\top) \coloneqq [\![\top]\!]$ and $\rho(\Psi) \coloneqq [\![\bigwedge_{j=1}^{m}(\ell_j \leq x_j \wedge x_j \leq u_j)]\!]$, where $-\infty \leq x_j$ and $x_j \leq +\infty$ are interpreted as $\top$.
- For a given $\Phi \in \mathrm{CQ}(\sigma)$ the function F returns the element of $\mathrm{dom}(\rho)$ computed (in polynomial time) as follows: Let x_n be the maximum variable occurring in Φ. For every $i \in [1, n]$, find the largest $\ell_i \in \mathbb{Z}$ and the smallest $u_i \in \mathbb{Z}$ such that $\ell_i \leq x_i$ and $x_i \leq u_i$ occur in Φ. If one of these inequalities does not appear, take the corresponding bound to be $\pm\infty$ instead. Return $\bigwedge_{i=1}^{n} x_i \in [\ell_i, u_i]$.

Let us now consider Item (F1) of Requirement 4.3. To establish this item, we must provide UXP reductions for computing, given $\Phi, \Psi \in \mathrm{dom}(\rho)$, (1) an element in $\mathrm{dom}(\rho)$ representing $\rho(\Phi) \wedge \rho(\Psi)$, and (2) whether $\rho(\Phi) \leq \rho(\Psi)$. Both problems can in fact be solved in polynomial time. Let $\Phi = \bigwedge_{j=1}^{m} x_i \in [\ell_i, u_i]$ and $\Psi = \bigwedge_{j=1}^{n} x_i \in [\ell'_i, u'_i]$. (Similar arguments hold when at least one among Φ or Ψ is $\top$ or $\bot$.) For the first problem, the algorithm simply returns $\bigwedge_{i=1}^{\max(n,m)} x_j \in [\max(\ell_j, \ell'_j), \min(u_j, u'_j)]$, where the values ℓ_j or ℓ'_j (resp. u_j or u'_j) not occurring in Φ or Ψ are defined as $-\infty$ (resp. $+\infty$). For the second problem, the algorithm returns true whenever either $u_i < \ell_i$ for some $i \in [1, m]$ (in this case, Φ is unsatisfiable), or $\ell'_i \leq \ell_i \leq u_i \leq u'_i$ for every $i \in [1, \max(n, m)]$.

To establish Item (F2), we must provide a $(\mathrm{len}(\mathbf{1})^2, \mathbf{1})$-UXP reduction for testing the inclusion $\mathrm{un}(\rho)(\Phi_1, \ldots, \Phi_j) \leq \mathrm{un}(\rho)(\Psi_1, \ldots, \Psi_k)$, where each Φ_i and Ψ_i belongs to $\mathrm{dom}(\rho)$. By definition of $\mathrm{un}(\rho)$, an algorithm for deciding $\rho(\Phi) \leq \mathrm{un}(\rho)(\Psi_1, \ldots, \Psi_k)$, with Φ in $\mathrm{dom}(\rho)$, suffices. Viewing elements of $\mathrm{dom}(\rho)$ as formulae in $\mathrm{CQ}(\sigma)$, this is equivalent to testing the unsatisfiability of $\Phi \wedge (\neg\Psi_1) \wedge \cdots \wedge (\neg\Psi_k)$. The parameter of the UXP reduction is $\mathrm{len}(\mathbf{1})$, and so it suffices to provide a $n^{\mathrm{poly}(k)}$ procedure. One such procedure consists of converting $\Phi \wedge (\neg\Psi_1) \wedge \cdots \wedge (\neg\Psi_k)$ into DNF, which can be done in $n^{\mathrm{poly}(k)}$ time, to then test the unsatisfiability of each disjunct. Since $\neg(x \leq k) \iff x \geq k + 1$, every disjunct belongs to $\mathrm{CQ}(\sigma)$, and its unsatisfiability can be decided in polynomial time. (Note: taking $k = 1$ provides another argument for solving the inclusion $\rho(\Phi) \wedge \rho(\Psi)$ required in Item (F1).) $\diamond$

Establishing Requirement 4.3 implies that the full Boolean algebra (including relative complementation) of $\mathrm{dfnf}(\mathrm{D})$ has the UXP signature that is required in Proposition 4.2.

Lemma 4.5 (Outcome of Requirement 4.3)**.** *Suppose that:*

(F1) *The structure* $(\mathrm{D}, \wedge, \leq)$ *has a* (ρ, θ)*-UXP signature.*
(F2) *The structure* $(\mathrm{un}(\mathrm{D}), \leq)$ *has a* $(\mathrm{un}(\rho), \mathrm{len}(\theta))$*-UXP signature.*

Then, the structure $(\mathrm{dfnf}(\mathrm{D}), \bot, \top, \vee, \wedge, -, \leq)$ *has a* $(\mathrm{dfnf}(\rho), \mathrm{dep}(\theta))$*-UXP signature.*

The proof of this lemma boils down to the definition of suitable UXP reductions implementing the binary operations $\wedge$, $\vee$ and $-$. We postpone this proof to Section 5, where we study in depth algorithmic aspects of the difference normal form.

Moving forward, we now consider projections and universal projections. Again, the goal is to minimise the efforts needed to add support for these operations. In this sense, the decision to adopt the difference normal form now becomes crucial. First, we need to introduce a variant of universal projection which we call relative universal projection. Given $Z \subseteq A^m$, $X \subseteq A^n$ and $\mathbf{i} = (i_1, \ldots, i_k) \in \mathbf{I}$, the *relative universal projection* $\pi^{\forall}_{Z}(\mathbf{i}, X)$ of X

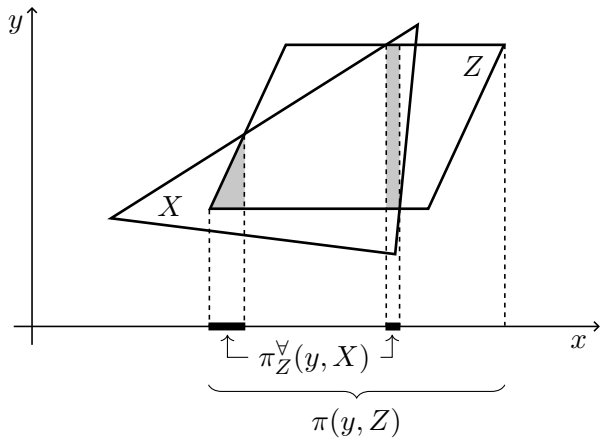

Figure 1: Illustration of the relative universal projection $\pi^{\forall}_Z(y, X)$.

with respect to Z is defined as follows, where $M \coloneqq \max(m, n)$:

$$\pi^{\forall}_Z(\mathbf{i}, X) \coloneqq \{(a_1, ..., a_M) \in A^M : \mathbf{a} \coloneqq (a_1, ..., a_m) \in \pi(\mathbf{i}, Z) \text{ and for all } \mathbf{b} \in A^k \\ \text{if } \mathbf{a}[\mathbf{i} \leftarrow \mathbf{b}] \in Z \text{ then } (a_1, ..., a_n)[\mathbf{i} \leftarrow \mathbf{b}] \in X\}.$$

Informally speaking, and illustrated in Figure 1, the set $\pi^{\forall}_Z(\mathbf{i}, X)$ acts as a universal projection for the part of X that lies inside Z. Note that $\pi^{\forall}(\mathbf{i}, X) = \pi^{\forall}_{\top}(\mathbf{i}, X)$.

The lemma below outlines a key "mutual distribution" property of projection and relative universal projection over relative complement. In the context of the difference normal form, this property allows us to disregard complementation when adding support for quantification.

Lemma 4.6. *Consider $X \subseteq A^n$, $Y \subseteq A^m$, $Z \subseteq A^r$, and $\mathbf{i} \in \mathbf{I}$. Then,*

$$\pi(\mathbf{i}, X - Y) = \pi(\mathbf{i}, X) - \pi^{\forall}_X(\mathbf{i}, Y) \quad \textit{and} \quad \pi^{\forall}_Z(\mathbf{i}, X - Y) = \pi^{\forall}_Z(\mathbf{i}, X) - \pi(\mathbf{i}, Y).$$

Example 4.7 (QBF)**.** Consider the formula Ψ from Example 4.1. Let $\mathbf{i} = (b, c)$. Following the equivalences in Lemma 4.6, we have

$$\pi(\mathbf{i}, [\![\Psi]\!]) \;=\; \pi(\mathbf{i}, [\![a \vee b]\!]) - \Big(\pi^{\forall}_{[\![a \vee b]\!]}(\mathbf{i}, [\![a \vee (b \wedge c)]\!]) - \big(\pi(\mathbf{i}, [\![a \wedge c]\!]) - \pi^{\forall}_{[\![a \wedge c]\!]}(\mathbf{i}[\![a \wedge b \wedge c]\!])\big)\Big).$$

It is easy to see that:

- $\pi(\mathbf{i}, [\![a \vee b]\!])$ corresponds to $[\![\top]\!]$. In other words, $\exists b, c : a \vee b$ is a valid formula.
- $\pi^{\forall}_{[\![a \vee b]\!]}(\mathbf{i}, [\![a \vee (b \wedge c)]\!])$ corresponds to $[\![a]\!]$; i.e., $a \iff (\forall b, c : a \vee b \Rightarrow a \vee (b \wedge c))$ is valid.
- $\pi(\mathbf{i}, [\![a \wedge c]\!])$ corresponds to $[\![a]\!]$, whereas $\pi^{\forall}_{[\![a \wedge c]\!]}(\mathbf{i}, [\![a \wedge b \wedge c]\!])$ corresponds to $[\![\bot]\!]$.

Therefore $\pi(\mathbf{i}, [\![\Psi]\!]) = [\![\top]\!] - ([\![a]\!] - ([\![a]\!] - [\![\bot]\!])) = [\![\top]\!]$. Moreover, since $\pi^{\forall}(a, [\![\top]\!]) = [\![\top]\!]$, it follows that the formula $\forall a \exists b \exists c\, \Phi$ from Example 4.1 is valid. ◇

We postpone the proof of Lemma 4.6 and of the forthcoming Lemma 4.9 to Section 6. Below, let us write $\dot{\pi}$ for the restriction of the projection operator π on inputs $(\mathbf{i}, X)$ where $X \in \mathrm{D}$, and write $\dot{\pi}^{\forall}$ for the restriction of the relativised universal projection $\pi^{\forall}$ on inputs $(Z, \mathbf{i}, X)$ where $Z \in \mathrm{D}$ and $X \in \mathrm{un}(\mathrm{D})$. Thanks to Lemma 4.6, adding to Requirement 4.3 the following requirement is sufficient to conclude that $\mathcal{D}$ has the UXP signature required by Proposition 4.2.

Requirement 4.8 (Projection and universal projection)**.** *Establish the following properties:*
(F3) *For every* $X \in \mathrm{D}$ *and* $\mathbf{i} \in \mathbf{I}$*,* $\dot{\pi}(\mathbf{i}, X)$ *belongs to* $\mathrm{dfnf}(\mathrm{D})$.
(F4) *For every* $Z \in \mathrm{D}$*,* $\mathbf{i} \in \mathbf{I}$ *and* $X \in \mathrm{un}(\mathrm{D})$*,* $\dot{\pi}^{\forall}_{Z}(\mathbf{i}, X)$ *belongs to* $\mathrm{dfnf}(\mathrm{D})$.
(F5) *There is a* $(\mathbf{1} \cdot \theta, \mathrm{dep}(\theta))$*-UXP reduction* $(\nu_{\mathbf{I}} \times \rho, \mathrm{dfnf}(\rho))$*-implementing* $\dot{\pi}$.
(F6) *There is a* $(\theta \cdot \mathbf{1} \cdot \mathrm{len}(\theta), \mathrm{dep}(\theta))$*-UXP reduction* $(\rho \times \nu_{\mathbf{I}} \times \mathrm{un}(\rho), \mathrm{dfnf}(\rho))$*-implementing* $\dot{\pi}^{\forall}$.

Lemma 4.9 (Outcome of Requirement 4.3 and 4.8)**.** *Assume (F1)–(F6) to hold. Then, the structure* $\mathcal{D}$ *from Proposition 4.2 has a* $(\mathrm{dfnf}(\rho), \mathrm{dep}(\theta))$*-UXP signature.*

Example 4.10 (PA with one variable per inequality, cont.)**.** We show how to implement Requirement 4.8 on the fragment of Presburger arithmetic introduced in Example 4.4. Recall that $\mathrm{D} \coloneqq \{[\![\Psi]\!] : \Psi \in \mathrm{CQ}(\sigma)\}$, and $\mathrm{dom}(\rho)$ is the set of systems of constraints of the form $\bigwedge_{i=1}^{m} x_i \in [\ell_i, u_i]$, where $\ell_i \in \mathbb{Z} \cup \{-\infty\}$ and $u_i \in \mathbb{Z} \cup \{+\infty\}$, for every $i \in [1, m]$, plus two elements $\top$ and $\bot$.

Establishing Items (F3) and (F5) is straightforward. Let Φ be an element of $\mathrm{dom}(\rho)$, and consider a vector of variables $\mathbf{x}$. Item (F3) asks to show that the set of solutions to $\exists \mathbf{x}\, \Phi$ is an element $S \in \mathrm{dfnf}(\mathrm{D})$. The next item asks for an algorithm to compute an element of $\mathrm{dom}(\mathrm{dfnf}(\rho))$ that represent S. In fact, we can compute in polynomial time an element of $\mathrm{dom}(\rho)$ that represents S. If Φ is $\top$ or $\bot$, we simply return Φ. Let $\Phi = \bigwedge_{i=1}^{m} x_i \in [\ell_i, u_i]$. If $u_i < \ell_i$ for some x_i in $\mathbf{x}$, then $\exists \mathbf{x}\, \Phi$ is unsatisfiable, and we can simply return $\bot$. Otherwise, remove $x_i \in [\ell_i, u_i]$ from the constraints in Φ. After repeating this step for all the variables in $\mathbf{x}$, the algorithm returns the resulting system.

Let us move to Items (F4) and (F6). These items ask for an algorithm that given in input $\Phi \in \mathrm{dom}(\rho)$, a vector of variables $\mathbf{x}$, and $(\Psi_1, \ldots, \Psi_k) \in \mathrm{dom}(\mathrm{un}(\rho))$, returns an element of $\mathrm{dom}(\mathrm{dfnf}(\rho))$ representing the set of solutions S to $(\exists \mathbf{x} \Phi) \wedge \forall \mathbf{x}(\Phi \Rightarrow \Psi_1 \vee \cdots \vee \Psi_k)$. For simplicity, let us suppose $\mathbf{x} = (x_{n+1}, \ldots, x_m)$, $\Phi = \bigwedge_{i=1}^{m} x_i \in [\ell_i, u_i]$ and $\Psi_j = \bigwedge_{i=1}^{m} x_i \in [\ell_{ji}, u_{ji}]$. We first present an algorithm that does not yield the correct UXP reduction, to illustrate potential pitfalls when instantiating the framework, and then provide the correct algorithm. The "flawed" algorithm tries to perform quantifier elimination on $\forall \mathbf{x}$ in a naïve way:

1: rewrite $\forall \mathbf{x}(\Phi \Rightarrow \Psi_1 \vee \cdots \vee \Psi_k)$ as $\neg \exists \mathbf{x}(\Phi \wedge \neg(\Psi_1) \wedge \ldots \neg(\Psi_k))$
2: bring $(\Phi \wedge \neg(\Psi_1) \wedge \ldots \neg(\Psi_k))$ in DNF, obtaining a formula $\bigvee_{j=1}^{M} \Phi_j$
3: for every $j \in [1, M]$ compute a system Φ'_j in $\mathrm{dom}(\rho)$, equivalent to $\exists \mathbf{x}\, \Phi_j$
4: compute a system Φ' in $\mathrm{dom}(\rho)$, equivalent to $\exists \mathbf{x}\, \Phi$
5: **return** $\Phi' - (\bigvee_{j=1}^{M} \Phi'_j)$

The problem with this algorithm is that, in the worst case, the number of disjuncts M is $\Omega(n^k)$; and so the parameter $\mathrm{dep}(\mathbf{1})$ of the output is not bounded by any function in the parameter $\mathbf{1} \cdot \mathbf{1} \cdot \mathrm{len}(\mathbf{1})$ of the input. In particular, the number of disjuncts must not depend on n. Note, however, that the running time $m^{\mathrm{poly}(k)}$ of the algorithm is unproblematic. Our main mistake in designing this algorithm is not using any property of the theory at hand.

We now provide a correct algorithm. Let $\mathbf{y} \coloneqq (x_1, \ldots, x_n)$ denote the variables we are not projecting away. A vector $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{Z}^n$ is in the set of solutions S whenever it satisfies $\exists \mathbf{x} \Phi$ and the sentence obtained from $\forall \mathbf{x}(\Phi \Rightarrow \Psi_1 \vee \cdots \vee \Psi_k)$ by replacing every variable in $\mathbf{y}$ with the corresponding value in $\mathbf{v}$ is a tautology. If $\mathbf{v}$ satisfies $\exists \mathbf{x} \Phi$, then the latter sentence is equivalent to $\forall \mathbf{x}(\Phi' \Rightarrow \Psi'_1 \vee \cdots \vee \Psi'_k)$ where $\Phi' = \bigwedge_{i=n+1}^{m} x_i \in [\ell_i, u_i]$ and each Ψ'_j is either $\bot$ (if $\mathbf{v}$ is not a solution to $\exists \mathbf{x} \Psi_j$) or it is $\bigwedge_{i=n+1}^{m} x_i \in [\ell_{ji}, u_{ji}]$. Crucially, there are only 2^k such sentences. This observation yields the following algorithm:

1: $T \leftarrow$ the empty list
2: **for** $J \subseteq [1,k]$ **do**
3: **if** $\forall \mathbf{x}((\bigwedge_{i=n+1}^{m} x_i \in [\ell_i, u_i]) \implies \bigvee_{j\in J} \bigwedge_{i=n+1}^{m} x_i \in [\ell_{ji}, u_{ji}])$ is a tautology **then**
4: let $\ell'_i \coloneqq \max\{\ell_i, \ell_{ji} : j \in J\}$ and $u'_i \coloneqq \min\{u_i, u_{ji} : j \in J\}$, for all $i \in [1,n]$
5: append $\bigwedge_{i=1}^{n} x_i \in [\ell'_i, u'_i]$ to the list T
6: **return** T ▷ T is an element of $\mathrm{dom}(\mathrm{un}(\rho))$

Intuitively, Lines 4 and 5 construct a formula whose solutions satisfy $\exists \mathbf{x}\Phi$ and ensure the sentence in Line 3 is a tautology. The ***for*** loop enumerates all such tautologies. The output T therefore contains 2^k systems of constraints; in other words, $\mathrm{dep}(\mathbf{1})(T) \leq 2^{(\mathbf{1}\cdot\mathbf{1}\cdot \mathrm{len}(\mathbf{1}))(\Phi,\mathbf{x},(\Psi_1,\ldots,\Psi_k))}$. Checking that the sentences in Line 3 are tautologies can be done using our "flawed" algorithm, which runs in $(m-n)^{\mathrm{poly}(k)}$ time. Hence, the overall running time is bounded by $m^{\mathrm{poly}(k)}$ (assuming $m \geq 2$). We conclude that this algorithm achieves the UXP reduction required in Items (F3) and (F5). ◇

5. The difference normal form of propositional logic

In this section, we explore computational aspects of the difference normal form. As a result of this exploration, we will establish Lemma 4.5 from Section 4. To simplify the presentation, we first focus our study on propositional logic, to then extend it to first-order theories. Recall that formulae in propositional logic are from the following grammar:

$$\Phi, \Psi \coloneqq \bot \mid \top \mid p \mid \neg\Phi \mid \Phi \wedge \Psi \mid \Phi \vee \Psi$$

where p is an atomic proposition from a countable set $\mathcal{AP}$. We write $[\![\Phi]\!]$ for the set of valuations $v\colon \mathcal{AP} \to \{\bot, \top\}$ making Φ true. Recall that $(\Phi - \Psi) \coloneqq \Phi \wedge \neg\Psi$.

We study the problem of bringing formulae in propositional logic into *strict* difference normal form (SDF). A formula $\Psi \coloneqq \Phi_1 - (\Phi_2 - (\cdots - (\Phi_{k-1} - \Phi_k)))$ in difference normal form is said to be strict whenever $[\![\Phi_k]\!] = \emptyset$ and $[\![\Phi_{i+1}]\!] \subsetneq [\![\Phi_i]\!]$, for every $i \in [1, k-1]$. Note that $[\![\Psi]\!] = \emptyset$ if and only if $k = 1$. Every Φ_i is a negation-free DNF formula, and so checking the strict entailment $[\![\Phi_j]\!] \subsetneq [\![\Phi_i]\!]$ can be done in polynomial time in the size of Φ_i and Φ_j: it suffices to iterate through the disjuncts of Φ_j, and check that each of them is syntactically contained in a disjunct of Φ_i, modulo associativity and commutativity of $\wedge$ (e.g., $a \wedge b$ is syntactically contained in $b \wedge c \wedge a$). We write $\mathrm{SDF}(\Sigma)$ for the set of all formulae in SDF having atomic propositions from a finite set $\Sigma \subseteq \mathcal{AP}$. We write $\mathrm{DNF}_+(\Sigma)$ for the set of all formulae in negation-free DNF having atomic propositions from Σ.

5.1. Operations on formulae in strict difference normal form.

We first study the *cons* operation $(:)\colon \mathrm{DNF}_+(\Sigma) \times \mathrm{SDF}(\Sigma) \to \mathrm{SDF}(\Sigma)$ that on an input (Φ, Ψ) returns an SDF representing $\Phi - \Psi$. Given $\Psi = \Phi_1 - (\cdots - \Phi_k)$, this operator is recursively defined as follows:

$$\Phi : \Psi \coloneqq \begin{cases} \bot & \text{if } [\![\Phi]\!] = \emptyset \\ \Phi - \bot & \text{if } [\![\Phi_1]\!] = \emptyset \\ \Phi - (\Phi_1 - (\cdots - \Phi_k)) & \text{else if } [\![\Phi_1]\!] \subsetneq [\![\Phi]\!] \\ \Phi - \big((\Phi \mathbin{\textcircled{\wedge}} \Phi_1) : (\Phi_2 - (\cdots - \Phi_k))\big) & \text{else if } [\![\Phi \mathbin{\textcircled{\wedge}} \Phi_1]\!] \subsetneq [\![\Phi]\!] \\ (\Phi \mathbin{\textcircled{\wedge}} \Phi_2) : (\Phi_3 - (\cdots - \Phi_k)) & \text{else if } k \geq 3 \\ \bot & \text{otherwise} \end{cases}$$

where, given $\Phi' = \bigvee_{i=1}^{r} \varphi'_i$ and $\Phi'' = \bigvee_{j=1}^{s} \varphi''_j$ in $\mathrm{DNF}_+(\Sigma)$ and having r and s disjuncts respectively, $\Phi' \mathbin{⩓} \Phi''$ is short for the negation-free DNF formula $\bigvee_{i=1}^{r} \bigvee_{j=1}^{s} \varphi'_i \wedge \varphi''_j$. A simple induction on k shows that (:) is well-defined.

Lemma 5.1. *The operator* (:) *respects the type* $\mathrm{DNF}_+(\Sigma) \times \mathrm{SDF}(\Sigma) \to \mathrm{SDF}(\Sigma)$. *Consider a formula* Φ *in* $\mathrm{DNF}_+(\Sigma)$, *and* $\Psi = \Phi_1 - (\cdots - \Phi_k)$ *in* $\mathrm{SDF}(\Sigma)$. *Then,* $\Phi : \Psi = \Phi'_1 - (\cdots - \Phi'_\ell)$ *with* $\ell \leq k+1$, $[\![\Phi'_1]\!] \subseteq [\![\Phi]\!]$ *and, for each* $j \in [1, \ell]$, $\Phi'_j = ⩓ Y_j$ *for some* $Y_j \subseteq \{\bot, \Phi, \Phi_1, \ldots, \Phi_k\}$. *Furthermore,* $[\![\Phi : \Psi]\!] = [\![\Phi]\!] \setminus [\![\Psi]\!]$.

