Journal of Logic, Language and Information
https://doi.org/10.1007/s10849-022-09384-9

Feferman–Vaught Decompositions for Prefix Classes of First Order Logic

Abhisekh Sankaran[1]

Accepted: 1 July 2022

Abstract

The Feferman–Vaught theorem provides a way of evaluating a first order sentence φ on a disjoint union of structures by producing a decomposition of φ into sentences which can be evaluated on the individual structures and the results of these evaluations combined using a propositional formula. This decomposition can in general be non-elementarily larger than φ. We introduce a "tree" generalization of the prenex normal form (PNF) for first order sentences, and show that for an input sentence in this form having a fixed number of quantifier alternations, a Feferman–Vaught decomposition can be obtained in time elementary in the size of the sentence. The sentences in the decomposition are also in tree PNF, and further have the same number of quantifier alternations and the same quantifier rank as the input sentence. We extend this result by considering binary operations other than disjoint union, in particular sum-like operations such as join, ordered sum and NLC-sum, that are definable using quantifier-free translation schemes.

Keywords Feferman–Vaught · Decomposition · Prefix classes · Elementary · Sum-like

Mathematics Subject Classification 03C13 · 03C40 · 03C52 · 05C05 · 05C38 · 05C62 · 05C69 · 05C75

1 Introduction

The Feferman–Vaught theorem (Feferman and Vaught 1959) is a classic result from model theory that gives a method to evaluate a first order (FO) sentence over a general-

Research supported by the Leverhulme Trust through a Research Project Grant on "Logical Fractals".

✉ Abhisekh Sankaran
abhisekh@gmail.com

[1] Department of Computer Science and Technology, University of Cambridge, Cambridge, UK

Published online: 12 November 2022

ized product of structures by reducing it to the evaluation of other first order sentences over the individual structures and the evaluation of a monadic second order (MSO) sentence over an index structure. Historically, the theorem was first shown by Mostowski (1952) for the special case of generalized products that are direct powers, before the full version of the theorem was established in Feferman and Vaught (1959). One of the simplest generalized products is disjoint union and here in the case of finitely many structures, one can replace the evaluation of the mentioned MSO sentence, with the evaluation of a propositional formula. One can also stratify the theorem by the rank of the FO sentence φ being evaluated on the disjoint union, that is, one can have the sentences in the alluded "decomposition" of φ to have the same bound on their rank as that for φ. These results and their generalizations to MSO have a variety of applications in computer science, such as in showing the decidability of theories, satisfiability checking and algorithmic meta theorems (see Makowsky 2004; Blumensath et al. 2008; Grohe 2008; Thomas 1997 for surveys on these results).

Computing a Feferman–Vaught decomposition of an FO sentence φ over the binary disjoint union of structures (finite or infinite) takes time that is bounded by an m-fold exponential in the size of φ, where m is the rank of φ (Makowsky 2004). This runtime is thus non-elementary in the size of φ, and cannot be improved in general, owing to a non-elementary lower bound for the size of the decomposition over all finite structures (indeed even over all finite directed rooted trees), and hence also over arbitrary structures (Dawar et al. 2007). The time complexity can however be improved by considering special classes of finite structures, such as those of bounded degree, where it takes at most 3-fold exponential time to compute the decomposition if the degree is at least 3, and 2-fold exponential time if the degree is at most 2 (Harwath et al. 2015).

In this paper, we take a different approach towards computing decompositions faster, by observing the syntax of the formulae considered. A well-studied normal form for FO sentences is the prenex normal form (PNF). A prenex sentence is an FO sentence which begins with a string of quantifiers that is followed by a quantifier-free formula. Every FO sentence is equivalent to a prenex sentence and can be brought into such a PNF in time polynomial in the size of the FO sentence (Harwath 2016). Let Σ_n and Π_n denote the classes of all PNF sentences that contain $n-1$ alternations of quantifiers (equivalently, n blocks of quantifiers) in the quantifier prefix, and whose leading quantifiers are existential and universal respectively. It turns out that various properties of interest in computer science can be expressed using Σ_n or Π_n sentences for very low values of n, indeed with n as just 2. Examples include parameterized problems such as k- Vertex cover, k-Clique, k-Dominating Set and (k, d)-Scattered Set which are all Σ_2 expressible. (More examples can be found in Appendix A of Sankaran 2018). In program verification, the Σ_2 fragment is called *Effectively Propositional Logic* (EPR) for which there exist practical implementations of DPLL-based decision procedures for checking satisfiability (Piskac et al. 2010; Emmer et al. 2010; Gulwani 2010). In databases, Π_2 sentences are the syntactic form of source-to-target dependencies in the data exchange setting, and also of views in data integration (Fagin et al. 2005; Lenzerini 2002). Again, over special classes of structures such as those of bounded degree as aforementioned, every FO sentence is equivalent to a Boolean combination of Σ_2 sentences. Thus, considering a fixed number of quantifier alternations is a well-motivated restriction.

Towards the central result of this paper, we consider a "tree" generalization of Σ_n and Π_n formulae, that we denote $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ respectively. For any FO formula, any root to leaf path in the parse tree of the formula can be seen as a word over the quantifier symbols $\exists$ and $\forall$, the logical connectives $\bigwedge$, $\bigvee$ and $\neg$, the predicate symbols of τ along with "=", and a set of variables. We define $\mathrm{T}\Sigma_n$ as the class of all FO formulae ψ in negation normal form (NNF, where negations appear only at the atomic level), such that the word corresponding to any root to leaf path in the parse tree of ψ has the form $\exists \cdot (\exists^* \bigwedge \forall^* \bigvee)^* w$, where the number of quantifier alternations in the word is at most $n-1$, and w contains no quantifiers. Likewise for $\mathrm{T}\Pi_n$, this word has the form $\forall \cdot (\forall^* \bigvee \exists^* \bigwedge)^* w$ with at most $n-1$ quantifier alternations and w as before. Clearly $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ generalize the Σ_n and Π_n classes of formulae considered in NNF, and therefore are normal forms for FO.

On the semantic front, we consider binary operations on structures that are definable using quantifier-free formulae. Given two structures $\mathfrak{A}$ and $\mathfrak{B}$ over a vocabulary τ, the *annotated disjoint union* of $\mathfrak{A}$ and $\mathfrak{B}$ is the disjoint union of these structures in which the elements of the (sub-)universe of $\mathfrak{A}$ are labeled using a new unary predicate. We can now specify binary operations on inputs $\mathfrak{A}$ and $\mathfrak{B}$, using quantifier-free scalar translation schemes that operate on the annotated disjoint union $\mathfrak{C}$ of $\mathfrak{A}$ and $\mathfrak{B}$. These schemes are sequences of FO formulae that enable constructing a τ-structure from $\mathfrak{A}$ and $\mathfrak{B}$ by interpreting the predicates of τ as the relations defined by the formulae in $\mathfrak{C}$. Here, scalar means that any element of the constructed τ-structure is an element of the union of the universes of $\mathfrak{A}$ and $\mathfrak{B}$ (as opposed to being a general tuple from the union). Such a binary operation as described is called a *quantifier-free sum-like* operation. A number of well-known operations on structures are quantifier-free and sum-like; examples of these include the disjoint union, the join of two graphs, the ordered sum of structures, and the NLC-sum of graphs (Makowsky 2004; Wanke 1994). One can then consider such operations as forming a more general setting than just the disjoint union to obtain Feferman-Vaught decompositions.

To be able to present the main result of the paper concretely, we describe in more detail the notion of a decomposition, and the classes $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$, that we have been referring to only in a semi-formal way so far. To describe decompositions we need the notion of a *reduction sequence*. Call a triple $(\Delta_1, \Delta_2, \beta)$ a reduction sequence if Δ_1 and Δ_2 are sequences of FO sentences, and β is a propositional formula over a set of variables that contains a unique propositional variable for every sentence appearing in Δ_1 or Δ_2. Call a pair $(\mathfrak{A}_1, \mathfrak{A}_2)$ of structures a model for a reduction sequence D as above if the formula β is true under the valuation that, for a variable p corresponding to a sentence ψ in Δ_i, assigns true to p if ψ is true in $\mathfrak{A}_i$ and false otherwise (here i is either 1 or 2). We now say that D is a Feferman–Vaught decomposition of φ over a binary operation $\circledast$ if for all $\mathfrak{A}_1, \mathfrak{A}_2$, the $\circledast$-composite of $\mathfrak{A}_1$ and $\mathfrak{A}_2$ satisfies φ precisely when $(\mathfrak{A}_1, \mathfrak{A}_2)$ is a model for D. We now come to the classes $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$. These are the classes of all FO formulae of the form $\exists \bar{x} \bigwedge \psi$ and $\forall \bar{x} \bigvee \chi$ respectively, where $\bigwedge \chi$ is a conjunction of $\mathrm{T}\Pi_{n-1}$ formulae and $\bigvee \psi$ is a disjunction of $\mathrm{T}\Sigma_{n-1}$ formulae. The classes $\mathrm{T}\Sigma_0$ and $\mathrm{T}\Pi_0$ are both the class of all quantifier-free formulae in NNF. We denote by $\mathrm{T}\Sigma_n[m]$ and $\mathrm{T}\Pi_n[m]$ respectively the classes of $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ formulae of quantifier rank at most m. We can now state our central result as below. The function $\mathsf{tower}(n, \cdot)$ is the n-fold exponential function given inductively as $\mathsf{tower}(0, k) = k$ and $\mathsf{tower}(n, k) = 2^{\mathsf{tower}(n-1,k)}$. The size of a formula φ, denoted $|\varphi|$, is the number of nodes in the parse tree of φ. The model of computation is the usual Turing model.

Theorem 1 (Theorem 3, Sect. 4) *Let $\mathcal{L}$ be one of the logics $\mathrm{T}\Sigma_n[m]$ or $\mathrm{T}\Pi_n[m]$ where $n, m \geq 0$. Let $\circledast$ be a quantifier-free sum-like binary operation on structures, whose defining (quantifier-free) translation scheme is Ξ. Then for every $\mathcal{L}$ sentence φ, there exists a Feferman–Vaught decomposition D of φ over $\circledast$ consisting of $\mathcal{L}$ sentences. Further, the decomposition D has size $\mathsf{tower}(n, O((n+1) \cdot |\varphi| \cdot |\Xi|))$, and can be computed from φ in time $\mathsf{tower}(n, O((n+1) \cdot (|\varphi| \cdot |\Xi|)^2))$.*

In other words, computing the Feferman–Vaught decomposition of φ over $\circledast$ has an elementary dependence on the size of φ when the number of quantifier alternations in the mentioned "tree PNF" form of φ is bounded. Further, this decomposition is stratified (in the sense mentioned earlier) by both the rank of φ as well as the number of quantifier alternations in the tree PNF form. As a consequence, we obtain a "composition" result that states that the $\mathrm{T}\Sigma_n[m]$ theory of the $\circledast$-composite of two structures is determined by the $\mathrm{T}\Sigma_n[m]$ theories of the individual structures; likewise for the $\mathrm{T}\Pi_n[m]$ theory (cf. Corollary 2). This result thus refines the known rank-preserving composition results for FO for quantifier-free sum-like operations (Makowsky 2004), to preserve the number of quantifier alternations as well. The proof of Theorem 1 goes via first showing an analogous result for the annotated disjoint union operation (Theorem 2). This is then transferred to general quantifier-free sum-like operations using the defining translation schemes of the latter. A similar approach is taken for showing the mentioned composition result (cf. Corollary 1 for the composition result for the annotated disjoint union).

An additional related investigation that we do concerns the number of $\mathrm{T}\Sigma_n$ or $\mathrm{T}\Pi_n$ sentences of a given quantifier rank considered modulo equivalence. It is well-known that the number of non-equivalent FO sentences of quantifier rank at most m is inherently a non-elementary function of m (Chapter 3 (Libkin, 2013)). We show that the number of $\mathrm{T}\Sigma_n[m]$ sentences is up to equivalence at most an $(n+2)$-fold exponential function of m; likewise for $\mathrm{T}\Pi_n[m]$ (cf. Proposition 2). The result holds more generally for $\mathrm{T}\Sigma_n[m]$ and $\mathrm{T}\Pi_n[m]$ formulae with a given number of free variables. Thus with a bound on the number of quantifier alternations, the number of FO sentences of rank m, and more generally FO formulae of rank m with a fixed number of free variables, is up to equivalence of formulae bounded by an elementary function of m.

Paper Organization In Sect. 2, we introduce terminology and notation, and formally define the classes $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$. In Sect. 3, we prove the decomposition result for $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ formulae over the annotated disjoint union operation (Theorem 2) and in Sect. 3 we transfer this result to general quantifier-free sum-like binary operations to show Theorem 1. In the latter section, we also show elementary bounds on the cardinalities of $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ considered up to equivalence. We conclude in Sect. 5 presenting various directions for future work.

Related Work It is known that bounding the number of quantifier alternations allows obtaining finite automata for MSO sentences over words, in elementary time (Thomas

1997), in contrast with general non-elementary lower bounds in this context (Stockmeyer and Meyer 1973). The same restriction on Presburger arithmetic again yields faster decision procedures (Reddy and Loveland 1978; Haase 2014). The two variable fragment of FO also admits an elementary (doubly exponential) Feferman–Vaught decomposition for disjoint union (Göller et al. 2015). Finally, other related decomposition results that do not necessarily give elementary bounds include Feferman–Vaught decompositions for weighted extensions of FO (Droste and Paul 2018; Bergerem and Schweikardt 2021) over (binary) disjoint union, and those for an extension of MSO called Guarded Second Order logic (GSO), over the disjoint union of arbitrarily many structures (Elberfeld et al. 2016).

2 Notation and Terminology

We assume the reader is familiar with the standard syntax and semantics of FO (Libkin 2013). Let $\mathbb{N}$ be the set of all natural numbers (including 0). For $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1, \ldots, n\}$. We will be concerned in this paper with only FO and only finite relational vocabularies τ, that is finite vocabularies τ containing only relation symbols. We denote FO formulae having free variables amongst a tuple $\bar{x}$ of variables as $\varphi(\bar{x})$, $\psi(\bar{x})$, etc. possibly with subscripts immediately succeeding the Greek letters. When $\bar{x}$ is clear from context, we denote the formulae simply as φ, ψ, etc. Structures over τ, or simply τ-structures, are denoted using the letters $\mathfrak{A}$, $\mathfrak{B}$, etc. possibly with subscripts. Given a τ-structure $\mathfrak{A}$, a finite tuple $\bar{a}$ of elements of $\mathfrak{A}$, and an FO formula $\varphi(\bar{x})$ over τ such that the length $|\bar{x}|$ of $\bar{x}$ is equal to $|\bar{a}|$, we denote by $(\mathfrak{A}, \bar{a}) \models \varphi(\bar{x})$ that φ is true in $\mathfrak{A}$ when $\bar{x}$ is instantiated as $\bar{a}$. We use the shorthand $\mathfrak{A} \models \varphi(\bar{a})$ for $(\mathfrak{A}, \bar{a}) \models \varphi(\bar{x})$ whenever it is convenient to do so. The size of φ, denoted $|\varphi|$, is the number of nodes in the parse tree of φ.

1. The Logics $\mathrm{T}\Sigma_n$ **and** $\mathrm{T}\Pi_n$: We define the classes $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ of FO formulae over τ that respectively generalize the Σ_n and Π_n prefix classes over this vocabulary when the quantifier-free parts of the formulae in the latter classes are in negation normal form (NNF). We define $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ using simultaneous induction.

