UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
MESTRADO E DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

LUIS HENRIQUE BUSTAMANTE DE MORAIS

PARAMETERIZED COMPLEXITY INVESTIGATIONS ON THE FIRST-ORDER SATISFIABILITY AND MATCHING PROBLEMS

FORTALEZA
2019

LUIS HENRIQUE BUSTAMANTE DE MORAIS

PARAMETERIZED COMPLEXITY INVESTIGATIONS ON THE FIRST-ORDER SATISFIABILITY AND MATCHING PROBLEMS

Tese apresentada ao Departamento de Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Ciência da Computação. Área de Concentração: Teoria da Computação

Orientadora: Prof.ª Dr.ª Ana Teresa de Castro Martins

Coorientador: Prof. Dr. Francicleber Martins Ferreira

FORTALEZA

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M825p Morais, Luis Henrique Bustamante de.
Parameterized Complexity Investigations on the First-order Satisfiability and Matching Problems / Luis Henrique Bustamante de Morais. – 2019.
69 f. : il.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação , Fortaleza, 2019.
Orientação: Profa. Dra. Ana Teresa de Castro Martins.
Coorientação: Prof. Dr. Francicleber Martins Ferreira.

1. Complexidade Parametrizada. 2. Satisfatibilidade. 3. Casamento de termos. 4. Lógica de primeira-ordem. I. Título.

CDD 005

LUIS HENRIQUE BUSTAMANTE DE MORAIS

PARAMETERIZED COMPLEXITY INVESTIGATIONS ON THE FIRST-ORDER SATISFIABILITY AND MATCHING PROBLEMS

Tese apresentada ao Departamento de Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Ciência da Computação. Área de Concentração: Teoria da Computação

Aprovada em:

BANCA EXAMINADORA

Prof.ª Dr.ª Ana Teresa de Castro Martins (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Francicleber Martins Ferreira (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Carlos Eduardo Fisch de Brito
Universidade Federal do Ceará (UFC)

Prof. Dr. Edward Hermann Haeusler
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

Prof. Dr. Mário R. Folhadela Benevides
Universidade Federal do Rio de Janeiro (UFRJ)

Dedico este trabalho a minha família.

AGRADECIMENTOS

Gostaria de agradecer a muitas pessoas que contribuíram para a finalização desta tese:

À Ana Teresa, pelo apoio dado durante todo o período do doutorado e mesmo antes do meu ingresso. Agradeço também a liberdade para a escolha do tema de pesquisa e pela paciência.

Ao Francicleber, pelo auxílio nos trabalhos e pelo estímulo durante todo o período.

Ao professores Carlos, Hermman e Mário pela leitura e considerações feitas no presente texto.

Ao meu pai, à minha mãe, e aos meus irmãos, pela motivação diária, que, por meio dos exemplos, me ajudaram a chegar até aqui. Também agradeço a ajuda financeira, sem isso, a conclusão deste trabalho não seria possível.

À Tamiris, pelo amor e todo incentivo para conclusão da Tese.

Por fim, aos meus amigos do laboratório, que compartilharam das difulculdades diárias da vida de doutorando, pelo convívio saudável e pelos momentos divertidos.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

“An expert is a person who has made all the mistakes that can be made in very narrow field.”

(Niels Bohr)

ABSTRACT

Parameterized complexity theory is a subarea of computational complexity theory in which the run-time analysis of a computational problem handles, besides the input size, an additional term that allows us to recognize "some kind of tractability" for many previously intractable problems. Many problems from Logic have been received attention by some parameterized analysis technique. We explore two logical tasks using the tools of the parameterized complexity. First, we study the parameterized complexity of the satisfiability problem for some prefix-vocabulary fragments of first-order logic. We consider the natural parameters emerging from the definition of these fragments, such as the quantifier rank, and the number of relation symbols. Following the classical classification of decidable prefix-vocabulary fragments, we observed that, when combining with the finite model property, many fragments have fixed-parameter tractability for the satisfiability concerning some of these parameters. Secondly, we apply parameterized complexity theory for classification for associative, commutative, and associative-commutative matching problems ({A, C, AC}-MATCHING) considering different parameterizations. We primarily consider the number of variables, the size of the substitution, and the size of the vocabulary as parameters. Combining the size of the substitution and the size of the vocabulary, we established the fixed-parameter tractability for these matching problems. For the other cases, we obtained the membership in W[P] for C-MATCHING for the number of variables and, for {A, AC}-MATCHING, when considering the size of the substitution.