Proof. The proof is by induction on the length k of Ψ.

base case: $k = 1$**:** We have $\Psi = \Phi_1$ and $[\![\Psi]\!] = \emptyset$. By definition of (:), either $\Phi : \Psi = \bot$ or $\Phi : \Psi = \Phi - \bot$, depending on whether $[\![\Phi]\!] = \emptyset$. In both cases, the lemma is satisfied.

induction step: $k \neq 1$**:** In this case, $[\![\Phi_1]\!] \neq \emptyset$. If $[\![\Phi]\!] = \emptyset$ then $\Phi : \Psi = \emptyset$, and again the lemma is trivially satisfied. Below, assume $[\![\Phi]\!] \neq \emptyset$. We divide the proof following the last three cases in the definition of (:) :

case: $[\![\Phi_1]\!] \subsetneq [\![\Phi]\!]$**:** By definition, $\Phi : \Psi = \Phi - (\Phi_1 - (\cdots - \Phi_k))$, which belongs to $\mathrm{SDF}(\Sigma)$ and respects all properties of the lemma.

case: $[\![\Phi \mathbin{⩓} \Phi_1]\!] \subsetneq [\![\Phi]\!]$ **and not** $[\![\Phi_1]\!] \subsetneq [\![\Phi]\!]$**:** Let $\Psi' \coloneqq \Phi_2 - (\cdots - \Phi_k)$. By definition, $\Phi : \Psi = \Phi - ((\Phi \mathbin{⩓} \Phi_1) : \Psi')$. Since Ψ' has length $k-1$, by the induction hypothesis, $(\Phi \mathbin{⩓} \Phi_1) : \Psi' = \Phi'_1 - (\cdots - \Phi'_\ell)$ where $\ell \leq k$, $[\![\Phi'_1]\!] \subseteq [\![\Phi \mathbin{⩓} \Phi_1]\!]$, and for every $j \in [1, \ell]$, $\Phi'_j = ⩓ Y_j$ for some $Y_j \subseteq \{\bot, (\Phi \mathbin{⩓} \Phi_1), \Phi_2, \ldots, \Phi_k\}$. Moreover, $[\![(\Phi \mathbin{⩓} \Phi_1) : \Psi']\!] = [\![\Phi \mathbin{⩓} \Phi_1]\!] \setminus [\![\Psi']\!]$. Since $[\![\Phi'_1]\!] \subseteq [\![\Phi \mathbin{⩓} \Phi_1]\!] \subsetneq [\![\Phi]\!]$, the formula $\Phi - (\Phi'_1 - (\cdots - \Phi'_\ell))$ is in $\mathrm{SDF}(\Sigma)$, and all properties required by the lemma are satisfied.

case: **not** $[\![\Phi \mathbin{⩓} \Phi_1]\!] \subsetneq [\![\Phi]\!]$ **and** $k \geq 3$**:** Let $\Psi' \coloneqq \Phi_3 - (\cdots - \Phi_k)$. By definition, $\Phi : \Psi = (\Phi \mathbin{⩓} \Phi_2) : \Psi'$. Since Ψ' has length $k-2$, by the induction hypothesis, $(\Phi \mathbin{⩓} \Phi_2) : \Psi' = \Phi'_1 - (\cdots - \Phi'_\ell)$ with $\ell \leq k-1$, $[\![\Phi'_1]\!] \subseteq [\![\Phi \mathbin{⩓} \Phi_2]\!]$, $[\![(\Phi \mathbin{⩓} \Phi_2) : \Psi']\!] = [\![\Phi \mathbin{⩓} \Phi_2]\!] \setminus [\![\Psi']\!]$, and for every $j \in [1, \ell]$, $\Phi'_j = ⩓ Y_j$ for some $Y_j \subseteq \{\bot, (\Phi \mathbin{⩓} \Phi_2), \Phi_3, \ldots, \Phi_k\}$. Since $[\![\Phi : \Psi]\!] = [\![(\Phi \mathbin{⩓} \Phi_2) : \Psi']\!]$, to conclude the proof of this case it suffices to show that $[\![\Phi \mathbin{⩓} \Phi_2]\!] \setminus [\![\Psi']\!] = [\![\Phi]\!] \setminus [\![\Psi]\!]$.
By definition of ⩓, $[\![\Phi \mathbin{⩓} \Phi_1]\!] \subseteq [\![\Phi]\!]$, and since $[\![\Phi \mathbin{⩓} \Phi_1]\!]$ is not strictly included in $[\![\Phi]\!]$, we have $[\![\Phi \mathbin{⩓} \Phi_1]\!] = [\![\Phi]\!]$, which implies $[\![\Phi]\!] \subseteq [\![\Phi_1]\!]$. From the validity "$A \subseteq B$ implies $A \setminus (B \setminus C) = A \cap C$" we get $[\![\Phi]\!] \setminus [\![\Psi]\!] = [\![\Phi]\!] \cap ([\![\Phi_2]\!] \setminus [\![\Psi']\!])$. Then, from the validity $A \cap (B \setminus C) = (A \cap B) \setminus C$ together with the definition of ⩓, we conclude that $[\![\Phi]\!] \setminus [\![\Psi]\!] = [\![\Phi \mathbin{⩓} \Phi_2]\!] \setminus [\![\Psi']\!]$.

case: **not** $[\![\Phi \mathbin{⩓} \Phi_1]\!] \subsetneq [\![\Phi]\!]$ **and** $k = 2$**:** In this case $\Psi = \Phi_1 - \bot$. By definition $\Phi : \Psi = \bot$, and to conclude the proof it suffices to show that $[\![\Phi]\!] \setminus [\![\Psi]\!] = \emptyset$. This follows from the fact that, as in the previous case, we have $[\![\Phi]\!] \subseteq [\![\Phi_1]\!]$. □

We introduce the operations of *union* $\curlyvee$, *intersection* $\curlywedge$ and *difference* $\frown$, with type $\mathrm{SDF}(\Sigma) \times \mathrm{SDF}(\Sigma) \to \mathrm{SDF}(\Sigma)$, that given a pair of SDFs (Φ, Ψ) compute an SDF representing $\Phi \vee \Psi$, $\Phi \wedge \Psi$ and $\Phi - \Psi$, respectively. These three operators have mutually recursive definitions. Below, whenever their length is not one (i.e., $[\![\Phi]\!] \neq \emptyset$ or $[\![\Psi]\!] \neq \emptyset$), we assume Φ and Ψ to be respectively $\Phi = \Phi_1 - \Phi'$ and $\Psi = \Psi_1 - \Psi'$, with $\Phi_1, \Psi_1 \in \mathrm{DNF}_+(\Sigma)$ and $\Phi', \Psi' \in \mathrm{SDF}(\Sigma)$.

$$\Phi \curlywedge \Psi := \begin{cases} \bot & \text{if } [\![\Phi]\!] = \emptyset \text{ or } [\![\Psi]\!] = \emptyset, \\ (\Phi_1 \mathbin{⩓} \Psi_1) : (\Phi' \curlyvee \Psi') & \text{otherwise.} \end{cases}$$

$$\Phi \frown \Psi := \begin{cases} \Phi & \text{if } [\![\Phi]\!] = \emptyset \text{ or } [\![\Psi]\!] = \emptyset, \\ \Phi_1 : (\Phi' \curlyvee \Psi) & \text{otherwise.} \end{cases}$$

$$\Phi \curlyvee \Psi := \begin{cases} \Psi & \text{if } [\![\Phi]\!] = \emptyset, \\ \Phi & \text{else if } [\![\Psi]\!] = \emptyset, \\ \Phi_1 : (\Phi' \frown \Psi) & \text{else if } [\![\Psi_1]\!] \subseteq [\![\Phi_1]\!], \\ (\Phi_1 \vee \Psi_1) : \big((\Phi' \curlyvee \Psi') \frown ((\Phi_1 \owedge \Psi_1) : (\Phi' \curlywedge \Psi'))\big) & \text{otherwise.} \end{cases}$$

We show that the operators $\curlywedge$, $-$ and $\curlyvee$ always yield an element of $\mathrm{SDF}(\Sigma)$. In particular, the computation of such an element always terminates.

Lemma 5.2. *The operators $\curlyvee$, $\curlywedge$, $\frown$ are of type $\mathrm{SDF}(\Sigma) \times \mathrm{SDF}(\Sigma) \to \mathrm{SDF}(\Sigma)$. Given two formulae Φ and Ψ in $\mathrm{SDF}(\Sigma)$, we have $[\![\Phi \curlyvee \Psi]\!] = [\![\Phi]\!] \cup [\![\Psi]\!]$, $[\![\Phi \curlywedge \Psi]\!] = [\![\Phi]\!] \cap [\![\Psi]\!]$, and $[\![\Phi \frown \Psi]\!] = [\![\Phi]\!] \setminus [\![\Psi]\!]$. Consider $\oplus \in \{\curlyvee, \curlywedge, \frown\}$, $\Phi = \Phi_1 - (\cdots - \Phi_i)$ and $\Psi = \Psi_1 - (\cdots - \Psi_j)$ and $\Phi \oplus \Psi = \Phi'_1 - (\cdots - \Phi'_\ell)$, and let $k \in [1, \ell]$. We have $[\![\Phi'_k]\!] = [\![\bigvee G_k]\!]$ where G_k is a subset of formulae from $\{\varphi,\ \psi,\ \varphi \wedge \psi : \varphi \in \{\Phi_1, \ldots, \Phi_i\},\ \psi \in \{\Psi_1, \ldots, \Psi_j\}\}$.*

Proof. We show the statement by well-founded induction. The domain of the induction is $(\mathrm{SDF}(\Sigma) \times \mathrm{SDF}(\Sigma), <)$, whose elements correspond to inputs of the operators $\curlyvee$, $\curlywedge$, $\frown$, where the order $<$ is defined as follows: for every (X, Y) and (W, Z) in $\mathrm{SDF}(\Sigma) \times \mathrm{SDF}(\Sigma)$, with $X = X_1 - (\cdots - X_{\ell_X})$, $Y = Y_1 - (\cdots - Y_{\ell_Y})$, $W = W_1 - (\cdots - W_{\ell_W})$ and $Z = Z_1 - (\cdots - Z_{\ell_Z})$,

$$(X, Y) < (W, Z) \text{ if and only if } \begin{array}{ll} \text{(i)} & [\![Y_1]\!] \subsetneq [\![Z_1]\!], \text{ or} \\ \text{(ii)} & \ell_X < \ell_W \text{ and } [\![Y_1]\!] \subseteq [\![Z_1]\!]. \end{array}$$

According to $<$, the only (required) base case corresponds to (X, Y) such that $\ell_X = 1$ and $[\![Y]\!] = \emptyset$. For simplicity, we treat as base cases all pairs (X, Y) where $\ell_X = 1$ or $[\![Y]\!] = \emptyset$.

Below, we refer to the objects in the statement of the lemma.

base case: $i = 1$: In this case, $[\![\Phi]\!] = \emptyset$. We have $\Phi \curlyvee \Psi = \Psi$, $\Phi \curlywedge \Psi = \bot$ and $\Phi \frown \Psi = \Phi$. In all cases, the statement of the lemma is satisfied.

base case: $[\![\Psi]\!] = \emptyset$: We have $\Phi \curlyvee \Psi = \Phi$, $\Phi \curlywedge \Psi = \bot$ and $\Phi \frown \Psi = \Phi$. Again, the statement of the lemma is trivially satisfied.

For the induction step, assume $i \geq 2$ and $[\![\Psi]\!] \neq \emptyset$. Note that then $[\![\Phi]\!] \neq \emptyset$ and $j \geq 2$, by definition of SDF. For brevity, let $\Phi' \coloneqq \Phi_2 - (\cdots - \Phi_i)$ and $\Psi' \coloneqq \Psi_2 - (\cdots - \Psi_j)$. We analyse all operators separately.

induction step: case $\Phi \curlywedge \Psi$: By definition, $\Phi \curlywedge \Psi = (\Phi_1 \owedge \Psi_1) : (\Phi' \curlyvee \Psi')$. From the fact that $[\![\Psi_2]\!] \subsetneq [\![\Psi_1]\!]$, and by the induction hypothesis, we conclude that the statement of the lemma holds for $\Phi' \curlyvee \Psi'$. In particular, this means that

(1) $[\![\Phi' \curlyvee \Psi']\!] = [\![\Phi']\!] \cup [\![\Psi']\!]$,
(2) $\Phi' \curlyvee \Psi' = (\Phi'_1 - (\cdots - \Phi'_\ell))$ where, for every $k \in [1, \ell]$, $[\![\Phi'_i]\!] = [\![\bigvee G_k]\!]$ with $G_k \subseteq \{\varphi,\ \psi,\ \varphi \wedge \psi : \varphi \in \{\Phi_2, \ldots, \Phi_i\},\ \psi \in \{\Psi_2, \ldots, \Psi_j\}\}$.

By Lemma 5.1, $(\Phi_1 \owedge \Psi_1) : (\Phi'_1 - (\cdots - \Phi'_\ell)) = (\Psi'_1 - (\cdots - \Psi'_r))$ where

(3) $[\![(\Phi_1 \owedge \Psi_1) : (\Phi'_1 - (\cdots - \Phi'_\ell))]\!] = [\![\Phi_1 \owedge \Psi_1]\!] \setminus [\![(\Phi'_1 - (\cdots - \Phi'_\ell))]\!]$,
(4) for all $k \in [1, r]$, $\Psi'_k = \owedge Y_k$ with $Y_k \subseteq \{\bot, (\Phi_1 \owedge \Psi_1), \Phi'_1, \ldots, \Phi'_\ell\}$.

From Items 1 and 3, together with identities of set theory, we get $[\![\Phi \curlywedge \Psi]\!] = [\![\Phi]\!] \cap [\![\Psi]\!]$:

$$\begin{aligned}
[\![\Phi \curlywedge \Psi]\!] &= [\![\Phi_1 \barwedge \Psi_1]\!] \setminus ([\![\Phi']\!] \cup [\![\Psi']\!]) && \text{by Items 1 and 3}\\
&= ([\![\Phi_1]\!] \cap [\![\Psi_1]\!]) \setminus ([\![\Phi']\!] \cup [\![\Psi']\!]) && \text{definition of } \barwedge\\
&= ([\![\Phi_1]\!] \setminus [\![\Phi']\!]) \cap ([\![\Psi_1]\!] \setminus [\![\Psi']\!]) && \text{set-theoretical validity}\\
& && (A \setminus B) \cap (C \setminus D) = (A \cap C) \setminus (B \cup D)\\
&= [\![\Phi]\!] \cap [\![\Psi]\!] && \text{definition of } \Phi \text{ and } \Psi.
\end{aligned}$$

From Items 2 and 4 and by definition of $\barwedge$, for every $k \in [1, r]$, $[\![\Psi'_k]\!] = [\![\bigvee D_k]\!]$ where D_k is a set of formulae of the form $\varphi_1 \wedge \cdots \wedge \varphi_n$ with $\varphi_p \in \{\Phi_1, \ldots, \Phi_i, \Psi_1, \ldots, \Psi_j\}$ for every $p \in [1, n]$ (recall here that $[\![\Psi_j]\!] = [\![\Phi_i]\!] = [\![\bot]\!] = \emptyset$, by definition of SDF). Since Φ and Ψ are in SDF, it is not necessary for $\varphi_1 \wedge \cdots \wedge \varphi_n$ to contain distinct formulae among $\Phi_1, \ldots, \Phi_j$, as one of them will imply the others. The same is true for formulae among $\Psi_1, \ldots, \Psi_j$. Therefore, $\varphi_1 \wedge \cdots \wedge \varphi_n$ can be simplified into a formula of the form ψ_1, ψ_2 or $\psi_1 \wedge \psi_2$, with $\psi_1 \in \{\Phi_1, \ldots, \Phi_i\}$ and $\psi_2 \in \{\Psi_1, \ldots, \Psi_j\}$. This completes the proof for the case $\curlywedge$.

induction step: case $\Phi \frown \Psi$: By definition, $\Phi \frown \Psi = \Phi_1 : (\Phi' \curlyvee \Psi)$. Since Φ' has length $i-1$, by the induction hypothesis the statement of the lemma holds for $\Phi' \curlyvee \Psi$. The rest of the proof follows similarly to the previous case. In particular, $[\![\Phi \frown \Psi]\!] = [\![\Phi]\!] \setminus [\![\Psi]\!]$ is established thanks to the set-theoretical validity $(A \setminus B) \setminus C = A \setminus (B \cup C)$ together with $[\![\Phi_1 : (\Phi' \curlyvee \Psi)]\!] = [\![\Phi_1]\!] \setminus ([\![\Phi']\!] \cup [\![\Psi]\!])$, which follows by the induction hypothesis and Lemma 5.1.

induction step: case $\Phi \curlyvee \Psi$, and $[\![\Psi_1]\!] \subseteq [\![\Phi_1]\!]$: By definition, $\Phi \curlyvee \Psi = \Phi_1 : (\Phi' \frown \Psi)$. Once more, since Φ' has length $i-1$, by the induction hypothesis the statement of the lemma holds for $\Phi' \frown \Psi$. The rest of the proof follows similarly to the previous cases. In particular, $[\![\Phi \curlyvee \Psi]\!] = [\![\Phi]\!] \cup [\![\Psi]\!]$ follows from the validity "$C \subseteq A$ implies $(A \setminus B) \cup C = (A \setminus (B \setminus C))$".

induction step: case $\Phi \curlyvee \Psi$, and $[\![\Psi_1]\!] \not\subseteq [\![\Phi_1]\!]$: This case is a bit more involved, so we give full details. By definition,

$$\Phi \curlyvee \Psi = (\Phi_1 \vee \Psi_1) : \big((\Phi' \curlyvee \Psi') \frown ((\Phi_1 \barwedge \Psi_1) : (\Phi' \curlywedge \Psi'))\big).$$

We proceed with a series of applications of Lemma 5.1 and the induction hypothesis. Since $[\![\Psi_2]\!] \subsetneq [\![\Psi_1]\!]$, by the induction hypothesis,

(1) $[\![\Phi' \curlywedge \Psi']\!] = [\![\Phi']\!] \cap [\![\Psi']\!]$,
(2) $\Phi' \curlywedge \Psi' = \Phi'_1 - (\cdots - \Phi'_\ell)$ where, for all $k \in [1, \ell]$, $[\![\Phi'_k]\!] = [\![\bigvee G_k]\!]$ with $G_k \subseteq \{\varphi, \psi, \varphi \wedge \psi : \varphi \in \{\Phi_2, \ldots, \Phi_i\},\ \psi \in \{\Psi_2, \ldots, \Psi_j\}\}$,
(3) $[\![\Phi' \curlyvee \Psi']\!] = [\![\Phi']\!] \cup [\![\Psi']\!]$,
(4) $\Phi' \curlyvee \Psi' = \Phi''_1 - (\cdots - \Phi''_s)$ where, for all $k \in [1, s]$, $[\![\Phi''_k]\!] = [\![\bigvee D_k]\!]$ with $D_k \subseteq \{\varphi, \psi, \varphi \wedge \psi : \varphi \in \{\Phi_2, \ldots, \Phi_i\},\ \psi \in \{\Psi_2, \ldots, \Psi_j\}\}$.

By Lemma 5.1, $(\Phi_1 \barwedge \Psi_1) : (\Phi' \curlywedge \Psi') = \Psi'_1 - (\cdots - \Psi'_r)$ where

(5) $[\![(\Phi_1 \barwedge \Psi_1) : (\Phi' \curlywedge \Psi')]\!] = [\![\Phi_1 \barwedge \Psi_1]\!] \setminus [\![\Phi' \curlywedge \Psi']\!]$,
(6) for all $k \in [1, r]$, $\Psi'_k = \barwedge Y_k$ where $Y_k \subseteq \{\bot, (\Phi_1 \barwedge \Psi_1), \Phi'_1, \ldots, \Phi'_\ell\}$,
(7) $[\![\Psi'_1]\!] \subseteq [\![\Phi_1 \barwedge \Psi_1]\!]$.

Since $[\![\Psi_1]\!] \not\subseteq [\![\Phi_1]\!]$ we have $[\![\Phi_1 \barwedge \Psi_1]\!] \subsetneq [\![\Psi_1]\!]$. Because of Item 7, by induction hypothesis we get $(\Phi''_1 - (\cdots - \Phi''_s)) \frown (\Psi'_1 - (\cdots - \Psi'_r)) = \Psi''_1 - (\cdots - \Psi''_t)$, where

(8) $[\![(\Phi''_1 - (\cdots - \Phi''_s)) \frown (\Psi'_1 - (\cdots - \Psi'_r))]\!] = \big([\![\Phi''_1 - (\cdots - \Phi''_s)]\!] \setminus [\![\Psi'_1 - (\cdots - \Psi'_r)]\!]\big)$,

(9) for every $k \in [1,\ell]$, $[\![\Phi'_k]\!] = [\![\bigvee E_k]\!]$
with $E_k \subseteq \{\varphi,\, \psi,\, \varphi \wedge \psi : \varphi \in \{\Phi''_1, \dots, \Phi''_s\},\ \psi \in \{\Psi'_1, \dots, \Psi'_r\}\}$.

By Lemma 5.1, $(\Phi_1 \vee \Psi_1) : (\Psi''_1 - (\cdots - \Psi''_t)) = \widetilde{\Phi}_1 - (\cdots - \widetilde{\Phi}_u)$ where

(10) $[\![(\Phi_1 \vee \Psi_1) : (\Psi''_1 - (\cdots - \Psi''_t))]\!] = [\![\Phi_1 \vee \Psi_1]\!] \setminus [\![\Psi''_1 - (\cdots - \Psi''_t)]\!]$,

(11) for all $k \in [1,u]$, $\widetilde{\Psi}_k = \bigwedge\!\!\!\bigcirc Z_k$ where $Z_k \subseteq \{\bot, (\Phi_1 \vee \Psi_1), \Psi''_1, \dots, \Psi''_t\}$.

Now, by Items 1, 3, 5, 8 and 10 and by definition of $\bigwedge\!\!\!\bigcirc$, we have

$$\begin{aligned}
[\![\Phi \curlyvee \Psi]\!] &= [\![(\Phi_1 \vee \Psi_1) : \big((\Phi' \curlyvee \Psi') \curvearrowright ((\Phi_1 \bigwedge\!\!\!\bigcirc \Psi_1) : (\Phi' \curlywedge \Psi'))\big)]\!] \\
&= ([\![\Phi_1]\!] \cup [\![\Psi_1]\!]) \setminus \big(([\![\Phi']\!] \cup [\![\Psi']\!]) \setminus (([\![\Phi_1]\!] \cap [\![\Psi_1]\!]) \setminus ([\![\Phi']\!] \cap [\![\Psi']\!]))\big) \\
&= ([\![\Phi_1]\!] \setminus [\![\Phi']\!]) \cup ([\![\Psi_1]\!] \setminus [\![\Psi']\!]) \\
&= [\![\Phi]\!] \cup [\![\Psi]\!],
\end{aligned}$$

where the second to last equivalence follows from the validity "$B \subseteq A$ and $D \subseteq C$ imply $(A \setminus B) \cup (C \setminus D) = (A \cup C) \setminus ((B \cup D) \setminus ((A \cap C) \setminus (B \cap D)))$".

From Items 2, 4, 6, 9 and 11, we conclude that, for every $k \in [1,u]$, $[\![\widetilde{\Psi}_k]\!] = [\![\bigvee F_k]\!]$ where F_k is a set of formulae of the form $\varphi_1 \wedge \cdots \wedge \varphi_n$ with $\varphi_p \in \{\Phi_1, \dots, \Phi_i, \Psi_1, \dots, \Psi_j\}$ for every $p \in [1,n]$. As explained in the first of the induction steps we considered, $\varphi_1 \wedge \cdots \wedge \varphi_n$ can be simplified into a formula of the form ψ_1, ψ_2 or $\psi_1 \wedge \psi_2$, with $\psi_1 \in \{\Phi_1, \dots, \Phi_i\}$ and $\psi_2 \in \{\Psi_1, \dots, \Psi_j\}$; concluding the proof. □

5.2. **Complexity of union, intersection and difference.** We now study the complexity of performing the various operations we have introduced.

Lemma 5.3. *Let $\Phi_1 - (\cdots - \Phi_i)$, $\Psi_1 - (\cdots - \Psi_j)$ and $\Phi'_1 - (\cdots - \Phi'_\ell)$ be formulae in* $\mathrm{SDF}(\Sigma)$. *Suppose that, for every $k \in [1,\ell]$, $[\![\Phi'_k]\!] = [\![\bigvee G_k]\!]$, where $G_k \subseteq \{\varphi,\, \psi,\, \varphi \wedge \psi : \varphi \in \{\Phi_1, \dots, \Phi_i\},\ \psi \in \{\Psi_1, \dots, \Psi_j\}\}$. Then, $\ell \leq i \cdot j$.*

Proof. Define $\Phi_0 \coloneqq \Psi_0 \coloneqq \top$. For $u \in [1,i]$ and $v \in [1,j]$, set $\alpha_{u,v} \coloneqq (\Phi_{u-i} - \Phi_u) \wedge (\Psi_{v-1} - \Psi_v)$. Every element of G_k, and thus every Φ'_k, can be represented as a union of elements from $\{\alpha_{u,v} : (u,v) \neq (1,1)\}$. Indeed, let $\ell \in [0,i]$ and $r \in [0,j]$. Then, $\Phi_\ell \wedge \Psi_r$ is equivalent to $(\bigvee_{u=\ell}^{i-1}(\Phi_u - \Phi_{u+1})) \wedge (\bigvee_{v=r}^{j-1}(\Psi_v - \Psi_{v+1}))$. (Note that for $\ell = i$ the left disjunction simplifies as $\bot$, as expected since $[\![\Phi_k]\!] = \emptyset$.) Expanding this formula into DNF yields a union of elements of the form $\alpha_{u,v}$. Since $\Phi_1 - (\cdots - \Phi_i)$, $\Psi_1 - (\cdots - \Psi_j)$ belong to $\mathrm{SDF}(\Sigma)$, the sets $[\![\alpha_{u,v}]\!]$ are pairwise disjoint: $[\![\alpha_{u,v}]\!] \cap [\![\alpha_{u',v'}]\!] = \emptyset$ whenever $(u,v) \neq (u',v')$. Finally, because $\Phi'_1 - (\cdots - \Phi'_\ell)$ belongs to $\mathrm{SDF}(\Sigma)$, and each Φ'_k is a union of elements from $\{\alpha_{u,v} : (u,v) \neq (1,1)\}$, we conclude that $k \leq i \cdot j$. □

Lemma 5.4. *There is an algorithm that given $\oplus \in \{\curlyvee, \curlywedge, \curvearrowright\}$, and $\Phi = \Phi_1 - (\cdots - \Phi_i)$ and $\Psi = \Psi_1 - (\cdots - \Psi_j)$ in* $\mathrm{SDF}(\Sigma)$, *computes $\Phi \oplus \Psi$ in time $2^{\mathrm{poly}(i,j)} \cdot \mathrm{poly}(n, |\Sigma|)$, where n is the maximum number of disjuncts in some of the* $\mathrm{DNF}_+(\Sigma)$ *formulae $\Phi_1, \dots, \Phi_i, \Psi_1, \dots, \Psi_j$.*

Proof. The algorithm, whose pseudocode is given in Figure 2 (function Apply), is simple. It starts by computing the set $\mathbb{G}$ of all $G \subseteq \{\varphi,\, \psi,\, \varphi \bigwedge\!\!\!\bigcirc \psi : \varphi \in \{\Phi_1, \dots, \Phi_i\}, \psi \in \{\Psi_1, \dots, \Psi_j\}\}$ representing formulae that might be needed in order to compute $\Phi \oplus \Psi$, according to Lemma 5.2. Afterwards (function Apply'), the algorithm simply follows the definitions of the operations $\curlywedge$, $\curvearrowright$ and $\curlyvee$ given in Section 5.1, making sure at each step to "normalise" the computed formula $\Phi'_1 - (\cdots - \Phi'_k)$ so that $\Phi'_1, \dots, \Phi'_k$ are formulae given by elements in $\mathbb{G}$. This normalisation is done by the function Normalise. In this function, the sentence "Let Φ'_1 be

function APPLY($\oplus$, Φ, Ψ)
input: $\oplus$: operation among $\curlywedge$, $\frown$ and $\curlyvee$; Φ and Ψ: two formulae in SDF.
output: A formula in SDF equivalent to $\Phi \oplus \Psi$.
    Let $\Phi = \Phi_1 - (\cdots - \Phi_i)$ and $\Psi = (\Psi_1 - (\cdots - \Psi_j))$.
    Compute the set $\mathbb{G}$ containing all sets G such that
    - $G \subseteq \{\varphi, \psi, \varphi \barwedge \psi : \varphi \in \{\Phi_1, \ldots, \Phi_i\}, \psi \in \{\Psi_1, \ldots, \Psi_j\}\}$,
    - for every $k \in [1, i]$, Φ_k occurs at most once in a formula of G *(we consider φ and ψ to be both occurring in $\varphi \barwedge \psi$)*, and
    - for every $k \in [1, j]$, Ψ_k occurs at most once in a formula of G.
    return APPLY'($\oplus$, Φ, Ψ, $\mathbb{G}$)
function APPLY'($\oplus$, Φ, Ψ, $\mathbb{G}$)
    if $[\![\Phi]\!] = \emptyset$ or $[\![\Psi]\!] = \emptyset$ **then**
        switch $\oplus$ **do**
            case $\curlywedge$ **: return** a representative of $\bot$ in $\mathbb{G}$
            case $\frown$ **: return** Φ
            case $\curlyvee$ **: return** $([\![\Phi]\!] = \emptyset \ ?\ \Psi : \Phi)$
    Let $\Phi = \Phi_1 - \Phi'$ and $\Psi = \Psi_1 - \Psi'$
    var result
    switch $\oplus$ **do**
        case $\curlywedge$ **:** result $\leftarrow (\Phi_1 \barwedge \Psi_1) :$ APPLY'$(\curlyvee, \Phi', \Psi', \mathbb{G})$
        case $\frown$ **:** result $\leftarrow \Phi_1 :$ APPLY'$(\curlyvee, \Phi', \Psi, \mathbb{G})$
        case $\curlyvee$ **:**
            if $[\![\Psi_1]\!] \subseteq [\![\Phi_1]\!]$ **then** result $\leftarrow \Phi_1 :$ APPLY'$(\frown, \Phi', \Psi, \mathbb{G})$
            else
                result $\leftarrow$ NORMALISE$((\Phi_1 \barwedge \Psi_1) :$ APPLY'$(\curlywedge, \Phi', \Psi', \mathbb{G}), \mathbb{G})$
                result $\leftarrow$ APPLY'$(\frown,$ APPLY'$(\curlyvee, \Phi', \Psi', \mathbb{G}),$ result, $\mathbb{G})$
                result $\leftarrow (\Phi_1 \vee \Psi_1) :$ result
    return NORMALISE(result, $\mathbb{G}$)
function NORMALISE($\Phi, \mathbb{G}$)
    if $[\![\Phi]\!] = \emptyset$ **then return** a representative of $\bot$ in $\mathbb{G}$
    Let $\Phi = \Phi_1 - \Phi'$
    Let Φ_1' be a representative of Φ_1 in $\mathbb{G}$
    return $\Phi_1' -$ NORMALISE$(\Phi', \mathbb{G})$

Figure 2: Computing Boolean operations on SDFs.

the representative of Φ_1 in $\mathbb{G}$" indicates that $\Phi_1' = \bigvee G$ for some $G \in \mathbb{G}$, and $[\![\Phi_1']\!] = [\![\Phi_1]\!]$. This normalisation is required for the $\mathrm{DNF}_+(\Sigma)$ formulae to stay of polynomial size during the procedure. The pseudocode of the cons function (:) follows exactly its mathematical definition given at the beginning of Section 5.1, and is thus omitted from Figure 2.