- For the base case of $n = 0$, the classes $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ are both equal to the class of all quantifier-free FO formulae over τ in NNF. So this class is built up from atomic formulae of the form $R(x_1, \ldots, x_k)$ and $x_1 = x_2$ and the negations of these, where R is a k-ary predicate symbol in τ and $x_1, \ldots, x_k$ are variables, using conjunctions and disjunctions of finite arity.
- Inductively assume $\mathrm{T}\Sigma_{n'}$ and $\mathrm{T}\Pi_{n'}$ have been defined for all $n' < n$. Then:
  - A $\mathrm{T}\Sigma_n$ formula is a $\mathrm{T}\Sigma_{n,r}$ formula for some $r \in \mathbb{N}$ where: (i) a formula is in $\mathrm{T}\Sigma_{n,0}$ if it is of the form $\bigwedge_{i\in[p]} \gamma_i$ where $p \in \mathbb{N}$ and γ_i is a $\mathrm{T}\Pi_{n''}$ formula for some $n'' < n$; (ii) a formula is in $\mathrm{T}\Sigma_{n,r}$ for $r > 0$ if it either is a $\mathrm{T}\Sigma_{n,r-1}$ formula, or is of the form $\exists y \varphi_1$ where φ_1 is a $\mathrm{T}\Sigma_{n,r-1}$ formula.
  - A $\mathrm{T}\Pi_n$ formula is a $\mathrm{T}\Pi_{n,r}$ formula for some $r \in \mathbb{N}$ where: (i) a formula is in $\mathrm{T}\Pi_{n,0}$ if it is of the form $\bigvee_{i\in[p]} \gamma_i$ where $p \in \mathbb{N}$ and γ_i is a $\mathrm{T}\Sigma_{n''}$ formula

for some $n'' < n$; (ii) a formula is in $\mathrm{T}\Pi_{n,r}$ for $r > 0$ if it either is a $\mathrm{T}\Pi_{n,r-1}$ formula, or is of the form $\forall y \varphi_1$ where φ_1 is a $\mathrm{T}\Pi_{n,r-1}$ formula.

We make a few simple observations about the classes defined above. Firstly, one can see using an easy induction that for $n' < n$, the classes $\mathrm{T}\Sigma_{n'}$ and $\mathrm{T}\Pi_{n'}$ are both contained inside each of $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$. The classes $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ are incomparable. The negation of a $\mathrm{T}\Sigma_n$ ($\mathrm{T}\Pi_n$) formula is equivalent to a $\mathrm{T}\Pi_n$ ($\mathrm{T}\Sigma_n$) formula. The prefix classes Σ_n and Π_n are respectively contained in $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$, whereby the latter classes constitute normal forms for FO.

We define in the usual way the rank of a formula $\varphi \in \{\mathrm{T}\Sigma_n, \mathrm{T}\Pi_n\}$ – this is the maximum of the number of quantifiers appearing in any root-to-leaf path in the parse tree of φ. For $\mathcal{L} \in \{\mathrm{T}\Sigma_n, \mathrm{T}\Pi_n, \mathrm{T}\Sigma_{n,r}, \mathrm{T}\Pi_{n,r}\}$, let $\mathcal{L}[m]$ denote the classes of all $\mathcal{L}$ formulae of rank at most m for $m \in \mathbb{N}$. Observe that the negation of a $\mathrm{T}\Sigma_n[m]$ ($\mathrm{T}\Pi_n[m]$) formula is equivalent to a $\mathrm{T}\Pi_n[m]$ ($\mathrm{T}\Sigma_n[m]$) formula; likewise with $\mathrm{T}\Sigma_{n,r}[m]$ and $\mathrm{T}\Pi_{n,r}[m]$ in place of $\mathrm{T}\Sigma_n[m]$ and $\mathrm{T}\Pi_n[m]$. Also, from the definition of $|\varphi|$, it follows for $\varphi \in \mathrm{T}\Sigma_n[m]$, that in the event that either φ is not in $\mathrm{T}\Sigma_{n'}$ for any $n' < n$ or φ is not in $\mathrm{T}\Sigma_n[m']$ for any $m' < m$, both n and m are at most $|\varphi|$. Finally $\mathrm{T}\Sigma_n[m] = \bigcup_{r \in \mathbb{N}} \mathrm{T}\Sigma_{n,r}[m]$ and $\mathrm{T}\Pi_n[m] = \bigcup_{r \in \mathbb{N}} \mathrm{T}\Pi_{n,r}[m]$.

Examples: We recall the examples mentioned in Sect. 1 and see below that they can be described by $\mathrm{T}\Sigma_n$ or $\mathrm{T}\Pi_n$ sentences for low values of n.

1. k- Vertex cover: The problem asks for a set S of at most k vertices in the given graph such that for every edge in the graph, one of its endpoints is in S. This problem can be expressed as the $\mathrm{T}\Sigma_{2,k}[k+2]$ sentence given below. The notation "$a \to b$" is a shorthand for "$\neg a \vee b$".

$$k\text{-VertexCover} := \exists x_1 \ldots \exists x_k \forall y_1 \forall y_2 \big(E(y_1, y_2) \to \bigvee_{j \in [2], i \in [k]} y_j = x_i\big)$$

2. k-Clique: The problem asks if there is a clique of size at least k in the input graph. The sentence below describing the problem belongs to $\mathrm{T}\Sigma_{1,k}[k]$.

$$k\text{-Clique} := \exists x_1 \ldots \exists x_k \bigwedge_{i,j \in [k], i \neq j} E(x_i, x_j)$$

3. k-Dominating Set: The problem asks if there is a set S of at most k vertices in the given graph such that every vertex not in S is adjacent to some vertex in S. This problem can be expressed as a $\mathrm{T}\Sigma_{2,k}[k+1]$ sentence as below.

$$k\text{-DominatingSet} := \exists x_1 \ldots \exists x_k \forall y \big(\bigvee_{i \in [k]} (y = x_i \vee E(y, x_i))\big)$$

4. (k, d)-Scattered Set: This problem asks for a set S of at least k vertices in the input graph such that the distance in the graph between any pair of vertices in S is at least d. This problem can be described using the $\mathrm{T}\Sigma_{2,k}[k+d]$ sentence given below. The formula "$\mathsf{dist}(x_i, x_j) \geq d$" is easier understood in terms of its negation

that says that there is path of length at most $d-1$ between x_i and x_j (but the natural syntactic form of the negation of this statement is not in NNF.)

$$\begin{array}{lll}(k,d)\text{-}\mathsf{ScatteredSet} & := & \exists x_1 \ldots \exists x_k \bigwedge_{i,j\in[k],i\neq j}(x_i \neq x_j \wedge \mathsf{dist}(x_i,x_j)\geq d)\\ \mathsf{dist}(x_i,x_j)\geq d & := & \forall y_1 \ldots \forall y_d\\ & & \big(y_1 = x_i \wedge \bigwedge_{l\in[d-1]}(y_l = y_{l+1} \vee E(y_l,y_{l+1}))\big)\\ & & \rightarrow \neg(y_d = x_j)\end{array}$$

5. Source-to-target tuple generating dependencies (STTGDs): These are special kinds of FO formulae that are used in the data exchange setting to define constraints on a *source schema* S and a *target schema* T. Here, a schema is a finite collection of relation symbols, so a finite relational vocabulary in our terminology. The typical form of an STTGD is as below, where ϕ_S is a conjunction of atomic formulae over S, and ψ_T is a conjunction of atomic formulae over T. Clearly, $\mathsf{STTGD} \in \mathrm{T}\Pi_2$ (assuming $\neg\phi_S$ in the expansion of $\rightarrow$ below, is taken in NNF).

$$\mathsf{STTGD} := \forall x_1 \ldots \forall x_k \big(\phi_S(x_1,\ldots,x_k) \rightarrow \exists y_1 \ldots \exists y_l \psi_T(x_1,\ldots,x_k,y_1,\ldots,y_l)\big)$$

2. Translation schemes: We recall the model-theoretic notion of translation schemes from the literature (Makowsky 2004), in particular its special case where the formulae are quantifier-free and contain no parameters. For vocabularies τ and σ, a *quantifier-free scalar* (τ,σ)*-translation scheme* Ξ, or simply a (τ,σ)-translation scheme Ξ, is a tuple $(\xi_U(x), (\xi_R(\bar{y}_R))_{R\in\sigma})$ of quantifier-free FO formulae in NNF over τ such that $|\bar{y}_R| = \mathrm{ar}(R)$ where $\mathrm{ar}(R)$ denotes the arity of R. The qualification "scalar" is used to indicate that the formula ξ_U has only one free variable (as opposed to "vectorized" in case of having more than one free variable). A (τ,σ)-translation scheme Ξ defines a function from τ-structures to σ-structures, called a *transduction*, that we denote again by Ξ and that is defined as follows. For a τ-structure $\mathfrak{A}$, let $\mathfrak{B}$ denote the σ-structure given by $\mathfrak{B} = \Xi(\mathfrak{A})$; then: (i) the universe B of $\mathfrak{B}$ is given by $B = \xi_U(\mathfrak{A}) = \{a \mid a \text{ is an element of } \mathfrak{A} \text{ such that } \mathfrak{A} \models \xi_U(a)\}$; (ii) a relation $R \in \sigma$ is interpreted in $\mathfrak{B}$ as $R^{\mathfrak{B}} = \xi_R(\mathfrak{A}) \cap B^{\mathrm{ar}(R)}$ where $\xi_R(\mathfrak{A}) = \{\bar{a} \mid \bar{a} \text{ is a } |\bar{y}_R|\text{-tuple from } \mathfrak{A} \text{ such that } \mathfrak{A} \models \xi_R(\bar{a})\}$. We say $\mathfrak{B}$ is Ξ*-transduced from* $\mathfrak{A}$, or simply, transduced from $\mathfrak{A}$. The transduction Ξ is clearly isomorphism-preserving: isomorphic τ-structures are mapped to isomorphic σ-structures. As an example, let $\tau = \sigma = \{E\}$ where E is a binary relation symbol, and let Ξ be the (τ,σ)-translation scheme given by $\Xi = (\xi_U(x), \xi_E(x,y))$ where $\xi_U(x) := \mathsf{True}$ and $\xi_E(x,y) = \neg E(x,y)$. Then the transduction defined by Ξ on undirected graphs is exactly graph complementation.

One can utilize the mechanism of translation schemes as defined above to not just construct unary operations on structures as seen above, but also binary operations. To be able to do so, we first define the *annotated* disjoint union of given disjoint τ-structures $\mathfrak{A}_1$ and $\mathfrak{A}_2$. Recall that the disjoint union of $\mathfrak{A}_1$ and $\mathfrak{A}_2$, denoted $\mathfrak{A}_1 \uplus \mathfrak{A}_2$, is the τ-structure whose universe is the disjoint union of the universes of $\mathfrak{A}_1$ and $\mathfrak{A}_2$, and in which every predicate of τ is interpreted as the union of its interpretations in $\mathfrak{A}_1$ and $\mathfrak{A}_2$. The *annotated* disjoint union of $\mathfrak{A}_1$ and $\mathfrak{A}_2$, denoted $\mathfrak{A}_1 \underline{\uplus} \mathfrak{A}_2$, is the structure defined by expanding $\mathfrak{A}_1 \cup \mathfrak{A}_2$ with a new unary predicate not in τ, that is interpreted as the universe of $\mathfrak{A}_1$. Formally, $\mathfrak{A}_1 \underline{\uplus} \mathfrak{A}_2$ is a $\underline{\tau}$-structure $\mathfrak{C}$ for $\underline{\tau} = \tau \uplus \{P\}$ and P a unary predicate (not in τ), such that: (i) the τ-reduct of $\mathfrak{C}$ is the structure $\mathfrak{A}_1 \uplus \mathfrak{A}_2$; the former is the τ-structure obtained from $\mathfrak{C}$ by removing the interpretation of the predicate P; (ii) the interpretation of P in $\mathfrak{C}$ is the universe of $\mathfrak{A}_1$.

We can now utilize the annotated disjoint union to define binary operations on τ-structures. In particular, each $(\underline{\tau}, \tau)$-translation scheme Ξ defines a binary operation $\circledast$ on disjoint τ-structures given by $\mathfrak{A}_1 \circledast \mathfrak{A}_2 = \Xi(\mathfrak{A}_1 \underline{\uplus} \mathfrak{A}_2)$. We call $\circledast$ a *quantifier-free sum-like binary operation* on τ-structures, and call Ξ its *quantifier-free definition*. Observe that every element of $\mathfrak{A}_1 \circledast \mathfrak{A}_2$ is an element of either $\mathfrak{A}_1$ or $\mathfrak{A}_2$ due to the sum-like nature of $\circledast$. Following are some well-known binary operations on structures that are quantifier-free and sum-like.

1. Disjoint union: A quantifier-free definition for this operation is $\Xi = (\xi_U(x), (\xi_R(\bar{y}_R))_{R \in \tau})$ where:
   - $\xi_U(x) :=$ True; and
   - $\xi_R(\bar{y}_R) := R(\bar{y}_R)$.
2. Ordered sum: Here τ is the vocabulary of ordered structures, and so is of the form $\tau = \{\leq\} \uplus \tau_1$ where $\leq$ is a binary predicate that is interpreted as a total linear order in an ordered τ-structure. The ordered sum of ordered τ-structures $\mathfrak{A}_1$ and $\mathfrak{A}_2$ is the disjoint union of the structures in which the interpretation of $\leq$ additionally contains the pairs (a_1, a_2) for $a_1 \in \mathfrak{A}_1$ and $a_2 \in \mathfrak{A}_2$ (thus making the interpretation a total order). Then a quantifier-free definition of the ordered sum is $\Xi = (\xi_U(x), (\xi_R(\bar{y}_R))_{R \in \tau})$ where:
   - $\xi_U(x) :=$ True;
   - $\xi_R(\bar{y}_R) := R(\bar{y}_R)$ for $R \in \tau_1$; and
   - $\xi_{\leq}(y_1, y_2) := (y_1 \leq y_2) \vee (P(y_1) \wedge \neg P(y_2))$.
3. NLC-sum: Here τ is the vocabulary of labeled undirected graphs, so is of the form $\tau = \{E\} \uplus \tau_1$ where $\tau_1 = \{Q_1, \ldots, Q_r\}$ and Q_i is a unary predicate for $i \in [r]$. An $[r]$-labeled graph is a τ-structure whose $\{E\}$-reduct is an undirected graph, and in which the interpretations of the Q_is form a partition of the vertex set of the graph (allowing empty parts in a slight abuse of the usual notion of a partition). The NLC-sum operation (Wanke 1994) is specified using a set $S \subseteq [r]^2$. It takes as input two $[r]$-labeled graphs G_1 and G_2, creates their disjoint union, and adds edges between vertices $u \in G_1$ and $v \in G_2$ such that $G_1 \models Q_i(u)$ and $G_2 \models Q_j(v)$ where $(i, j) \in S$. This operation has a quantifier-free definition $\Xi = (\xi_U(x), (\xi_R(\bar{y}_R))_{R \in \tau})$ where:
   - $\xi_U(x) :=$ True;
   - $\xi_{Q_i}(y) := Q_i(y)$ for $i \in [r]$; and
   - $\xi_E(y_1, y_2) := E(y_1, y_2) \vee \bigvee_{(i,j) \in S}(P(y_1) \wedge Q_i(y_1) \wedge \neg P(y_2) \wedge Q_j(y_2))$

   The join of two graphs is essentially the special case of the NLC-sum when r equals 1.