Keywords: Parameterized Complexity. Satisfiability. Matching. First-order Logic.

RESUMO

A teoria da complexidade parametrizada é uma sub-área da teoria da complexidade computacional em que a análise de tempo computacional considera, além do tamanho da entrada, um termo adicional e que permite perceber um "certa tratabilidade" para muitos problemas outrora intratáveis. Muitos problemas da Lógica tem sido tratados por alguma técnica de análise parametrizadas. Nós exploramos dois problemas da Lógica usando ferramentas da complexidade parametrizada. Inicialmente, nós estudamos a complexidade parametrizada do problema da satisfatibilidade para alguns fragmentos definidos por prefixo e o vocabulário da lógica de primeira-ordem. Nós consideramos parâmetros naturais retirados da definição desses fragmentos tais como o posto de quantificadores e o número de símbolos relacionais. Seguindo a classificação clássica dos fragmentos decidíveis definidos pelo prefixo e pelo vocabulário, nós observamos que, quando combinados com a propriedade de modelo finito, muitos fragmentos tem a satisfatibilidade tratável por um parâmetro fixo com respeito a um desses parâmetros. Em um segundo momento, nós aplicamos a complexidade parametrizada para a classificação dos problemas de *matching* associativo, comutativo e associativo-comutativo ({A, C, AC}-MATCHING) para diferentes parametrizações. Nós inicialmente consideramos o número de variáveis, o tamanho da substituição e o tamanho do vocabulário como parâmetros. Combinando o tamanho da substiuição e o tamanho do vocabulário, nós obtivemos a tratabilidade por um parâmetro fixo para esses problemas. Para os outros casos, nós obtivemos a pertinência em W[P] para o *matching* comutativo (C-MATCHING) considerando o número de variáveis e, para {A, AC}-MATCHING, quando consideramos o tamanho da substituição.

Palavras-chave: Complexidade Parametrizada. Satisfatibilidade. Casamento de termos. Lógica de primeira-ordem.

SUMÁRIO

1 INTRODUCTION

Parameterized complexity theory (DOWNEY; FELLOWS, 2012; FLUM; GROHE, 2006) is a branch of computational complexity theory dedicated to the analysis of computational problems regarding an additional term, the *parameter*. The parameter can be seen as a particular data arising from the structure of the problem, or a particular choice of input in the context of a multivariate analysis.

For example, consider the k vertex cover in a graph, an NP-complete problem, which we have to decide if there exists a subset of k vertices such that, each edge of the graph, one of its endpoints is on this subset. There is an algorithm running in $\mathcal{O}(1.2738^k + k \cdot n)$ time where n is the size of the graph (CHEN *et al.*, 2010) i.e., for all fixed values of k, there is an algorithm running in $\mathcal{O}(n)$. Moreover, for small values of k, this algorithm is comparatively better than the brute-force algorithm with $\mathcal{O}(n^k)$ running time, while for many NP-complete problems like k-Dominating-Set, it is unlikely that an algorithm with a similar running time exists.

The central notion of parameterized complexity theory is the *fixed-parameter tractability* that corresponds to a relaxed version of classical tractability where the "intractability" is confined to some expression in terms of the parameter. A parameterized problem is said to be fixed-parameter tractable (see Chapter 2), if there exists an algorithm that runs in time $f(k) \cdot |x|^c$, where $|x|$ is the input size, c is some constant, k is the parameter, and f is some arbitrary computable function (sometimes exponential or even worse). The definitions briefly presented in this introduction will be more precisely presented in the following chapters. Here, we introduce them to give a good understanding of the problem.