Since the algorithm simply follows the recursive definitions of the various operations, and by Lemma 5.2 we can restrict the formulae of the output to be given by elements in $\mathbb{G}$, its correctness is immediate.

Let us discuss the runtime of Apply. First, note that computing $\mathbb{G}$ requires $2^{\mathrm{poly}(i,j)}$ time: to construct the sets in $\mathbb{G}$, we first iterate through the subsets $A \subseteq \{\Phi_1, \dots, \Phi_i\}$ and $B \subseteq \{\Psi_1, \dots, \Psi_j\}$. Every such pair (A, B) indicates what formulae appear in the set of $\mathbb{G}$ we are constructing. Afterwards, we iterate through all possible injections $f \colon A' \to B$ such that $A' \subseteq A$ and $|A'| \leq |B|$, and for a choice of f construct the set $\{\varphi : \varphi \in A \setminus A'\} \cup \{\varphi \wedge \psi : \varphi \in A' \text{ and } f(\varphi) = \psi\} \cup \{\psi : \psi \in B \setminus f(A')\}$. The total number of iterations (and hence of sets in $\mathbb{G}$) is bounded by

$$\begin{aligned}
&\underbrace{\sum_{l=0}^{i}\sum_{r=0}^{j}\left(\binom{i}{l}\binom{j}{r}\right)}_{\text{subsets } A \text{ and } B}\ \underbrace{\sum_{d=0}^{\min(l,r)}\left(\binom{l}{d}\binom{r}{d}d!\right)}_{\text{injections } f}\Bigg) \\
&\leq 3 \cdot \sum_{l=0}^{i}\sum_{r=0}^{j}\Big(\frac{i! \cdot j!}{(i-l)! \cdot (j-r)!}\Big) && \text{using} \quad \sum_{d=0}^{\min(l,r)} \frac{1}{d!} < e \\
&\leq 3 \cdot (i+1)! \cdot (j+1)! \leq 2^{\mathrm{poly}(i,j)}.
\end{aligned}$$

Let us now move to the function Normalise, that given $\mathbb{G}$ computed as above, and a formula $\Phi'' = \Phi''_1 - (\cdots - \Phi''_k)$ in $\mathrm{SDF}(\Sigma)$ where each Φ''_ℓ ($\ell \in [1,k]$) is a can be represented with a set in $\mathbb{G}$, returns a formula $(\bigvee G_1) - (\cdots - (\bigvee G_k))$ in $\mathrm{SDF}(\Sigma)$ such that $G_1, \dots, G_k \in \mathbb{G}$ and $[\![\Phi''_\ell]\!] = [\![\bigvee G_\ell]\!]$ for all $\ell \in [1,k]$. This function runs in time

$$\underbrace{k \cdot 2^{\mathrm{poly}(i,j)}}_{\text{iterate through } \Phi'' \text{ and } \mathbb{G}} \quad \cdot \quad \underbrace{\mathrm{poly}(i,j,n,m,|\Sigma|)}_{\text{comparison of DNF}_+(\Sigma) \text{ formulae}}$$

where m is the maximum number of disjuncts in some $\mathrm{DNF}_+(\Sigma)$ formula Φ''_ℓ. Note that an upper bound to the number of disjuncts that formulae $\bigvee G$, with $G \in \mathbb{G}$, is given by $i \cdot n^2 + j$.

Lastly, consider the function Apply'. Given $\mathbb{G}$, an operation $\oplus \in \{\curlywedge, \curlyvee, \curlyvee\}$, and two formulae Φ and Ψ in $\mathrm{SDF}(\Sigma)$ made of $\mathrm{DNF}_+(\Sigma)$ formulae of the form $\bigvee G$ with $G \in \mathbb{G}$, this function returns a formula in $\mathrm{SDF}(\Sigma)$ that is equivalent to $\Phi \oplus \Psi$ and that is made of $\mathrm{DNF}_+(\Sigma)$ formulae of the form $\bigvee G$ with $G \in \mathbb{G}$. Because of Lemma 5.3, we know that all the recursive calls done by Apply' return a formula $\Phi'_1 - (\cdots - \Phi'_\ell)$ with $\ell \leq i \cdot j$. Because of this, we can bound the time required to perform the various calls to (:) and Normalise throughout the procedure. For example, in line 17, computing $(\Phi_1 \mathbin{\bigcirc\!\!\!\!\wedge} \Psi_1) : \textsc{Apply'}(\curlyvee, \Phi', \Psi', \mathbb{G})$ once the result of $\textsc{Apply'}(\curlyvee, \Phi', \Psi', \mathbb{G})$ is given takes time $\mathrm{poly}(i,j,n,|\Sigma|)$.

Below, given $\ell \in [1, i \cdot j]$ and $r \in [0, i \cdot j]$, let us write $\mathtt{R}(\ell, r)$ for the maximal running time of Apply' on an input $(\oplus, \Phi', \Psi', \mathbb{G})$ with $\oplus \in \{\curlywedge, \curlyvee, \curlyvee\}$, $\mathbb{G}$ as above, $\Phi' = \Phi'_1 - (\cdots - \Phi'_\ell)$, and $\Psi' = \Psi'_1 - \Psi''$, with Ψ'' formula in $\mathrm{SDF}(\Sigma)$, such that there are r elements $G \in \mathbb{G}$ satisfying $[\![\bigvee G]\!] \subsetneq [\![\Psi'_1]\!]$. The fact that r can be restricted to be at most $i \cdot j$ follows from Lemma 5.3. A simple inspection of Apply' yield the following inequalities:

$$\begin{aligned}
\mathtt{R}(1, r) &\leq O(1) && \text{case: } [\![\Phi]\!] = \emptyset \\
\mathtt{R}(\ell, 0) &\leq O(1) && \text{case: } [\![\Psi]\!] = \emptyset \\
\mathtt{R}(\ell+1, r+1) &\leq \max \begin{cases}
\mathtt{R}(\ell, r) + \mathrm{poly}(i,j,n,|\Sigma|) & \text{case: line 17} \\
\mathtt{R}(\ell, r+1) + \mathrm{poly}(i,j,|\Sigma|) & \text{case: line 18 or line 20} \\
2 \cdot \mathtt{R}(\ell, r) + \mathtt{R}(i \cdot j, r) & \\
\qquad + 2^{\mathrm{poly}(i,j)} \cdot \mathrm{poly}(n, |\Sigma|) & \text{case: lines 22–24.}
\end{cases}
\end{aligned}$$

Let us write $C(i,j,n,|\Sigma|)$ for a function in $2^{\mathrm{poly}(i,j)} \cdot \mathrm{poly}(n,|\Sigma|)$ that upper bounds all functions $O(1)$, $\mathrm{poly}(i,j,n,|\Sigma|)$ and $2^{\mathrm{poly}(i,j)} \cdot \mathrm{poly}(n,|\Sigma|)$ appearing in the inequalities above. Note that, in giving an upper bound to $\mathbb{R}(\ell,r)$, we can treat i,j,n and $|\Sigma|$ as constants, hence below we simply write C instead of $C(i,j,n,|\Sigma|)$. We have $\mathbb{R}(1,r) \leq C$, $\mathbb{R}(\ell,0) \leq C$ and otherwise, given $\ell \geq 1$ and $r \geq 0$, $\mathbb{R}(\ell+1,r+1) \leq C + \max(\mathbb{R}(\ell,r+1), 2 \cdot \mathbb{R}(\ell,r) + \mathbb{R}(i \cdot j, r))$. A simple check shows that $\mathbb{R}(\ell,r) \leq C \cdot 3^{r \cdot i \cdot j + \ell}$. This inequality is clearly true for the base cases $\mathbb{R}(1,r)$ and $\mathbb{R}(\ell,0)$, and otherwise

$$\begin{aligned}
\mathbb{R}(\ell+1,r+1) &\leq C + \max(\mathbb{R}(\ell,r+1), 2 \cdot \mathbb{R}(\ell,r) + \mathbb{R}(i \cdot j, r)) \\
&\leq C + \max(C \cdot 3^{(r+1) \cdot i \cdot j + \ell}, 2 \cdot C \cdot 3^{r \cdot i \cdot j + \ell} + C \cdot 3^{r \cdot i \cdot j + i \cdot j}) \\
&\leq C \cdot 3^{(r+1) \cdot i \cdot j + \ell + 1}.
\end{aligned}$$

We conclude that, on the formulae Φ and Ψ of the statement of the lemma, $\textsc{Apply'}(\oplus, \Phi, \Psi, \mathbb{G})$ runs in time $2^{\mathrm{poly}(i,j)} \cdot \mathrm{poly}(n,|\Sigma|)$. Then, taking into account the computation of $\mathbb{G}$, $\textsc{Apply}(\oplus, \Phi, \Psi)$ runs in $2^{\mathrm{poly}(i,j)} \cdot \mathrm{poly}(n,|\Sigma|)$. □

5.3. **Proof of Lemma 4.5.** We now rely on the strict difference normal form for propositional logic to establish Lemma 4.5. Let us fix a structure $\mathcal{A} = (A, \sigma, I)$ and consider its first-order theory $\mathrm{FO}(\mathcal{A})$. Moreover, let ρ be a representation of $\mathrm{D} := \bigcup_{n \in \mathbb{N}} \mathrm{D}_n$, where, for all $n \in \mathbb{N}$, $\mathrm{D}_n \subseteq \mathcal{P}(A^n)$ is such that $[\![\Psi]\!]_{\mathcal{A}} \in \mathrm{D}_n$ for every $\Psi \in \mathrm{CQ}(\sigma)$ having maximum variable x_n. In order to establish Lemma 4.5, we assume that:

(1) The structure $(\mathrm{D}, \wedge, \leq)$ has a (ρ,θ)-UXP signature;
(2) The structure $(\mathrm{un}(\mathrm{D}), \leq)$ has a $(\mathrm{un}(\rho), \mathrm{len}(\theta))$-UXP signature,

and show that, then, $(\mathrm{dfnf}(\mathrm{D}), \bot, \top, \vee, \wedge, -, \leq)$ has a $(\mathrm{dfnf}(\rho), \mathrm{dep}(\theta))$-UXP signature.

The functions $\bot$ and $\top$ are constants, and thus can be implemented with computable functions running in constant time. Consider a binary function $\oplus \in \{\vee, \wedge, -\}$. The pseudocode of the $(\mathrm{dep}(\theta)^2, \mathrm{dep}(\theta))$-UXP reduction that is a $(\mathrm{dfnf}(\rho)^2, \mathrm{dfnf}(\rho))$-implementation of $\oplus$ is given in Figure 3. In a nutshell, given as input two sequences $(u_1, \ldots, u_i)$ and $(v_1, \ldots, v_j)$ in $\mathrm{dom}(\mathrm{dfnf}(\rho))$, the procedure in Figure 3 computes a sequence $(w_1, \ldots, w_k)$ representing $(\mathrm{dfnf}(\rho)(u_1, \ldots, u_i)) \oplus (\mathrm{dfnf}(\rho)(v_1, \ldots, v_j))$ by translating the input sequences into propositional formulae, computing $\oplus$ with respect to propositional calculus, and translating back into an element of $\mathrm{dom}(\mathrm{dfnf}(\rho))$.

Lemma 5.5. *The function in Figure 3 respects its specification.*

Proof. Let $p_1, \ldots, p_i$ and $q_1, \ldots, q_j$ be the fresh propositional symbols introduced in line 1, and let $\Sigma = \{p_1, \ldots, p_i, q_1, \ldots, q_j\}$. Given $\ell \in [1,i]$, the procedure associates to u_ℓ the propositional symbol p_ℓ. Similarly, given $r \in [1,j]$, it associates to v_r the propositional symbol q_r. Recall that elements in $\mathrm{dom}(\mathrm{dfnf}(\rho))$ are tuples representing formulae in difference normal form that might not be *strict* (which is instead required in the notion of $\mathrm{SDF}(\Sigma)$). For example, the sequence $(u_1, \ldots, u_i)$ might be such that, for some $k \in [1, i-1]$, $\mathrm{un}(\rho)(u_{k+1}) \not\leq \mathrm{un}(\rho)(u_k)$. However, the following equivalence holds directly from set-theoretical validities:

$$\mathrm{dfnf}(\rho)(u_1, \ldots, u_i) = \mathrm{dfnf}(\rho)(u_1,\ u_1 \wedge u_2,\ \ldots,\ u_1 \wedge u_2 \wedge \cdots \wedge u_i, \bot).$$

At the propositional level, it suffices then to consider the formulae Φ and Ψ defined in lines 3 and 4, which are in $\mathrm{SDF}(\Sigma)$ and correspond to $(u_1, \ldots, u_i)$ and $(v_1, \ldots, v_j)$, respectively. By Lemma 5.4, the formula $\Phi'_1 - (- \ldots \Phi'_k)$ computed in line 5 corresponds

Input: $(u_1, \dots, u_i)$ and $(v_1, \dots, v_j)$ in $\mathrm{dom}(\mathrm{dnf}(\rho))$.
Output: $(w_1, \dots, w_k)$ belonging to $\mathrm{dom}(\mathrm{dnf}(\rho))$, and such that $k \leq (i+1)\cdot(j+1)$ and $(\mathrm{dnf}(\rho)(u_1, \dots, u_i)) \oplus (\mathrm{dnf}(\rho)(v_1, \dots, v_j)) = \mathrm{dnf}(\rho)(w_1, \dots, w_k)$.

1: Let $p_1, \dots, p_i$ and $q_1, \dots, q_j$ be $i+j$ fresh propositional symbols
2: Let $\oplus_{\mathbb{B}}$ be equal to $\curlywedge$, $\frown$ or $\curlyvee$, depending on $\oplus$ being $\wedge$, $-$ or $\vee$, respectively
3: $\Phi \leftarrow (p_1 - (p_1 \wedge p_2 - (\cdots - (\bigwedge_{\ell=1}^{i} p_\ell - \bot))))$
4: $\Psi \leftarrow (q_1 - (q_1 \wedge q_2 - (\cdots - (\bigwedge_{r=1}^{j} q_r - \bot))))$
5: $\Phi'_1 - (- \dots \Phi'_k) \leftarrow \textsc{Apply}(\oplus_{\mathbb{B}}, \Phi, \Psi)$ ▷ Figure 2
6: $(w_1, \dots, w_k) \leftarrow ((), \dots, ())$ ▷ to be populated
7: **for** t in $[1, k]$ **do**
8: Let $\Phi'_t = \bigvee_{s=1}^{d} \Psi'_s$
9: **for** s in $[1, d]$ such that $\bot$ does not occur in Ψ'_s **do**
10: $m \leftarrow$ element of $\mathrm{dom}(\mathrm{un}(\rho))$ corresponding to $\top$
11: **for** $\ell \in [1, i]$ such that p_ℓ occurs in Ψ'_s **do** $m \leftarrow u_\ell \wedge m$
12: **for** $r \in [1, j]$ such that q_r occurs in Ψ'_s **do** $m \leftarrow v_r \wedge m$
13: $w_t \leftarrow m \vee w_t$
14: **return** $(w_1, \dots, w_k)$

Figure 3: Implementation of $\oplus \in \{\vee, \wedge, -\}$ as a $(\mathrm{dep}(\theta)^2, \mathrm{dep}(\theta))$-UXP reduction.

to $\Phi \oplus_{\mathbb{B}} \Psi$. Moreover, as described at the beginning of the proof of Lemma 5.4 (and according to Lemma 5.2), for every $t \in [1, k]$, the formula Φ'_t is equal to $\bigvee G$ for some $G \subseteq \{\varphi, \psi, \varphi \barwedge \psi : \varphi \in \{\Phi_1, \dots, \Phi_i, \bot\}, \psi \in \{\Psi_1, \dots, \Psi_j, \bot\}\}$. Note that, given $\varphi \in \{\Phi_1, \dots, \Phi_i\}$ and $\psi \in \{\Psi_1, \dots, \Psi_j\}$, $\varphi \barwedge \psi$ is a disjunction of formulae of the form $p_1 \wedge \cdots \wedge p_\ell \wedge q_1 \wedge \cdots \wedge q_r$, for some $\ell \in [1, i]$ and $r \in [1, j]$. Then, lines 7–14 simply translate back the formulae p_ℓ and q_r into u_ℓ and v_r, respectively, and compute the necessary conjunction and disjunctions of these elements of $\mathrm{un}(\rho)$. In particular, the operations $\wedge$ and $\vee$ appearing in lines 11–13 are the binary functions implementing conjunction and disjunction on elements of $\mathrm{un}(\rho)$. Then, the fact that translating back from propositional calculus yield an element of representing $(\mathrm{dnf}(\rho)(u_1, \dots, u_i)) \oplus (\mathrm{dnf}(\rho)(v_1, \dots, v_j))$ stems directly from the definitions of $\curlywedge$, $\frown$ and $\curlyvee$, that only rely on set-theoretical tautologies. □

Lemma 5.6. *The function in Figure 3 is a* $(\mathrm{dep}(\theta)^2, \mathrm{dep}(\theta))$*-UXP reduction.*

Proof. Directly from Lemma 5.4, the formula $\Phi_1 - (- \cdots - \Phi'_k)$ of line 5 can be computed in polynomial time when the lengths i and j of $(u_1, \dots, u_i)$ and $(v_1, \dots, v_j)$ are considered fixed (as it is the case for the parameter $\mathrm{dep}(\theta)$). Note moreover that $k \leq i \cdot j$, by Lemma 5.3, and that d in line 9 is bounded in $\mathrm{poly}(i, j)$, as every Φ'_t is equal to $\bigvee G$ for some $G \subseteq \{\varphi, \psi, \varphi \barwedge \psi : \varphi \in \{\Phi_1, \dots, \Phi_i, \bot\}, \psi \in \{\Psi_1, \dots, \Psi_j, \bot\}\}$ such that for every $k \in [1, i]$ $\Phi_k \coloneqq p_1 \wedge \cdots \wedge p_k$ occurs at most once in a formula of G, and for every $k \in [1, j]$, $\Psi_k \coloneqq q_1 \wedge \cdots \wedge q_k$ occurs at most once in a formula of G. Therefore, the **for** loops of lines 7–13 perform overall a constant number of iteration, when taking into account the parameter $\mathrm{dep}(\theta)$. This implies a constant number of calls to the functions computing $\wedge$ and $\vee$ on elements of $\mathrm{dom}(\mathrm{un}(\rho))$. To conclude the proof it suffices then to show that these functions are $(\mathrm{len}(\theta)^2, \mathrm{len}(\theta))$-UXP reductions. For $\vee$, the analysis is simple: given $u = (a_1, \dots, a_n)$ and $v = (b_1, \dots, b_m)$ in $\mathrm{dom}(\mathrm{un}(\rho))$, $u \vee v = (a_1, \dots, a_n, b_1, \dots, b_m)$, and therefore $\vee$ is a constant time operation when the lengths n and m are fixed. For $\wedge$, $u \wedge v$ correspond to the sequence of all $a_\ell \wedge b_r$ with $\ell \in [1, n]$ and

Input: $(u_1, \ldots, u_i)$ and $(v_1, \ldots, v_j)$ in $\mathrm{dom}(\mathrm{dfnf}(\rho))$.
Output: **true** iff $(\mathrm{dfnf}(\rho)(u_1, \ldots, u_i)) \leq (\mathrm{dfnf}(\rho)(v_1, \ldots, v_j))$.

1: $(w_1, \ldots, w_k) \leftarrow (u_1, \ldots, u_i) - (v_1, \ldots, v_j)$ ▷ Figure 3
2: $\ell = 1$
3: **while** $\ell \leq k$ **do**
4: **if** $w_\ell \leq ()$ **then return true** ▷ i.e., $\mathrm{un}(\rho)(w_\ell) = \emptyset$
5: **else if** $\ell < k$ and $w_\ell \leq w_{\ell+1}$ **then** $\ell \leftarrow \ell + 2$
6: **else return false**
7: **return true**

Figure 4: Implementation of $\leq$ as a $(\mathrm{dep}(\theta)^2, \mathbf{1})$-UXP reduction.

$r \in [1, m]$. Since n and m are considered fixed, this is a constant number of calls to the function computing $\wedge$ on elements of $\mathrm{dom}(\rho)$, which from the properties of Requirement 4.3 runs in polynomial time when accounting for the parameter θ. □

To conclude the proof of Lemma 4.5, it now suffices to provide a $(\mathrm{dep}(\theta)^2, \mathbf{1})$-UXP reduction that implements $\leq$. Its pseudocode is given in Figure 4. Briefly, given $(u_1, \ldots, u_i)$ and $(v_1, \ldots, v_j)$ in $\mathrm{dom}(\mathrm{dfnf}(\rho))$, this function establishes whether $(\mathrm{dfnf}(\rho)(u_1, \ldots, u_i)) \leq (\mathrm{dfnf}(\rho)(v_1, \ldots, v_j))$ by checking if $(\mathrm{dfnf}(\rho)(u_1, \ldots, u_i)) - (\mathrm{dfnf}(\rho)(v_1, \ldots, v_j)) = \emptyset$.

Lemma 5.7. *The function in Figure 4 respects its specification.*

Proof. By Lemma 5.5, the sequence $(w_1, \ldots, w_k)$ computed in line 1 belongs to $\mathrm{dom}(\mathrm{dfnf}(\rho))$ and represents $(\mathrm{dfnf}(\rho)(u_1, \ldots, u_i)) - (\mathrm{dfnf}(\rho)(v_1, \ldots, v_j))$. By definition, $\mathrm{dfnf}(\rho)(w_1, \ldots, w_k)$ is equivalent to $\mathrm{un}(\rho)(w_1) - (\mathrm{un}(\rho)(w_2) - (\cdots - \mathrm{un}(\rho)(w_k)))$. Hence, $\mathrm{dfnf}(\rho)(w_1, \ldots, w_k) = \emptyset$ if and only if one of the following holds

- $k = 0$, i.e., $(w_1, \ldots, w_k)$ is an empty sequence of elements in $\mathrm{un}(\rho)$,
- $\mathrm{un}(\rho)(w_1) = \emptyset$, or
- $k \geq 2$, $\mathrm{un}(\rho)(w_1) \leq \mathrm{un}(\rho)(w_2)$ and $\mathrm{dfnf}(\rho)(w_3, \ldots, w_k) = \emptyset$.