Just as a (τ, σ)-translation scheme $\Xi = (\xi_U(x), (\xi_R(\bar{y}_R))_{R\in\sigma})$ gives rise to a function from τ-structures to σ-structures, it also gives rise to a function from $\mathcal{L}$ formulae over σ to $\mathcal{L}$ formulae over τ, for $\mathcal{L} \in \{\mathrm{T}\Sigma_n, \mathrm{T}\Pi_n\}$. This function, called a *translation*, that we again denote as Ξ for the ease of readability, is defined as follows. Let $\varphi(\bar{z})$ be an $\mathcal{L}$ formula over σ. Then $\Xi(\varphi)(\bar{z})$ is the $\mathcal{L}$ formula over τ defined inductively over the structure of $\varphi(\bar{z})$ as follows:

1. If $\varphi(\bar{z}) := R(z_1, \ldots, z_r)$ for $R \in \sigma \cup \{=\}$, then $\Xi(\varphi)(\bar{z}) := \xi_R(z_1, \ldots, z_r) \wedge \bigwedge_{i\in[r]} \xi_U(z_i)$.
2. If $\varphi(\bar{z}) := \neg R(z_1, \ldots, z_r)$ for $R \in \sigma \cup \{=\}$, then $\Xi(\varphi)(\bar{z}) := \neg\xi_R(z_1, \ldots, z_r) \wedge \bigwedge_{i\in[r]} \xi_U(z_i)$, where $\neg\xi_R(z_1, \ldots, z_r)$ is considered in NNF.
3. If $\varphi(\bar{z}) := \circledast_{i\in I}\varphi_i(\bar{z})$ for $\circledast \in \{\bigwedge, \bigvee\}$, then $\Xi(\varphi)(\bar{z}) := \circledast_{i\in I}\Xi(\varphi_i)(\bar{z})$.
4. If $\varphi(\bar{z}) := Q\bar{x}\varphi_1(\bar{z}, \bar{x})$ where $\varphi_1(\bar{z}, \bar{x}) := \circledast_{i\in I}\varphi'_i(\bar{z}, \bar{x})$ and $(Q, \circledast) \in \{(\exists, \bigwedge), (\forall, \bigvee)\}$, then for $\bar{x} = (x_1, \ldots, x_r)$ for some $r \geq 0$, we have the following. (Here '$Q\bar{x}$' is a shorthand for '$Qx_1 \ldots Qx_r$'.)
   - $\Xi(\varphi)(\bar{z}) := \exists\bar{x}(\bigwedge_{j\in[r]} \xi_U(x_j) \wedge \Xi(\varphi_1)(\bar{z}, \bar{x}))$ if $(Q, \circledast) = (\exists, \bigwedge)$.
   - $\Xi(\varphi)(\bar{z}) := \forall\bar{x}((\bigvee_{j\in[r]} \neg\xi_U(x_j)) \vee \Xi(\varphi_1)(\bar{z}, \bar{x}))$ if $(Q, \circledast) = (\forall, \bigvee)$, where $\neg\xi_U(x_j)$ is taken in NNF. Observe that $\Xi(\varphi)(\bar{z})$ here is equivalent to the formula $\forall\bar{x}((\bigwedge_{j\in[r]} \xi_U(x_j)) \rightarrow \Xi(\varphi_1)(\bar{z}, \bar{x}))$.

Remark 1 We observe that the scalar nature of $\circledast$ ensures that the free variables of $\Xi(\varphi)$ are exactly those of φ, and that the quantifier-free nature of $\circledast$ ensures that $\Xi(\varphi)$ has the same quantifier rank as φ.

We can now state the following equivalence [This is a special case of a more general result called the *fundamental theorem of translation schemes* (Theorem 2.6 Makowsky 2004)]. Let Ξ be a (τ, σ)-translation scheme, and $\varphi(\bar{x})$ be an $\mathcal{L}$ formula over σ. Let $\mathfrak{A}$ be a τ-structure, and $\bar{a}$ be a tuple from $\Xi(\mathfrak{A})$ of length $|\bar{x}|$. Then

$$\Xi(\mathfrak{A}) \models \varphi(\bar{a}) \quad \Leftrightarrow \quad \mathfrak{A} \models \Xi(\varphi)(\bar{a}).$$

Applying this to the special setting of Ξ being the definition of a quantifier-free sum-like binary operation $\circledast$, we get the following.

Proposition 1 *Let $\varphi(\bar{x}_1, \bar{x}_2)$ be an $\mathcal{L}$ formula over τ, for $\mathcal{L} \in \{\mathrm{T}\Sigma_n, \mathrm{T}\Pi_n\}$. For $i \in [2]$, let $\mathfrak{A}_i$ be a τ-structure and let $\bar{a}_i$ be a tuple from $\mathfrak{A}_1 \circledast \mathfrak{A}_2$ such that $|\bar{a}_i| = |\bar{x}_i|$ and all the elements of $\bar{a}_i$ belong to $\mathfrak{A}_i$. Then it holds that:*

$$\mathfrak{A}_1 \circledast \mathfrak{A}_2 \models \varphi(\bar{a}_1, \bar{a}_2) \quad \Leftrightarrow \quad \mathfrak{A}_1 \underline{\uplus} \mathfrak{A}_2 \models \Xi(\varphi)(\bar{a}_1, \bar{a}_2).$$

Proposition 1 in conjunction with Remark 1 will be useful for us in transferring results about $\underline{\uplus}$ to similar results about $\circledast$.

3. Reduction sequences: We now recall the notions of reduction sequences and models for these from the literature. We mention that reduction sequences as we present them below are an adaptation of the special case of 2-reduction sequences from Harwath et al. (2015), and the adaptation follows the ideas in Grohe (2008).

Let $\mathcal{L}$ be any one of the logics defined earlier in this section. Given numbers $r \geq 0$ and $j \in [2]$, and an index set I and an element $i \in I$, let $\psi_{i,j}$ be an $\mathcal{L}$ formula over a vocabulary τ, whose free variables are contained in a (finite) sequence $\bar{x}_j$ of variables. We assume $\bar{x}_1$ and $\bar{x}_2$ to be disjoint. Let $\Delta_j(\bar{x}_j) = (\psi_{i,j})_{i \in I}$. Let $X_{i,j}$ be a propositional variable, $\mathcal{X} = \{X_{i,j} \mid i \in I, j \in [2]\}$, and β be a propositional formula over the variables of $\mathcal{X}$. We call the triple $D(\bar{x}_1, \bar{x}_2) = (\Delta_1(\bar{x}_1), \Delta_2(\bar{x}_2), \beta)$ an $\mathcal{L}$ *reduction sequence over* τ. If τ is clear from context, then we call $D(\bar{x}_1, \bar{x}_2)$ simply an $\mathcal{L}$ reduction sequence.

Let $\mathfrak{A}_1$ and $\mathfrak{A}_2$ be τ-structures that are disjoint, and for each $j \in [2]$, let $\bar{a}_j$ be a (finite) tuple of elements from $\mathfrak{A}_j$. We say that $(\mathfrak{A}_1, \mathfrak{A}_2, \bar{a}_1, \bar{a}_2)$ is a model of the $\mathcal{L}$ reduction sequence $D(\bar{x}_1, \bar{x}_2) = (\Delta_1(\bar{x}_1), \Delta_2(\bar{x}_2), \beta)$, denoted $(\mathfrak{A}_1, \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models D(\bar{x}_1, \bar{x}_2)$, if $|\bar{a}_j| = |\bar{x}_j|$ for all $j \in [2]$, and there exists an assignment $\zeta : \mathcal{X} \to \{0, 1\}$ such that $\zeta \models \beta$ and for all $i \in I$ and $j \in [2]$,

$$\zeta(X_{i,j}) = 1 \quad \Leftrightarrow \quad (\mathfrak{A}_j, \bar{a}_j) \models \psi_{i,j}(\bar{x}_j).$$

Let $\circledast$ be a quantifier-free sum-like binary operation on τ-structures, and let $\otimes$ be one of the operations $\circledast$ or $\underline{\uplus}$. Let $\mathcal{L}$ be a logic over τ as before, and let $\mathsf{L}_{\otimes}$ be a logic over τ, respectively over $\underline{\tau}$, if $\otimes = \circledast$, respectively $\otimes = \underline{\uplus}$. Given an $\mathsf{L}_{\otimes}$ formula $\varphi(\bar{x}_1, \bar{x}_2)$, we say that an $\mathcal{L}$ reduction sequence $D(\bar{x}_1, \bar{x}_2)$ is a *Feferman–Vaught decomposition of* $\varphi(\bar{x}_1, \bar{x}_2)$ *over* $\otimes$ if it holds that for any two τ-structures $\mathfrak{A}_1$ and $\mathfrak{A}_2$, if $\bar{a}_j$ is a $|\bar{x}_j|$-tuple from $\mathfrak{A}_j$ for $j \in [2]$, then

$$(\mathfrak{A}_1 \otimes \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models \varphi(\bar{x}_1, \bar{x}_2) \quad \Leftrightarrow \quad (\mathfrak{A}_1, \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models D(\bar{x}_1, \bar{x}_2)$$

We also say that φ β-*factorizes* over $\circledast$ into its factors $\Delta_1(\bar{x}_1)$ and $\Delta_2(\bar{x}_2)$.

A function that will be important for us in this paper is $\mathsf{tower} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ which is the function defined as follows: $\mathsf{tower}(0, k) = k$, and $\mathsf{tower}(n, k) = 2^{\mathsf{tower}(n-1,k)}$. Thus $\mathsf{tower}(n, k)$ is an n-fold exponential function of k. Recall that the size of a formula φ is denoted $|\varphi|$. By extension, the size of a (τ, σ)-translation scheme Ξ, defined as the sum of the sizes of the formulae appearing in Ξ, is denoted $|\Xi|$; the size of a reduction sequence D is defined analogously and is denoted $|D|$. Finally, we abbreviate in the standard way the expressions ‘if and only if’ as ‘iff’, ‘such that’ as ‘s.t.’, and ‘respectively’ as ‘resp.’.

3 Decompositions over the Annotated Disjoint Union of Structures

In this section, we show that the logics $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ admit Feferman–Vaught decompositions for their formulae over the annotated disjoint union. The formulae in the decompositions have the same rank and number of quantifier alternations as the input formulae, and are obtained inductively in keeping with the inductive structure of the latter. While quantifications in the input cause only a linear increase in the sizes of the formulae in the decompositions, every conjunction and disjunction of (already decomposed) non-atomic formulae in the input causes an exponential blow-up. This gives the n-fold exponential upper bound in the result below. Theorem 2 is the technical core of this paper, which enables proving its analogue for general quantifier-free sum-like operations in Theorem 3 of Sect. 4. Recall that for a vocabulary τ, the vocabulary $\underline{\tau} = \tau \cup \{P\}$ for a unary predicate P not in τ is the vocabulary of the annotated disjoint unions of τ-structures.

Theorem 2 *Let $\mathcal{L}$ be one of the logics $\mathrm{T}\Sigma_n[m]$ or $\mathrm{T}\Pi_n[m]$ for $m, n \in \mathbb{N}$. Let τ be a vocabulary. Then for every $\mathcal{L}$ formula $\varphi(\bar{x}_1, \bar{x}_2)$ over $\underline{\tau}$, there is an $\mathcal{L}$ reduction sequence $D(\bar{x}_1, \bar{x}_2)$ over τ such that:*

1. *$D(\bar{x}_1, \bar{x}_2)$ is a Feferman–Vaught decomposition of $\varphi(\bar{x}_1, \bar{x}_2)$ over the annotated disjoint union operation.*
2. *The size of $D(\bar{x}_1, \bar{x}_2)$ is $\mathsf{tower}(n, O((n+1) \cdot |\varphi|))$, and $D(\bar{x}_1, \bar{x}_2)$ can be computed from $\varphi(\bar{x}_1, \bar{x}_2)$ in time $\mathsf{tower}(n, O((n+1) \cdot |\varphi|^2))$.*

Proof We prove the theorem by showing the stronger statement $\mathcal{P}(n, \mathcal{L})$. Below, c is a suitably large constant that will be specified shortly.
$\mathcal{P}(n, \mathcal{L}) \equiv$ "For each $\mathcal{L}$ formula $\varphi(\bar{x}_1, \bar{x}_2)$ over $\underline{\tau}$, there is an $\mathcal{L}$ reduction sequence $D(\bar{x}_1, \bar{x}_2) = (\Delta_1(\bar{x}_1), \Delta_2(\bar{x}_2), \beta)$ over τ that is a Feferman–Vaught decomposition of $\varphi(\bar{x}_1, \bar{x}_2)$ over the annotated disjoint union operation, and is such that:

(i) β contains no negations, and if $\mathcal{X}$ is the set of propositional variables corresponding to the formulae appearing in $\Delta_1(\bar{x}_1)$ and $\Delta_2(\bar{x}_2)$ (c.f. Sect. 2), then every variable in $\mathcal{X}$ appears in β exactly once.
(ii) if $n > 0$ and $\varphi(\bar{x}_1, \bar{x}_2) \in \mathrm{T}\Sigma_n$ ($\mathrm{T}\Pi_n$), then the formula β is a disjunction (conjunction) of conjuncts (disjuncts) that are each a conjunction (disjunction) of exactly two positive literals, one a variable corresponding to a formula in Δ_1 and the other a variable corresponding to a formula in Δ_2;
(iii) the size of $D(\bar{x}_1, \bar{x}_2)$ at most $\mathsf{tower}(n, c \cdot (n+1) \cdot |\varphi|)$; and
(iv) $D(\bar{x}_1, \bar{x}_2)$ can be computed in time at most $\mathsf{tower}(n, c \cdot (n+1) \cdot |\varphi|^2)$."

Our proof goes via showing $\mathcal{P}(n, \mathrm{T}\Sigma_n[m])$ and $\mathcal{P}(n, \mathrm{T}\Pi_n[m])$ by simultaneous induction as n increases, for all $m \geq 0$. The analysis in the proof builds on the exposition in Grohe (2008).

A. Base case: The base case is when $n = 0$. Note that in this case $\mathrm{T}\Sigma_0[m] = \mathrm{T}\Pi_0[m] = \mathrm{T}\Sigma_0[0] = \mathrm{T}\Pi_0[0]$. We have the following sub-cases:

1. $\varphi(\bar{x}_1, \bar{x}_2) := A(\bar{z})$ where $A(\bar{z})$ is an atomic formula of the form $R(\bar{z})$ or $z_1 = z_2$ or the negations of these, for a predicate $R \in \tau$.
   (a) If $\bar{z}$ is a sub-tuple of $\bar{x}_1$, then define the reduction sequence $D(\bar{x}_1, \bar{x}_2) = (\Delta_1(\bar{x}_1), \Delta_2(\bar{x}_2), \beta)$ as such that $\Delta_1(\bar{x}_1) = (A(\bar{z}))$, $\Delta_2(\bar{x}_2) = (\mathsf{True})$ and $\beta = X_{1,1} \wedge X_{1,2}$.
   (b) If $\bar{z}$ is a sub-tuple of $\bar{x}_2$, then define the reduction sequence $D(\bar{x}_1, \bar{x}_2)$ such that $\Delta_1(\bar{x}_1) = (\mathsf{True})$, $\Delta_2(\bar{x}_2) = (A(\bar{z}))$ and $\beta = X_{1,1} \wedge X_{1,2}$.
   (c) If $\bar{z}$ is a neither a sub-tuple of $\bar{x}_1$ nor of $\bar{x}_2$, then the reduction sequence is $D(\bar{x}_1, \bar{x}_2)$ where $\Delta_1(\bar{x}_1) = \Delta_2(\bar{x}_2) = ()$ (= the empty tuple), and $\beta = \mathsf{False}$ if A does not contain negation, and $\beta = \mathsf{True}$ if A contains negation.