From the earliest research in the intractability of computational problems and the search for more efficient algorithms, it was clear that finer analysis was needed. Take, for example, the work of Moshe Vardi (VARDI, 1982) who considered three types of analyzes for the query evaluation in databases: data-complexity, expression-complexity and combined-complexity. However, it was only in the works of Rod Downey and Michael Fellows that the area of parameterized complexity was introduced as a research field by their own (DOWNEY; FELLOWS, 1992b; DOWNEY; FELLOWS, 1992a), and it has been applied extensively to many logical problems (GOTTLOB *et al.*, 2002; SZEIDER, 2004; ACHILLEOS *et al.*, 2012; PFANDLER *et al.*, 2015; HAAN; SZEIDER, 2016; LÜCK *et al.*, 2017; MEIER *et al.*, 2019).

At the birth of parameterized complexity, the weighted satisfiability problem was central in the definition of the fixed-parameter intractable class W[1]. Many problems for which no efficient algorithms were known within the theory were classified employing parameterized reductions for the weighted satisfiability problem. Moreover, the model checking problem for FO, the question of verifying whether a first-order sentence φ holds in a given finite structure $\mathfrak{A}$, plays a central role in the characterization of fixed-parameter intractability (FLUM; GROHE, 2001; FLUM; GROHE, 2006; HAAN; SZEIDER, 2017). As we will see in Section 2.2, model checking for fragments of first-order logic based on the alternation of quantifiers defines a canonical family of parameterized complexity classes.

Many applications justify the effort in the area of parameterized complexity. Many intractable problems have polynomial behavior when someone confines to a particular kind of instances. For example, the model checking problem for FO is decidable in PSPACE in the general case. However, when we consider the problem over graphs with bounded degree, the problem can be solved in linear time (SEESE, 1996) (For a graph G with degree $d \geq 3$ and a first-order formula φ, the problem $G \models \varphi$ can be solved in $2^{2^{2^{\mathcal{O}(k)}}} \cdot n$ time where $k = |\varphi|$ and n is the size of G). A similar result holds for Monadic Second-Order (MSO). In the general case, model checking in MSO is in PSPACE. Courcelle's theorem (COURCELLE, 1990) says that it is possible to decide in linear time whether an MSO definable property holds for a given graph when restricted to the class of bounded tree-width graphs.

In the next section, we introduce the parameterized analysis of the satisfiability problem of some decidable first-order fragments concerning different parameters and, for almost all fragments, we obtain a fixed-parameter result.

1.1 Satisfiability of Prefix-Vocabulary Fragments

The satisfiability problem for some fragments of first-order logic was initially explored in the seminal work of Löwenheim (LÖWENHEIM, 1915) where he showed that satisfiability of the monadic fragment of FO is decidable and that formulas with binary predicate have satisfiability problem as hard as the class of all first-order formulas, i.e., they form a *reduction class* for the satisfiability problem. Then, the decision problem (*Entscheidungsproblem*) was placed at a central point within the Mathematical Logic (HILBERT; ACKERMAN, 1928), and the classification of classes of formulas into reduction classes or decidable classes flourished. After this, many reduction classes for satisfiability were established, for example, the $\forall^*\exists^*$ sentences, as a corollary of Skolem's normal form in (SKOLEM, 1920), and it was improved to $\forall^3\exists^*$ by K. Gödel (GÖDEL, 1933). A complete description of these developments of this problem is provided in the book "The Classical Decision Problem" BÖRGER *et al.*, but we are mainly concerned with decidable classes.