Lines 3–7 implement this check, leading to the correctness of the procedure. □

Lemma 5.8. *The function in Figure 4 is a* $(\mathrm{dep}(\theta)^2, \mathbf{1})$*-UXP reduction.*

Proof. By Lemma 5.6, there is a $(\mathrm{dep}(\theta)^2, \mathrm{dep}(\theta))$-UXP reduction computing the sequence $(w_1, \ldots, w_k)$ in line 1. By Lemma 5.5, $k \leq (i+1) \cdot (j+1)$, and thus the **while** loop of line 3 iterates $O(i \cdot j)$ times. Following lines 4 and 5, each iteration might require a comparison between two elements of $\mathrm{dom}(\mathrm{un}(\rho))$. Because of the properties required by Requirement 4.3, this comparison can be implemented by a $(\mathrm{len}(\theta)^2, \mathbf{1})$-UXP reduction. Putting all together, we conclude that the function in Figure 4 is a $(\mathrm{dep}(\theta)^2, \mathbf{1})$-UXP reduction. □

6. Proofs Lemmas 4.6 and 4.9 from Section 4

We present the proofs of Lemmas 4.6 and 4.9 deferred from Section 4. Given a set A, $Z \subseteq A^j$ and $k \in \mathbb{N}$, we define $Z^c \coloneqq A^j \setminus Z$ and write

$$Z{\uparrow^k} \coloneqq Z \times A^{\max(0,k-j)} = \{(v_1, \ldots, v_{\max(j,k)}) \in \mathrm{seq}(A) : (v_1, \ldots, v_j) \in Z\}$$

for the set of tuples obtained from Z by "appending" $\max(0, k-j)$ dimensions.

Lemma 4.6. *Consider $X \subseteq A^n$, $Y \subseteq A^m$, $Z \subseteq A^r$, and $\mathbf{i} \in \mathbf{I}$. Then,*

$$\pi(\mathbf{i}, X - Y) = \pi(\mathbf{i}, X) - \pi^{\forall}_X(\mathbf{i}, Y) \quad \textit{and} \quad \pi^{\forall}_Z(\mathbf{i}, X - Y) = \pi^{\forall}_Z(\mathbf{i}, X) - \pi(\mathbf{i}, Y).$$

Proof. Let $M \coloneqq \max(m, n)$ and $\mathbf{i} = (i_1, \ldots, _k)$. For the first equivalence:

$$\begin{aligned}
\pi(\mathbf{i}, X - Y) &= \{\gamma \in A^M : \exists \mathbf{a} \in A^k \text{ s.t. } \gamma[\mathbf{i} \leftarrow \mathbf{a}] \in (X - Y)\} \\
&= \{\gamma \in A^M : \forall \mathbf{a} \in A^k \text{ s.t. } \gamma[\mathbf{i} \leftarrow \mathbf{a}] \notin (X - Y)\}^c \\
&= \{\gamma \in A^M : \forall \mathbf{a} \in A^k \text{ s.t. } \gamma[\mathbf{i} \leftarrow \mathbf{a}] \in (X^c \vee Y)\}^c \\
&= \Big(\{\gamma \notin \pi(\mathbf{i}, X)\uparrow^m : \forall \mathbf{a} \in A^k \text{ s.t. } \gamma[\mathbf{i} \leftarrow \mathbf{a}] \in (X^c \vee Y)\} \vee \\
&\qquad \{\gamma \in \pi(\mathbf{i}, X)\uparrow^m : \forall \mathbf{a} \in A^k \text{ s.t. } \gamma[\mathbf{i} \leftarrow \mathbf{a}] \in (X^c \vee Y)\}\Big)^c \\
&= \big(\pi(\mathbf{i}, X)^c \vee \pi^{\forall}_X(\mathbf{i}, Y)\big)^c \\
&= \pi(\mathbf{i}, X) - \pi^{\forall}_X(\mathbf{i}, Y).
\end{aligned}$$

where in the second-last equality we have used the fact that if $\gamma \notin \pi(\mathbf{i}, X)\uparrow^m$ then for every $\mathbf{a} \in A^k$ we have $\gamma[\mathbf{i} \leftarrow \mathbf{a}] \notin (X^c)\uparrow^m$.

The second equivalence is proven in a similar way. Let $N \coloneqq \max(M, r)$.

$$\begin{aligned}
&\pi^{\forall}_Z(\mathbf{i}, X - Y) \\
&= \{\gamma \in \pi(\mathbf{i}, Z\uparrow^N) : \forall \mathbf{a} \in A^k, \gamma' \coloneqq \gamma[\mathbf{i} \leftarrow \mathbf{a}] \in Z\uparrow^N \text{ implies } \gamma' \in (X - Y)\uparrow^N\} \\
&= \{\gamma \in \pi(\mathbf{i}, Z\uparrow^N) : \forall \mathbf{a} \in A^k \text{ and } \gamma' \coloneqq \gamma[\mathbf{i} \leftarrow \mathbf{a}], (\gamma' \in Z\uparrow^N \text{ implies } \gamma' \in X\uparrow^N) \\
&\qquad\qquad \text{and } (\gamma' \in Z\uparrow^N \text{ implies } \gamma' \in (Y^c)\uparrow^N)\} \\
&= \{\gamma \in \pi(\mathbf{i}, Z\uparrow^N) : \forall \mathbf{a} \in A^k \text{ and } \gamma' \coloneqq \gamma[\mathbf{i} \leftarrow \mathbf{a}], \gamma' \in Z\uparrow^N \text{ implies } \gamma' \in X\uparrow^N\} \\
&\quad \wedge \{\gamma \in \pi(\mathbf{i}, Z\uparrow^N) : \forall \mathbf{a} \in A^k \text{ and } \gamma' \coloneqq \gamma[\mathbf{i} \leftarrow \mathbf{a}], \gamma' \in Z\uparrow^N \text{ implies } \gamma' \in Y^c\uparrow^N\} \\
&= \pi^{\forall}_Z(\mathbf{i}, X)\uparrow^N \wedge \{\gamma \in A^N : \gamma \notin \pi(\mathbf{i}, Z\uparrow^N), \text{ or } \gamma[\mathbf{i} \leftarrow \mathbf{a}] \in (Z\uparrow^N \wedge Y\uparrow^N) \text{ for some } \mathbf{a} \in A^k\}^c \\
&= \pi^{\forall}_Z(\mathbf{i}, X)\uparrow^N \wedge \Big(\big(\pi(\mathbf{i}, Z)^c \vee \pi(\mathbf{i}, Z \wedge Y)\big)^c\Big)\uparrow^N \\
&= \pi^{\forall}_Z(\mathbf{i}, X) \wedge \big(\pi(\mathbf{i}, Z)^c \vee \pi(\mathbf{i}, Z \wedge Y)\big)^c && \text{by definition of } \wedge \text{ and } N \\
&= \pi^{\forall}_Z(\mathbf{i}, X) \wedge \big(\pi(\mathbf{i}, Z) \wedge \pi(\mathbf{i}, Z \wedge Y)^c\big) \\
&= \pi^{\forall}_Z(\mathbf{i}, X) - \pi(\mathbf{i}, Z \wedge Y) && \text{as } \pi(\mathbf{i}, Z) \subseteq \pi^{\forall}_Z(\mathbf{i}, X) \text{ by def. of } \pi^{\forall}_{_} \\
&= \pi^{\forall}_Z(\mathbf{i}, X) - \pi(\mathbf{i}, Y) && \text{again from } \pi(\mathbf{i}, Z) \subseteq \pi^{\forall}_Z(\mathbf{i}, X) \quad \square
\end{aligned}$$

Before moving to the proof of Lemma 4.9 we need the following intermediate result.

Lemma 6.1. *Let $X \subseteq A^n$, $Z_1, \ldots, Z_r$ with $Z_j \subseteq A^{m_j}$, and $\mathbf{i} = (i_1, \ldots, i_k) \in \mathbf{I}$. Let $Z \coloneqq \bigvee_{j=1}^r Z_j$. Then, $\pi^{\forall}_Z(\mathbf{i}, X) = \bigwedge_{j=1}^r \Big(\pi^{\forall}_{Z_j}(\mathbf{i}, X) \vee \big(\pi(\mathbf{i}, Z) - \pi(\mathbf{i}, Z_j)\big)\Big)$.*

Proof. Let $M \coloneqq \max_{i=1}^{r}(n, m_i)$. The lemma follows from a simple calculation:

$$
\begin{aligned}
&\pi^{\forall}_{Z}(\mathbf{i}, X)\\
&= \{\gamma \in \pi(\mathbf{i}, Z{\uparrow^n}) : \forall \mathbf{a} \in A^k \text{ and } \gamma' \coloneqq \gamma[\mathbf{i} \leftarrow \mathbf{a}], \gamma' \in Z{\uparrow^n} \text{ implies } \gamma' \in X{\uparrow^M}\}\\
&= \pi(\mathbf{i}, Z{\uparrow^n}) \wedge \{\gamma \in A^M : \text{for every } j \in [1, r], \text{ every } \mathbf{a} \in A^k \text{ and } \gamma' \coloneqq \gamma[\mathbf{i} \leftarrow \mathbf{a}],\\
&\qquad\qquad \gamma' \in Z_j{\uparrow^M} \text{ implies } \gamma' \in X{\uparrow^M}\}\\
&= \pi(\mathbf{i}, Z{\uparrow^n}) \wedge \bigwedge_{j=1}^{r} \{\gamma \in A^M : \text{for every } \mathbf{a} \in A^k \text{ and } \gamma' \coloneqq \gamma[\mathbf{i} \leftarrow \mathbf{a}],\\
&\qquad\qquad \gamma' \in Z_j{\uparrow^M} \text{ implies } \gamma' \in X{\uparrow^M}\}\\
&= \bigwedge_{j=1}^{r} \Big(\pi(\mathbf{i}, Z{\uparrow^n}) \wedge \{\gamma \in A^M : \forall \mathbf{a} \in A^k \text{ and } \gamma' \coloneqq \gamma[\mathbf{i} \leftarrow \mathbf{a}],\\
&\qquad\qquad \gamma' \in Z_j{\uparrow^M} \text{ implies } \gamma' \in X{\uparrow^M}\}\Big)\\
&\qquad \textit{(Note: if } \gamma \notin \pi(\mathbf{i}, Z_j{\uparrow^M}) \textit{ then } \gamma' \notin Z_j{\uparrow^M} \textit{ for all } \mathbf{a} \in A^k \textit{ and } \gamma' \coloneqq \gamma[\mathbf{i} \leftarrow \mathbf{a}].\textit{)}\\
&= \bigwedge_{j=1}^{r} \Big(\pi(\mathbf{i}, Z{\uparrow^n}) \wedge \big(\pi(\mathbf{i}, Z_j)^c \vee \{\gamma \in \pi(\mathbf{i}, Z_j{\uparrow^M}) : \forall \mathbf{a} \in A^k \text{ and } \gamma' \coloneqq \gamma[\mathbf{i} \leftarrow \mathbf{a}]\\
&\qquad\qquad \gamma' \in Z_j{\uparrow^M} \text{ implies } \gamma' \in X{\uparrow^M}\}\big)\Big)\\
&= \bigwedge_{j=1}^{r} \Big(\pi(\mathbf{i}, Z{\uparrow^n}) \wedge \big(\pi(\mathbf{i}, Z_j)^c \vee \pi^{\forall}_{Z_j}(\mathbf{i}, X)\big)\Big)\\
&= \bigwedge_{j=1}^{r} \Big((\pi(\mathbf{i}, Z{\uparrow^n}) \wedge \pi^{\forall}_{Z_j}(\mathbf{i}, X)) \vee \big(\pi(\mathbf{i}, Z{\uparrow^n}) \wedge \pi(\mathbf{i}, Z_j)^c\big)\Big)\\
&= \bigwedge_{j=1}^{r} \Big(\pi^{\forall}_{Z_j}(\mathbf{i}, X) \vee \big(\pi(\mathbf{i}, Z) \wedge \pi(\mathbf{i}, Z_j)^c\big)\Big)\\
&= \bigwedge_{j=1}^{r} \Big(\pi^{\forall}_{Z_j}(\mathbf{i}, X) \vee \big(\pi(\mathbf{i}, Z) - \pi(\mathbf{i}, Z_j)\big)\Big).
\end{aligned}
$$

□

We recall Requirement 4.8 of the framework.

Requirement 4.8 (Projection and universal projection)**.** *Establish the following properties:*

(F3) *For every* $X \in \mathrm{D}$ *and* $\mathbf{i} \in \mathbf{I}$, $\dot{\pi}(\mathbf{i}, X)$ *belongs to* $\mathrm{dfnf}(\mathrm{D})$.
(F4) *For every* $Z \in \mathrm{D}$, $\mathbf{i} \in \mathbf{I}$ *and* $X \in \mathrm{un}(\mathrm{D})$, $\dot{\pi}^{\forall}_{Z}(\mathbf{i}, X)$ *belongs to* $\mathrm{dfnf}(\mathrm{D})$.
(F5) *There is a* $(\mathbf{1} \cdot \theta, \mathrm{dep}(\theta))$*-UXP reduction* $(\nu_{\mathbf{I}} \times \rho, \mathrm{dfnf}(\rho))$*-implementing* $\dot{\pi}$.
(F6) *There is a* $(\theta \cdot \mathbf{1} \cdot \mathrm{len}(\theta), \mathrm{dep}(\theta))$*-UXP reduction* $(\rho \times \nu_{\mathbf{I}} \times \mathrm{un}(\rho), \mathrm{dfnf}(\rho))$*-implementing* $\dot{\pi}^{\forall}$.

Lemma 4.9 (Outcome of Requirement 4.3 and 4.8)**.** *Assume (F1)–(F6) to hold. Then, the structure* $\mathcal{D}$ *from Proposition 4.2 has a* $(\mathrm{dfnf}(\rho), \mathrm{dep}(\theta))$*-UXP signature.*

Proof. By Lemma 4.5, the structure $\mathcal{D}' \coloneqq (\mathrm{dfnf}(\mathrm{D}), \bot, \top, \vee, \wedge, -, \leq)$ has a $(\mathrm{dfnf}(\rho), \mathrm{dep}(\theta))$-UXP signature. It suffices to show that there are $(\mathbf{1} \cdot \mathrm{dep}(\theta), \mathrm{dep}(\theta))$-UXP reductions implementing π and $\pi^{\forall}$ with respect to the representation $\mathrm{dfnf}(\rho)$.

Consider the projection π. We describe a $(\mathbf{1} \cdot \mathrm{dep}(\theta), \mathrm{dep}(\theta))$-UXP reduction that given $\mathbf{x} \in \mathrm{dom}(\mathrm{dfnf}(\rho))$ computes $\mathbf{y} \in \mathrm{dom}(\mathrm{dfnf}(\rho))$ satisfying $\mathrm{dfnf}(\rho)(\mathbf{y}) = \pi(\mathbf{i}, \mathrm{dfnf}(\rho)(\mathbf{x}))$. Let

$\mathbf{i} \in \mathbf{I}$ and $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_\ell) \in \mathrm{dom}(\mathrm{dfnf}(\rho))$, where each $\mathbf{x}_r$ is in $\mathrm{dom}(\mathrm{un}(\rho))$, and it is thus of the form $(x_{r,1}, \dots, x_{r,j_r})$ with every $x_{r,j} \in \mathrm{dom}(\rho)$. The reduction computes $\mathbf{y}$ as follows:

1: **for** $r \in [1, \ell]$ **do**
2: **if** r is odd **then** $\mathbf{x}'_r \leftarrow \bigvee_{k=1}^{j_r} \dot{\pi}(\mathbf{i}, x_{r,k})$
3: **else** $\mathbf{x}'_r \leftarrow \bigwedge_{k=1}^{j_{r-1}} \left(\dot{\pi}^{\forall}{}_{x_{r-1,k}}(\mathbf{i}, \mathbf{x}_r) \vee \left(\mathbf{x}'_{r-1} - \dot{\pi}(\mathbf{i}, x_{r-1,k}) \right) \right)$
4: **return** $\mathbf{x}'_1 - (\mathbf{x}'_2 - \ldots (\ldots - \mathbf{x}'_\ell))$

Above, note that all calls to the projections $\dot{\pi}$ and $\dot{\pi}^{\forall}$ are with respect to elements in $\mathrm{dom}(\rho)$ and $\mathrm{dom}(\mathrm{un}(\rho))$, respectively, and thus return elements of $\mathrm{dom}(\mathrm{dfnf}(\rho))$, accordingly to the first two properties in Requirement 4.8. The correctness of the procedure stems directly from Lemma 4.6 and Lemma 6.1. For its complexity, note that because of the parameter $\mathrm{dep}(\theta)$, the lengths $\ell, j_1, \dots, j_\ell$ are to be considered constant. Hence, the procedure above boils down to a constant number of calls to suitable UXP reductions with respect to the parameter $\mathrm{dep}(\theta)$, and it is therefore a $(\mathbf{1} \cdot \mathrm{dep}(\theta), \mathrm{dep}(\theta))$-UXP reduction.

The arguments are analogous for $\pi^{\forall}$. In particular, given $\mathbf{i} \in \mathbf{I}$ and $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_\ell) \in \mathrm{dom}(\mathrm{dfnf}(\rho))$, the procedure for $\pi^{\forall}$ simply computes $\dot{\pi}^{\forall}{}_{\top}(\mathbf{i}, \mathbf{x}_1) - \pi(\mathbf{i}, (\mathbf{x}_2, \dots, \mathbf{x}_\ell))$. □

7. Instantiation of the framework to weak linear real arithmetic

In this section, we provide a first non-trivial instantiation of our framework. We consider weak linear real arithmetic (weak LRA), the first-order theory of the structure $(\mathbb{R}, 0, 1, +, =)$, and show that its k negations satisfiability problem lies in PTIME. In fact, since the first-order theories of weak LRA over the reals and rationals are known be elementary equivalent, for technical convenience this section considers the structure $\mathcal{Q} = (\mathbb{Q}, 0, 1, +, =)$ instead.

7.1. **Setup.** According to Proposition 4.2, instantiating the framework requires first to define the domain D, its representation ρ and the change of representation $F\colon \mathrm{CQ}(\sigma) \to \mathrm{dom}(\rho)$. In weak LRA, conjunctions of atomic formulae are systems of affine equations, which over $\mathbb{Q}$ are known to define *affine subspaces* (AS). We define D_n be the set of all affine subspaces of $\mathbb{Q}^n$. Then, $\mathrm{D} = \bigcup_{n \in \mathbb{N}} \mathrm{D}_n$ is the set of all affine subspaces over $\mathbb{Q}^n$, for some n.

Following our notation, let $\nu_{\mathbb{Q}}$ be the (canonical) representation of the rational numbers as pairs (n, d) where $n \in \mathrm{dom}(\nu_{\mathbb{Z}})$ and $d \in \mathrm{dom}(\nu_{\mathbb{N}})$, and $\nu_{\mathbb{Q}}(n, d) = \frac{\nu_{\mathbb{Z}}(n)}{\nu_{\mathbb{N}}(d)}$. Note that the structure $(\mathbb{Q}, +, \times, -, /, =, \leqslant)$ has a $(\nu_{\mathbb{Q}}, \mathbf{1})$-UXP signature since arithmetic operations on $\mathbb{Q}$ are in PTIME. Vectors are simply represented as tuples: $\nu_{\mathbb{Q}^n} := \nu_{\mathbb{Q}}^n$ for all n. To ease the presentation, we do not make a distinction between $\mathbb{Q}^n$ and $\mathrm{dom}(\nu_{\mathbb{Q}^n})$. We represent the affine subspaces by an offset and a basis (i.e., a set of linearly independent vectors) of the linear part. Formally, for every $n \in \mathbb{N}$, if v_0 represents a vector in $\mathbb{Q}^n$, and $v_1, \dots, v_k$ represent linearly independent vectors in $\mathbb{Q}^n$, then we let

$$\rho_{\mathrm{AS}}(n, v_0, \cdots, v_k) \coloneqq \nu_{\mathbb{Q}^n}(v_0) + \mathrm{span}_{\mathbb{Q}}\{\nu_{\mathbb{Q}^n}(v_1), \cdots, \nu_{\mathbb{Q}^n}(v_k)\}.$$

Here, $+$ stands for the Minkowski sum, and given $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbb{Q}^n$, $\mathrm{span}_{\mathbb{Q}}\{\mathbf{v}_1, \dots, \mathbf{v}_k\} \coloneqq \{\lambda_1 \cdot \mathbf{v}_1 + \cdots + \lambda_k \cdot \mathbf{v}_k : \lambda_1, \dots, \lambda_k \in \mathbb{Q}\}$ is a *linear subspace*. We call the vectors $v_0, \dots, v_k$ above a V-representation of the affine subspace $\rho_{\mathrm{AS}}(n, v_0, \cdots, v_k)$. The presence of the dimension n as first argument of ρ_{AS} is required to map the vectors to a shifted lattice of the right dimension. The definition of ρ_{AS} is surjective but not injective, as every affine subspace admits an infinite number of V-representations. We also consider a constant symbol

$\varnothing$ to represent the empty set, i.e., $\rho_{\mathrm{AS}}(\varnothing) \coloneqq \emptyset$, and assume $\emptyset \in \mathrm{D}_n$ for every $n \in \mathbb{N}$. Note that $(0, ())$ stands instead for the only vector space of dimension 0; hence in particular $\rho_{\mathrm{AS}}(0, ()) = \{()\} \neq \rho_{\mathrm{AS}}(\varnothing)$. The map ρ_{AS} is the map ρ required by Proposition 4.2.

Finally, as a preliminary step for instantiating the framework, we must provide a change of representation F from conjunctions of atomic formulae to elements in $\mathrm{dom}(\rho_{\mathrm{AS}})$. This function is essentially given by the PTIME algorithm to compute the echelon form of a matrix with rational entries:

Proposition 7.1 [Edm67]**.** *There is a PTIME algorithm computing the echelon form, along with the transformation matrices, of a given matrix with rational entries.*

Since F runs in PTIME, the parameter ξ from Proposition 4.2 equals $\mathbf{1}$. To instantiate the framework, it now suffices to show that the structure $\mathcal{D}$ from Proposition 4.2 has a $(\mathrm{dfnf}(\rho_{\mathrm{SL}}), \mathrm{dep}(\mathbf{1}))$-UXP signature, by establishing Requirement 4.3 and 4.8.

7.2. **Requirement 4.3: Boolean connectives.** The next lemma establishes Item (F1) of Requirement 4.3. Therein, $+$ stands for the Minkowski sum, which we later need to implement subsequent parts of the framework.

Lemma 7.2. *The structure* $(\mathrm{D}, \wedge, +, \leq)$ *has a* $(\rho_{\mathrm{AS}}, \mathbf{1})$*-UXP signature.*

Proof. Observe that it suffices to show how to compute $\leq$, $\wedge$ and $+$ on affine subspaces of the same dimension: given $(n, \mathbf{v}_0, \ldots, \mathbf{v}_k)$ and $(m, \mathbf{w}_0, \ldots, \mathbf{w}_j)$ from $\mathrm{dom}(\rho_{\mathrm{AS}})$ with $n > m$, we append $n - m$ many zeros to each vector $\mathbf{w}_i$ in order to obtain an affine subspace of dimension n. (In the case of subspaces having same dimension, $\leq$ and $\wedge$ correspond to $\subseteq$ and $\cap$, so we often use these symbols interchangeably.) Moreover, all operations are straightforward when at least one of the two arguments is $\varnothing$. Below, let $X \coloneqq (n, \mathbf{v}_0, \ldots, \mathbf{v}_k)$ and $Y \coloneqq (n, \mathbf{w}_0, \ldots, \mathbf{w}_j)$. For brevity, we write V and W for the matrices having as columns $\mathbf{v}_1, \ldots, \mathbf{v}_k$ and $\mathbf{w}_1, \ldots, \mathbf{w}_j$, respectively.

The algorithm for testing $\rho_{\mathrm{AS}}(X) \subseteq \rho_{\mathrm{AS}}(Y)$ is simple: the inclusion holds if and only if $\mathbf{v}_0 \in \mathbf{w}_0 + W \cdot \mathbb{Q}^j$ and, for every $i \in [1, k]$, $\mathbf{v}_i \in W \cdot \mathbb{Q}^j$. These membership queries boil down to solving systems of equations over $\mathbb{Q}$, which can be done in polynomial time by, e.g., Gaussian elimination or by appealing to Proposition 7.1.

Computing the Minkowski sum is also simple: we have $\rho_{\mathrm{AS}}(X) + \rho_{\mathrm{AS}}(Y) = (\mathbf{v}_0 + \mathbf{w}_0) + U$ with $U \coloneqq \mathrm{span}_{\mathbb{Q}}(\mathbf{v}_1, \ldots, \mathbf{v}_k, \mathbf{w}_1, \ldots, \mathbf{w}_j)$. Hence, the algorithm consists in computing a basis (of independent vectors) $\mathbf{t}_1, \ldots, \mathbf{t}_r$ for the vector space U, and return $(n, (\mathbf{v}_0 + \mathbf{w}_0), \mathbf{t}_1, \ldots, \mathbf{t}_r)$. The basis can be directly extracted from the echelon form of the matrix $\begin{bmatrix} V & W \end{bmatrix}$, so the algorithm runs in polynomial time (Proposition 7.1).