2. $\varphi(\bar{x}_1, \bar{x}_2) := A(z)$ where $A(z)$ is the atomic formula $P(z)$ or its negation. Then the reduction sequence is $D(\bar{x}_1, \bar{x}_2) = (\Delta_1(\bar{x}_1), \Delta_2(\bar{x}_2), \beta)$ where $\Delta_1(\bar{x}_1) = \Delta_2(\bar{x}_2) = ()$ and

- $\beta = \mathsf{True}$ if either z is a variable of $\bar{x}_1$ and $A(z) = P(z)$, or z is a variable of $\bar{x}_2$ and $A(z) = \neg P(z)$.
- $\beta = \mathsf{False}$ otherwise

3. $\varphi(\bar{x}_1, \bar{x}_2) := \varphi_1(\bar{x}_{1,1}, \bar{x}_{2,1}) \circledast \varphi_2(\bar{x}_{1,2}, \bar{x}_{2,2})$ where $\circledast \in \{\wedge, \vee\}$, φ_k is quantifier-free, and $\bar{x}_{j,k}$ is a sub-tuple of $\bar{x}_j$, for $j, k \in [2]$. Assume that there exist $\mathrm{T}\Sigma_0$ reduction sequences $D_k(\bar{x}_{1,k}, \bar{x}_{2,k}) = (\Delta_1^k(\bar{x}_{1,k}), \Delta_2^k(\bar{x}_{2,k}), \beta_k)$ that witness $\mathcal{P}(0, \mathrm{T}\Sigma_0)$ for $\varphi_k(\bar{x}_{1,k}, \bar{x}_{2,k})$, for $k \in [2]$. Then the desired reduction sequence for $\varphi(\bar{x}_1, \bar{x}_2)$ is $D(\bar{x}_1, \bar{x}_2) = (\Delta_1(\bar{x}_1), \Delta_2(\bar{x}_2), \beta)$ where $\Delta_j(\bar{x}_j) = \Delta_j^1(\bar{x}_{j,1}) \cdot \Delta_j^2(\bar{x}_{j,2})$ for $j \in [2]$, and $\beta = \beta_1 \circledast \beta_2$. Here $\cdot$ denotes concatenation of tuples.

With the above definitions of $D(\bar{x}_1, \bar{x}_2)$, one observes that parts (i) and (ii) of $\mathcal{P}(0, \mathrm{T}\Sigma_0)$ are indeed true. We show below that parts (iii) and (iv) of $\mathcal{P}(0, \mathrm{T}\Sigma_0)$ also hold to complete the base case analysis.

1. In cases (1) and (2), the size of $D(\bar{x}_1, \bar{x}_2)$ and the time taken to compute it are both at most some suitably large constant in all cases. We take this constant to be $c > 0$.
2. In case (3), assume as the structural induction hypothesis, that the size of $D_k(\bar{x}_{1,k}, \bar{x}_{2,k})$ is at most $\mathsf{tower}(0, c \cdot |\varphi_k|)$, and that the time taken to compute $D_k(\bar{x}_{1,k}, \bar{x}_{2,k})$ is at most $\mathsf{tower}(0, c \cdot |\varphi_k|^2)$ for $k \in [2]$. Then the size of $D(\bar{x}_1, \bar{x}_2)$ is

$$\begin{aligned} &= O(1) + \sum_{k \in [2]} \text{Size of } D_k(\bar{x}_{1,k}, \bar{x}_{2,k}) \\ &\leq O(1) + \sum_{k \in [2]} \mathsf{tower}(0, c \cdot |\varphi_k|) \\ &\leq \mathsf{tower}(0, c \cdot |\varphi|) \end{aligned}$$

since c is sufficiently large. The time taken to compute $D(\bar{x}_1, \bar{x}_2)$ is

$$\begin{aligned} &\leq \sum_{k \in [2]} \text{Time taken to compute } D_k(\bar{x}_{1,k}, \bar{x}_{2,k}) \ + \\ &\quad\ \text{Time taken to write } D(\bar{x}_1, \bar{x}_2) \\ &\leq \sum_{k \in [2]} \mathsf{tower}(0, c \cdot |\varphi_k|^2) \ + \ \sum_{k \in [2]} \text{Size of } D_k(\bar{x}_{1,k}, \bar{x}_{2,k}) \ + \ O(1) \\ &\leq \sum_{k \in [2]} \mathsf{tower}(0, c \cdot |\varphi_k|^2) \ + \ \sum_{k \in [2]} \mathsf{tower}(0, c \cdot |\varphi_k|) \ + \ O(1) \\ &\leq \mathsf{tower}(0, c \cdot |\varphi|^2) \end{aligned}$$

B. Induction: Assume as induction hypothesis, that $\mathcal{P}(n', \mathcal{L}')$ holds for $\mathcal{L}'$ that is one of the logics $\mathrm{T}\Sigma_{n'}[m']$ or $\mathrm{T}\Pi_{n'}[m']$ over $\underline{\tau}$, for all $n', m' \geq 0$ such that $n' < n$ for a given $n > 0$. We show below that $\mathcal{P}(n, \mathcal{L})$ holds for the case when $\mathcal{L} = \mathrm{T}\Sigma_n[m]$ for a given $m \geq 0$. The reasoning when $\mathcal{L} = \mathrm{T}\Pi_n[m]$ can be similarly done (by considering disjunctions in place of conjunctions and vice-versa, and universal quantifiers in place of existential quantifiers and vice-versa) to complete the induction.

We recall from Sect. 2 that $\mathrm{T}\Sigma_n[m] = \bigcup_{r \in \mathbb{N}} \mathrm{T}\Sigma_{n,r}[m]$. Our proof below goes via showing $\mathcal{P}(n, \mathrm{T}\Sigma_{n,r}[m])$ by a nested induction on r.

Nested base case: The base case is when $r = 0$. Then $\varphi(\bar{x}_1, \bar{x}_2)$ is given by $\varphi(\bar{x}_1, \bar{x}_2) := \bigwedge_{i \in I} \varphi_i(\bar{x}_{1,i}, \bar{x}_{2,i})$ where I is a finite index set, φ_i is a formula of $\mathrm{T}\Pi_{n_i}[m]$ over $\underline{\tau}$ for some $n_i < n$, and $\bar{x}_{l,i}$ is a sub-tuple of $\bar{x}_l$ for all $l \in [2]$ and $i \in I$. From the (outer) induction hypothesis above, let $D_i(\bar{x}_{1,i}, \bar{x}_{2,i}) = (\Delta_1^i(\bar{x}_{1,i}), \Delta_2^i(\bar{x}_{2,i}), \beta_i)$ be the $\mathrm{T}\Pi_{n_i}[m]$ reduction sequence over τ that witnesses $\mathcal{P}(n_i, \mathrm{T}\Pi_{n_i}[m])$ for $\varphi_i(\bar{x}_{1,i}, \bar{x}_{2,i})$, for $i \in I$. We have two cases as below depending on whether $n = 1$ or $n > 1$. We analyse the latter first, then the former.

(**a**) $\mathbf{n} > \mathbf{1}$: Here β_i is of the form $\bigwedge_{j \in J_i}(X_1^{(i,j)} \vee X_2^{(i,j)})$ where J_i is a finite index set, and if $X_l^{(i,j)}$ corresponds to the formula $\psi_l^{(i,j)}$, then $\Delta_l^i(\bar{x}_{l,i}) = (\psi_l^{(i,j)}(\bar{x}_{l,i}))_{j \in J_i}$ for $l \in [2]$. The sets $\mathcal{X}_i = \{X_l^{(i,j)} \mid j \in J_i, l \in [2]\}$ are all disjoint.

Let $J = \{(i, j) \mid i \in I, j \in J_i\}$. Consider the formula $\beta' = \bigwedge_{i \in I} \beta_i$. Writing this formula as an OR of ANDs, we have that

$$\beta' \Leftrightarrow \beta'' := \bigvee_{f \in \{1,2\}^J} C_f \quad \text{where} \quad C_f := \bigwedge_{k \in S_{f,1}} X_1^k \wedge \bigwedge_{k \in S_{f,2}} X_2^k \tag{1}$$

Above $\{1, 2\}^J$ denotes the set of all functions $f : J \to \{1, 2\}$, the set J is partitioned into $S_{f,1}$ and $S_{f,2}$ (allowing empty parts in a slight abuse of the usual notion of a partition), where $S_{f,l} = \{p \in J \mid f(p) = l\}$ for $l \in [2]$. We now define the formulae $\xi_{f,l}(\bar{x}_l)$ for $f \in \{1, 2\}^J$ and $l \in [2]$ as below.

$$\xi_{f,l}(\bar{x}_l) := \bigwedge_{\substack{k \in S_{f,l} \\ k=(i,j)}} \psi_l^k(\bar{x}_{l,i}) \tag{2}$$

In the event that $S_{f,l} = \emptyset$, we put $\xi_{f,l}(\bar{x}_l) := \mathsf{True}$. Let $Y_{f,l}$ be a new propositional variable for $f \in \{1, 2\}^J$ and $l \in [2]$. Consider the reduction sequence $D(\bar{x}_1, \bar{x}_2) = (\Delta_1(\bar{x}_1), \Delta_2(\bar{x}_2), \beta)$ where for $l \in [2]$

$$\Delta_l(\bar{x}_l) = (\xi_{f,l}(\bar{x}_l))_{f \in \{1,2\}^J} \quad ; \quad \beta := \bigvee_{f \in \{1,2\}^J} (Y_{f,1} \wedge Y_{f,2}) \tag{3}$$

We claim that $D(\bar{x}_1, \bar{x}_2)$ witnesses $\mathcal{P}(n, \mathrm{T}\Sigma_{n,0}[m])$ for $\varphi(\bar{x}_1, \bar{x}_2)$.

1. Firstly, the formula $\psi_l^k(\bar{x}_{l,i})$ in (2) belongs to $\mathrm{T}\Pi_{n_i}[m]$ over τ by induction hypothesis. Since $\xi_{f,l}(\bar{x}_l)$ is a conjunction of the ψ_l^ks, it follows that $\xi_{f,l}(\bar{x}_l)$ is in $\mathrm{T}\Sigma_{n,0}[m]$ over τ. Then $D(\bar{x}_1, \bar{x}_2)$ is a $\mathrm{T}\Sigma_{n,0}[m]$-reduction sequence over τ. Further, $D(\bar{x}_1, \bar{x}_2)$ is a Feferman–Vaught decomposition for $\varphi(\bar{x}_1, \bar{x}_2)$ as seen via the following equivalences. Below, $\bar{a}_l$ is a tuple from $\mathfrak{A}_l$ of length $|\bar{x}_l|$ for all $l \in [2]$; $\bar{a}_{l,i}$ is the sub-tuple of $\bar{a}_l$ corresponding to $\bar{x}_{l,i}$; the third equivalence is by the induction hypothesis; $\mathcal{X}_i = \{X_l^{(i,j)} \mid j \in J_i, l \in [2]\}$ for $i \in I$; and $\mathcal{Y} = \{Y_{f,l} \mid f \in \{1, 2\}^J, l \in [2]\}$.

$$
\begin{array}{cc}
 & (\mathfrak{A}_1 \uplus \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models \varphi(\bar{x}_1, \bar{x}_2) \\
\Leftrightarrow & (\mathfrak{A}_1 \uplus \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models \bigwedge_{i \in I} \varphi_i(\bar{x}_{1,i}, \bar{x}_{2,i}) \\
\Leftrightarrow & \text{For all } i \in I, \text{ it holds that } (\mathfrak{A}_1 \uplus \mathfrak{A}_2, \bar{a}_{1,i}, \bar{a}_{2,i}) \models \varphi_i(\bar{x}_{1,i}, \bar{x}_{2,i}) \\
\Leftrightarrow & \text{For all } i \in I, \text{ it holds that } (\mathfrak{A}_1, \mathfrak{A}_2, \bar{a}_{1,i}, \bar{a}_{2,i}) \models D_i(\bar{x}_{1,i}, \bar{x}_{2,i}) \\
\Leftrightarrow & \text{For all } i \in I, \text{ there exists } \zeta_i : \mathcal{X}_i \to \{0, 1\} \text{ s.t. } \zeta_i \models \beta_i \text{ and} \\
 & \zeta_i(X_l^{(i,j)}) = 1 \text{ iff } (\mathfrak{A}_l, \bar{a}_{l,i}) \models \psi_l^{(i,j)}(\bar{x}_{l,i}) \quad \text{for all } j \in J_i \text{ and } l \in [2] \\
\Leftrightarrow & \text{For all } i \in I, \text{ there exists } \zeta_i : \mathcal{X}_i \to \{0, 1\} \text{ s.t. for all } j \in J_i, \text{ there exists} \\
 & l \in [2] \text{ s.t. (recalling the form of } \beta_i) \\
 & \zeta_i \models X_l^{(i,j)} \quad \text{and} \quad \zeta_i(X_l^{(i,j)}) = 1 \text{ iff } (\mathfrak{A}_l, \bar{a}_{l,i}) \models \psi_l^{(i,j)}(\bar{x}_{l,i}) \\
\Leftrightarrow & \text{There exists } \zeta : \mathcal{Y} \to \{0, 1\} \text{ s.t. } \zeta \models Y_{f,1} \wedge Y_{f,2} \text{ for some } f \in \{1, 2\}^J, \text{ and} \\
 & \zeta(Y_{f,l}) = 1 \text{ iff } (\mathfrak{A}_l, \bar{a}_l) \models \xi_{f,l}(\bar{x}_l) \quad \text{for all } l \in [2] \\
\Leftrightarrow & \text{There exists } \zeta : \mathcal{Y} \to \{0, 1\} \text{ s.t. } \zeta \models \beta \text{ and} \\
 & \zeta(Y_{f,l}) = 1 \text{ iff } (\mathfrak{A}_l, \bar{a}_l) \models \xi_{f,l}(\bar{x}_l) \quad \text{for all } f \in \{1, 2\}^J \text{ and } l \in [2] \\
\Leftrightarrow & (\mathfrak{A}_1, \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models D(\bar{x}_1, \bar{x}_2)
\end{array}
$$

2. The formula β is indeed of the form required by parts (i) and (ii) of $\mathcal{P}(n, \mathrm{T}\Sigma_{n,0}[m])$.
3. Towards showing part (iii) of $\mathcal{P}(n, \mathrm{T}\Sigma_{n,0}[m])$, we first observe that for $f \in \{1, 2\}^J$, every pair $(\xi_{f,1}, \xi_{f,2})$ corresponds to a unique subset of the set $\{\psi_l^k \mid k \in J, l \in [2]\}$; the latter set is the same as $\bigcup_{i \in I, l \in [2]} \Delta_l^i$ viewing Δ_l^i as the set (instead of as a sequence) of its constituent formulas. Then the size of the pair $(\xi_{f,1}, \xi_{f,2})$ is at most the size of $\bigcup_{i \in I, l \in [2]} \Delta_l^i$ which is at most $\sum_{i \in I} |D_i|$. (Recall that $|D_i|$ denotes the size of D_i, namely the sum of the sizes of the formulae appearing in it.) Also since the size of J_i is at most the size of D_i, the size of J, which is $\sum_{i \in I} |J_i|$, is at most $\sum_{i \in I} |D_i|$. Using these observations and the induction hypothesis, we have the following.