We consider the satisfiability problem of prefix-vocabulary classes of first-order logic. A *prefix-vocabulary class* $[\Pi, \bar{p}, \bar{f}]$ (respectively $[\Pi, \bar{p}, \bar{f}]_=$) is a set of first-order logic formulas φ in the prenex normal form without equality (respectively, with equality) where Π is a string in $\{\exists, \forall, \exists^*, \forall^*\}$ denoting a set of quantifier prefixes, $\bar{p}$ is a relation arity sequence $(p_1, p_2, \dots)$ such that φ has at most p_a relation symbols of arity a, and $\bar{f}$ is a function arity sequence $(f_1, f_2, \dots)$ such that φ has at most f_a function symbols of arity a (with $a \leq \omega$). (see Definition 3.1). We occasionally use some abbreviations: we exclude an infinite sequence of zeros, i.e., we write (1) instead of $(1, 0, 0, \dots)$, and (ω) instead $(\omega, 0, 0, \dots)$. Moreover we write "all" if there is no restrictions. For example, $[\forall^3\exists^*, (0,1), (\omega)]$ is the class of all first-order sentences, without equality, of form $\forall x_1 \forall x_2 \forall x_3 \exists y_1 \exists y_2 \dots \exists y_n \varphi$, with arbitrary n, where φ is quantifier-free and whose vocabulary consists of a binary relation and an arbitrary number of unary functions. The *relational classes* are the prefix-vocabulary classes of first-order formulas without function symbols, and denoted by $[\Pi, \bar{p}]$.

In (BÖRGER *et al.*, 2001, Chapters (6-7)), the authors present a complete investigation of the decidability and the complexity for many prefix-vocabulary classes. *Maximum decidable fragments*, depicted in Table **??**, are those that are maximum with respect to decidability, i.e if we extend the definition of the class in any dimension, then it turns into a reduction class. An essential tool for some decidability results is the *finite model property* that guarantees a finite model for all satisfiable formulas within the considered class. However, the classes $[\text{all}, (\omega), (1)]_=$ (RABIN, 1969) and $[\exists^*\forall\exists^*, \text{all}, (1)]_=$ (SHELAH, 1977) are decidable without finite model property.

We drawn our attention to the fragments that are *maximal for the finite model property* depicted in Table 2. Then, we consider the parameterized satisfiability problem p-κ-SAT(X) for some prefix-vocabulary class X, and parameterized by some function κ. We take into account the parameter that are derived from the definition of prefix class like **the quantifier rank**, **the number of relations**, and **the arity of the vocabulary**.

In Section 3.2, we address the parameterized analysis of the satisfiability of relational classes from Table 2 (1-4). For these classes, we could establish fixed-parameter tractability with respect to some parameters (Theorems(3.4)-(3.5)-(3.8)-(3.9)). Additionally, we express a lower bound for the satisfiability problem of $[\text{all}, (\omega)]$, when parameterized by **the quantifier rank**

Tabela 1 – Maximal Prefix-Vocabulary classes

Prefix-Vocabulary Class	**Reference**
(1) $[\exists^*\forall^*, \text{all}]_=$,	(Bernays, Schönfinkel 1928) (BERNAYS; SCHÖNFINKEL, 1928)
(2) $[\exists^*\forall^2\exists^*, \text{all}]$,	(Gödel 1932, Kalmár 1933, Schütte 1934) (GÖDEL, 1932; KALMÁR, 1933; SCHÜTTE, 1934)
(3) $[\text{all}, (\omega), (\omega)]$,	(Löb 1967, Gurevich 1969) (LÖB, 1967; GUREVICH, 1969)
(4) $[\exists^*\forall\exists^*, \text{all}, \text{all}]$,	(Gurevich 1973) (GUREVICH, 1973)
(5) $[\exists^*, \text{all}, \text{all}]_=$,	(Gurevich 1976) (GUREVICH, 1976)
(6) $[\text{all}, (\omega), (1)]_=$,	(Rabin 1969) (RABIN, 1969)
(7) $[\exists^*\forall\exists^*, \text{all}, (1)]_=$,	(Shelah 1977) (SHELAH, 1977)

Fonte: “The Classical Decision Problem” (BÖRGER *et al.*, 2001).

Tabela 2 – Prefix-Vocabulary classes maximal for the finite model property.