For the intersection $\rho_{\mathrm{AS}}(X) \cap \rho_{\mathrm{AS}}(Y)$, we have:

$$\begin{aligned}
\rho_{\mathrm{AS}}(X) \cap \rho_{\mathrm{AS}}(Y) &= \{\mathbf{v}_0 + V \cdot \mathbf{y} : \mathbf{y} \in \mathbb{Q}^k \text{ and } \mathbf{v}_0 + V \cdot \mathbf{y} = \mathbf{w}_0 + W \cdot \mathbf{z} \text{ for some } \mathbf{z} \in \mathbb{Q}^j\} \\
&= \mathbf{v}_0 + \{V \cdot \mathbf{y} : \mathbf{y} \in \mathbb{Q}^k \text{ and } V \cdot \mathbf{y} - W \cdot \mathbf{z} = \mathbf{w}_0 - \mathbf{v}_0 \text{ for some } \mathbf{z} \in \mathbb{Q}^j\} \\
&= \mathbf{v}_0 + V \cdot \pi(U),
\end{aligned}$$

where $\pi(\mathbf{y}, \mathbf{z}) \coloneqq \mathbf{y}$ is the projection from $\mathbb{Q}^{k+j}$ to $\mathbb{Q}^k$, and $U \coloneqq \{(\mathbf{y}, \mathbf{x}) : V \cdot \mathbf{y} - W \cdot \mathbf{z} = \mathbf{w}_0 - \mathbf{v}_0\}$ is an affine subspace whose representation $(k + j, \mathbf{u}_0, \ldots, \mathbf{u}_\ell) \in \mathrm{dom}(\rho_{\mathrm{AS}})$ can be computed in polynomial time by putting the matrix $\begin{bmatrix} V^T & -W^T \end{bmatrix}^T$ in echelon form with Proposition 7.1. Let $\mathbf{u}'_i$ be the vector obtained from $\mathbf{u}_i$ by projecting away the last j entries. We have $\rho_{\mathrm{AS}}(X) \cap \rho_{\mathrm{AS}}(Y) = (\mathbf{v}_0 + V \cdot \mathbf{u}'_0) + U$, with $U \coloneqq \mathrm{span}_{\mathbb{Q}}((V \cdot \mathbf{u}'_1), \ldots, (V \cdot \mathbf{u}'_\ell))$. The algorithm

computes a basis $\mathbf{t}_1, \ldots, \mathbf{t}_r$ for the vector space U (as done for the Minkowski sum), and returns $(n, (\mathbf{v}_0 + V \cdot \mathbf{u}'_0), \mathbf{t}_1, \ldots, \mathbf{t}_r)$. □

Item (F2) requires an algorithm for inclusion testing of unions of affine subspaces represented as $\mathrm{un}(\rho_{\mathrm{AS}})$. The algorithm relies on the following well-known result.

Lemma 7.3. *Let X be an affine subspace and $Y = \bigcup_{i \in I} Y_i$ be a (finite) union of affine subspaces. Then, $X \subseteq Y$ if and only if $X \subseteq Y_i$ for some $i \in I$.*

Proof. This can be shown by induction on the size of I. If $|I| = 1$ then the result is trivial. By induction, suppose the result to be true up to a certain size k and let I of size $k+1$. Let $i_0 \in I$ and $J = I \setminus \{i_0\}$. Assume that $X \subseteq \bigcup_{i \in I} Y_i$. If $X \subseteq Y_{i_0}$ then the result is proven. Otherwise, $X \not\subseteq Y_{i_0}$ and we will show that $X \subseteq \bigcup_{j \in J} Y_j$ which will conclude by induction.

Pick $x \in X \setminus Y_{i_0}$ and let $y \in X$ be arbitrary. The case $y \in \bigcup_{j \in J} Y_j$ is trivial. Otherwise, it must be the case that $y \in Y_{i_0}$ since $X \subseteq \bigcup_{i \in I} Y_i$. In particular, $x \neq y$ because $x \notin Y_{i_0}$. Since $x, y \in X$ and $x \neq y$, the affine line L that passes through x and y is contained in X (X being an affine subspace) and contains infinitely many points. As a result, there exists $i_1 \in I$ such that $L \cap Y_{i_1}$ contains at least two points. But Y_{i_1} is an affine subspace so Y_{i_1} contains L. This implies that $i_0 \neq i_1$ because otherwise we would have $x \in L \subseteq Y_{i_1} = Y_{i_0}$ which is not possible. Therefore $i_1 \in J$ and $y \in L \subseteq Y_{i_1} \subseteq \bigcup_{j \in J} Y_j$. □

Lemma 7.3 allows reducing inclusion testing of union of affine subspaces to inclusion testing on affine subspaces, which is in polynomial time by Lemma 7.2. Item (F2) follows.

Lemma 7.4. *The structure $(\mathrm{un}(\mathrm{D}), \leq)$ has a $(\mathrm{un}(\rho_{\mathrm{AS}}), \mathbf{1})$-UXP signature.*

7.3. **Requirement 4.8: Projections and universal projections.** We now move to the second Requirement of the framework, which adds support for projections and universal projections. Establishing the Items (F3) and (F5) of Requirement 4.8 is trivial; in V-representation, (existential) projection is a simple operation. Indeed, given $X \in \mathrm{dom}(\rho_{\mathrm{AS}})$ and $\mathbf{i} \in \mathbf{I}$, $\dot{\pi}(\mathbf{i}, X)$ can be computed by simply crossing out the entries of all vectors of X corresponding to the indices in $\mathbf{i}$. The resulting vectors $\mathbf{v}_0, \ldots, \mathbf{v}_\ell$ are such that $\dot{\pi}(\mathbf{i}, \rho_{\mathrm{AS}}(X)) = \nu_{\mathbb{Q}^n}(\mathbf{v}_0) + V$ with $V \coloneqq \mathrm{span}_{\mathbb{Q}}\{\nu_{\mathbb{Q}^n}(\mathbf{v}_1), \cdots, \nu_{\mathbb{Q}^n}(\mathbf{v}_\ell)\}$. It then suffices to compute, starting from $\mathbf{v}_1, \ldots, \mathbf{v}_\ell$, a basis for V, which can be done in polynomial time following Proposition 7.1. The following result is thus immediate.

Lemma 7.5. *The structure $(\mathrm{D}, (\dot{\pi}, \mathbf{I}))$ has a $(\rho_{\mathrm{AS}}, \mathbf{1})$-UXP signature.*

On the contrary, universal projections are not *a priori* easy to compute. Our main observation is that affine subspaces and their unions have very special properties that allow us to express the universal projection in terms of projections. For simplicity, below we index entries in vectors starting from one, and instead of considering projections over arbitrary vectors of indices $\mathbf{i}$, we assume $\mathbf{i} = [1, k]$ for some $k \in \mathbb{N}$ so that $\pi^{\forall}_Z(\mathbf{i}, X)$ projects over the first k dimensions. This is w.l.o.g., as we can reorder components appropriately. So, let $\pi^{\forall}_Z(k.X) \coloneqq \pi^{\forall}_Z([1, k], X)$, and $\pi(k, X) \coloneqq \pi([1, k], X)$. Given a set $X \subseteq \mathbb{Q}^n$ and $k \leqslant n$, it will be useful to introduce the following set, for any $\mathbf{p} \in \mathbb{Q}^{n-k}$:

$$X_{\mathbf{p}} \coloneqq \{\mathbf{t} \in \mathbb{Q}^k : (\mathbf{p}, \mathbf{t}) \in X\}.$$

Note that the value of k is implicit in the notation, but it will always be clear from the context. We start with an auxiliary lemma.

Lemma 7.6. *Let $X \subseteq \mathbb{Q}^n$ be an affine subspace, and $k \leq n$. There is a linear subspace $V \subseteq \mathbb{Q}^k$ such that for every $\mathbf{p} \in \pi(k, X)$, there is $\mathbf{b} \in \mathbb{Q}^k$ s.t. $X_{\mathbf{p}} = \mathbf{b} + V$. Furthermore, there is a polynomial time algorithm that given in input an element of $\mathrm{dom}(\rho_{\mathrm{AS}})$ representing X, and k (encoded in unary), computes an element of $\mathrm{dom}(\rho_{\mathrm{AS}})$ representing V.*

Proof. If $\pi(k, X) = \emptyset$, then the statement allows to take an arbitrary linear space $V \subseteq \mathbb{Q}^k$. Otherwise, consider $\mathbf{p}_0 \in \pi(k, X)$. Then $X_{\mathbf{p}_0} \neq \emptyset$ is an affine subspace, hence $X_{\mathbf{p}_0} = \mathbf{t}_0 + V$ for some $t_0 \in \mathbb{Q}^k$ and $V \subseteq \mathbb{Q}^k$ a linear subspace. We show that V is the linear subspace in the statement of the lemma. Let $\mathbf{p} \in \pi(k, X)$, and $\mathbf{t}' \in \mathbb{Q}^k$ such that $(\mathbf{p}, \mathbf{t}') \in X$. Denote by $\mathrm{Lin}(X)$ the *linear part* of X, i.e., $X = \mathbf{v} + \mathrm{Lin}(X)$ for some vector $\mathbf{v}$. The following chain of equivalences shows that, for every $\mathbf{x} \in \mathbb{Q}^k$, $\mathbf{x} \in X_{\mathbf{p}}$ if and only if $\mathbf{x} \in \mathbf{t}' + V$.

$$
\begin{aligned}
\mathbf{x} \in X_{\mathbf{p}} &\Leftrightarrow (\mathbf{p}, \mathbf{x}) \in X \\
&\Leftrightarrow (\mathbf{p}, \mathbf{x}) - (\mathbf{p}, \mathbf{t}') \in \mathrm{Lin}(X) && \text{since } X \text{ is an affine subspace and } (\mathbf{p}, \mathbf{t}') \in X \\
&\Leftrightarrow (\mathbf{0}, \mathbf{x} - \mathbf{t}') \in \mathrm{Lin}(X) \\
&\Leftrightarrow (\mathbf{p}_0, \mathbf{t}_0) + (\mathbf{0}, \mathbf{x} - \mathbf{t}') \in X && \text{since } X \text{ is an affine subspace and } (\mathbf{p}_0, \mathbf{t}_0) \in X \\
&\Leftrightarrow (\mathbf{p}_0, \mathbf{t}_0 + \mathbf{x} - \mathbf{t}') \in X \\
&\Leftrightarrow \mathbf{t}_0 + \mathbf{x} - \mathbf{t}' \in X_{\mathbf{p}_0} \\
&\Leftrightarrow \mathbf{x} - \mathbf{t}' \in V && \text{since } X_{\mathbf{p}_0} = \mathbf{t}_0 + V \\
&\Leftrightarrow \mathbf{x} \in \mathbf{t}' + V
\end{aligned}
$$

Consider a representation $(n, \mathbf{v}_0, \ldots, \mathbf{v}_\ell)$ of X. In order to compute V, we first compute (an arbitrary) $\mathbf{p}_0 \in \pi(k, X)$, and then consider the quantified system of equalities

$$\exists y_1, \ldots, y_\ell : \begin{bmatrix} \mathbf{x} \\ \mathbf{p}_0 \end{bmatrix} = \mathbf{v}_0 + \mathbf{v}_1 \cdot y_1 + \ldots \mathbf{v}_n \cdot y_\ell.$$

Note that this is a formula fo weak LRA. Following Proposition 7.1 and Lemma 7.5, we already know how to compute an affine subspace for the quantifier-free part of this formula (which has variables $\mathbf{x}, y_1, \ldots, y_\ell$), to then project away the coordinates corresponding to $y_1, \ldots, y_\ell$. The result is a family of vectors $\mathbf{w}_0, \ldots, \mathbf{w}_r$ such that $X_{\mathbf{p}_0} = \mathbf{w}_0 + V$ with $V \coloneqq \mathrm{span}_{\mathbb{Q}}\{\mathbf{w}_1, \ldots, \mathbf{w}_r\}$. Therefore, it suffices to return $(k, \mathbf{0}, \mathbf{w}_1, \ldots, \mathbf{w}_r)$ as a representation of V. □

Lemma 7.6 allows us to tackle relative universal projections $\pi^{\forall}_Z(k, X)$ in the cases where both Z and X are affine subspaces.

Lemma 7.7. *Let X and Z be affine subspaces, and let $k \in \mathbb{N}$. Then either $\pi^{\forall}_Z(k, X) = \emptyset$ or $\pi^{\forall}_Z(k, X) = \pi(k, X \wedge Z)$. Moreover, there is an algorithm that given in input $X, Z \in \mathrm{dom}(\rho_{\mathrm{AS}})$ and $k \in \mathbb{N}$ in unary, returns $Y \in \mathrm{dom}(\rho_{\mathrm{AS}})$ such that $\rho_{\mathrm{AS}}(Y) = \pi^{\forall}_{\rho_{\mathrm{AS}}(Z)}(k, \rho_{\mathrm{AS}}(X))$. The algorithm runs in polynomial time.*

Proof. Following the discussion at the beginning of the proof of Lemma 7.2, it is easy to see that the only interesting case is when X and Z are affine subspaces having the same dimension. First note that by definition, we must have $\pi^{\forall}_Z(k, X) \subseteq \pi(k, X \cap Z)$. Indeed, if $\mathbf{x} \in \pi^{\forall}_Z(k, X)$ then $\mathbf{x} \in \pi(k, Z)$ so there is $\mathbf{t}$ such that $(\mathbf{x}, \mathbf{t}) \in Z$ and then $(\mathbf{x}, \mathbf{t}) \in X$ so $(\mathbf{x}, \mathbf{t}) \in X \cap Z$. Consider then the reverse inclusion. By Lemma 7.6, there are linear subspaces V and W such that

$$\forall \mathbf{p} \in \pi(k, Z), \exists \mathbf{b}_{\mathbf{p}} \in \mathbb{Q}^k, X_{\mathbf{p}} = \mathbf{b}_{\mathbf{p}} + V \qquad \text{and} \qquad \forall \mathbf{p} \in \pi(k, X), \exists \mathbf{g}_{\mathbf{p}} \in \mathbb{Q}^k, Z_{\mathbf{p}} = \mathbf{g}_{\mathbf{p}} + W.$$

It follows that

$$\begin{aligned}\pi^{\forall}_{Z}(k,X) &= \{\mathbf{x} \in \pi(k,Z) : \forall \mathbf{t} \in \mathbb{Q}^k, (\mathbf{x},\mathbf{t}) \in Z \Rightarrow (\mathbf{x},\mathbf{t}) \in X\} \\ &= \{\mathbf{x} \in \pi(k,Z) : Z_{\mathbf{x}} \subseteq X_{\mathbf{x}}\} \\ &= \{\mathbf{x} \in \pi(k,Z) : \mathbf{g}_{\mathbf{x}} - \mathbf{b}_{\mathbf{x}} + W \subseteq V\}. \end{aligned} \tag{7.1}$$

We now distinguish two cases:

(1) If $W \not\subseteq V$ then it cannot be the case that $\mathbf{g}_{\mathbf{x}} - \mathbf{b}_{\mathbf{x}} + W \subseteq V$. This is true for every $\mathbf{x} \in \pi(k,Z)$, and therefore $\pi^{\forall}_{Z}(k,X) = \emptyset$.
(2) If $W \subseteq V$, then consider $\mathbf{x} \in \pi(k, X \cap Z)$. There is $\mathbf{t}$ such that $(\mathbf{x},\mathbf{t}) \in X \cap Z$, that is, $\mathbf{t} \in X_{\mathbf{x}} \cap Z_{\mathbf{x}}$. In particular, $\mathbf{t} = \mathbf{b}_{\mathbf{x}} + \mathbf{v} = \mathbf{g}_{\mathbf{x}} + \mathbf{w}$ for some $\mathbf{v} \in V$ and $\mathbf{w} \in W$. But then $\mathbf{g}_{\mathbf{x}} - \mathbf{b}_{\mathbf{x}} = \mathbf{v} - \mathbf{w} \in V$ since $W \subseteq V$ and W is a linear subspace. Therefore, $\mathbf{g}_{\mathbf{x}} - \mathbf{b}_{\mathbf{x}} + W \subseteq \mathbf{g}_{\mathbf{x}} - \mathbf{b}_{\mathbf{x}} + V = V$. This shows that $\mathbf{x} \in \pi^{\forall}_{Z}(k,X)$ by Equation (7.1).

Algorithmically, given $X, Z \in \mathrm{dom}(\rho_{\mathrm{AS}})$, one computes an element of $\mathrm{dom}(\rho_{\mathrm{AS}})$ representing $\pi^{\forall}_{\rho_{\mathrm{AS}}(Z)}(k, \rho_{\mathrm{AS}}(X))$ as follows. First, if one among X or Z is $\varnothing$, the algorithm simply returns $\varnothing$. Otherwise, it computes representations of the linear subspaces V and W above, by using the algorithm from Lemma 7.6. By Lemma 7.2, testing $W \subseteq V$ can be performed in polynomial time. If the inclusion is not true, the algorithm returns $\emptyset$. Otherwise, it returns $\pi(k, X \wedge Z)$, which can again be computed in polynomial time by Lemma 7.5. □

We extend Lemma 7.7 to union of affine subspaces, completing Requirement 4.8:

Lemma 7.8. *Let Z be an affine subspace, and $X = \bigvee_{j=1}^{m} X_j$, with X_j affine subspace. Let $k \in \mathbb{N}$. Then, $\pi^{\forall}_{Z}(k,X) = \bigvee_{j=1}^{m} \pi^{\forall}_{Z}(k,X_j)$. Moreover, there is an algorithm that given in input $Z \in \mathrm{dom}(\rho_{\mathrm{AS}})$, $X \in \mathrm{dom}(\mathrm{un}(\rho_{\mathrm{AS}}))$ and $k \in \mathbb{N}$ in unary, returns $Y \in \mathrm{dom}(\mathrm{un}(\rho_{\mathrm{AS}}))$ such that $\mathrm{un}(\rho_{\mathrm{AS}})(Y) = \pi^{\forall}_{\rho_{\mathrm{AS}}(Z)}(k, \mathrm{un}(\rho_{\mathrm{AS}})(X))$. The algorithm runs in polynomial time.*

Proof. Again, without loss of generality we assume Z and each X_j to have the same dimension. Note that

$$\pi^{\forall}_{Z}(k,X) = \{\mathbf{x} \in \pi(k,Z) : Z_{\mathbf{x}} \subseteq X_{\mathbf{x}}\} = \{\mathbf{x} \in \pi(k,Z) : X_{\mathbf{x}} \subseteq \bigcup\nolimits_{j=1}^{m} (X_j)_{\mathbf{x}}\}.$$

Recall that $Z_{\mathbf{x}}$ and $(X_j)_{\mathbf{x}}$ are affine subspaces (Lemma 7.6). By Lemma 7.3, $Z_{\mathbf{x}} \subseteq \bigcup_{j=1}^{m} (X_j)_{\mathbf{x}}$ if and only if $Z_{\mathbf{x}} \subseteq (X_j)_{\mathbf{x}}$ for some $j \in [1,m]$. Hence,

$$\begin{aligned}\pi^{\forall}_{Z}(k,X) &= \{\mathbf{x} \in \pi(k,Z) : X_{\mathbf{x}} \subseteq (X_j)_{\mathbf{x}} \text{ for some } j \in [1,m]\} \\ &= \bigcup\nolimits_{j=1}^{m} \{x \in \pi(k,Z) : Z_{\mathbf{x}} \subseteq (X_j)_{\mathbf{x}}\} = \bigcup\nolimits_{j=1}^{m} \pi^{\forall}_{Z}(k,X_j).\end{aligned}$$

Algorithmically, given $Z \in \mathrm{dom}(\rho_{\mathrm{AS}})$ and $(X_1, \ldots, X_m) \in \mathrm{dom}(\mathrm{un}(\rho_{\mathrm{AS}}))$, by Lemma 7.7 we can compute in polynomial time a representation $Y_j \in \mathrm{dom}(\rho_{\mathrm{AS}})$ of $\pi^{\forall}_{\rho_{\mathrm{AS}}(Z)}(k, \rho_{\mathrm{AS}}(X_j))$, for every $j \in [1,m]$. Then, it suffices to return $Y \coloneqq (Y_1, \ldots, Y_m)$. □

The main result of the section follows:

Theorem 7.9. *The k negations satisfiability problem for weak LRA is in PTIME.*

Proof. Lemmas 7.2 and 7.4 imply Requirement 4.3, and Lemmas 7.5 and 7.8 imply Requirement 4.8. By Lemma 4.9, $\mathcal{D}$ has a $(\mathrm{dfnf}(\rho), \mathrm{dep}(\mathbf{1}))$-UXP signature. Then, the theorem follow from Proposition 4.2. □

8. Instantiation of the framework to weak linear integer arithmetic

In this section, we apply our framework to show that the k negations satisfiability problem for weak Presburger arithmetic (weak PA), i.e. the FO theory of the structure $\mathcal{Z} = (\mathbb{Z}, 0, 1, +, =)$, is in PTIME. Our presentation follows quite closely of Section 7, although the details are now more intricate, particularly those involving universal projection.

8.1. **Setup.** We define the domain D, its representation ρ and the change of representation $F\colon \mathrm{CQ}(\sigma) \to \mathrm{dom}(\rho)$ required by Proposition 4.2. In weak PA, conjunctions of atomic formulae are systems of affine equations, which over $\mathbb{Z}$ define *shifted (integer) lattices* (SL), which are not necessarily fully dimensional. We let D_n be the set of all shifted lattices of $\mathbb{Z}^n$, so that D is the set of all shifted lattices of $\mathbb{Z}^n$ for some n. We represent elements in D with the standard representation of shifted lattice as a *base* point together with a collection of linearly independent vectors (the *periods* of the lattice). Recall that we write $\nu_{\mathbb{Z}^n}$ for the canonical representation of $\mathbb{Z}^n$ (see Section 3). Formally, we define the representation ρ_{SL} for shifted lattices as follows. For every $n \in \mathbb{N}$, if v_0 represents a vector in $\mathbb{Z}^n$, and $v_1, \ldots, v_k$ represent linearly independent vectors in $\mathbb{Z}^n$, then $(n, v_0, \ldots, v_k) \in \mathrm{dom}(\rho_{\mathrm{SL}})$ and

$$\rho_{\mathrm{SL}}(n, v_0, \cdots, v_k) \coloneqq \nu_{\mathbb{Z}^n}(v_0) + \mathrm{span}_{\mathbb{Z}}\{\nu_{\mathbb{Z}^n}(v_1), \cdots, \nu_{\mathbb{Z}^n}(v_k)\},$$

which is a shifted lattice in D_n. As in Section 7, $+$ stands for the Minkowski sum, and given $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{Z}^n$, $\mathrm{span}_{\mathbb{Z}}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\} \coloneqq \{\lambda_1 \cdot \mathbf{v}_1 + \cdots + \lambda_k \cdot \mathbf{v}_k : \lambda_1, \ldots, \lambda_k \in \mathbb{Z}\}$ is a (non-shifted) *lattice*. We use a constant symbol $\varnothing$ to represent the empty lattice, i.e., $\rho_{\mathrm{SL}}(\varnothing) \coloneqq \emptyset$, and assume $\emptyset \in \mathrm{D}_n$ for every $n \in \mathbb{N}$.

Below, to ease the presentation, we do not make a distinction between elements of $\mathbb{Z}^n$ and elements of $\mathrm{dom}(\nu_{\mathbb{Z}^n})$, that is we assume $\mathbb{Z}$ to be the set of integers encoded in binary (hence, all algorithm we give assume integers represented via $\nu_{\mathbb{Z}^n}$). Because of this (usual) assumption, occurrences of the map $\nu_{\mathbb{Z}^n}$ are often omitted, and ρ_{SL} is seen as a function taking as input $n \in \mathbb{Z}$ together with linearly independent vectors of $\mathbb{Z}^n$ (or $\varnothing$).

A polynomial time function F allowing to change representation from conjunctions of atomic formulae of $\mathrm{FO}(\mathcal{Z})$ to elements in $\mathrm{dom}(\rho_{\mathrm{SL}})$ can be obtained due to the Hermite normal form of an integer matrix being computable in polynomial time. See [Sch99, Chapter 4] for an introduction to the Hermite normal form. Briefly, recall that the Hermite normal form $H \in \mathbb{Z}^{n\times d}$ of a matrix $A \in \mathbb{Z}^{n\times d}$ is unique and has (among others) the following properties:

- H is lower triangular (so, its non-zero columns are linearly independent) and every pivot of a non-zero column is positive,
- $H = A \cdot U$ for some unimodular matrix $U \in \mathbb{Z}^{d\times d}$ (i.e. a matrix with determinant ± 1),
- H generates the same (non-shifted) lattice as A, i.e., $H \cdot \mathbb{Z}^d = A \cdot \mathbb{Z}^d$.

Proposition 8.1 [KB79]**.** *There is a polynomial time algorithm to compute the Hermite normal form H, along with the transformation matrix U, of a given matrix $A \in \mathbb{Z}^{n\times d}$.*

Lemma 8.2. *There is a polynomial time function F that given in input a system of equations $A \cdot \mathbf{x} = \mathbf{b}$ in d variables, returns $\varnothing$ if the system is unsatisfiable, and otherwise it returns a tuple $(d, \mathbf{v}_0, \ldots, \mathbf{v}_k)$ where $\mathbf{v}_0 \in \mathbb{Z}^d$, $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{Z}^d$ are k linearly independent vectors, and $\{\mathbf{x} \in \mathbb{Z}^d : A \cdot \mathbf{x} = \mathbf{b}\} = \mathbf{v}_0 + \mathrm{span}_{\mathbb{Z}}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$.*

Proof. The function F relies on the algorithm of Proposition 8.1 to compute H in Hermite normal form and the unimodular matrix U such that $H = A \cdot U$. We have $\{\mathbf{x} \in \mathbb{Z}^d : A \cdot \mathbf{x} = \mathbf{b}\} = \{U \cdot \mathbf{y} : \mathbf{y} \in \mathbb{Z}^d \text{ and } H \cdot \mathbf{y} = \mathbf{b}\}$, as setting $\mathbf{x} = U \cdot \mathbf{y}$ yields $\mathbf{y} = U^{-1} \cdot \mathbf{x}$ and $H \cdot U^{-1} = A$.