$$
\begin{aligned}
\sum_{i \in I} |D_i| &\leq \sum_{i \in I} \mathsf{tower}(n_i, c \cdot (n_i + 1) \cdot |\varphi_i|) \\
&\leq \sum_{i \in I} \mathsf{tower}(n - 1, c \cdot n \cdot |\varphi_i|) \\
&\leq \mathsf{tower}(n - 1, c \cdot n \cdot \sum_{i \in I} |\varphi_i|) \\
&\leq \mathsf{tower}(n - 1, c \cdot n \cdot |\varphi|) \\
|\Delta_1(\bar{x}_1)| + |\Delta_2(\bar{x}_2)| &\leq \sum_{f \in \{1,2\}^J} \text{Size of the pair } (\xi_{f,1}, \xi_{f,2}) \\
&\leq |\{1, 2\}^J| \cdot \sum_{i \in I} |D_i|
\end{aligned}
$$

$$
\begin{aligned}
& \leq 2^{|J|} \cdot \sum_{i \in I} |D_i| \\
& \leq 2^{\sum_{i \in I} |D_i|} \cdot \sum_{i \in I} |D_i| \\
|\beta| & \leq \sum_{f \in \{1,2\}^J} \cdot \text{Size of } (Y_{f,1} \wedge Y_{f,2}) \\
& \leq 3 \cdot 2^{|J|} \\
& \leq 3 \cdot 2^{\sum_{i \in I} |D_i|} \\
& \leq 3 \cdot 2^{\sum_{i \in I} |D_i|} \cdot \sum_{i \in I} |D_i| \\
\therefore \quad |D(\bar{x}_1, \bar{x}_2)| & \leq |\Delta_1(\bar{x}_1)| + |\Delta_2(\bar{x}_2)| + |\beta| \\
& \leq 4 \cdot 2^{\sum_{i \in I} |D_i|} \cdot \sum_{i \in I} |D_i| \\
& \leq 4 \cdot 2^{\mathsf{tower}(n-1, c \cdot n \cdot |\varphi|)} \cdot \mathsf{tower}(n-1, c \cdot n \cdot |\varphi|) \\
& \leq 4 \cdot \mathsf{tower}(n, c \cdot n \cdot |\varphi|) \cdot \mathsf{tower}(n-1, c \cdot n \cdot |\varphi|) \\
& \leq \mathsf{tower}(n, c \cdot (n+1) \cdot |\varphi|)
\end{aligned}
$$

4. Finally, to see part (iv) of $\mathcal{P}(n, \mathrm{T}\Sigma_{n,0}[m])$, we observe that there is no need to explicitly generate β''; we can directly write out the pair $(\xi_{f,1}, \xi_{f,2})$ by performing $|\{1,2\}^J|$ many passes over the formulae of D_i for $i \in I$, and extracting out the relevant ψ_j^k's in each pass. That would give us the sequences $\Delta_l(\bar{x}_l)$ for $l \in [2]$. We next directly write out β by introducing the new variables $Y_{f,l}$ since we already know J by a single pass over all the D_is. We assume that it takes unit time to introduce any of the variables $Y_{f,l}$. Then the total time taken to generate $D(\bar{x}_1, \bar{x}_2)$ is

$$
\begin{aligned}
\leq \; & \sum_{i \in I} \text{Time taken to compute } D_i \; + \\
& \text{Time taken for } |\{1,2\}^J| \text{ passes over the } D_i\text{s and obtaining} \\
& \Delta_1(\bar{x}_1) \text{ and } \Delta_2(\bar{x}_2) \; + \\
& \text{Time taken to write } \beta \\
\leq \; & \sum_{i \in I} \mathsf{tower}(n_i, c \cdot (n_i + 1) \cdot |\varphi_i|^2) \; + \\
& d \cdot |\{1,2\}^J| \cdot \sum_{i \in I} |D_i| \; + \qquad \text{(for some constant } d > 0) \\
& d \cdot 3 \cdot |\{1,2\}^J| \qquad \Big(\because \; \beta := \bigvee_{f \in \{1,2\}^J} (Y_{f,1} \wedge Y_{f,2})\Big) \\
\leq \; & \sum_{i \in I} \mathsf{tower}(n-1, c \cdot n \cdot |\varphi_i|^2) \; + \; d \cdot 2^{|J|} \cdot \sum_{i \in I} |D_i| \; + \; 3 \cdot d \cdot 2^{|J|}
\end{aligned}
$$

$$
\begin{aligned}
&\leq \ \mathsf{tower}(n-1, c \cdot n \cdot \sum_{i \in I} |\varphi_i|^2) \\
&\quad + \ 4 \cdot d \cdot 2^{\sum_{i \in I} |D_i|} \sum_{i \in I} |D_i| \qquad \big(\because\ |J| \leq \sum_{i \in I} |D_i|\big) \\
&\leq \ \mathsf{tower}(n-1, c \cdot n \cdot |\varphi|^2) \ + \ 4 \cdot d \cdot \mathsf{tower}(n, c \cdot (n+1) \cdot |\varphi|) \\
&\leq \ \mathsf{tower}(n, c \cdot (n+1) \cdot |\varphi|^2) \qquad \text{(since } c \text{ is sufficiently large)}
\end{aligned}
$$

This establishes the nested base case for the sub-case of $n > 1$.

(b) $\mathbf{n = 1}$: Recall that $\varphi(\bar{x}_1, \bar{x}_2) := \bigwedge_{i \in I} \varphi_i(\bar{x}_{1,i}, \bar{x}_{2,i})$ where φ_i is a formula of $\mathrm{T}\Pi_{n_i}[m]$ over $\underline{\tau}$ for $n_i < n$, and $\bar{x}_{l,i}$ is a sub-tuple of $\bar{x}_l$ for $l \in [2]$ and $i \in I$. Also $D_i(\bar{x}_{1,i}, \bar{x}_{2,i}) = (\Delta_1^i(\bar{x}_{1,i}), \Delta_2^i(\bar{x}_{2,i}), \beta_i)$ is the $\mathrm{T}\Pi_{n_i}[m]$ reduction sequence over τ that witnesses $\mathcal{P}(n_i, \mathrm{T}\Pi_{n_i}[m])$ for $\varphi_i(\bar{x}_{1,i}, \bar{x}_{2,i})$, for $i \in I$. The set of variables appearing in β_i is $\mathcal{X}_i = \{X_l^{(i,j)} \mid j \in J_i, l \in [2]\}$ for some finite set J_i, and $\Delta_l^i(\bar{x}_{l,i}) = (\psi_l^{(i,j)}(\bar{x}_{l,i}))_{j \in J_i}$ for all $i \in I$, and $l \in [2]$. The sets $\mathcal{X}_i$ are all disjoint.

Now given that $n = 1$, we can make a few observations. Firstly, we see that $n_i = 0$ for all $i \in I$; so that φ_i and all the formulae of $\Delta_l^i(\bar{x}_{l,i})$ are quantifier-free for all $l \in [2]$. Also β_i constructed inductively for φ_i need not be structured as an AND of ORs as we had in the case when $n > 1$ (recall part (ii) of $\mathcal{P}(n_i, \mathrm{T}\Pi_{n_i}[m])$). Therefore, as opposed to the $n > 1$ case where J_i was the set indexing the conjuncts in β_i and hence enabled enumerating the formulae of $\Delta_l^i(\bar{x}_{l,i})$, here J_i is just some finite index set enumerating formulae of $\Delta_l^i(\bar{x}_{l,i})$.

Given that β_i is potentially an arbitrary propositional formula, one approach to dealing with the present case is to process β_i to bring it in the AND of ORs form as in the $n > 1$ case for all i, construct β' as the conjunction of the processed β_i's, and then construct $D(\bar{x}_1, \bar{x}_2)$ from β' as in the $n > 1$ case. While the last of these steps is essentially how we construct $D(\bar{x}_1, \bar{x}_2)$ from β' even in this case, the processing of β_i's as mentioned runs the risk, for our computational result, of introducing an extra exponential in the time taken to compute $D(\bar{x}_1, \bar{x}_2)$ as well as the size of $D(\bar{x}_1, \bar{x}_2)$, since the AND to OR conversion would be followed by an OR to AND conversion of the formula β'.

To avoid this additional exponential, we construct the formula β' directly as $\beta' := \bigwedge_{i \in I} \beta_i$. Writing this formula now as an OR of ANDs, we have that

$$
\beta' \leftrightarrow \beta'' := \bigvee_{p \in [N]} C_p \quad \text{where} \quad C_p := \Big(\bigwedge_{i \in I} \bigwedge_{j \in S_{p,1}^i} X_1^{(i,j)}\Big) \wedge \Big(\bigwedge_{i \in I} \bigwedge_{j \in S_{p,2}^i} X_2^{(i,j)}\Big) \tag{4}
$$

Above N denotes the number of conjuncts in β'', the sets $S_{p,l}^i \subseteq J_i$ (which could be overlapping and some empty) are such that $J_i = \bigcup_{p \in [N], l \in [2]} S_{p,l}^i$. We now define the formulae $\xi_{p,l}(\bar{x}_l)$ for $p \in [N]$ and $l \in [2]$ as below.

$$\xi_{p,l}(\bar{x}_l) := \bigwedge_{i \in I} \bigwedge_{j \in S^i_{p,l}} \psi_l^{(i,j)}(\bar{x}_{l,i}) \tag{5}$$

In the event that $\bigcup_{i \in I} S^i_{p,l} = \emptyset$, we put $\xi_{p,l}(\bar{x}_l) := \mathsf{True}$. Let $Y_{p,l}$ be a new propositional variable for $p \in [N]$ and $l \in [2]$. Consider the reduction sequence $D(\bar{x}_1, \bar{x}_2) = (\Delta_1(\bar{x}_1), \Delta_2(\bar{x}_2), \beta)$ where

$$\Delta_l(\bar{x}_l) = (\xi_{p,l}(\bar{x}_l))_{p \in [N]} \quad ; \quad \beta := \bigvee_{p \in [N]} (Y_{p,1} \wedge Y_{p,2}) \tag{6}$$

We claim that $D(\bar{x}_1, \bar{x}_2)$ witnesses $\mathcal{P}(n, \mathrm{T}\Sigma_{n,0}[0])$ for $\varphi(\bar{x}_1, \bar{x}_2)$.

1. Each of the formulae $\xi_{p,l}$ is a conjunction of quantifier-free formulae over τ, and hence belongs to $\mathrm{T}\Sigma_{n,0}[0]$ over τ (since $n = 1$); then $D(\bar{x}_1, \bar{x}_2)$ is a $\mathrm{T}\Sigma_{n,0}[0]$ reduction sequence over τ. That $D(\bar{x}_1, \bar{x}_2)$ is a Feferman–Vaught decomposition of $\varphi(\bar{x}_1, \bar{x}_2)$ can be shown analogously as in the previous sub-case of $n > 1$.
2. The formula β has no negations, contains exactly one occurrence of the propositional variable corresponding to any formula of $\Delta_1(\bar{x}_1)$ or $\Delta_2(\bar{x}_2)$, and is a finite OR of conjuncts of the form required by $\mathcal{P}(n, \mathrm{T}\Sigma_{n,0}[0])$; then parts (i) and (ii) of the latter are true.
3. To show part (iii) of $\mathcal{P}(n, \mathrm{T}\Sigma_{n,0}[0])$, we do our computations analogously as done in the $n > 1$ case. We observe that every pair $(\xi_{p,1}, \xi_{p,2})$ corresponds to a unique subset of the set $\bigcup_{i \in I, l \in [2]} \Delta^i_l$, so that the size of $(\xi_{p,1}, \xi_{p,2})$ is at most $\sum_{i \in I} |D_i|$. Also N is at most $2^{|\mathcal{X}|}$ where $\mathcal{X} = \bigcup_{i \in I} \mathcal{X}_i$ and the size of each $\mathcal{X}_i$ is at most $|D_i|$. Using these observations and the induction hypothesis, and nearly the same calculations as in the $n > 1$ case, we have the following.

$$\sum_{i \in I} |D_i| \leq \mathsf{tower}(0, c \cdot |\varphi|) \qquad \sum_{l \in [2]} |\Delta_l(\bar{x}_l)| \leq 2^{\sum_{i \in I} |D_i|} \cdot \sum_{i \in I} |D_i|$$

$$4 \cdot 2^{\sum_{i \in I} |D_i|} \cdot \sum_{i \in I} |D_i| \leq \mathsf{tower}(1, c \cdot 2 \cdot |\varphi|) \qquad |\beta| \leq 3 \cdot 2^{\sum_{i \in I} |D_i|} \cdot \sum_{i \in I} |D_i|$$

$$\therefore \quad |D(\bar{x}_1, \bar{x}_2)| \leq \mathsf{tower}(1, c \cdot 2 \cdot |\varphi|)$$

4. Finally, to show part (iv) of $\mathcal{P}(n, \mathrm{T}\Sigma_{n,0}[0])$, we observe that as opposed to the $n > 1$ case, we would need to generate β'' to be able to know the number N and the individual conjuncts C_p. The time taken to do this is (singly) exponential in the sum of the sizes of the β_is, which in turn is at most exponential in the sum of the sizes of the D_is. Once β'' is obtained, generating each pair $(\xi_{p,1}, \xi_{p,2})$ takes a single pass over all the D_is taken together. That would give us the sequences $\Delta_l(\bar{x}_l)$ for $l \in [2]$. Finally we directly write out β by introducing the new variables $Y_{p,l}$. We assume it takes unit time to introduce each variable $Y_{p,l}$. Recalling that $N \leq 2^{\sum_{i \in I} |D_i|}$, the total time taken to generate $D(\bar{x}_1, \bar{x}_2)$ is

$$\leq \sum_{i \in I} \text{Time taken to compute } D_i \; + \; \text{Time taken to obtain } \beta'' \; +$$

$$
\begin{aligned}
&\text{Time taken to otain } \Delta_1(\bar{x}_1) \text{ and } \Delta_2(\bar{x}_2) \;+\; \text{Time taken to write } \beta \\
&\leq \sum_{i \in I} \mathsf{tower}(0, c \cdot |\varphi_i|^2) \;+\; d \cdot 2^{\sum_{i \in I} |D_i|} \;+\; \quad \text{(for some constant } d > 0) \\
&\quad d \cdot N \cdot \sum_{i \in I} |D_i| \;+\; d \cdot 3 \cdot N \qquad \left(\because \;\; \beta := \bigvee_{p \in [N]} (Y_{p,1} \wedge Y_{p,2}) \right) \\
&\leq \mathsf{tower}(0, c \cdot \sum_{i \in I} |\varphi_i|^2) \;+\; 6 \cdot d \cdot 2^{\sum_{i \in I} |D_i|} \cdot \sum_{i \in I} |D_i| \\
&\leq \mathsf{tower}(0, c \cdot \sum_{i \in I} |\varphi_i|^2) \;+\; 6 \cdot d \cdot \mathsf{tower}(1, c \cdot 2 \cdot |\varphi|) \\
&\leq \mathsf{tower}(1, c \cdot 2 \cdot |\varphi|^2) \quad \text{(since } c \text{ is sufficiently large)}
\end{aligned}
$$

This completes the analysis of the sub-case $n = 1$ of the nested base case, and thus establishes the latter in all cases.