Prefix-Vocabulary Class	**Reference**
(1) $[\text{all}, (\omega)]_=$	(Löwenheim 1915) (LÖWENHEIM, 1915)
(2) $[\exists^*\forall^*, \text{all}]_=$	(Bernays-Schönfinkel-Ramsey 1930) (BERNAYS; SCHÖNFINKEL, 1928; RAMSEY, 1987)
(3) $[\exists^*\forall\exists^*, \text{all}]_=$	(Ackermann 1928) (ACKERMANN, 1928)
(4) $[\exists^*\forall^2\exists^*, \text{all}]$	(Gödel 1932, Schütte 1934) (GÖDEL, 1932; SCHÜTTE, 1934)
(5) $[\text{all}, (\omega), (\omega)]$	(Löb 1967, Gurevich 1969) (LÖB, 1967; GUREVICH, 1969)
(6) $[\exists^*\forall\exists^*, \text{all}, \text{all}]$	(Gurevich 1973) (GUREVICH, 1973)
(7) $[\exists^*, \text{all}, \text{all}]_=$	(Gurevich 1976) (GUREVICH, 1976)
(8) $[\forall^*, (\omega), (1)]_=$	(Ash 1975) (ASH, 1975)
(9) $[\exists^*\forall, \text{all}, (1)]_=$	(Grädel 1996) (GRÄDEL, 1989)

Fonte: “The Classical Decision Problem” (BÖRGER *et al.*, 2001).

only, the problem is unlikely to be fixed-parameter tractable (Proposition 3.6).

In Section 3.3, we also address the fixed-parameter tractability of the *classes with modest complexity* (BÖRGER *et al.*, 2001, Sec. 6.4), classes with the satisfiability problem placed on P, NP, CoNP, PSPACE, Σ_2^p and Π_2^p.

The strategy applied in the previous results is to define a set of fixed-parameter reductions from p-κ-SAT(X) to the propositional parameterized satisfiability p-SAT. This method will imply that the problem is fixed-parameter tractable due to the closure of the class FPT under this kind of reduction. The results are based on the classical conversion of first-order sentences into propositional sentences in a finite domain. We summarize them in Tables 6 and 7.

To close our contributions in this topic, in Section 3.4, we extend the analysis of parameterized complexity to the functional classes $[\text{all}, (\omega), (\omega)]$ and $[\exists^*, \text{all}, \text{all}]_=$.

In the next section, we describe our approach into parameterized complexity of the matching problem for first-order terms under associative, commutative, and associative-commutative equational theories.

1.2 Matching of First-Order Terms

Unification and matching problems have an essential place in many areas like, for example, term rewriting and resolution-based theorem proving (BAADER; SNYDER, 2001). Historically, unification had already appeared in the works of Emil Post during his postdoctoral year of 1920 (URQUHART, 2009; MOL, 2006), and in the PhD Thesis of Jacques Herbrand (HERBRAND, 1930). However, unification has only been proposed explicitly in theorem-proving context as a necessary procedure in the resolution step in the seminal paper of Robinson (ROBINSON, 1965).

The unification problem of first-order terms is related to some identification between two symbolic expressions that it could be strictly *syntactical* or *equational*. For example, in the case of *syntactic unification*, in what case are $f(x,a)$ and $f(b,y)$ syntactically equal, such that f is an arbitrary function symbol, a,b are constants, and x,y are variables? This question could be answered by a method that decides whether is there any substitution for x and y by some other expressions. Here we consider the decision version of this problem, and someone may check that the substitution $\theta := \{x \mapsto b, y \mapsto a\}$ is a solution for the example. The *matching problem* for first-order terms is a restriction of the unification problem such that just the first term has variables to be replaced.

From computational complexity, the unification problem obtained some progress after Robinson's algorithm with exponential time complexity. First, a quadratic algorithm was proposed (ZILLI, 1975). Then a linear algorithm was presented in (PATERSON; WEGMAN, 1978). Finally, the polynomial completeness under log-space reductions was achieved (DWORK *et al.*, 1984). From the perspective of parameterized complexity theory, the syntactic unification is not very attractive once it is fixed-parameter tractable for every parameterized version of the decision problem.