Let $\mathbf{y} = (y_1, \dots, y_d)$. Since H is triangular, we have $H = [\mathbf{b}_1 \mid \cdots \mid \mathbf{b}_j \mid \mathbf{0} \mid \cdots \mid \mathbf{0}]$ where the columns $\mathbf{b}_1, \dots, \mathbf{b}_j$ are non-zero. Moreover, if $A \cdot \mathbf{x} = \mathbf{b}$ has a solution, then there is a unique way to generate $\mathbf{b}$ as a linear combination of $\mathbf{b}_1, \dots, \mathbf{b}_j$. Finding the values $y_1^*, \dots, y_j^*$ for $y_1, \dots, y_j$ that generate $\mathbf{b}$ is trivial (briefly, y_1^* is the only integer making the first non-zero entry of $\mathbf{b}$ equal to the entry of $\mathbf{b}_1 \cdot y_1^*$ in the same position; update then $\mathbf{b}$ to $\mathbf{b} - \mathbf{b}_1 \cdot y_1^*$ and recursively find values for $y_2, \dots, y_j$). If $y_1^*, \dots, y_j^*$ do not exist, then F outputs $\varnothing$. Otherwise, let $U = [\mathbf{u}_1 \mid \cdots \mid \mathbf{u}_d]$. We set $\mathbf{v}_0 \coloneqq \mathbf{u}_1 \cdot y_1^* + \cdots + \mathbf{u}_j \cdot y_j^*$. Note that, because of the shape of H, the values assigned to the variables $y_{j+1}, \dots, y_d$ do not influence the satisfaction of $H \cdot \mathbf{y} = \mathbf{b}$. Then, F returns $(d, \mathbf{v}_0, \mathbf{u}_{j+1}, \dots, \mathbf{u}_d)$. □

Since the function F runs in polynomial time, the parameter ξ in Proposition 4.2 equals $\mathbf{1}$, and to instantiate the framework we need to show that $\mathcal{D}$ has a $(\mathrm{dfnf}(\rho_{\mathrm{SL}}), \mathrm{dep}(\mathbf{1}))$-UXP signature, by establishing Requirement 4.3 and 4.8.

8.2. **Requirement 4.3, Item (F1): the structure $(\mathrm{D}, \wedge, \leq)$ has a $(\rho_{\mathrm{SL}}, \mathbf{1})$-UXP signature.** Briefly, both the problems of computing intersections and testing inclusions for two shifted lattices represented as in ρ_{SL} reduces to solving systems of linear equations over $\mathbb{Z}$, which can be done in polynomial time again thanks to Proposition 8.1 (alternatively, Lemma 8.2). We now formalise these reductions.

Lemma 8.3. *$(\mathrm{D}, \wedge, \leq)$ has a $(\rho_{\mathrm{SL}}, \mathbf{1})$-UXP signature.*

Proof. First, note that in case one of the inputs given to $\wedge$ or $\leq$ is $\varnothing$, computing the output is trivial. Hence, we only consider the case of non-empty shifted lattices. Moreover, note that we can restrict ourselves to shifted lattices having the same dimension. (As in the previous section, this restriction means that $\wedge$ and $\leq$ are equivalent to $\cap$ and $\subseteq$.) Indeed, consider elements $X = (n, \mathbf{v}_0, \dots, \mathbf{v}_k)$ and $Y = (m, \mathbf{w}_0, \dots, \mathbf{w}_j)$ from $\mathrm{dom}(\rho_{\mathrm{SL}})$, with $n < m$. Recall that, from the definition of FO theory given in Section 2.3, the shifted lattice $\rho_{\mathrm{SL}}(X)$ can be extended into a shifted lattice in $\mathbb{Z}^m$ as $\rho_{\mathrm{SL}}(X) \times \mathbb{Z}^{m-n}$. At the level of representation, this corresponds to adding $m - n$ vectors $\mathbf{v}_{k+1}, \dots, \mathbf{v}_{k+m-n}$ such that, for $i \in [1, m-n]$, $\mathbf{v}_i$ has a 1 in position $n + i$ and zeros in all other position (i.e., essentially adding an identity matrix for the last $m - n$ dimensions). One can then consider $X' = (m, \mathbf{v}_0, \dots, \mathbf{v}_k, \dots, \mathbf{v}_{k+m-n})$ instead of X to compute $\wedge$ and $\leq$. Adding the new vectors can be done in polynomial time.

Consider $X = (n, \mathbf{v}_0, \dots, \mathbf{v}_k)$ and $Y = (n, \mathbf{w}_0, \dots, \mathbf{w}_j)$ from $\mathrm{dom}(\rho_{\mathrm{SL}})$. We show that computing an element of $\mathrm{dom}(\rho_{\mathrm{SL}})$ representing $\rho_{\mathrm{SL}}(X) \cap \rho_{\mathrm{SL}}(Y)$ can be done in polynomial time. Let $A \coloneqq [\mathbf{v}_1 \mid \cdots \mid \mathbf{v}_k]$ and $B \coloneqq [\mathbf{w}_1 \mid \cdots \mid \mathbf{w}_j]$ be the matrices whose columns correspond to the periods of the shifted lattices represented by X and Y, respectively. Then,

$$\begin{aligned}
\rho_{\mathrm{SL}}(X) \cap \rho_{\mathrm{SL}}(Y) &= \left\{ \mathbf{v} \in \mathbb{Z}^m : \mathbf{v} = \mathbf{v}_0 + A \cdot \mathbf{y} = \mathbf{w}_0 + B \cdot \mathbf{z} \text{ for some } \mathbf{y} \in \mathbb{Z}^k \text{ and } \mathbf{z} \in \mathbb{Z}^j \right\} \\
&= \mathbf{v}_0 + \left\{ A \cdot \mathbf{y} : \mathbf{y} \in \mathbb{Z}^k \text{ and } A\mathbf{y} - B\mathbf{z} = \mathbf{w}_0 - \mathbf{v}_0 \text{ for some } \mathbf{z} \in \mathbb{Z}^j \right\} \\
&= \mathbf{v}_0 + A \cdot p(Z),
\end{aligned}$$

where $p \colon \mathbb{Z}^k \times \mathbb{Z}^j \to \mathbb{Z}^k$ is the projection $p(\mathbf{y}, \mathbf{z}) = \mathbf{y}$ and $Z \coloneqq \{(\mathbf{y}, \mathbf{z}) \in \mathbb{Z}^k \times \mathbb{Z}^j : A\mathbf{y} - B\mathbf{z} = \mathbf{w}_0 - \mathbf{v}_0\}$. In particular, Z is a shifted lattice given by the set of solutions of the weak PA formula $A\mathbf{y} - B\mathbf{z} = \mathbf{w}_0 - \mathbf{v}_0$. By appealing to Lemma 8.2, we can compute a representation $(k + j, \mathbf{z}_0, \dots, \mathbf{z}_\ell)$ in $\mathrm{dom}(\rho_{\mathrm{SL}})$ of Z. The projection p simply

removes the last j components of $\mathbf{z}_0, \ldots, \mathbf{z}_\ell$, resulting in vectors $\mathbf{z}'_0, \ldots, \mathbf{z}'_\ell \in \mathbb{Z}^k$. We have $\rho_{\mathrm{SL}}(X) \cap \rho_{\mathrm{SL}}(Y) = \mathbf{v}_0 + A \cdot \mathbf{z}'_0 + \mathrm{span}_{\mathbb{Z}}\{A \cdot \mathbf{z}'_1, \ \ldots, \ A \cdot \mathbf{z}'_r\}$. A representation for $\rho_{\mathrm{SL}}(X) \cap \rho_{\mathrm{SL}}(Y)$ is then given by $(n, \mathbf{v}_0 + A \cdot \mathbf{z}'_0, \mathbf{u}_1, \ldots, \mathbf{u}_r)$, where $\mathbf{u}_1, \ldots, \mathbf{u}_r$ form a basis for $\mathrm{span}_{\mathbb{Z}}\{A \cdot \mathbf{z}'_1, \ \ldots, \ A \cdot \mathbf{z}'_r\}$. To compute such a basis (in polynomial time) it suffices to put the matrix $[A \cdot \mathbf{z}'_1 \mid \cdots \mid A \cdot \mathbf{z}'_r]$ in Hermite normal form and take as $\mathbf{u}_1, \ldots, \mathbf{u}_r$ all its non-zero columns.

Consider now the case of establishing whether $\rho_{\mathrm{SL}}(X) \subseteq \rho_{\mathrm{SL}}(Y)$. It is easy to see that this inclusion holds if and only if $\mathbf{v}_0 \in \rho_{\mathrm{SL}}(Y)$ and for every $i \in [1, k]$, $\mathbf{v}_i \in \mathrm{span}_{\mathbb{Z}}\{\mathbf{w}_1, \ldots, \mathbf{w}_j\}$. The right to left direction is trivial: as $\mathrm{span}_{\mathbb{Z}}\{\mathbf{w}_1, \ldots, \mathbf{w}_j\}$ is closed under linear combinations, we conclude $\mathrm{span}_{\mathbb{Z}}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\} \subseteq \mathrm{span}_{\mathbb{Z}}\{\mathbf{w}_1, \ldots, \mathbf{w}_j\}$ which, together with $\mathbf{v}_0 \in \rho_{\mathrm{SL}}(Y)$, implies $\rho_{\mathrm{SL}}(X) \subseteq \rho_{\mathrm{SL}}(Y)$. For the left to right direction, $\rho_{\mathrm{SL}}(X) \subseteq \rho_{\mathrm{SL}}(Y)$ directly implies $\mathbf{v}_0 \in \rho_{\mathrm{SL}}(Y)$ and $\mathbf{v}_0 + \mathbf{v}_i \in \rho_{\mathrm{SL}}(Y)$, for every $i \in [1, k]$. Then,

$$\begin{aligned}
\mathbf{v}_i &= (\mathbf{v}_0 + \mathbf{v}_i) - \mathbf{v}_0 \\
&= \mathbf{w}_0 + B \cdot \mathbf{z}_1 - (\mathbf{w}_0 + B \cdot \mathbf{z}_2) \text{ for some } \mathbf{z}_1, \mathbf{z}_2 \in \mathbb{Z}^j \\
&= B \cdot (\mathbf{z}_1 - \mathbf{z}_2) \text{ for some } \mathbf{z}_1, \mathbf{z}_2 \in \mathbb{Z}^j \\
&\in \mathrm{span}_{\mathbb{Z}}\{\mathbf{w}_1, \ldots, \mathbf{w}_j\}.
\end{aligned}$$

The various membership queries $\mathbf{v}_0 \in \rho_{\mathrm{SL}}(Y)$ and, for all $i \in [1, k]$, $\mathbf{v}_i \in \mathrm{span}_{\mathbb{Z}}\{\mathbf{w}_1, \ldots, \mathbf{w}_j\}$ ask for the feasibility of some systems of linear equations. This problem can be solved in polynomial time by Gaussian elimination (or by checking if F in Lemma 8.2 returns $\varnothing$). □

8.3. **Auxiliary lemmas on lattices.** Before moving to Item (F2) of Requirement 4.3, a few more notions and lemmas on lattices are required. Consider linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{Z}^d$ and the lattice $L \coloneqq \mathrm{span}_{\mathbb{Z}}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$. The determinant of L is defined by $\det(L) \coloneqq \sqrt{\det(B^\intercal B)}$ where $B \coloneqq [\mathbf{v}_1 \mid \cdots \mid \mathbf{v}_k]$. When L is fully-dimensional (i.e., $k = d$), we have $\det(L) = |\det(B)|$.

Proposition 8.4 [Mic12, Lecture 1, Theorem 16]**.** *Let $L \subseteq \mathbb{Z}^d$ be a fully-dimensional lattice. Then,* $\det(L) \cdot \mathbb{Z}^d \subseteq L$.

We let $L^\perp$ denote the *orthogonal lattice* to L, given by $L^\perp \coloneqq \{\mathbf{y} \in \mathbb{Z}^d : \forall \mathbf{x} \in L, \langle \mathbf{y}, \mathbf{x} \rangle = 0\}$, where $\langle \cdot, \cdot \rangle$ stands for the dot product. See e.g. [NS97].

Lemma 8.5. *Let $L \subseteq \mathbb{Z}^d$ be a lattice of dimension k. Then, $L^\perp$ is a lattice of dimension $d - k$, and $L + L^\perp$ is a fully-dimensional lattice. Moreover, there is a polynomial time algorithm that on input $X = (d, \mathbf{0}, \mathbf{v}_1, \ldots, \mathbf{v}_k) \in \mathrm{dom}(\rho_{\mathrm{SL}})$ returns a basis $\mathbf{v}_{k+1}, \ldots, \mathbf{v}_d \in \mathbb{Z}^d$ of $L^\perp$, where $L \coloneqq \rho_{\mathrm{SL}}(X)$; and the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ form a basis of $L + L^\perp$.*

Proof. For $k = 0$, i.e., when L has dimension 0, $L^\perp = \mathbb{Z}^d$ and the lemma is trivial. Below, we show the result for $k = 1$, and then generalise it to arbitrary $k \geq 2$.

Let $L = \mathbb{Z} \cdot \mathbf{v}_1$ where $\mathbf{v}_1 = (u_1, \ldots, u_d) \in \mathbb{Z}^d \setminus \{\mathbf{0}\}$. Without loss of generality, since $\mathbf{v}_1 \neq 0$, we can permute the coordinates so that $u_1 \neq 0$. For $j \in [1, d]$, let $g_j \coloneqq \gcd(u_1, \ldots, u_j)$, which is non-zero as $u_1 \neq 0$. Note that $g_{j+1} = \gcd(g_j, u_{j+1})$ (for $j < d$). We rely on the extended Euclidean algorithm for GCD to compute in polynomial time each g_j and the Bézout's coefficients $a_1, \ldots, a_j \in \mathbb{Z}$ such that $g_j = a_{j,1} u_1 + \cdots + a_{j,j} u_j$. For $j \in [1, d-1]$, let

$$\mathbf{v}_{j+1} \coloneqq (-\beta_j a_{j,1}, \ \ldots, -\beta_j a_{j,j}, \ \tfrac{g_j}{g_{j+1}}, \ 0, \ \ldots, \ 0), \text{ where } \beta_j \coloneqq \tfrac{u_{j+1}}{g_{j+1}}.$$

Note that both β_j and $\frac{g_j}{g_{j+1}}$ belong to $\mathbb{Z}$, by definition of g_j and g_{j+1}.

Observe that $\mathbf{v}_2, \dots, \mathbf{v}_d$ is a family of $d-1$ linearly independent vectors, as they form an echelon family of vectors. Let $L' \coloneqq \operatorname{span}_{\mathbb{Z}}\{\mathbf{v}_2, \dots, \mathbf{v}_d\}$, which is a lattice of dimension $d-1$. We claim that $L' = L^\perp$.

First, we show that $L' \subseteq L^\perp$. Let $j \in \{1, \dots, d-1\}$, we need to show that $\langle \mathbf{v}_1, \mathbf{v}_{j+1} \rangle = 0$. This is simple to check:

$$\begin{aligned}
\langle \mathbf{v}_1, \mathbf{v}_{j+1} \rangle &= u_1 \cdot (-\beta_j a_{j,1}) + \cdots + u_j \cdot (-\beta_j a_{j,j}) + u_{j+1} \cdot \tfrac{g_j}{g_{j+1}} \\
&= \beta_j \left(g_j - u_1 a_{1,j} - \cdots - u_j a_{j,j}\right) && \text{since } u_{j+1} \cdot \tfrac{g_j}{g_{j+1}} = \beta_j g_j \\
&= 0 && \text{by def. of } g_j.
\end{aligned}$$

Let us now show that $L^\perp \subseteq L'$. For any $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{Z}^d$, define $k(\mathbf{x}) \coloneqq \max(\{0\} \cup \{i : x_i \neq 0\})$. We show that if $\mathbf{x} \in L^\perp$ then $\mathbf{x} \in L'$, for every $\mathbf{x} \in \mathbb{Z}^d$, by induction on $k(\mathbf{x})$. If $k(\mathbf{x}) = 0$ then $\mathbf{x} = 0$ so the result is trivial. Otherwise, let $\ell \coloneqq k(\mathbf{x})$ and observe that $\langle \mathbf{v}_1, \mathbf{x} \rangle = 0$ since $\mathbf{x} \in L^\perp$, that is

$$u_1 x_1 + \cdots + u_{\ell-1} x_{\ell-1} = -u_\ell x_\ell.$$

By Bézout's identity, $\gcd(u_1, \dots, u_{\ell-1}) = g_{\ell-1}$ divides $u_\ell x_\ell$. Furthermore, note that $g_\ell = \gcd(g_{\ell-1}, u_\ell)$ by definition. Note that, for all integers a, b, c, if a divides $b \cdot c$ then a divides $\gcd(a, b) \cdot c$. Hence, $g_{\ell-1}$ divides $g_\ell x_\ell$, and $N \coloneqq \frac{g_\ell x_\ell}{g_{\ell-1}}$ is an integer. Now consider the vector $\mathbf{x}' = (x'_1, \dots, x'_k) = \mathbf{x} - N\mathbf{v}_\ell$. Recall that $\mathbf{v}_\ell \in L' \subseteq L^\perp$ by the previous inclusion, so $\mathbf{x}' \in L^\perp$ since $L^\perp$ is a lattice. Furthermore, we claim that $k(\mathbf{x}') < k(\mathbf{x})$. Indeed, the coordinates of $\mathbf{x}$ above $\ell = k(\mathbf{x})$ are all zero, and the same is true for $\mathbf{v}_\ell$ by definition. The ℓ-th coordinate is

$$x'_\ell \;=\; x_\ell - N \cdot \tfrac{g_{\ell-1}}{g_\ell} \;=\; x_\ell - \tfrac{g_\ell x_\ell}{g_{\ell-1}} \cdot \tfrac{g_{\ell-1}}{g_\ell} \;=\; 0.$$

This shows that $k(\mathbf{x}') < \ell = k(\mathbf{x})$ so by induction $\mathbf{x}' \in L'$. Since L' is a lattice and $\mathbf{v}^{\ell-1} \in L'$, we conclude that $\mathbf{x} = \mathbf{x}' + N\mathbf{v}_\ell \in L'$.

In summary, we have shown that $L^\perp = L'$ when $L = \mathbb{Z}\mathbf{v}_1$, and that $\dim(L^\perp) = d-1$. It follows immediately by the orthogonality of the vectors that $\dim(L + L^\perp) = \dim(L) + \dim(L^\perp) = d$. Finally, computing $\mathbf{v}_2, \dots, \mathbf{v}_k$ can be done in polynomial time. Hence, the lemma is proven when $k = 1$.

For the case $k \geq 2$, observe that given $L = \operatorname{span}_{\mathbb{Z}}\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$, we have $L^\perp = \bigcap_{i=1}^k (\mathbb{Z}\mathbf{v}_i)^\perp$. It is clear then that $\dim(L^\perp) \geqslant d-k$ since every $(\mathbb{Z}\mathbf{v}_i)^\perp$ has dimension $d-1$. On the other hand, it is easy to see that the orthogonality yields $\dim(L + L^\perp) = \dim(L) + \dim(L^\perp)$ and therefore $\dim(L) + \dim(L^\perp) \leqslant d$, so we conclude that $\dim(L^\perp) = d-k$ and $\dim(L + L^\perp) = \dim(L) + \dim(L^\perp) = d$. To compute a basis for $L^\perp$, we first compute a basis for each $(\mathbb{Z}\mathbf{v}_i)^\perp$ in polynomial time by relying on the argument used above for the case $k = 1$. Afterwards, the intersection of the d resulting lattices can be computed in polynomial time by slightly extending the arguments used in Lemma 8.3 to compute $\rho_{\mathrm{SL}}(X) \wedge \rho_{\mathrm{SL}}(Y)$. Given, for every $i \in [1, k]$, the matrix $A_i \in \mathbb{Z}^{d \times (d-1)}$ whose columns form a basis of $(\mathbb{Z}\mathbf{v}_i)^\perp$, we have

$$\begin{aligned}
L^\perp &= \left\{ A_1 \cdot \mathbf{y}_1 : \mathbf{y}_1 \in \mathbb{Z}^{d-1} \text{ and } A_1 \cdot \mathbf{y}_1 = A_2 \cdot \mathbf{y}_2 = \cdots = A_d \cdot \mathbf{y}_d \right. \\
&\qquad\qquad \left. \text{for some } \mathbf{y}_2, \dots, \mathbf{y}_d \in \mathbb{Z}^{d-1} \right\} \\
&= A \cdot \pi(Z),
\end{aligned}$$

where $\pi\colon (\mathbb{Z}^{(d-1)})^d \to \mathbb{Z}^{d-1}$ is here the projection into the first vector of dimension $d-1$, and $Z \coloneqq \{(\mathbf{y}_1, \dots, y_d) \in (\mathbb{Z}^{(d-1)})^d : A_1 \cdot \mathbf{y}_1 = A_2 \cdot \mathbf{y}_2 = \cdots = A_d \cdot \mathbf{y}_d\}$. Then, the computation of a basis for $L^\perp$ proceeds as in Lemma 8.3, by appealing to Proposition 8.1. □

Below, given an assertion Φ, we write $\mathbb{1}[\Phi]$ for the indicator function defined as $\mathbb{1}[\Phi] = 1$ if Φ is true and $\mathbb{1}[\Phi] = 0$ otherwise.

Lemma 8.6. *Let $L = \mathrm{span}_{\mathbb{Z}}\{\mathbf{v}_1, \dots, \mathbf{v}_k\} \subseteq \mathbb{Z}^d$ and $s \geq 1$ such that $s \cdot \mathbb{Z}^d \subseteq L$. For every $\mathbf{v}_0 \in \mathbb{Z}^d$, $|(\mathbf{v}_0 + L) \cap [0, s)^d| = \frac{s^d}{\det(L)}$.*

Proof. Note that the hypotheses of the lemma imply $s \in \mathbb{N}$ and that L is fully-dimensional. Considering $\mathbf{v}_0 = \mathbf{0}$ suffices to show the lemma. Indeed, given $\mathbf{w} \in \mathbb{Z}^d$, we have

$$\begin{aligned}
|(\mathbf{w} + L) \cap [0, s)^d| &= \sum_{\mathbf{y} \in [0,s)^d} \mathbb{1}[\mathbf{y} \in (\mathbf{w} + L)] \\
&= \sum_{\mathbf{y} \in [0,s)^d} \mathbb{1}[\mathbf{y} - \mathbf{w} \in L] \\
&= \sum_{\mathbf{y} \in [0,s)^d} \mathbb{1}[(\mathbf{y} - \mathbf{w} \bmod s) \in L] && \text{since } s \cdot \mathbb{Z}^d \subseteq L \\
&= \sum_{\mathbf{y} \in [0,s)^d} \mathbb{1}[\mathbf{y} \in L] && \begin{array}{r}\text{since } f\colon [0,s)^d \to [0,s)^d \\ \text{defined as } f(\mathbf{y}) \coloneqq \mathbf{y} - \mathbf{x} \bmod s \\ \text{is a bijection}\end{array} \\
&= |L \cap [0, s)^d|.
\end{aligned}$$

Let us show that $|L \cap [0, s)^d| = \frac{s^d}{\det(L)}$. Let H be the Hermite normal form of the matrix $A \coloneqq [\mathbf{v}_1 \mid \cdots \mid \mathbf{v}_k]$, and U be the unimodular matrix such that $H = A \cdot U$. Since L is fully-dimensional, H is invertible and (from the fact that H is triangular and has positive pivots), we conclude that

$$H = \begin{bmatrix} p_1 & 0 & 0 & \cdots & 0 \\ a_{2,1} & p_2 & 0 & \cdots & 0 \\ a_{3,1} & a_{3,2} & p_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{d,1} & a_{d,2} & a_{d,3} & \cdots & p_d \end{bmatrix}.$$

Here, p_i is strictly positive, for every $i \in [1, d]$. We claim that p_i divides s. Indeed, since $s \cdot \mathbb{Z}^d \subseteq L$, we have $s \cdot \mathbf{e}_i \in L$, where $\mathbf{e}_i$ is the ith unit vector of the canonical basis of $\mathbb{Z}^d$. Let $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{Z}^d$ such that $H \cdot \mathbf{x} = s \cdot e_i$. Because of the shape of H, it must be the case that $x_j = 0$ for every $j \in [1, i-1]$. This implies $p_i \cdot x_i = s$, i.e., p_i divides s. Since U is unimodular, the function $f(\mathbf{u}) \coloneqq U \cdot \mathbf{u}$ is a bijection on $\mathbb{Z}^d$, that is, $U \cdot \mathbb{Z}^d = \mathbb{Z}^d$. We get:

$$\begin{aligned}
|L \cap [0, s)^d| &= |\{\mathbf{y} \in [0, s)^d : \exists \mathbf{x} \in \mathbb{Z}^d, \mathbf{y} = A \cdot \mathbf{x}\}| \\
&= |\{\mathbf{y} \in [0, s)^d : \exists \mathbf{x} \in \mathbb{Z}^d, \mathbf{y} = H \cdot \mathbf{x}\}|.
\end{aligned}$$

Let us denote with $(H \cdot \mathbf{x})_i$ the ith entry of $H \cdot \mathbf{x}$. We have,

$$0 \leq (H \cdot \mathbf{x})_i < s \;\Leftrightarrow\; 0 \leq \underbrace{\sum_{j=1}^{i-1} a_{i,j} \cdot x_j + p_i \cdot x_i}_{\alpha_i} < s \;\Leftrightarrow\; -\frac{\alpha_i}{p_i} \leq x_i < \frac{s}{p_i} - \frac{\alpha_i}{p_i}.$$

Note that $\frac{s}{p_i}$ is a positive integer, since $p_i, s \geq 1$ and p_i divides s. Therefore, the rightmost inequality above is of the form $\beta \leq x_i < N + \beta$ for some positive integer N and rational β. This system of inequalities has always N integer solutions for x_i, independently of the value of β. It therefore follows that the cardinality of the set $\{\mathbf{y} \in [0,s)^d : \exists \mathbf{x} \in \mathbb{Z}^d, \mathbf{y} = H \cdot \mathbf{x}\}$ is

$$\frac{s^d}{p_1 \cdot \ldots \cdot p_k} = \frac{s^d}{|\det(H)|} = \frac{s^d}{\det(L)}. \qquad \square$$

The following two lemmas are related to the problem of testing inclusion of union of lattices, but will also play a role when considering the universal projection in Requirement 4.8.