Nested induction: Assume as the nested induction hypothesis that $\mathcal{P}(n, \mathrm{T}\Sigma_{n,r'}[m'])$ holds for all r' such that $0 \leq r' < r$ for a given $r \geq 0$ and all m' such that $0 \leq m' \leq m$, where m and n are as in the outer induction step. Consider a formula $\varphi(\bar{x}_1, \bar{x}_2)$ of $\mathrm{T}\Sigma_{n,r}[m]$ over $\underline{\tau}$. The formula has the form $\varphi(\bar{x}_1, \bar{x}_2) := \exists z \varphi_1(\bar{x}_1, \bar{x}_2, z)$ where φ_1 is a formula of $\mathrm{T}\Sigma_{n,r-1}[m-1]$ over $\underline{\tau}$. We observe that the free variables of φ_1 can be seen as being amongst the tuple $\bar{y}_1 \cdot \bar{y}_2$ where either $\bar{y}_1 = \bar{x}_1 \cdot z$ and $\bar{y}_2 = \bar{x}_2$, or $\bar{y}_1 = \bar{x}_1$ and $\bar{y}_2 = \bar{x}_2 \cdot z$. Corresponding to each of these views, we have by the nested induction hypothesis that there exist $\mathrm{T}\Sigma_{n,r-1}[m-1]$ reduction sequences $D_1(\bar{x}_1 \cdot z, \bar{x}_2) = (\Delta_1^1(\bar{x}_1 \cdot z), \Delta_2^1(\bar{x}_2), \beta_1)$ and $D_2(\bar{x}_1, \bar{x}_2 \cdot z) = (\Delta_1^2(\bar{x}_1), \Delta_2^2(\bar{x}_2 \cdot z), \beta_2)$ over τ witnessing $\mathcal{P}(n, \mathrm{T}\Sigma_{n,r-1}[m-1])$ resp. for $\varphi_1(\bar{x}_1 \cdot z, \bar{x}_2)$ and $\varphi_1(\bar{x}_1, \bar{x}_2 \cdot z)$.

Let $\Delta_1^1(\bar{x}_1 \cdot z) = (\psi_1^{(i,1)}(\bar{x}_1 \cdot z))_{i \in I_1}$, $\Delta_2^1 = (\psi_2^{(i,1)}(\bar{x}_2))_{i \in I_1}$, $\Delta_1^2 = (\psi_1^{(i,2)}(\bar{x}_1))_{i \in I_2}$, and $\Delta_2^2 = (\psi_2^{(i,2)}(\bar{x}_2 \cdot z))_{i \in I_2}$ where I_1 and I_2 are finite index sets. Let $\beta_j := \bigvee_{i \in I_j} (X_1^{(i,j)} \wedge X_2^{(i,j)})$ – observe that by the nested induction hypothesis this is the form of β_j – for $j \in [2]$, where $X_l^{(i,j)}$ corresponds to the formula $\psi_l^{(i,j)}$ for $i \in I_j, l \in [2]$.

We now define the formulae $\xi_l^{(i,j)}(\bar{x}_j)$ for $j, l \in [2]$ and $i \in I_j$ as below.

$$
\begin{array}{ll}
\xi_1^{(i,1)}(\bar{x}_1) := \exists z \psi_1^{(i,1)}(\bar{x}_1, z) & \xi_2^{(i,1)}(\bar{x}_2) := \psi_2^{(i,1)}(\bar{x}_2) \\
\xi_1^{(i,2)}(\bar{x}_1) := \psi_1^{(i,2)}(\bar{x}_1) & \xi_2^{(i,2)}(\bar{x}_2) := \exists z \psi_2^{(i,2)}(\bar{x}_2, z)
\end{array}
\tag{7}
$$

Let $Y_l^{(i,j)}$ be a new propositional variable for $j, l \in [2]$ and $i \in I_j$. Consider the reduction sequence $D(\bar{x}_1, \bar{x}_2) = (\Delta_1(\bar{x}_1), \Delta_2(\bar{x}_2), \beta)$ where for $l \in [2]$

$$
\Delta_l(\bar{x}_l) = (\xi_l^{(i,1)})_{i \in I_1} \cdot (\xi_l^{(i,2)})_{i \in I_2} \quad ; \quad \beta := \bigvee_{j \in [2]} \bigvee_{i \in I_j} (Y_1^{(i,j)} \wedge Y_2^{(i,j)}) \tag{8}
$$

We claim that $D(\bar{x}_1, \bar{x}_2)$ witnesses $\mathcal{P}(n, \mathrm{T}\Sigma_{n,r}[m])$ for $\varphi(\bar{x}_1, \bar{x}_2)$.

1. The formula φ_1 is in $\mathrm{T}\Sigma_{n,r-1}[m-1]$ over $\underline{\tau}$; so by the nested induction hypothesis, $\psi_l^{(i,j)}$ is a $\mathrm{T}\Sigma_{n,r-1}[m-1]$ formula over τ for all $l \in [2]$. Then $\xi_l^{(i,j)}$ is a formula

of $\mathrm{T}\Sigma_{n,r}[m]$ over τ. Hence $D(\bar{x}_1, \bar{x}_2)$ is a $\mathrm{T}\Sigma_{n,r}[m]$ reduction sequence over τ. Further, the reduction sequence $D(\bar{x}_1, \bar{x}_2)$ is a Feferman–Vaught decomposition for $\varphi(\bar{x}_1, \bar{x}_2)$, which we show using the following equivalences. Below, $\bar{a}_l$ is a tuple of $\mathfrak{A}_l$ of length $|\bar{x}_l|$ for all $l \in [2]$; b is an element of $\mathfrak{A}_1 \uplus \mathfrak{A}_2$; the third equivalence is by the induction hypothesis; $\mathcal{X}_j = \{X_l^{(i,j)} \mid i \in I_j, l \in [2]\}$ for $j \in [2]$; and $\mathcal{Y} = \{Y_l^{(i,j)} \mid j, l \in [2], i \in I_j\}$.

$$
\begin{aligned}
& (\mathfrak{A}_1 \uplus \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models \varphi(\bar{x}_1, \bar{x}_2) \\
\Leftrightarrow\ & (\mathfrak{A}_1 \uplus \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models \exists z \varphi_1(\bar{x}_1, \bar{x}_2, z) \\
\Leftrightarrow\ & (\mathfrak{A}_1 \uplus \mathfrak{A}_2, \bar{a}_1 \cdot b, \bar{a}_2) \models \varphi_1(\bar{x}_1 \cdot z, \bar{x}_2) \bigvee (\mathfrak{A}_1 \uplus \mathfrak{A}_2, \bar{a}_1, \bar{a}_2 \cdot b) \models \varphi_1(\bar{x}_1, \bar{x}_2 \cdot z) \\
\Leftrightarrow\ & (\mathfrak{A}_1, \mathfrak{A}_2, \bar{a}_1 \cdot b, \bar{a}_2) \models D_1(\bar{x}_1 \cdot z, \bar{x}_2) \bigvee (\mathfrak{A}_1, \mathfrak{A}_2, \bar{a}_1, \bar{a}_2 \cdot b) \models D_2(\bar{x}_1, \bar{x}_2 \cdot z) \\
\Leftrightarrow\ & \text{For some } j \in [2],\ \text{there exists } \zeta_j : \mathcal{X}_j \to \{0,1\} \text{ s.t. } \zeta_j \models \beta_j \text{ and} \\
& \text{if } j = 1,\ \text{then} \\
& \zeta_1(X_1^{(i,1)}) = 1 \text{ iff } (\mathfrak{A}_1, \bar{a}_1 \cdot b) \models \psi_1^{(i,1)}(\bar{x}_1 \cdot z) \quad \text{for all } i \in I_1 \\
& \zeta_1(X_2^{(i,1)}) = 1 \text{ iff } (\mathfrak{A}_2, \bar{a}_2) \models \psi_2^{(i,1)}(\bar{x}_2) \quad \text{for all } i \in I_1 \\
& \text{else} \\
& \zeta_2(X_1^{(i,2)}) = 1 \text{ iff } (\mathfrak{A}_1, \bar{a}_1) \models \psi_1^{(i,2)}(\bar{x}_1) \quad \text{for all } i \in I_2 \\
& \zeta_2(X_2^{(i,2)}) = 1 \text{ iff } (\mathfrak{A}_2, \bar{a}_2 \cdot b) \models \psi_2^{(i,2)}(\bar{x}_2 \cdot z) \quad \text{for all } i \in I_2 \\
\Leftrightarrow\ & \text{For some } j \in [2],\ \text{there exists } \zeta_j : \mathcal{X}_j \to \{0,1\} \text{ s.t. for some } i \in I_j, \\
& \zeta_j \models (X_1^{(i,j)} \wedge X_2^{(i,j)}) \text{ and} \\
& \text{if } j = 1,\ \text{then} \\
& \zeta_1(X_1^{(i,1)}) = 1 \text{ iff } (\mathfrak{A}_1, \bar{a}_1 \cdot b) \models \psi_1^{(i,1)}(\bar{x}_1 \cdot z) \quad \text{for all } i \in I_1 \\
& \zeta_1(X_2^{(i,1)}) = 1 \text{ iff } (\mathfrak{A}_2, \bar{a}_2) \models \psi_2^{(i,1)}(\bar{x}_2) \quad \text{for all } i \in I_1 \\
& \text{else} \\
& \zeta_1(X_1^{(i,2)}) = 1 \text{ iff } (\mathfrak{A}_1, \bar{a}_1) \models \psi_1^{(i,2)}(\bar{x}_1) \quad \text{for all } i \in I_2 \\
& \zeta_1(X_2^{(i,2)}) = 1 \text{ iff } (\mathfrak{A}_2, \bar{a}_2 \cdot b) \models \psi_2^{(i,2)}(\bar{x}_2 \cdot z) \quad \text{for all } i \in I_2 \\
\Leftrightarrow\ & \text{There exists } \zeta : \mathcal{Y} \to \{0,1\} \text{ s.t. } \zeta \models (Y_1^{(i,j)} \wedge Y_2^{(i,j)}) \text{ for some } j \in [2] \text{ and} \\
& \text{some } i \in I_j \text{ and} \\
& \zeta(Y_l^{(i,j)}) = 1 \text{ iff } (\mathfrak{A}_l, \bar{a}_l) \models \xi_l^{(i,j)}(\bar{x}_l) \quad \text{for all } l \in [2] \\
\Leftrightarrow\ & \text{There exists } \zeta : \mathcal{Y} \to \{0,1\} \text{ s.t. } \zeta \models \beta \text{ and} \\
& \zeta(Y_l^{(i,j)}) = 1 \text{ iff } (\mathfrak{A}_l, \bar{a}_l) \models \xi_l^{(i,j)}(\bar{x}_l) \quad \text{for all } i \in I_j \text{ and } j, l \in [2] \\
\Leftrightarrow\ & (\mathfrak{A}_1, \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models D(\bar{x}_1, \bar{x}_2)
\end{aligned}
$$

2. It is easy to verify that β is indeed as required by parts (i) and (ii) of $\mathcal{P}(n, \mathrm{T}\Sigma_{n,r}[m])$ for $\varphi(\bar{x}_1, \bar{x}_2)$.
3. For part (iii) of $\mathcal{P}(n, \mathrm{T}\Sigma_{n,r}[m])$, we first observe that there is a 1-1 correspondence between the formulae of D and the formulae of the reduction sequences D_1 and D_2 taken together, and that the size of each formula of D (so $\xi_l^{(i,j)}$) is at most two more than, and hence at most twice, the size of the corresponding formula in D_1 or D_2 (which is $\psi_l^{(i,j)}$). Further we see that the size of β (which is "essentially" $\beta_1 \vee \beta_2$) is one more than, and hence at most twice, the sum of the sizes of β_1 and β_2. Then

$$\begin{aligned}|D(\bar{x}_1, \bar{x}_2)| &= |\Delta_1(\bar{x}_1)| + |\Delta_2(\bar{x}_2)| + |\beta| \\ &\leq 2 \cdot \sum_{j \in [2]} |D_j| \\ &\leq 2 \cdot 2 \cdot \mathsf{tower}(n, c \cdot (n+1) \cdot |\varphi_1|) \\ &\leq \mathsf{tower}(n, c \cdot (n+1) \cdot |\varphi|).\end{aligned}$$

4. Finally, for part (iv) of $\mathcal{P}(n, \mathrm{T}\Sigma_{n,r}[m])$, we see that the time taken to compute $D(\bar{x}_1, \bar{x}_2)$ is

$$\begin{aligned}&\leq \text{Time taken to compute } D_1 \text{ and } D_2 + \text{Time taken to write out } D(\bar{x}_1, \bar{x}_2) \\ &\leq 2 \cdot \mathsf{tower}(n, c \cdot (n+1) \cdot |\varphi_1|^2) + d \cdot \mathsf{tower}(n, c \cdot (n+1) \cdot |\varphi|) \quad (\text{for some } d > 1) \\ &\leq \mathsf{tower}(n, c \cdot (n+1) \cdot |\varphi|^2)\end{aligned}$$

This completes the nested induction, and hence the outer induction and the proof. □

A corollary of the above decomposition theorem (Theorem 2) is the following "composition" result. We need some terminology to state the result. Let $\mathcal{L}$ be a logic as in Theorem 2. For $i \in [2]$, let $\mathfrak{A}_i$ be a τ-structure and $\bar{a}_i$ be a (finite) tuple from $\mathfrak{A}_i$. The $\mathcal{L}$ *theory* of $(\mathfrak{A}_i, \bar{a}_i)$ is the class of all $\mathcal{L}$ formulae $\varphi(\bar{x})$ over τ such that $|\bar{x}| = |\bar{a}_i|$ and $\mathfrak{A}_i \models \varphi(\bar{a}_i)$. We say the $\mathcal{L}$ theory of $(\mathfrak{A}_1 \underline{\uplus} \mathfrak{A}_2, \bar{a}_1, \bar{a}_2)$ is *determined by* the $\mathcal{L}$ theories of $(\mathfrak{A}_1, \bar{a}_1)$ and $(\mathfrak{A}_2, \bar{a}_2)$, if for all $i \in [2]$ and all τ-structures $\mathfrak{A}'_i$ and tuples $\bar{a}'_i$ of the same length as $\bar{a}_i$, it holds that

$$(\mathfrak{A}_i, \bar{a}_i) \equiv_{\mathcal{L}} (\mathfrak{A}'_i, \bar{a}'_i) \text{ for all } i \in [2] \quad \Rightarrow \quad (\mathfrak{A}_1 \underline{\uplus} \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \equiv_{\mathcal{L}} (\mathfrak{A}'_1 \underline{\uplus} \mathfrak{A}'_2, \bar{a}'_1, \bar{a}'_2)$$

where $\mathfrak{C} \equiv_{\mathcal{L}} \mathfrak{D}$ denotes that the $\mathcal{L}$ theories of $\mathfrak{C}$ and $\mathfrak{D}$ are the same. Finally, recall that 'arbitrary' means 'finite or infinite'.