However, the unification/matching problems get a little more complicated when we consider equality modulo some equational theory like, for example, associativity, commutativity, or distribuitivity. If someone consider a certain axiomatization for which a function symbol must be interpreted, there are more solutions for the unification problem. For example, consider a commutative function f such that $f(x,y) \approx f(y,x)$, and the previous equivalence example modulo commutativity of f: $f(x,g(a,b)) =_C f(g(y,b),x)$, then there are many solutions with respect to the substitution of x while y is substituted by a. In this case, we are dealing with *equational unification/matching*.

In Chapter 4, we are mainly concerned with the equational matching problem for associative (A), commutative (C) and associative-commutative (AC) terms which are known to be NP-complete (BENANAV *et al.*, 1987). More precisely, for $s \in T(\mathscr{F},\mathscr{V})$ and $t \in T(\mathscr{F})$ (a term without variables), the problem asks for a substitution θ such that $s\theta =_E t$ for some equational theory $E \in \{A,C,AC\}$. They are said to *match* if there is a substitution θ such that $s\theta = t$.

We evoke the parameterized complexity theory as a framework able to distinguish the fine-grained complexity of these matching problems for different parameters and, in some sense, to detect the source of their hidden complexity. In addition to the concept of fixed-parameter tractability, a diversified collection of classes describes the parameterized intractability, and it is best represented by the classes W[1] and W[2], the lowest levels of the W-Hierarchy (Definition 2.19). On the top of this, we have the class W[P] (Definition 2.16) which is the class of parameterized problems decidable by an algorithm in $f(k) \cdot |x|^c$ but with at most $h(k) \cdot \log|x|$ non-deterministic steps for some computable functions f, h and a constant c.

In (AKUTSU *et al.*, 2017), the parameterized complexity of the {A, C, AC}-unification/matching was studied with respect to $|\text{var}(s)|$. They obtained that p-$|\text{var}(s)|$-E-MATCHING are W[1]-hard for E $\in$ {A, AC}, and they conjectured that p-$|\text{var}(s)|$-C-MATCHING is in FPT by a dynamic programming algorithm. The process passes through, in a dag representation of the input terms, all pairs of vertices of the same level to the root checking whether they match.

In Section 4.4, we give an algorithm in W[P] for p-$|\text{var}(s)|$-C-MATCHING when parameterized by $|\text{var}(s)|$. Although, for {A, AC}-MATCHING, we would like to answer if these problems are within W[1] concluding their W[1]-completeness, we could only show the W[P] membership with $|\theta|$ as the parameter.

The relevance to locate a problem within W[1] is related to an algorithmic solvability faster than the exhaustive search over all $\binom{n}{k}$ subsets. For example, p-CLIQUE, a W[1]-complete problem, has an algorithm that runs in time $\mathscr{O}(n^{(\omega/3)k})$ (NEŠETŘIL; POLJAK, 1985), achieved with the use of a $n \times n$ matrix multiplication algorithm with running time in $\mathscr{O}(n^{\omega})$ (best known value for ω is 2.3728639 (GALL, 2014)). For p-DOMINATING-SET, a W[2]-complete problem, we cannot do anything better than an algorithm running in $\mathscr{O}(n^{k+1})$ unless CNF satisfiability has an $2^{\delta n}$ time algorithm for some $\delta < 1$ (PĂTRAŞCU; WILLIAMS, 2010).

It seems that the size of the substitution $|\theta| = |\text{var}(s)| + \sum_{i=1}^{|\text{var}(s)|} |t_i|$ (see Section 4.3) determine a more natural parameter since it represents the size of the solution. In this case, we may also provide a membership in W[P] for p-$|\theta|$-{A, AC}-MATCHING. In Section 4.5, we observe that with $|\mathscr{F}| + |\theta|$ as the parameter we can construct a brute-force algorithm that attests the fixed-parameter tractability. This idea was applied in (FERNAU *et al.*, 2016) to the string morphism problem, with respect to different parameters.

In the next section, we summarize the results included here and refer to publications resulting from the studies previously mentioned.

1.3 Overview and Contributions

We apply the parameterized complexity theory to two decision problems relevant to first-order logic (FO):

1. the satisfiability of decidable fragments of FO; and
2. the matching for first-order terms with associative, commutative, and associative-commutative function symbols.