Lemma 8.7. *Consider ℓ lattices $L_0, \ldots, L_\ell \subseteq \mathbb{Z}^d$ and vectors $\mathbf{v}_0, \ldots, \mathbf{v}_\ell \in \mathbb{Z}^d$. Suppose $s \cdot \mathbb{Z}^d \subseteq L_0, \ldots, L_\ell$ for some $s \geq 1$. Then, $\mathbf{v}_0 + L_0 = \bigcup_{i=1}^{\ell}(\mathbf{v}_i + L_i)$ if and only if $(\mathbf{v}_0 + L_0) \cap [0,s)^d = \bigcup_{i=1}^{\ell}(\mathbf{v}_i + L_i) \cap [0,s)^d$.*

Proof. The left to right direction is immediate. For the other direction, suppose $(\mathbf{v}_0 + L_0) \cap [0,s)^d = \bigcup_{i=1}^{\ell}(\mathbf{v}_i + L_i) \cap [0,s)^d$. We prove the two inclusions.

($\subseteq$): We consider $\mathbf{v} \in L_0$ and show $\mathbf{v}_0 + \mathbf{v} \in \bigcup_{i=1}^{\ell}(\mathbf{v}_i + L_i)$. Since $s \cdot \mathbb{Z}^d \subseteq L_0$, there is $\mathbf{u} \in \mathbb{Z}^d$ such that $\mathbf{v}_0 + \mathbf{v} + s \cdot \mathbf{u} \in (\mathbf{v}_0 + L_0) \cap [0,s)^d$. Then, by hypothesis, $\mathbf{v}_0 + \mathbf{v} + s \cdot \mathbf{u} \in \bigcup_{i=1}^{\ell}(\mathbf{v}_i + L_i) \cap [0,s)^d$. Since $s \cdot \mathbb{Z}^d \subseteq L_i$ for every $i \in [1,\ell]$, we conclude that $\mathbf{v}_0 + \mathbf{v} \in \bigcup_{i=1}^{\ell}(\mathbf{v}_i + L_i) \cap [0,s)^d$.

($\supseteq$): This inclusion is similar to the previous one. Consider $i \in [1,\ell]$ and $\mathbf{v} \in L_i$. We show that $\mathbf{v}_i + \mathbf{v} \in \mathbf{v}_0 + L_0$. Since $s \cdot \mathbb{Z}^d \subseteq L_i$, there is $\mathbf{u} \in \mathbb{Z}^d$ such that $\mathbf{v}_i + \mathbf{v} + s \cdot \mathbf{u} \in (v_i + L_i) \cap [0,s)^d$. Then, $\mathbf{v}_i + \mathbf{v} + s \cdot \mathbf{u} \in (v_0 + L_0) \cap [0,s)^d$, and since $s \cdot \mathbb{Z}^d \subseteq L_0$ we conclude that $\mathbf{v}_i + \mathbf{v} \in \mathbf{v}_0 + L_0$. $\square$

Lemma 8.8. *Consider ℓ lattices $L_0, \ldots, L_\ell \subseteq \mathbb{Z}^d$ and vectors $\mathbf{v}_0, \ldots, \mathbf{v}_\ell \in \mathbb{Z}^d$, with L_0 fully-dimensional. Then,*

$$\mathbf{v}_0 + L_0 \subseteq \bigcup_{i=1}^{\ell} \mathbf{v}_i + L_i \;\Leftrightarrow\; \mathbf{v}_0 + L_0 \subseteq \bigcup_{i \in I} \mathbf{v}_i + L_i,$$

where $I \coloneqq \{i \in [1,\ell] : L_i$ is fully-dimensional$\}$.

Proof. The lemma is trivial for $d = 0$, hence assume $d \geq 1$. The right to left direction is straightforward. For the left to right direction, *ad absurdum* suppose $\mathbf{v}_0 + L_0 \subseteq \bigcup_{i=1}^{\ell} \mathbf{v}_i + L_i$ but $\mathbf{v}_0 + L_0 \not\subseteq \bigcup_{i \in I} \mathbf{v}_i + L_i$. Since, for every $i \in I$, L_i is fully-dimensional, by Proposition 8.4 we can find $s_i \in \mathbb{N}$ such that $s_i \cdot \mathbb{Z}^d \subseteq L_i$. Similarly, we can find $s_0 \in \mathbb{N}$ such that $s_0 \cdot \mathbb{Z}^d \subseteq L_0$. Let $s \coloneqq \operatorname{lcm}\{s_i : i \in I \cup \{0\}\}$, so that $s \cdot \mathbb{Z}^d \subseteq L_i$ for every $i \in I \cup \{0\}$. By properties of lattices, this means that for every $i \in I \cup \{0\}$ there is a (finite) set $U_i \subseteq [0,s)^d$ such that $\mathbf{v}_i + L_i = U_i + s \cdot \mathbb{Z}^d$. Now, on the one hand,

$$\mathbf{v}_0 + L_0 \subseteq \bigcup_{i=1}^{\ell} \mathbf{v}_i + L_i = (U + s \cdot \mathbb{Z}^d) \cup \bigcup_{j \in [1,\ell] \setminus I} (\mathbf{v}_j + L_j),$$

where $U \coloneqq \bigcup_{i\in I} U_i$. On the other hand, we have

$$\mathbf{v}_0 + L_0 \not\subseteq \bigcup_{i\in I} \mathbf{v}_i + L_i = U + s\cdot\mathbb{Z}^d.$$

Recall that $\mathbf{v}_0 + L_0 = U_0 + s\cdot\mathbb{Z}^d$, so we must have $U_0 \not\subseteq U$. Since L_0 is fully-dimensional, $U_0 \neq \emptyset$. Then, take $\mathbf{z} \in U_0 \setminus U$. Note $U_0, U \subseteq [0,s)^d$ and $\mathbf{z} \in [0,s)^d$, and therefore $\mathbf{z} + s\cdot\mathbb{Z}^d \subseteq (U_0 + s\cdot\mathbb{Z}^d) \setminus (U + s\cdot\mathbb{Z}^d)$. This means $\mathbf{z} + s\cdot\mathbb{Z}^d \subseteq \bigcup_{j\in[1,\ell]\setminus I}(\mathbf{v}_j + L_j)$. We claim that this is not possible by a dimension argument. To see that, fix an integer M. The last inclusion implies that

$$|(\mathbf{z} + s\cdot\mathbb{Z}^d) \cap [0, M\cdot s)^d| \leqslant \sum\nolimits_{j\in[1,\ell]\setminus I} |(\mathbf{v}_j + L_j) \cap [0, M\cdot s)^d|.$$

However, observe that $|(\mathbf{v} + s\cdot\mathbb{Z}^d) \cap [0, M\cdot s)^d| = (M-1)^d \in \Omega(M^d)$ whereas, since $\dim(L_i) \leqslant d-1$ for every $i \in [1,\ell]\setminus I$, $|(\mathbf{v}_j + L_j) \cap [0, M\cdot s)^d| = O_{M\to\infty}(M^{k-1})$. As $M \to \infty$, we can see that the left hand-side of the above equation grows much faster than the right-hand side, and so we have reached a contradiction. □

8.4. **Requirement 4.3, Item (F2): $(\mathrm{un}(\mathrm{D}), \leq)$ has a $(\mathrm{un}(\rho_{\mathrm{SL}}), \mathrm{len}(\mathbf{1}))$-UXP signature.** We are ready to present an algorithm to solve inclusion between union of shifted lattices that runs in polynomial time when the length of the union is considered fixed (as it is the case when taking into account the parameter $\mathrm{len}(\mathbf{1})$).

Lemma 8.9. *The structure $(\mathrm{un}(\mathrm{D}), \leq)$ has a $(\mathrm{un}(\rho_{\mathrm{SL}}), \mathrm{len}(\mathbf{1}))$-UXP signature.*

Proof. As in Lemma 8.3, without loss of generality we assume all shifted lattices to be non-empty and have the same dimension. We describe an algorithm that given $X \coloneqq (X_1, \ldots, X_n)$ and $Y = (Y_1, \ldots, Y_m)$ in $\mathrm{dom}(\mathrm{un}(\rho_{\mathrm{SL}}))$, with $X_i = (d, \mathbf{v}_{i,0}, \ldots, \mathbf{v}_{i,k_i})$ and $Y_j = (d, \mathbf{w}_{j,0}, \ldots, \mathbf{w}_{j,\ell_j})$, checks if $\mathrm{un}(\rho)(X) \leq \mathrm{un}(\rho)(Y)$. The algorithm runs in time $2^m \cdot \mathrm{poly}(|X|, |Y|)$. The inclusion is checked by verifying that, for every $i \in [1,n]$, $\rho_{\mathrm{SL}}(X_i) \leq \mathrm{un}(\rho)(Y)$.

For every $i \in [1,n]$, the algorithm first computes the following objects:

(1) Using Lemma 8.3, for all $j \in [1,m]$, we compute $\mathbf{v}'_j \in \mathbb{Z}^d$ and a basis for the lattice $L'_j \subseteq \mathbb{Z}^d$ such that $\rho_{\mathrm{SL}}(Y_j \wedge X_i) = \mathbf{v}'_j + L'_j$. This step only requires polynomial time.
(2) A basis for the lattice $L^\perp$ orthogonal to $L \coloneqq \mathrm{span}_{\mathbb{Z}}(\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,k_i})$, using Lemma 8.5. This step only requires polynomial time.
(3) The set $I \coloneqq \{j \in [1,m] : L^\perp + \rho_{\mathrm{SL}}(Y_j \wedge X_i) \text{ is fully-dimensional}\}$. To check whether $L^\perp + \rho_{\mathrm{SL}}(Y_j \wedge X_i)$ is fully-dimensional, it suffices to check that $Y_j \wedge X_i$ and X_i have the same dimension, i.e., the same number of periods. This can be done in polynomial time.
(4) The positive integer $s \coloneqq \det(L + L^\perp) \cdot \prod_{j\in I} \det(L'_j + L^\perp)$. Since we have bases for all these lattices, computing the determinant only requires polynomial time (e.g., bring the matrix in Hermite normal form and multiply all the elements in the diagonal).

By Proposition 8.4, we have $s\cdot\mathbb{Z}^d \subseteq L + L^\perp$ and $s\cdot\mathbb{Z}^d \subseteq L'_j + L^\perp$ for every $j \in I$. At this stage, observe the following equivalences that stem from the previous lemmas on lattices:

$$\begin{aligned} &\rho_{\mathrm{SL}}(X_i) \leq \mathrm{un}(\rho)(Y) \\ \Leftrightarrow\ &\rho_{\mathrm{SL}}(X_i) = \bigcup_{j=1}^{m} (\rho_{\mathrm{SL}}(Y_j \wedge X_j)) \end{aligned}$$

$$\begin{aligned}
&\Leftrightarrow Z = \bigcup_{j=1}^{m} W_j && Z \coloneqq \mathbf{v}_{i,0} + L + L^{\perp} \text{ and}\\
&&& W_j \coloneqq \mathbf{v}'_j + L'_j + L^{\perp}\\
&\Leftrightarrow Z = \bigcup_{j\in I} W_j && \text{Lemma 8.8 and the fact}\\
&&& \text{that } W_j \subseteq Z \text{ for all } j \in [1,m]\\
&\Leftrightarrow Z \cap [0,s)^d = \bigcup_{j\in I} W_j \cap [0,s)^d && \text{Lemma 8.7}\\
&\Leftrightarrow |Z \cap [0,s)^d| = |\bigcup_{j\in I} W_j \cap [0,s)^d| && \Leftarrow \text{ follows from the fact}\\
&&& \text{that } W_j \subseteq Z \text{ for all } j \in I\\
&\Leftrightarrow |Z \cap [0,s)^d| = \sum_{J\subseteq I} (-1)^{|J|+1} |\bigcap_{j\in J} W_j \cap [0,s)^d| && \text{inclusion-exclusion.}
\end{aligned}$$

The algorithm then computes, for every $J \subseteq I$ a shifted lattice V_J from $\mathrm{dom}(\rho_{\mathrm{SL}})$ representing $\bigcap_{j\in J} W_j$. If $V_J \neq \varnothing$, below let $V_J \coloneqq (d, \mathbf{u}_{J,0}, \dots, \mathbf{u}_{J,r_J})$. As described in the last part of the proof of Lemma 8.5, computing such a representation can be done in polynomial time. Since, by Proposition 8.4, $s \cdot \mathbb{Z}^d \subseteq L'_j + L^{\perp}$ for every $j \in I$, we have $s \cdot \mathbb{Z}^d \subseteq \mathrm{span}_{\mathbb{Z}}\{\mathbf{u}_{J,1}, \dots, \mathbf{u}_{J,r_J}\}$ for every $J \subseteq I$ with $V_J \neq \varnothing$. Note now the following:

$$\begin{aligned}
&|Z \cap [0,s)^d| = \sum_{J\subseteq I} (-1)^{|J|+1} |\bigcap_{j\in J} W_j \cap [0,s)^d|\\
\Leftrightarrow\ & \frac{s^d}{\det(L+L^{\perp})} = \sum_{J\subseteq I} (-1)^{|J|+1} \frac{s^d \cdot \mathbb{1}[V_J \neq \varnothing]}{\det(\mathrm{span}_{\mathbb{Z}}\{\mathbf{u}_{J,1}, \dots, \mathbf{u}_{J,r_J}\})} && \text{Lemma 8.6}\\
\Leftrightarrow\ & 1 = \sum_{J\subseteq I} (-1)^{|J|+1} \frac{\det(L+L^{\perp}) \cdot \mathbb{1}[V_J \neq \varnothing]}{|\det([\mathbf{u}_{J,1} \mid \cdots \mid \mathbf{u}_{J,r_J}])|}.
\end{aligned}$$

Hence, the algorithm computes $\sum_{J\subseteq I} (-1)^{|J|+1} \frac{\det(L+L^{\perp}) \cdot \mathbb{1}[V_J \neq \varnothing]}{|\det([\mathbf{u}_{J,0}|\cdots|\mathbf{u}_{J,r_J}])|}$ and returns true if and only if it equals 1. Each determinant computation requires polynomial time, but $O(2^{|I|})$ such computations are required. Overall, we observe that the exponential blow-up in the algorithm is limited to the iterations of all subsets J of I. All other operations are in polynomial time, resulting in a $2^m \cdot \mathrm{poly}(|X|, |Y|)$ running time. □

8.5. **Requirement 4.8: both projections are in** $(\mathrm{dfnf}(\rho), \mathrm{dep}(\theta))$**-UXP.** Establishing the Items (F3) and (F5) of Requirement 4.8 is trivial: thanks to our choice of representation based on ρ_{SL}, given $X \in \mathrm{dom}(\rho_{\mathrm{SL}})$ and $\mathbf{i} \in \mathbf{I}$, $\dot{\pi}(\mathbf{i}, X)$ can be computed by simply crossing out the entries of all vectors of X corresponding to the indices in $\mathbf{i}$, bringing the resulting matrix of periods in Hermite normal form and removing all its zero columns (to force the periods to be linearly independent). The following result is thus immediate.

Lemma 8.10. *The structure* $(\mathrm{D}, (\dot{\pi}, \mathbf{I}))$ *has a* $(\rho_{\mathrm{SL}}, \mathbf{1})$*-UXP signature.*

On the contrary, computing the universal projections $\dot{\pi}^{\forall}_Z(\mathbf{i}, X)$, as required by the Items (F4) and (F6), is computationally expensive. For simplicity, below we index entries in vectors starting from one, and instead of considering projections over arbitrary vectors of indices $\mathbf{i}$, we assume $\mathbf{i} = [1,k]$ for some $k \in \mathbb{N}$ so that $\pi^{\forall}_Z(\mathbf{i}, X)$ projects over the first k dimensions. This is w.l.o.g., as we can reorder components appropriately. So,

let $\pi^{\forall}_Z(k.X) \coloneqq \pi^{\forall}_Z([1,k],X)$, and $\pi(k,X) \coloneqq \pi([1,k],X)$. For a set $S \subseteq \mathbb{Z}^d$ and $\mathbf{x} \in \mathbb{R}^k$, with $k \leq d$ we define the *slice* of S at $\mathbf{x}$, denoted by $S\restriction_{\mathbf{x}}$ as the set

$$S\restriction_{\mathbf{x}} \coloneqq \{\mathbf{t} \in \mathbb{Z}^{d-k} : (\mathbf{x},\mathbf{t}) \in S\}.$$

Before giving the algorithm for universal projection, we need the following result.

Lemma 8.11. *Let $L \subseteq \mathbb{Z}^d$ be a lattice and $\mathbf{v}_0 \in \mathbb{Z}^d$. There is a lattice $L' \subseteq \mathbb{Z}^d$ such that for all $\mathbf{x} \in \pi(k, \mathbf{v}_0 + L)$ there is $\mathbf{t}_\mathbf{x} \in \mathbb{Z}^k$ such that $(\mathbf{v}_0 + L)\restriction_{\mathbf{x}} = \mathbf{t}_\mathbf{x} + L'$. Moreover, there is an algorithm that given $X = (d, \mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n) \in \mathrm{dom}(\rho_{\mathrm{SL}})$ and $k \in \mathbb{N}$ written in unary, returns a basis of L', with respect to $L \coloneqq \rho_{\mathrm{SL}}(X)$. The algorithm runs in polynomial time.*

Proof. Let $L \coloneqq \mathrm{span}_{\mathbb{Z}}(\mathbf{v}_1, \ldots, \mathbf{v}_n)$, with $\mathbf{v}_1, \ldots, \mathbf{v}_n$ being linearly independent. Note that $\pi(k, \mathbf{v}_0 + L)$ cannot be empty. Let $\mathbf{x}_0$ be the vector in $\mathbb{Z}^{d-k}$ that is obtained from $\mathbf{v}_0$ by removing the first k components, so that $\mathbf{x}_0 \in \pi(k, \mathbf{v}_0 + L)$. The non-empty set $(\mathbf{v}_0 + L)\restriction_{\mathbf{x}_0}$ corresponds to the set of solutions $\mathbf{x} \in \mathbb{Z}^{d-k}$ to the following weak PA formula $\Phi(\mathbf{x})$

$$\exists y_1, \ldots, y_n : \begin{bmatrix}\mathbf{x}_0 \\ \mathbf{x}\end{bmatrix} = \mathbf{v}_0 + \mathbf{v}_1 \cdot y_1 + \ldots \mathbf{v}_n \cdot y_n.$$

By Lemma 8.2 and Proposition 8.1, we can compute in polynomial time with respect to $\mathbf{v}_0, \ldots, \mathbf{v}_k$ a family of vectors $\mathbf{w}_0, \ldots, \mathbf{w}_j$ such that $\mathbf{w}_1, \ldots, \mathbf{w}_j$ are linearly independent and $(\mathbf{v}_0 + L)\restriction_{\mathbf{x}_0} = [\![\Phi(\mathbf{x})]\!]_{\mathcal{Z}} = \mathbf{w}_0 + \mathrm{span}_{\mathbb{Z}}(\mathbf{w}_1, \ldots, \mathbf{w}_j)$. Then, L' in the statement of the lemma is given by $\mathrm{span}_{\mathbb{Z}}(\mathbf{w}_1, \ldots, \mathbf{w}_j)$.

To conclude the proof, we must check that for every $\mathbf{x} \in \pi(k, \mathbf{v}_0 + L)$ there is $\mathbf{t}_\mathbf{x} \in \mathbb{Z}^k$ such that $(\mathbf{v}_0 + L)\restriction_{\mathbf{x}} = \mathbf{t}_\mathbf{x} + L'$. To this end, consider $\mathbf{x} \in \pi(k, \mathbf{v}_0 + L)$ and pick as $\mathbf{t}_\mathbf{x} \in \mathbb{Z}^k$ the only vector in $\{\mathbf{v}_0\}\restriction_{\mathbf{x}}$. Given $\mathbf{s} \in \mathbb{Z}^k$, we have

$$\begin{aligned}
&\mathbf{s} \in (\mathbf{v}_0 + L)\restriction_{\mathbf{x}} \\
\Leftrightarrow\ &(\mathbf{x},\mathbf{s}) \in \mathbf{v}_0 + L \\
\Leftrightarrow\ &(\mathbf{x},\mathbf{s}) - (\mathbf{x},\mathbf{t}_\mathbf{x}) \in L && \text{since } (\mathbf{x},\mathbf{t}_\mathbf{x}) = \mathbf{v}_0 \text{ and } L \text{ is a lattice} \\
\Leftrightarrow\ &(\mathbf{x}_0,\mathbf{w}_0) + (\mathbf{0},\mathbf{t}_\mathbf{x} - \mathbf{s}) \in \mathbf{v}_0 + L && \text{as } (\mathbf{x}_0,\mathbf{w}_0) \in \mathbf{v}_0 + L \text{ and } L \text{ is a lattice} \\
\Leftrightarrow\ &(\mathbf{x}_0,\mathbf{t}_\mathbf{x} - \mathbf{s} + \mathbf{w}_0) \in \mathbf{v}_0 + L \\
\Leftrightarrow\ &\mathbf{t}_\mathbf{x} - \mathbf{s} + \mathbf{w}_0 \in (\mathbf{v}_0 + L)\restriction_{\mathbf{x}_0} \\
\Leftrightarrow\ &\mathbf{t}_\mathbf{x} - \mathbf{s} \in L' && \text{by def. of } \mathbf{w}_0 \\
\Leftrightarrow\ &\mathbf{s} \in \mathbf{t}_\mathbf{x} + L' && \text{as } L' \text{ is a lattice.}
\end{aligned}$$

□

Back to the problem of performing universal projection, intuitively, we need to count points in unions of shifted lattices (similarly to inclusion testing) but in a parametric way. This means that given a union of shifted lattices $X = \bigcup_{i=1}^m (\mathbf{v}_i + L_i)$ every intersection $\bigcap_{j \in J}(\mathbf{v}_i + L_i)$ with $J \subseteq L$ in the inclusion-exclusion formula may or may not need to be accounted for, depending on the value of a parameter $f\colon 2^{[1,m]} \to \{0,1\}$ belonging of a certain set of parameters $\mathcal{F}$ (see the lemma below, the exact definition of $\mathcal{F}$ is technical and only given in the proof; $2^{[1,m]}$ stands for the powerset of $[1,m]$). The algorithm therefore considers all possible ways in which intersections may or may not be taken, which is roughly 2^{2^m}; i.e., the number of functions in $[2^{[1,m]} \to \{0,1\}]$. Our algorithm allows us to conclude a rather surprising fact: the relative universal projection can be expressed as a complex combination of unions, intersections, projections and the relative complementations that are

exclusively applied to the initial sets in input. The number of these operations only depends on m, resulting in an algorithm that runs in polynomial time when m is fixed.

Lemma 8.12. *Let $Z, X_1, \dots, X_m \subseteq \mathbb{Z}^d$ be shifted lattices, $X = \bigcup_{i=1}^m X_i$ and $k \in \mathbb{N}$. There are $I \subseteq [1,m]$ and $\mathcal{F} \subseteq [2^I \to \{0,1\}]$ such that*

$$\pi^{\forall}_Z(k,X) = \bigcup_{f\in\mathcal{F}} \Big(\Big(\bigcap_{J:f(J)=1} \bigcap_{j\in J} \pi(k, X_j \cap Z) \Big) - \Big(\bigcup_{J:f(J)=0} \bigcap_{j\in J} \pi(k, X_j \cap Z) \Big) \Big).$$

Fix $m \in \mathbb{N}$. There is an algorithm that given in input $X = (X_1, \dots, X_m) \in \mathrm{dom}(\mathrm{un}(\rho_{\mathrm{SL}}))$, $Z \in \mathrm{dom}(\rho_{\mathrm{SL}})$, and $k \in \mathbb{N}$ in unary, returns $Y \in \mathrm{dom}(\mathrm{dfnf}(\rho_{\mathrm{SL}}))$ such that $\mathrm{dfnf}(\rho_{\mathrm{SL}})(Y) = \pi^{\forall}_{\rho_{\mathrm{SL}}(Z)}(k, \mathrm{un}(\rho_{\mathrm{SL}})(X))$. The algorithm runs in polynomial time.

Proof. Let us focus on the first part of the lemma. We first show the result under the following additional hypothesis:

$$X \subseteq Z \text{ and for every } \mathbf{x} \in \pi(k,Z),\ Z\restriction_{\mathbf{x}} \subseteq \mathbb{Z}^k \text{ is fully-dimensional.} \qquad (\dagger)$$

Note that, in the statement of the lemma, the intersections $X_j \cap Z$ can then be replaced by X_j. We later show how to discharge this additional hypothesis.