Corollary 1 *Let $\mathcal{L}$ be one of the logics $\mathrm{T}\Sigma_n[m]$ or $\mathrm{T}\Pi_n[m]$ over a vocabulary τ, for some $n, m \geq 0$. For $i \in [2]$, let $\mathfrak{A}_i$ be an arbitrary τ-structure and $\bar{a}_i$ be a (finite) tuple from $\mathfrak{A}_i$. Then the $\mathcal{L}$ theory of $(\mathfrak{A}_1 \underline{\uplus} \mathfrak{A}_2, \bar{a}_1, \bar{a}_2)$ is determined by the $\mathcal{L}$ theories of $(\mathfrak{A}_1, \bar{a}_1)$ and $(\mathfrak{A}_2, \bar{a}_2)$.*

Proof For $i \in [2]$, let $\mathfrak{A}'_i$ be an arbitrary τ-structure and $\bar{a}'_i$ be a (finite) tuple from $\mathfrak{A}'_i$ of length $|\bar{a}_i|$, such that $(\mathfrak{A}_i, \bar{a}_i) \equiv_{\mathcal{L}} (\mathfrak{A}'_i, \bar{a}'_i)$. Let $\varphi(\bar{x}_1, \bar{x}_2)$ be an $\mathcal{L}$ formula over $\underline{\tau}$ such that $|\bar{x}_i| = |\bar{a}_i|$. We show the following to complete the proof.

$$\mathfrak{A}_1 \underline{\uplus} \mathfrak{A}_2 \models \varphi(\bar{a}_1, \bar{a}_2) \quad \Leftrightarrow \quad \mathfrak{A}'_1 \underline{\uplus} \mathfrak{A}'_2 \models \varphi(\bar{a}'_1, \bar{a}'_2) \tag{9}$$

Towards showing (9), let $D(\bar{x}_1, \bar{x}_2) = (\Delta_1(\bar{x}_1), \Delta_2(\bar{x}_2), \beta)$ be the $\mathcal{L}$ reduction sequence for φ over $\underline{\uplus}$ that is a Feferman–Vaught decomposition of φ as given by Theorem 2. Let $\psi_{i,j}(\bar{x}_j)$ for $i \in I$, $j \in [2]$ for some index set I, be $\mathcal{L}$ formulae over τ such that $\Delta_j(\bar{x}_j) = (\psi_{i,j}(\bar{x}_j))_{i \in I}$. Let $X_{i,j}$ be propositional variables such that β is a propositional formula over $\mathcal{X} = \{X_{i,j} \mid i \in I, j \in [2]\}$. Then there exist assignments $\zeta, \zeta' : \mathcal{X} \to \{0, 1\}$ such that for $i \in I$ and $j \in [2]$,

$$\zeta(X_{i,j}) = 1 \Leftrightarrow \mathfrak{A}_j \models \psi_{i,j}(\bar{a}_j) \text{ and } \zeta'(X_{i,j}) = 1 \Leftrightarrow \mathfrak{A}'_j \models \psi_{i,j}(\bar{a}'_j) \tag{10}$$

$$\mathfrak{A}_1 \underline{\uplus} \mathfrak{A}_2 \models \varphi(\bar{a}_1, \bar{a}_2) \Leftrightarrow \zeta \models \beta \text{ and } \mathfrak{A}'_1 \underline{\uplus} \mathfrak{A}'_2 \models \varphi(\bar{a}'_1, \bar{a}'_2) \Leftrightarrow \zeta' \models \beta \tag{11}$$

Since $(\mathfrak{A}_j, \bar{a}_j) \equiv_{\mathcal{L}} (\mathfrak{A}'_j, \bar{a}'_j)$, it follows for all $i \in I$, that $\mathfrak{A}_j \models \psi_{i,j}(\bar{a}_j)$ iff $\mathfrak{A}'_j \models \psi_{i,j}(\bar{a}'_j)$; whereby $\zeta(X_{i,j}) = 1$ iff $\zeta'(X_{i,j}) = 1$ from (10). Then $\zeta = \zeta'$, so by (11), we indeed have (9). □

4 Decompositions over Definable Operations on Structures

We now consider quantifier-free sum-like operations on structures as defined in Sect. 2, and show that these admit analogues of Theorem 2 and Corollary 1 as the results below show. Recall that $|\Xi|$ denotes the size of Ξ, that is, the sum of the sizes of the formulae of Ξ.

Theorem 3 *Let $\mathcal{L}$ be one of the logics $\mathrm{T}\Sigma_n[m]$ or $\mathrm{T}\Pi_n[m]$ over a vocabulary τ, for some $n, m \geq 0$. Let $\circledast$ be a quantifier-free sum-like binary operation on τ-structures. Let Ξ be a quantifier-free definition of $\circledast$. Then for every $\mathcal{L}$ formula $\varphi(\bar{x}_1, \bar{x}_2)$ over τ, there is an $\mathcal{L}$ reduction sequence $D(\bar{x}_1, \bar{x}_2)$ such that the following hold:*

1. *$D(\bar{x}_1, \bar{x}_2)$ is a Feferman–Vaught decomposition of $\varphi(\bar{x}_1, \bar{x}_2)$ over $\circledast$.*
2. *The size of $D(\bar{x}_1, \bar{x}_2)$ is $\mathsf{tower}(n, O((n+1) \cdot |\varphi| \cdot |\Xi|))$, and $D(\bar{x}_1, \bar{x}_2)$ can be computed from $\varphi(\bar{x}_1, \bar{x}_2)$ in time $\mathsf{tower}(n, O((n+1) \cdot (|\varphi| \cdot |\Xi|)^2))$.*

Proof Consider the formula $\psi(\bar{x}_1, \bar{x}_2) := \Xi(\varphi)(\bar{x}_1, \bar{x}_2)$ over $\underline{\tau}$ as defined in Sect. 2; by Remark 1, $\psi(\bar{x}_1, \bar{x}_2)$ is an $\mathcal{L}$ formula whose free variables are exactly those of $\varphi(\bar{x}_1, \bar{x}_2)$. Let $D(\bar{x}_1, \bar{x}_2)$ be the $\mathcal{L}$ reduction sequence for $\psi(\bar{x}_1, \bar{x}_2)$ as given by Theorem 2. We show below that $D(\bar{x}_1, \bar{x}_2)$ is also the desired reduction sequence for the present theorem.

Part 1: Since $D(\bar{x}_1, \bar{x}_2)$ is as given by Theorem 2, it is a Feferman–Vaught decomposition of $\psi(\bar{x}_1, \bar{x}_2)$ over the annotated disjoint union operation. The following equivalences show that $D(\bar{x}_1, \bar{x}_2)$ is also a Feferman–Vaught decomposition of $\varphi(\bar{x}_1, \bar{x}_2)$ over $\circledast$. For $i \in [2]$, let $\mathfrak{A}_i$ be a τ-structure and $\bar{a}_i$ be a tuple from $\mathfrak{A}_1 \circledast \mathfrak{A}_2$ of length $|\bar{x}_i|$ such that all the elements of $\bar{a}_i$ belong to $\mathfrak{A}_i$.

$$\begin{aligned}
& (\mathfrak{A}_1 \circledast \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models \varphi(\bar{x}_1, \bar{x}_2) \\
\Leftrightarrow\ & (\mathfrak{A}_1 \underline{\uplus} \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models \Xi(\varphi)(\bar{x}_1, \bar{x}_2) \quad && \text{(by Proposition 1)} \\
\Leftrightarrow\ & (\mathfrak{A}_1 \underline{\uplus} \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models \psi(\bar{x}_1, \bar{x}_2) \quad && (\text{since } \psi := \Xi(\varphi)) \\
\Leftrightarrow\ & (\mathfrak{A}_1, \mathfrak{A}_2, \bar{a}_1, \bar{a}_2) \models D(\bar{x}_1, \bar{x}_2)
\end{aligned}$$

Part 2: We first show below that the size of ψ and the time taken to compute it are both $O(|\varphi| \cdot |\Xi|)$. We show the calculations for the size of ψ; the calculations for the time taken to compute ψ can be done similarly. We assume τ is a fixed vocabulary and show by induction on the structure of an $\mathcal{L}$ formula γ over τ, that $|\Xi(\gamma)| \leq k \cdot |\Xi| \cdot |\gamma|$ for a suitable constant $k > 0$.

1. In the base case of γ being an atomic formula or its negation, we see that the size of $\Xi(\gamma)$ is bounded above by $k \cdot q$ for a suitably large constant $k > 0$ where q is the maximum length of any formula in Ξ. This is because the arity of any predicate of τ is bounded by a constant since τ is a fixed vocabulary. Since $q \leq |\Xi|$, we have that $|\Xi(\gamma)| \leq k \cdot |\Xi| \cdot |\gamma|$.
2. Suppose $\gamma = \circledast_{i \in I} \gamma_i$ where $\circledast \in \{\bigwedge, \bigvee\}$, that $|\gamma_i| \leq k \cdot |\Xi| \cdot |\gamma_i|$ for all $i \in I$. Then:

$$\begin{aligned} |\Xi(\gamma)| &= 1 + \sum_{i \in I} |\Xi(\gamma_i)| \\ &\leq k \cdot |\Xi| + \sum_{i \in I} k \cdot |\Xi| \cdot |\gamma_i| \quad \text{(by induction hypothesis)} \\ &\leq k \cdot |\Xi| \cdot |\gamma| \end{aligned}$$

3. Suppose $\gamma = Q\bar{x}\gamma_1$ for $\gamma_1 = \circledast_{i \in I} \gamma_i$ where $(Q, \circledast) \in \{(\exists, \bigwedge), (\forall, \bigvee)\}$ and $\bar{x} = (x_1, \ldots, x_r)$. Suppose that $|\gamma_i| \leq k \cdot |\Xi| \cdot |\gamma_i|$ for all $i \in I$. We show the calculations below when $(Q, \circledast) = (\forall, \bigvee)$; the calculations when $(Q, \circledast) = (\exists, \bigwedge)$ are similar. The "3" in the equation below is for the three top level disjunctions in $\Xi(\gamma)$.

$$\begin{aligned} |\Xi(\gamma)| &= |\text{``}\forall\bar{x}\text{"}| + 3 + \sum_{j \in [r]} |\neg\xi_U(x_j)| + \sum_{i \in I} |\Xi(\gamma_i)| \\ &\leq 2 \cdot r + 3 + 2 \cdot r \cdot |\Xi| + \sum_{i \in I} k \cdot |\Xi| \cdot |\gamma_i| \quad \text{(by induction hypothesis)} \\ &\leq k \cdot |\Xi| \cdot (2r + 1 + \sum_{i \in I} |\gamma_i|) \quad \text{(since } k \text{ is sufficiently large)} \\ &\leq k \cdot |\Xi| \cdot |\gamma| \end{aligned}$$

This completes the structural induction. Putting $\gamma := \varphi$, we get that $|\psi| = |\Xi(\varphi)| \leq k \cdot |\Xi| \cdot |\varphi|$.

We can now complete the proof as below recalling that $D(\bar{x}_1, \bar{x}_2)$ satisfies Theorem 2(part (2)) for $\psi(\bar{x}_1, \bar{x}_2)$.

- The size of $D(\bar{x}_1, \bar{x}_2)$ is in

$$\begin{aligned} &\mathsf{tower}(n, O((n+1) \cdot |\psi|)) \\ &\subseteq \mathsf{tower}(n, O((n+1) \cdot |\varphi| \cdot |\Xi|)) \end{aligned}$$

- The time taken to compute $D(\bar{x}_1, \bar{x}_2)$ for φ is

$$\begin{aligned} &= \text{the time taken to compute } \psi + \\ &\quad \text{the time taken to compute } D(\bar{x}_1, \bar{x}_2) \text{ for } \psi \\ &\in O(|\varphi| \cdot |\Xi|) + \mathsf{tower}(n, O((n+1) \cdot |\psi|^2)) \\ &\subseteq \mathsf{tower}(n, O((n+1) \cdot (|\varphi| \cdot |\Xi|)^2)) \end{aligned}$$

□

Corollary 2 *Let $\mathcal{L}$ be one of the logics $\mathrm{T}\Sigma_n[m]$ or $\mathrm{T}\Pi_n[m]$ over a vocabulary τ, for some $n, m \geq 0$. Let $\mathfrak{A}_1$ and $\mathfrak{A}_2$ be arbitrary τ-structures. For $i \in [2]$, let $\bar{a}_i$ be a (finite) tuple from $\mathfrak{A}_1 \circledast \mathfrak{A}_2$, all of whose elements belong to $\mathfrak{A}_i$. Let $\circledast$ be a quantifier-free sum-like binary operation on τ-structures. Then the $\mathcal{L}$ theory of $(\mathfrak{A}_1 \circledast \mathfrak{A}_2, \bar{a}_1, \bar{a}_2)$ is determined by the $\mathcal{L}$ theories of $(\mathfrak{A}_1, \bar{a}_1)$ and $(\mathfrak{A}_2, \bar{a}_2)$.*

Proof For $i \in [2]$, let $\mathfrak{A}'_i$ be an arbitrary τ-structure and $\bar{a}'_i$ be a tuple from $\mathfrak{A}'_1 \circledast \mathfrak{A}'_2$ of length $|\bar{a}_i|$, such that all the elements of $\bar{a}'_i$ belong to $\mathfrak{A}'_i$, and $(\mathfrak{A}_i, \bar{a}_i) \equiv_{\mathcal{L}} (\mathfrak{A}'_i, \bar{a}'_i)$. Let $\varphi(\bar{x}_1, \bar{x}_2)$ be an $\mathcal{L}$ formula over τ such that $|\bar{x}_i| = |\bar{a}_i|$. We have the following equivalences. The first and third of these are by Proposition 1, and the second is by Corollary 1 observing that $\Xi(\varphi)$ is an $\mathcal{L}$ formula by Remark 1.

$$\begin{aligned} &\mathfrak{A}_1 \circledast \mathfrak{A}_2 \models \varphi(\bar{a}_1, \bar{a}_2) \\ \Leftrightarrow\ &\mathfrak{A}_1 \underline{\uplus} \mathfrak{A}_2 \models \Xi(\varphi)(\bar{a}_1, \bar{a}_2) \\ \Leftrightarrow\ &\mathfrak{A}'_1 \underline{\uplus} \mathfrak{A}'_2 \models \Xi(\varphi)(\bar{a}'_1, \bar{a}'_2) \\ \Leftrightarrow\ &\mathfrak{A}'_1 \circledast \mathfrak{A}'_2 \models \varphi(\bar{a}'_1, \bar{a}'_2) \end{aligned}$$

□

4.1 The Sizes of $\mathrm{T}\Sigma_n[m]$ and $\mathrm{T}\Pi_n[m]$ up to Equivalence

Our final result is concerning the calculation of bounds on the cardinalities of the classes of $\mathrm{T}\Sigma_n[m]$ and $\mathrm{T}\Pi_n[m]$ formulae in a given number of free variables, considered up to equivalence. As Proposition 2 shows, when n is fixed, these cardinalities are upper bounded by an elementary function of m. This is in contrast to the inherent non-elementariness in m of these cardinalities when considering all of FO in place of $\mathrm{T}\Sigma_n$ or $\mathrm{T}\Pi_n$.

Proposition 2 *Let $\mathcal{L}$ be one of the logics $\mathrm{T}\Sigma_n[m]$ or $\mathrm{T}\Pi_n[m]$ over a vocabulary τ, for some $n, m \geq 0$. Let p be the maximum arity of the predicates of τ. Then up to logical equivalence, given $t \geq 0$, the number of formulae in $\mathcal{L}$ with at most t free variables is bounded above by $\mathsf{tower}(n+2, (|\tau|+1) \cdot (n+1) \cdot (m+t)^p)$.*

Proof Let $\mathrm{T}\Sigma_n[m,t]$ and $\mathrm{T}\Pi_n[m,t]$ resp. denote the classes of all $\mathrm{T}\Sigma_n[m]$ and $\mathrm{T}\Pi_n[m]$ formulae having at most t free variables. We show the proposition by simultaneous induction on $\mathrm{T}\Sigma_n[m,t]$ and $\mathrm{T}\Pi_n[m,t]$ as n increases. We prove the result for

$\mathrm{T}\Sigma_n[m,t]$; the proof for $\mathrm{T}\Pi_n[m,t]$ is similar. Let $\mathrm{T}\Sigma_{n,r}[m,t]$ denote the class of all $\mathrm{T}\Sigma_{n,r}[m]$ formulae having at most t free variables. For $\mathcal{L} \in \{\mathrm{T}\Sigma_n[m,t], \mathrm{T}\Sigma_{n,r}[m,t]\}$, let $||\mathcal{L}||$ denote the number of formulae in $\mathcal{L}$ up to equivalence. We calculate $||\mathcal{L}||$ inductively following the inductive definition of $\mathcal{L}$ as given in Sect. 2.