For the first problem, we obtained the fixed-parameter tractability for satisfiability of prefix-vocabulary classes well explored in “The Classical Decision Problem” of Börger, Grädel, and Gurevich (BÖRGER *et al.*, 2001, Chapters (6-7)). We extracted parameters from the definition of these fragments that are based on the prefix and the vocabulary, like **the quantifier rank** and **the size of the vocabulary**. The first fixed-parameter tractability results concerning some relational classes, given in Chapter 3, were published in (BUSTAMANTE *et al.*, 2018), and the remaining cases, classes with function symbols, were presented at 19th Brazilian Logic Conference (BUSTAMANTE *et al.*, 2019a).

For the second problem, we explored different parameterized versions of the matching problem for first-order terms with associative, commutative, associative-commutative function symbols. The parameters were extracted from the structure of the problem, namely: **the number of variables**, **the size of the substitution**, and **the size of the vocabulary**. The results characterize the membership in an intractable parameterized class and the fixed-parameter tractability. The results of Chapter 4 were presented at the 33rd International Workshop on

Unification (BUSTAMANTE *et al.*, 2019b).

2 PARAMETERIZED COMPLEXITY

In this chapter, we review some basic definitions and results of parameterized complexity theory. Our primary reference for this chapter is the textbook "Parameterized Complexity Theory" (FLUM; GROHE, 2006) from where we extract the main tools of fixed-parameter tractability to develop the following chapters.

We assume that the reader has some knowledge in computational complexity theory (PAPADIMITRIOU, 2003), and mathematical logic (EBBINGHAUS *et al.*, 2013).

We denote the set of integers, non-negative integers, natural numbers by $\mathbb{Z}, \mathbb{N}_0, \mathbb{N}$, respectively. For $m, n \in \mathbb{Z}$, we define $[m,n] := \{l \in \mathbb{Z} \mid m \leq l \leq n\}$ and $[n] := [1,n]$. We denote tuples of elements $(v_1 \ldots, v_k)$ by $\bar{v}$.

2.1 Elements of Fixed-Parameter Tractability

Many computational problems were defined in a multivariate setting, and with some natural parameterizations. Take for example the definition of Vertex Cover, or Dominating Set whose instances consist of a graph $G = (V, E)$, and a natural $k \leq |V|$ (JOHNSON; GAREY, 1979). Recall that all these problems are NP-complete.

For a graph $G = (V, E)$, a *vertex cover* C is a subset of V such that for all $\{u, v\} \in E$ at least one of u and v belong to C.

VERTEX-COVER	
Instance:	A graph $G = (V, E)$, a natural k.
Problem:	Decide if G has a vertex cover $C \subset V$ of size k.

For a graph $G = (V, E)$, a *dominating set* D is a subset of V such that for all $u \in V \backslash D$ there is a $v \in D$ for which $\{u, v\} \in E$.

DOMINATING-SET	
Instance:	A graph $G = (V, E)$, a natural k.
Problem:	Decide if G has a dominant set with size k.

For the first problem, searching for a vertex cover of size k, the naive algorithm that tries all possible solutions runs in time $\mathscr{O}(n^k)$. However, a more clever solution exists. Someone could build a recursive algorithm that for each edge $\{u, v\}$ call two recursive procedures, one that considers u in the solution set and others that consider v. It is easy to implement each recursive call to run in $\mathscr{O}(||G||)$. This procedure leads to an algorithm running in time $\mathscr{O}(2^k \cdot ||G||)$ (see Example 2.5).

For the second problem, there is nothing better than the brute-force algorithm with running time $\mathscr{O}(n^{k+1})$ assuming that there is no better algorithm for the Boolean satisfiability in conjunctive normal form CNF-SAT (PĂTRAŞCU; WILLIAMS, 2010). Note that for all values of $k > 0$ both problems have a polynomial-time algorithm. The difference in the running time leads to a linear-time algorithm for all values of k in the first case while, in the second case, the exponent in the running time depends on k. For small values of k, the difference between $\mathscr{O}(2^k \cdot n)$ and $\mathscr{O}(n^k)$ is dramatic and characterizes the central aspect of parameterized complexity theory that confines the combinatorial explosion to k.