Starting from representations of X and Z in $\mathrm{dom}(\mathrm{un}(\rho_{\mathrm{SL}}))$ and $\mathrm{dom}(\rho_{\mathrm{SL}})$ respectively, we apply Lemma 8.11 to compute bases for the lattices $L'_i \subseteq \mathbb{Z}^k$ ($i \in [0,m]$) such that for every $i \in [1,m]$ and $\mathbf{x} \in \pi(k, X_i)$ there is $\mathbf{t}_{\mathbf{x}} \in \mathbb{Z}^k$ such that $X_i\restriction_{\mathbf{x}} = \mathbf{t}_{\mathbf{x}} + L'_i$, and for every $\mathbf{y} \in \pi(k,Z)$ there is $\mathbf{t}_{\mathbf{y}} \in \mathbb{Z}^k$ such that $Z\restriction_{\mathbf{y}} = \mathbf{t}_{\mathbf{y}} + L'_0$. Note that the assumption that $Z\restriction_{\mathbf{x}}$ is fully-dimensional implies that L'_0 is fully-dimensional. Let $I \coloneqq \{i \in [1,m] : L'_i \text{ is fully-dimensional}\}$. Then,

$$\begin{aligned}
\pi^{\forall}_Z(k,X) &= \{\mathbf{x} \in \pi(k,Z) : \forall \mathbf{t} \in \mathbb{Z}^k, (\mathbf{x},\mathbf{t}) \in Z \text{ implies } (\mathbf{x},\mathbf{t}) \in \textstyle\bigcup_{i=1}^m X_i\} \\
&= \{\mathbf{x} \in \pi(k,Z) : Z\restriction_{\mathbf{x}} \subseteq \textstyle\bigcup_{i=1}^m X_i\restriction_{\mathbf{x}}\} \\
&= \{\mathbf{x} \in \pi(k,Z) : Z\restriction_{\mathbf{x}} \subseteq \textstyle\bigcup_{i\in I} X_i\restriction_{\mathbf{x}}\} && \text{by Lemma 8.8} \\
&= \{\mathbf{x} \in \pi(k,Z) : Z\restriction_{\mathbf{x}} = \textstyle\bigcup_{i\in I} X_i\restriction_{\mathbf{x}}\} && \text{since } X \subseteq Z
\end{aligned}$$

Let $s \coloneqq \prod_{i\in I\cup\{0\}} |\det(L'_i)|$ (since we have a basis for each L'_i, computing s requires polynomial time). By Proposition 8.4, $s \cdot \mathbb{Z}^{d-k} \subseteq L'_i$ for every $i \in I \cup \{0\}$. We now replay the proof of Lemma 8.9 to derive that, for every $\mathbf{x} \in \mathbb{Z}^{d-k}$,

$$Z\restriction_{\mathbf{x}} = \textstyle\bigcup_{i\in I} X_i\restriction_{\mathbf{x}} \iff |Z\restriction_{\mathbf{x}} \cap [0,s)^k| = \sum_{J\subseteq I} (-1)^{|J|+1} |\bigcap_{j\in J} X\restriction_{\mathbf{x}} \cap [0,s)^k|.$$

Consider $J \subseteq I$. If $\bigcap_{j\in J} X\restriction_{\mathbf{x}} \neq \emptyset$, i.e. when $\mathbf{x} \in \bigcap_{j\in J} \pi(k, X_j)$, by $s \cdot \mathbb{Z}^{d-k} \subseteq L'_i$ we conclude that $|\bigcap_{j\in J} X\restriction_{\mathbf{x}} \cap [0,s)^k| = |\bigcap_{j\in J} L'_i \cap [0,s)^k|$. Similarly, $|Z\restriction_{\mathbf{x}} \cap [0,s)^k| = |L'_0 \cap [0,s)^k|$. Define

$$N_0 \coloneqq |L'_0 \cap [0,s)^k|, \quad N_J \coloneqq |\textstyle\bigcap_{i\in J} L'_i \cap [0,d)^k|, \quad T_J(\mathbf{x}) \coloneqq \mathbb{1}\Big[\mathbf{x} \in \textstyle\bigcap_{j\in J} \pi(k, X_j)\Big].$$

Hence, $Z\restriction_{\mathbf{x}} = \bigcup_{i\in I} X_i\restriction_{\mathbf{x}}$ holds if and only if $N(\mathbf{x}) = 1$, where

$$N(\mathbf{x}) \coloneqq \sum_{J\subseteq I} \Big(T_J(\mathbf{x}) \cdot \frac{(-1)^{|J|+1} \cdot N_J}{N_0} \Big).$$

Observe that the quantity $N(\mathbf{x})$ ultimately depends on the values of $T_J(\mathbf{x})$, which are in $\{0,1\}$. More precisely, given $f\colon 2^I \to \{0,1\}$, define

$$N(f) \coloneqq \sum_{J\subseteq I} \Big(f(J) \cdot \frac{(-1)^{|J|+1} \cdot N_J}{N_0} \Big).$$

Then, for all $\mathbf{x} \in \mathbb{Z}^{d-k}$ and $J \subseteq I$, given $f_{\mathbf{x}}(J) \coloneqq T_J(\mathbf{x})$ we have $N(\mathbf{x}) = N(f_{\mathbf{x}})$. Let $\mathcal{F} \coloneqq \{f \colon 2^I \to \{0,1\} : N(f) = 1\}$. We have $N(\mathbf{x}) = 1$ if and only if $f_{\mathbf{x}} \in \mathcal{F}$; and so to summarise $\pi^{\forall}_Z(k, X) = \bigcup_{f \in \mathcal{F}} \{\mathbf{x} \in \pi(k, Z) : f_{\mathbf{x}} = f\}$.

Given $f \in \mathcal{F}$ and $\mathbf{x} \in \pi(k, Z)$ we have

$$
\begin{aligned}
& f_{\mathbf{x}} = f \\
\iff & \text{for all } J \subseteq I, T_J(\mathbf{x}) = f(J) && \text{by def. of } f_{\mathbf{x}} \\
\iff & \text{for all } J \subseteq I, \mathbb{1}\Big[\mathbf{x} \in \textstyle\bigcap_{j \in J} \pi(k, X_j)\Big] = f(J) && \text{by def. of } T_J(\mathbf{x}) \\
\iff & \text{for all } J \in \{J \subseteq I : f(J) = 1\}, \mathbf{x} \in \textstyle\bigcap_{i \in J} \pi(k, X_i) \text{ and} \\
& \text{for all } J \in \{J \subseteq I : f(J) = 0\}, \mathbf{x} \notin \textstyle\bigcap_{i \in J} \pi(k, X_i) \\
\iff & \mathbf{x} \in \Big(\bigcap_{J : f(J) = 1} \bigcap_{j \in J} \pi(k, X_j) \Big) - \Big(\bigcup_{J : f(J) = 0} \bigcap_{j \in J} \pi(k, X_j) \Big).
\end{aligned}
$$

This concludes the proof of the equivalence in the statement of the lemma, subject to the additional hypothesis (†).

Before showing how to remove the hypothesis (†), we consider the second part of the lemma and study the complexity of computing the element $Y \in \mathrm{dom}(\mathrm{dfnf}(\rho_{\mathrm{SL}}))$ that represents $\pi^{\forall}_{\rho_{\mathrm{SL}}(Z)}(k, \mathrm{un}(\rho_{\mathrm{SL}}(X)))$ (again assuming (†)). First of all not that w.l.o.g. we can assume all elements of $\mathrm{dom}(\rho_{\mathrm{SL}})$ in X, and Z, to be different from $\varnothing$. Indeed, if $Z = \varnothing$ or $X = (\varnothing)$ then Y can be set as $\varnothing$, and we can remove from X every element equal to $\varnothing$ as this does not change the set $\pi^{\forall}_{\rho_{\mathrm{SL}}(Z)}(k, \mathrm{un}(\rho_{\mathrm{SL}}(X)))$. Referring to the objects in the first part of the proof, we see that computing bases for the lattices L'_i ($i \in [0, m]$) can be done in polynomial time, by Lemma 8.11. Given these bases it is trivial to compute the set I. Then, by Lemma 8.6, computing N_0 and N_J for any $J \subseteq I$ also requires polynomial time (there are however $2^{|I|}$ many such J). To compute the set $\mathcal{F}$ it suffices to list all $f \colon 2^I \to \{0,1\}$, compute $N(f)$ and check that this integer equals 1. Hence, by definition of $N(f)$, we conclude that constructing $\mathcal{F}$ takes time $2^{2^{O(m)}}\mathrm{poly}(|X|, |Z|)$. Given $\mathcal{F}$, the element Y is computed as

$$
\bigvee_{f \in \mathcal{F}} \Big(\Big(\bigwedge_{J : f(J) = 1} \bigwedge_{j \in J} \pi(k, X_j) \Big) - \Big(\bigvee_{J : f(J) = 0} \bigwedge_{j \in J} \pi(k, X_j) \Big) \Big).
$$

When m is considered fixed, this corresponds to a constant number of applications of the operators $\vee$, $\wedge$, $-$ and π to the sets $X_1, \ldots, X_m$. Then, Y can be computed in polynomial time thanks to Lemma 8.10 and Lemma 4.5; where the latter holds for weak PA as we have already established Requirement 4.3 of the framework.

What is missing is to get rid of the hypothesis (†) without incurring an exponential blow-up with respect to $\max\{|X_1|, \ldots, |X_m|, |Z|\}$. To this end, consider again shifted lattices $Z, X_1, \ldots, X_m$ (assumed for simplicity to be elements in $\mathrm{dom}(\rho_{\mathrm{SL}})$) as in the first statement of the lemma (but now without assuming (†)). We will define shifted lattices $\widetilde{X}_1, \ldots, \widetilde{X}_m, \widetilde{Z}$ such that

(A) $\widetilde{X} \coloneqq \bigcup_{j=1}^m \widetilde{X}_j \subseteq \widetilde{Z}$ and for every $\mathbf{x} \in \pi(k, \widetilde{Z})$, $\widetilde{Z}\restriction_{\mathbf{x}} \subseteq \mathbb{Z}^k$ is fully-dimensional (that is, the sets $\widetilde{X}$ and $\widetilde{Z}$ satisfy (†)),

(B) $\pi(k, \widetilde{X}_j) = \pi(k, X_j \cap Z)$ and $\pi^{\forall}_{\widetilde{Z}}(k, \widetilde{X}) = \pi^{\forall}_Z(k, X)$.

Thanks to Items (A) and (B), to compute a representation in $\mathrm{dom}(\mathrm{dfnf}(\rho_{\mathrm{SL}}))$ of the set $\pi^{\forall}_{\rho_{\mathrm{SL}}(Z)}(k, \mathrm{un}(\rho_{\mathrm{SL}}(X)))$, it suffices to apply the equivalence in the statement of the lemma on the fully-dimensional sets $\widetilde{X}_1, \dots, \widetilde{X}_m, \widetilde{Z}$.

Let us define $\widetilde{X}_1, \dots, \widetilde{X}_m$ and $\widetilde{Z}$. First, apply Lemma 8.11 to obtain a basis for a lattice $L \subseteq \mathbb{Z}^{d-k}$ such that for all $\mathbf{x} \in \pi(k, Z)$ there is $\mathbf{t}_{\mathbf{x}} \in \mathbb{Z}^k$ such that $Z\restriction_{\mathbf{x}} = \mathbf{t}_{\mathbf{x}} + L$. We compute a basis for the lattice $L^\perp$ orthogonal to L, according to Lemma 8.5. Recall that $L + L^\perp$ is fully-dimensional. We define:

$$\widetilde{Z} \coloneqq Z + \{\mathbf{0}\} \times L^\perp, \qquad \widetilde{X}_j \coloneqq (X_j \cap Z) + \{\mathbf{0}\} \times L^\perp, \qquad \widetilde{X} \coloneqq \bigcup_{j=1}^{m} \widetilde{X}_j,$$

where $\mathbf{0}$ stands here for the zero vector of $\mathbb{Z}^{d-k}$. By definition, $\widetilde{Z}$ and all $\widetilde{X}_j$ are shifted lattices, and moreover $\widetilde{X} \subseteq \widetilde{Z}$.

Observe that for every $\mathbf{x} \in \mathbb{Z}^{n-k}$ and every $A \subseteq \mathbb{Z}^d$,

$$\begin{aligned}(A + \{\mathbf{0}\} \times L^\perp)\restriction_{\mathbf{x}} &= \{\mathbf{t} \in \mathbb{Z}^k : (\mathbf{x}, \mathbf{t}) \in A + \{\mathbf{0}\} \times L^\perp\} \\ &= \{\mathbf{t} \in \mathbb{Z}^k : \mathbf{t} \in A\restriction_{\mathbf{x}} + L^\perp\} \\ &= A\restriction_{\mathbf{x}} + L^\perp.\end{aligned}$$

Hence, for every $\mathbf{x} \in \pi(k, Z)$ we have $\widetilde{Z}\restriction_{\mathbf{x}} = Z\restriction_{\mathbf{x}} + L^\perp = \mathbf{t}_{\mathbf{x}} + L + L^\perp$, and therefore $\widetilde{Z}\restriction_{\mathbf{x}}$ is fully-dimensional. This establishes Item (A).

Moving towards Item (B), we observe that for every $A \subseteq \mathbb{Z}^d$,

$$\begin{aligned}\pi(A + \{\mathbf{0}\} \times L^\perp) &= \{\mathbf{x} \in \mathbb{Z}^{d-k} : (A + \{\mathbf{0}\} \times L^\perp)\restriction_{\mathbf{x}} \neq \emptyset\} \\ &= \{\mathbf{x} \in \mathbb{Z}^{d-k} : A\restriction_{\mathbf{x}} + L^\perp \neq \emptyset\} \\ &= \{\mathbf{x} \in \mathbb{Z}^{d-k} : A\restriction_{\mathbf{x}} \neq \emptyset\} && \text{as } L^\perp \neq \emptyset \\ &= \pi(k, A).\end{aligned}$$

Therefore, $\pi(k, \widetilde{X}_j) = \pi(k, X_j \cap Z)$ for every $j \in [1, m]$. Lastly, let us show the equivalence $\pi^{\forall}_{\widetilde{Z}}(k, \widetilde{X}) = \pi^{\forall}_{Z}(k, X)$. Recall that for $\mathbf{x} \in \pi(k, Z) = \pi(k, \widetilde{Z})$, we have $Z\restriction_{\mathbf{x}} = \mathbf{t}_{\mathbf{x}} + L$. Furthermore, $X \cap Z \subseteq Z$, so $(X \cap Z)\restriction_{\mathbf{x}} \subseteq Z\restriction_{\mathbf{x}}$ and we can write $(X \cap Z)\restriction_{\mathbf{x}} = \mathbf{t}_{\mathbf{x}} + A_{\mathbf{x}}$ for some set $A_{\mathbf{x}} \subseteq L$. We have

$$\begin{aligned}\pi^{\forall}_{\widetilde{Z}}(k, \widetilde{X}) &= \{\mathbf{x} \in \pi(k, \widetilde{Z}) : \widetilde{Z}\restriction_{\mathbf{x}} \subseteq \widetilde{X}\restriction_{\mathbf{x}}\} \\ &= \{\mathbf{x} \in \pi(k, Z) : Z\restriction_{\mathbf{x}} + L^\perp \subseteq (X \cap Z)\restriction_{\mathbf{x}} + L^\perp\} \\ &= \{\mathbf{x} \in \pi(k, Z) : \mathbf{t}_{\mathbf{x}} + L + L^\perp \subseteq \mathbf{t}_{\mathbf{x}} + A_{\mathbf{x}} + L^\perp\} \\ &= \{\mathbf{x} \in \pi(k, Z) : \mathbf{t}_{\mathbf{x}} + L \subseteq \mathbf{t}_{\mathbf{x}} + A_{\mathbf{x}}\} && \text{see below} \\ &= \{\mathbf{x} \in \pi(k, Z) : Z\restriction_{\mathbf{x}} \subseteq (X \cap Z)\restriction_{\mathbf{x}}\} \\ &= \{\mathbf{x} \in \pi(k, Z) : Z\restriction_{\mathbf{x}} \subseteq X\restriction_{\mathbf{x}}\} \\ &= \pi^{\forall}_{Z}(k, X).\end{aligned}$$

Above, we have used the fact that $\mathbf{t}_{\mathbf{x}} + L + L^\perp \subseteq \mathbf{t}_{\mathbf{x}} + A_{\mathbf{x}} + L^\perp$ if and only if $\mathbf{t}_{\mathbf{x}} + L \subseteq \mathbf{t}_{\mathbf{x}} + A_{\mathbf{x}}$; which is trivially equivalent to

$$L + L^\perp \subseteq A_{\mathbf{x}} + L^\perp \text{ if and only if } L \subseteq A_{\mathbf{x}}.$$

The right to left direction of this double implication is straightforward. For the other direction, suppose $L + L^\perp \subseteq A_{\mathbf{x}} + L^\perp$ and pick $\mathbf{u} \in L$. Since $\mathbf{0} \in L^\perp$, we have $\mathbf{u} \in L + L^\perp \subseteq A_{\mathbf{x}} + L^\perp$, so there are $\mathbf{v}_1 \in A_{\mathbf{x}}$ and $\mathbf{v}_2 \in L^\perp$ such that $\mathbf{u} = \mathbf{v}_1 + \mathbf{v}_2$. By $A_{\mathbf{x}} \subseteq L$ we get $\mathbf{u} - \mathbf{v}_1 \in L$. From $\mathbf{u} - \mathbf{v}_1 = \mathbf{v}_2$ and the fact that L and $L^\perp$ are orthogonal lattices we conclude that $\mathbf{v}_2 = \mathbf{0}$, and thus $\mathbf{u} \in A_{\mathbf{x}}$. This completes the proof of Item (B).

To conclude, let us discuss how to compute representations of $\widetilde{Z}$ and $\widetilde{X}_j$ in $\mathrm{dom}(\rho_{\mathrm{SL}})$ in polynomial time. Starting from the representations $(d, \mathbf{v}_1, \ldots, \mathbf{v}_\ell)$ and $(d, \mathbf{w}_0, \ldots, \mathbf{w}_{r_j})$ of Z and X_j, respectively, we compute bases $\mathbf{u}_0, \ldots, \mathbf{u}_i$ and $\mathbf{u}_{i+1}, \ldots, \mathbf{u}_d$ of the lattices L and $L^\perp$, respectively, in polynomial time by relying on Lemma 8.11 and Lemma 8.5. So, $\widetilde{Z} = \mathbf{v}_0 + \mathrm{span}_{\mathbb{Z}}\{\mathbf{v}_0, \ldots, \mathbf{v}_\ell, \mathbf{0} \times \mathbf{u}_{i+1}, \ldots, \mathbf{0} \times \mathbf{u}_d\}$ and to find one of its representations in $\mathrm{dom}(\rho_{\mathrm{SL}})$ it suffices to put the matrix $[\mathbf{v}_1 \mid \cdots \mid \mathbf{v}_\ell \mid \mathbf{0} \times \mathbf{u}_{i+1} \mid \cdots \mid \mathbf{0} \times \mathbf{u}_d]$ in Hermite normal form using Proposition 8.1, to then take as period all its non-zero columns; and $\mathbf{v}_0$ as its base point. The computation of $\widetilde{X}_j$ is analogous, but we must first compute a representation for $X_j \wedge Z$, which can be done in polynomial time by Lemma 8.3. □

As the parameter $\mathrm{len}(\mathbf{1})$ fixes the number of shifted lattices of an element in $\mathrm{dom}(\mathrm{un}(\rho_{\mathrm{SL}}))$, Lemma 8.12 establishes Items (F4) and (F6) of Requirement 4.8. Then, by appealing to Lemma 4.9 and Proposition 4.2, the Lemmas 8.3, 8.9, 8.10 and 8.12 yield the main result of the section.

Theorem 8.13. *Fix $k \in \mathbb{N}$. The k negations satisfiability problem for weak Presburger arithmetic is decidable in polynomial time.*

By Proposition 4.2, we also conclude that there is a polynomial-time procedure that given a weak PA formula Φ only having k negations, returns an element of $\mathrm{dom}(\mathrm{dfnf}(\rho_{\mathrm{SL}}))$ representing the set of solutions $[\![\Phi]\!]_{\mathcal{Z}}$.

9. Conclusion

We developed a framework to establish polynomial-time decidability of fixed negation sentences of first-order theories whose signatures enjoy certain fixed-parameter tractability properties. A key feature of the framework is that it treats complementation in a general way, and considers universal projection as a first-class citizen. Note that, a priori, the latter operation might be easier than the former to decide, as shown for instance in [CH17].

We instantiated our framework to show that the fixed negation satisfiability problems for weak linear real arithmetic and weak Presburger arithmetic are decidable in PTIME. This is in sharp contrast with standard Presburger arithmetic, which is known to be NP-hard even when the Boolean structure and the number of variables in the formula is fixed [NP22]. We believe that our framework also provides a sensible approach to study fixed negation fragments of FO extensions of, e.g., certain abstract domains. An interesting extension of our running example in Section 4 to further test our framework is *octagon arithmetic*, where inequalities take the form $\pm x \pm y \leq c$, with $c \in \mathbb{Z}$ [Min06]. Over the integers, the full first-order theory of octagon arithmetic is known to be PSPACE-complete [BCM23]. More generally, as the various requirements to instantiate the framework relate to natural computational problems (deciding inclusion and computing projections), we are confident that our framework can also be appied to logical theories outside the world of arithmetic.

Acknowledgement

This work is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (GA 852769, ARiAT). Alessio Mansutti is co-funded by the European Union (GA 101154447), MICIU/AEI (GA CEX2024-001471-M and PID2022-138072OB-I00) and FEDER, UE. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them.

References

[ANVB98] Hajnal Andréka, István Németi, and Johan Van Benthem. Modal languages and bounded fragments of predicate logic. *J. Philos. Log.*, 27(3):217–274, 1998. doi:10.1023/A:1004275029985.

[BCM23] Michael Benedikt, Dmitry Chistikov, and Alessio Mansutti. The complexity of Presburger arithmetic with power or powers. In *ICALP*, 2023. doi:10.4230/LIPIcs.ICALP.2023.112.

[BCS15] Vince Bárány, Balder Ten Cate, and Luc Segoufin. Guarded negation. *J. ACM*, 62(3):1–26, 2015. doi:10.1145/2701414.

[BMMM18] Manuel Bodirsky, Barnaby Martin, Marcello Mamino, and Antoine Mottet. The complexity of disjunctive linear diophantine constraints. In *MFCS*, 2018. doi:10.4230/LIPIcs.MFCS.2018.33.

[BS81] S. Burris and H. P. Sankappanavar. *A Course in Universal Algebra*. Graduate Texts in Mathematics. Springer, 1981.

[BS96] Eric Bach and Jeffrey Shallit. *Algorithmic Number Theory, Vol 1: Efficient Algorithms*. Foundations of Computing. MIT Press, 1996.

[BT76] Itshak Borosh and Leon B. Treybing. Bounds on positive integral solutions of linear Diophantine equations. *Proc. Am. Math. Soc.*, 55:299–304, 1976. doi:10.2307/2041711.

[Büc60] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Math. Logic Quart.*, 6(1–6):66–92, 1960. doi:10.1002/malq.19600060105.

[CH17] Dmitry Chistikov and Christoph Haase. On the complexity of quantified integer programming. In *ICALP*, 2017. doi:10.4230/LIPIcs.ICALP.2017.94.

[Che09] Hubie Chen. A rendezvous of logic, complexity, and algebra. *ACM Comput. Surv.*, 42(1):2:1–2:32, 2009. doi:10.1145/1592451.1592453.

[CHHM22] Dmitry Chistikov, Christoph Haase, Zahra Hadizadeh, and Alessio Mansutti. Higher-order quantified Boolean satisfiability. In *MFCS*, 2022. doi:10.4230/LIPIcs.MFCS.2022.33.

[CHM22] Dmitry Chistikov, Christoph Haase, and Alessio Mansutti. Geometric decision procedures and the VC dimension of linear arithmetic theories. In *LICS*, 2022. doi:10.1145/3531130.3533372.

[DF99] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. doi:10.1007/978-1-4612-0515-9.

[Edm67] Jack Edmonds. Systems of distinct representatives and linear algebra. *J. Res. Natl. Bur. Stand., Sec. B, Mathematics and Mathematical Physics*, page 241, 1967.

[Grä91] Erich Grädel. Simple sentences that are hard to decide. *Inf. Comput.*, 94(1):62–82, 1991. doi:10.1016/0890-5401(91)90033-X.

[Hau14] Felix Hausdorff. *Grundzüge der Mengenlehre*. Veit and Company, 1914. https://archive.org/details/grundzgedermen00hausuoft.

[HMP23] Christoph Haase, Alessio Mansutti, and Amaury Pouly. On polynomial-time decidability of k-negations fragments of FO theories. In *MFCS*, 2023. doi:10.4230/LIPIcs.MFCS.2023.52.

[JLO24] Peter Jonsson, Victor Lagerkvist, and George Osipov. CSPs with few alien constraints. In *CP*, 2024. doi:10.4230/LIPIcs.CP.2024.15.

[Jun00] Markus Junker. A note on equational theories. *J. Symb. Log.*, 65(4):1705–1712, 2000. doi:10.2307/2695070.

[Kan90] Ravindran Kannan. Test sets for integer programs, $\forall\exists$ sentences. *Polyhedral Combinatorics, Proc. of a DIMACS workshop*, pages 38–48, 1990.

[KB79] Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comput.*, 8(4):499–507, 1979. doi:10.1137/0208040.

[KKS87] Marek Karpinski, Hans Kleine Büning, and Peter H. Schmitt. On the computational complexity of quantified Horn clauses. In *CSL*, 1987. doi:10.1007/3-540-50241-6_34.

[Koz81] Dexter Kozen. Positive first-order logic is NP-complete. *IBM Journal of Research and Development*, 25(4):327–332, 1981.

[Len83] Hendrik W. Lenstra Jr. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983. doi:10.1287/moor.8.4.538.

[Mic12] Daniele Micciancio. Lattices algorithms and applications, 2012. https://cseweb.ucsd.edu/classes/wi12/cse206A-a/.

[Min06] Antoine Miné. The octagon abstract domain. *High. Order Symb. Comput.*, 19(1):31–100, 2006. doi:10.1007/s10990-006-8609-1.

[NP22] Danny Nguyen and Igor Pak. Short Presburger arithmetic is hard. *SIAM J. Comput.*, 51(2):17:1–30, 2022. doi:10.1137/17M1151146.

[NS97] Phong Nguyen and Jacques Stern. Merkle-Hellman revisited: A cryptanalysis of the Qu-Vanstone cryptosystem based on group factorizations. In *CRYPTO*, 1997. doi:10.1007/BFb0052236.

[Sca84] Bruno Scarpellini. Complexity of subcases of Presburger arithmetic. *Trans. Am. Math. Soc.*, 284:203–218, 1984. doi:10.2307/1999283.

[Sch97] Uwe Schöning. Complexity of Presburger arithmetic with fixed quantifier dimension. *Theory Comput. Syst.*, 30(4):423–428, 1997. doi:10.1007/s002240000059.

[Sch99] Alexander Schrijver. *Theory of linear and integer programming*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1999.

[Sto76] Larry J. Stockmeyer. The polynomial-time hierarchy. *Theor. Comput. Sci.*, 3(1):1–22, 1976. doi:10.1016/0304-3975(76)90061-X.

[TCS13] Balder Ten Cate and Luc Segoufin. Unary negation. *Log. Methods Comput. Sci.*, 9(3), 2013. doi:10.2168/LMCS-9(3:25)2013.

[Vor99] Andrei Voronkov. The ground-negative fragment of first-order logic is Π_2^p-complete. *J. Symb. Log.*, 64(3):984–990, 1999. doi:10.2307/2586615.

[vzGS78] Joachim von zur Gathen and Malte Sieveking. A bound on solutions of linear integer equalities and inequalities. *Proc. Am. Math. Soc.*, 72(1):155–158, 1978. doi:10.1080/00029890.1978.11994639.

[Wei00] Klaus Weihrauch. *Computable analysis: an introduction*. Springer Science & Business Media, 2000.