Base case: For the base case of $n = 0$, we observe that since un-negated atomic formulae over τ are only of the form $R(x_1, \ldots, x_k)$ for a k-ary predicate $R \in \tau \cup \{=\}$, the total number of possible un-negated atomic formulae one can construct with at most t free variables, is up to equivalence at most $(|\tau|+1) \cdot t^p$ where p is the maximum arity of any predicate in τ. Since $\mathrm{T}\Sigma_0$ $(= \mathrm{T}\Pi_0)$ is the class of all propositional formulae in NNF over the mentioned atomic formulae, we have the following relation which verifies the base case.

$$||\mathrm{T}\Sigma_0[m,t]|| = ||\mathrm{T}\Sigma_{0,r}[m,t]|| = ||\mathrm{T}\Sigma_0[0,t]|| \leq \mathsf{tower}(2, (|\tau|+1) \cdot t^p).$$

Induction: Assume as induction hypothesis that the statement of the proposition is true for all $n' < n$ and all $m', t' \geq 0$ for both logics $\mathrm{T}\Sigma_{n'}[m']$ and $\mathrm{T}\Pi_{n'}[m']$. We calculate $||\mathrm{T}\Sigma_n[m,t]||$ for any given $m \geq 1$ and $t \geq 0$. Towards this, we first show by induction on r that

$$(\dagger) \quad ||\mathrm{T}\Sigma_{n,r}[m,t]|| \leq (m+t+2)^r \cdot \mathsf{tower}(n+2, (|\tau|+1) \cdot n \cdot (m+t)^p).$$

We have the following cases depending on the value of r.

- $r = 0$: The class $\mathrm{T}\Sigma_{n,0}[m,t]$ is the class of finite conjunctions of formulae from $\mathrm{T}\Pi_{n-1}[m,t]$ all of whose free variables are contained in the same t-tuple of variables. The latter class is contained in the class of all finite conjunctions of formulae from $\mathrm{T}\Pi_{n-1}[m,t]$. Then the following inequalities hold which verify $(\dagger)$.

$$\begin{aligned} ||\mathrm{T}\Sigma_{n,0}[m,t]|| &\leq 2^{||\mathrm{T}\Pi_{n-1}[m,t]||} \\ &\leq 2^{\mathsf{tower}(n+1, (|\tau|+1) \cdot n \cdot (m+t)^p)} \\ &\leq \mathsf{tower}(n+2, (|\tau|+1) \cdot n \cdot (m+t)^p) \end{aligned}$$

- $r > 0$: Assume that $(\dagger)$ holds for all r' such that $0 \leq r' < r$. Consider the class $\mathrm{T}\Sigma_{n,r}[m,t]$ – this class is the union of the classes $\mathrm{T}\Sigma_{n,r-1}[m,t]$ and the class of formulae obtained from the formulae of $\mathrm{T}\Sigma_{n,r-1}[m-1,t+1]$ by existentially quantifying out one of the (at most) $t+1$ free variables. Then we have the following inequalities that verify $(\dagger)$.

$$\begin{aligned} ||\mathrm{T}\Sigma_{n,r}[m,t]|| &\leq ||\mathrm{T}\Sigma_{n,r-1}[m,t]|| + (t+1) \cdot ||\mathrm{T}\Sigma_{n,r-1}[m-1,t+1]|| \\ &\leq (m+t+2)^{r-1} \cdot \mathsf{tower}(n+2, (|\tau|+1) \cdot n \cdot (m+t)^p) + \\ &\quad (t+1) \cdot ((m-1)+(t+1)+2)^{r-1} \cdot \\ &\quad \mathsf{tower}(n+2, (|\tau|+1) \cdot n \cdot ((m-1)+(t+1))^p) \quad (by(\dagger)) \\ &\leq (1+(t+1)) \cdot (m+t+2)^{r-1} \cdot \\ &\quad \mathsf{tower}(n+2, (|\tau|+1) \cdot n \cdot (m+t)^p) \end{aligned}$$

$$\leq (m+t+2)^r \cdot \mathsf{tower}(n+2, (|\tau|+1) \cdot n \cdot (m+t)^p)$$

We can now calculate $||\,\mathrm{T}\Sigma_n[m,t]\,||$. We observe that if $r > m$, then $\mathrm{T}\Sigma_{n,r}[m,t] \setminus \mathrm{T}\Sigma_{n,r-1}[m,t] = \emptyset$. Then the following hold:

$$\begin{aligned}
\mathrm{T}\Sigma_n[m,t] &= \bigcup_{r\geq 0} \mathrm{T}\Sigma_{n,r}[m,t] \\
&= \mathrm{T}\Sigma_{n,m}[m,t] \\
\therefore\ ||\,\mathrm{T}\Sigma_n[m,t]\,|| &= ||\,\mathrm{T}\Sigma_{n,m}[m,t]\,|| \\
&\leq (m+t+2)^m \cdot \mathsf{tower}(n+2, (|\tau|+1)\cdot n \cdot (m+t)^p) \quad \text{(by } (\dagger)) \\
&\leq \mathsf{tower}(n+2, (|\tau|+1)\cdot(n+1)\cdot(m+t)^p)
\end{aligned}$$

This completes the induction and the proof. □

5 Conclusion and Future Work

In this paper, we introduced a "tree" generalization of prefix classes of FO formulae. These classes, denoted $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$, are defined simultaneously as the classes of all FO formulae of the form $\exists\bar{x} \bigwedge \psi$ and $\forall\bar{x} \bigvee \chi$ respectively, where $\bigwedge \psi$ is a finite conjunction of $\mathrm{T}\Pi_{n-1}$ formulae and $\bigvee \chi$ is a finite disjunction of $\mathrm{T}\Sigma_{n-1}$ formulae. We showed Feferman–Vaught decompositions for formulae in these classes over quantifier-free sum-like operations, that preserve the quantifier-alternation structure as well as bounds on the rank of the formulae. As a consequence, the mentioned operations satisfy composition results by which the $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ theories of the structures input to the operations respectively determine the $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ theories of the operations' outputs. These results are further true stratified by the ranks of the formulae of the theories. While rank-preserving decomposition and composition results for FO are folklore in the literature, such results that additionally preserve the quantifier alternation structure are new to the best of our knowledge. On the computational side, we showed that the mentioned decompositions are computable in time that is elementary in the sizes of the input formulae. This adds to the (currently limited) toolbox of results from the literature showing scenarios where Feferman–Vaught decompositions can be obtained in elementary time. Our addition to this set of results is via exploiting a syntactic restriction on FO formulae, that of a low number of quantifier alternations, which is a feature of the FO descriptions of a wide range of interesting properties and problems in computer science (Appendix A (Sankaran, 2018)).

For future work, we would like to take ahead the results of this paper in various directions as mentioned below.

1. Theorem 3 shows an n-fold exponential upper bound on the sizes of the factors in the decomposition of any $\mathrm{T}\Sigma_n$ or $\mathrm{T}\Pi_n$ formula taken as input. We would like to investigate if this number of folds of the exponential is tight, i.e. whether the sizes of the factors cannot be bounded by any n'-fold exponential function in the size of the input formula, for $n' < n$. We would like to do a similar investigation for the tower height in Proposition 2.

2. We would like to generalize Theorems 2 and 3 to sum-like operations that are not necessarily binary, possibly even having infinite arity (as in Feferman and Vaught 1959; Elberfeld et al. 2016). We would also like to generalize the said theorems to operations that are product-like, specifically the Cartesian and tensor products.
3. Monadic second order logic (MSO) in a logic of much interest in algorithmic settings given that many important decision problems, such as 3-colorability, have natural descriptions in MSO. We would like to investigate extensions of Theorems 2 and 3 to suitably defined MSO analogues of $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$. We again observe that even with second order quantifiers, the number of quantifier alternations required to express interesting algorithmic problems, is low, and typically at most 1. For example, for 3-colorability, the number of second order quantifier alternations is 0, and the total number of quantifier alternations (first and second order quantifiers included) is 1.
4. Finally, we are interested in investigating the model checking problem for $\mathrm{T}\Sigma_n$ and $\mathrm{T}\Pi_n$ over graphs of bounded clique-width. These graphs are constructed from labeled point graphs, using simple unary and sum-like binary operations that are quantifier-free. A celebrated algorithmic meta theorem of Courcelle, Courcelle et al. (2000) states that the model checking problem for MSO is fixed parameter tractable over bounded clique-width graphs. That is, checking if an MSO formula φ holds over a graph from a bounded clique-width class, is solvable in time $f(|\varphi|) \cdot n^{O(1)}$ where n is the number of vertices in the graph, and f is some computable function of $|\varphi|$. However, under believed complexity theoretic assumptions, it was shown by Frick and Grohe (2004) that the function f is unavoidably non-elementary, and that this holds even when considering FO instead of MSO over the class of all finite trees (which have clique-width at most 3).
   Intuitively, it seems that the unrestricted number of quantifier alternations in the input FO sentences has a role to play in the mentioned non-elementariness, given the fact that the number of FO sentences modulo equivalence, of a given rank and arbitrary quantifier alternations, is non-elementary in the rank. In this light, Proposition 2 motivates asking the following question which can be seen as a stratification via the number of quantifier alternations, of the above mentioned algorithmic meta theorem of Courcelle et al. (2000) restricted to FO.

   Problem 1 For any fixed $k, n \geq 0$, does there exist an algorithm that, given a graph G of clique-width at most k and a $\mathrm{T}\Sigma_n$ or $\mathrm{T}\Pi_n$ sentence φ, decides whether G satisfies φ in time $f_{k,n}(|\varphi|) \cdot |G|^{O(1)}$ where $|G|$ is the number of vertices of G, and $f_{k,n}$ is an elementary function of $|\varphi|$?

Acknowledgements I thank the anonymous referees of this paper, and those of the conference version of the paper (Sankaran 2021), for their helpful comments in improving the presentation of the results and for pointing to related literature.

References

Blumensath, A., Colcombet, T., & Löding, C. (2008). Logical theories and compatible operations. *Logic and Automata, 2*, 73–106.

Courcelle, B., Makowsky, J. A., & Rotics, U. (2000). Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems, 33*(2), 125–150.

Dawar, A., Grohe, M., Kreutzer, S., & Schweikardt, N. (2007). Model theory makes formulas large. In *International Colloquium on Automata, Languages, and Programming* (pp. 913–924).

Droste, M., & Paul, E. (2018). A feferman-vaught decomposition theorem for weighted mso logic. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik*.

Elberfeld, M., Grohe, M., & Tantau, T. (2016). Where first-order and monadic second-order logic coincide. *ACM Transactions on Computational Logic (TOCL), 17*(4), 1–18.

Emmer, M., Khasidashvili, Z., Korovin, K., & Voronkov, A. (2010). Encoding industrial hardware verification problems into effectively propositional logic. In *Proceedings of Formal Methods in Computer Aided Design, FMCAD 2010, Lugano, Switzerland, October 20–23, 2010* (pp. 137–144).

Fagin, R., Kolaitis, P. G., Miller, R. J., & Popa, L. (2005). Data exchange: Semantics and query answering. *Theoretical Computer Science, 336*(1), 89–124.

Feferman, S., & Vaught, R. (1959). The first order properties of products of algebraic systems. *Fundamenta Mathematicae, 47*(1), 57–103.

Frick, M., & Grohe, M. (2004). The complexity of first-order and monadic second-order logic revisited. *Annals of Pure and Applied Logic, 130*(1–3), 3–31.

Göller, S., Jung, J. C., & Lohrey, M. (2015). The complexity of decomposing modal and first-order theories. *ACM Transactions on Computational Logic (TOCL), 16*(1), 1–43.

Grohe, M. (2008). Logic, graphs, and algorithms. *Logic and Automata, 2*, 357–422.

Gulwani, S. (2010). Dimensions in program synthesis. In *Proceedings of the 12th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming, ACM, PPDP '10* (pp 13–24).

Haase, C. (2014). Subclasses of Presburger arithmetic and the weak EXP hierarchy. In *Proceedings of the joint meeting of the 23rd EACSL annual conference on computer science logic (CSL) and the 29th annual ACM/IEEE symposium on logic in computer science (LICS)* (pp. 1–10).

Harwath, F. (2016). A note on the size of Prenex normal forms. *Information Processing Letters, 116*(7), 443–446.

Harwath, F., Heimberg, L., & Schweikardt, N. (2015). Preservation and decomposition theorems for bounded degree structures. *Logical Methods in Computer Science, 11*(4), 1. https://doi.org/10.2168/LMCS-11(4:17)2015

Lenzerini, M. (2002). Data integration: A theoretical perspective. In *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems, ACM, PODS '02* (pp. 233–246).

Libkin, L. (2013). *Elements of finite model theory*. Springer.

Makowsky, J. A. (2004). Algorithmic uses of the Feferman–Vaught theorem. *Annals of Pure and Applied Logic, 126*(1–3), 159–213.

Mostowski, A. (1952). On direct products of theories. *The Journal of Symbolic Logic, 17*(1), 1–31.

Piskac, R., de Moura, L. M., & Bjørner, N. (2010). Deciding effectively propositional logic using DPLL and substitution sets. *Journal of Automated Reasoning, 44*(4), 401–424.

Reddy, C. R., & Loveland, D. W. (1978). Presburger arithmetic with bounded quantifier alternation. In *Proceedings of the 10th annual ACM symposium on theory of computing* (pp. 320–325).

Sankaran, A. (2018). *A generalization of the Łoś–Tarski preservation theorem—Dissertation summary*. arXiv:1811.01014

Sankaran, A. (2021). Feferman–Vaught decompositions for prefix classes of first order logic. In *ICLA 2021 Proceedings, 9th Indian Conference on Logic and its Applications* (pp. 111–116). March 4–7, 2021. https://www.isichennai.res.in/~sujata/icla2021/proceedings.pdf.

Stockmeyer, L. J., & Meyer, A. R. (1973). Word problems requiring exponential time (preliminary report). In *Proceedings of the 5th annual ACM symposium on theory of computing, association for computing machinery, New York, NY, USA, STOC '73* (p. 1–9).

Thomas, W. (1997). Ehrenfeucht games, the composition method, and the monadic theory of ordinal words. In *Structures in logic and computer science, a selection of essays in honor of A. Ehrenfeucht, number 1261 in lecture notes in computer science* (pp. 118–143). Springer.

van Bergerem, S., & Schweikardt, N. (2021). Learning concepts described by weight aggregation logic. In Baier, C., Goubault-Larrecq, J. (Eds) *29th EACSL Annual Conference on Computer Science Logic, CSL 2021, January 25–28, 2021, Ljubljana, Slovenia (Virtual Conference), Schloss Dagstuhl - Leibniz-Zentrum für Informatik, LIPIcs* (Vol. 183, pp. 10:1–10:18).

Wanke, E. (1994). k-nlc graphs and polynomial algorithms. *Discrete Applied Mathematics, 54*(2–3), 251–266.