Parameterized complexity theory (DOWNEY; FELLOWS, 2012; FLUM; GROHE, 2006) introduces an additional dimension to the analysis of computational complexity. The first building block of this theory is the idea of the parameterized problem.

Definition 2.1 (Parameterized problem). A *parameterized problem* is a pair (Q, κ) where $Q \subseteq \Sigma^*$, for some finite alphabet Σ, is a decision problem[1] and κ is a polynomial-time computable function from Σ^* to natural numbers $\mathbb{N}$, called the *parameterization*[1]. For an *instance* $x \in \Sigma^*$ of Q, $\kappa(x) = k$ is the *parameter* of x.

Fixing a value of the parameter, we obtain the important concept of *slices* that will be useful for the connection with classical complexity theory.

Definition 2.2 (Slice of a parameterized problems). A *slice* of a parameterized problem (Q, κ) is the decision problem $(Q, \kappa)_\ell := \{x \in Q \mid \kappa(x) = \ell \in \mathbb{N}\}$.

Example 2.3 (Parameterized satisfiability)**.** A canonical example is the parameterized satisfiability problem for propositional logic where a propositional formula φ is encoded over some finite alphabet Σ and $\kappa(\varphi)$ equals to the number of propositional variables of φ.

p-SAT	
Instance:	A propositional formula α.
Parameter:	Number of variables of α.
Problem:	Decide whether α is satisfiable.

[1] As is common in complexity theory, a decision problem is described as a language over finite alphabets. We always assume Σ to be nonempty.

[1] The definition of “parameterization” can be slightly different if we consider problems with a particular structure. For example, in the Courcelle’s Theorem, the parameterization is computable by an algorithm in fpt-time (FLUM; GROHE, 2006, Chapter 11).

We can also look at the satisfiability problem from the perspective of different parameterizations like **the number of clauses** or **structural parameters** from different representations of the clausal form of propositional formulas (SZEIDER, 2004).

The second and maybe the central concept of the parameterized analysis is a relaxed notion of tractability.

Definition 2.4 (Fixed-parameter tractability). We say that a problem (Q, κ) is *fixed-parameter tractable* (fpt) if there is an algorithm that decides $x \in Q$ in time bounded by $f(\kappa(x)) \cdot |x|^{\mathcal{O}(1)}$ for some computable and non-decreasing function f. The class of all fixed-parameter tractable problems is denoted by FPT.

We can verify the fixed-parameter tractability of p-SAT by the exhaustive search iterating over all 2^k propositional truth values, and for each iteration, evaluate it in the input formula. This procedure runs in $\mathcal{O}(2^k \cdot n)$ time such that k is the number of propositional variables, and n is the size of the propositional formula. The number of propositional variables is not expected to be small in this case, but it exposes the source of the difficulty of the problem.

The status of fixed-parameter tractable is the goal for many parameterized analysis of many NP-hard problems. The list of problems that are settled in FPT is considerable, and it is still increasing (CESATI, 2006).

Example 2.5. One of the most explored problems in parameterized complexity is p-VERTEX-COVER (ABU-KHZAM *et al.*, 2004; CHEN *et al.*, 2010). Recall the definition of the problem. For a graph $G = (V, E)$, a *vertex cover* C is a subset of V such that, for all edge $\{u, v\} \in E$, $u \in C$ or $v \in C$. Then, the parameterized version of this problem consider **the size of** C as a parameter.

p-VERTEX-COVER	
Instance:	A graph G, and a natural k.
Parameter:	k.
Problem:	Decide if G has a vertex cover C of size k.

As we can see, the size of the solution $|C|$ is a natural parameter, and it is an element of the problem (JOHNSON; GAREY, 1979).

The method of *bounded search trees* of height k can be applied to obtain a solution with size at most k in $\mathcal{O}(2^k \cdot \|G\|)$ time (DOWNEY; FELLOWS, 1995b). Consider a binary search tree of height k such that, each node of the tree is labeled with a possible solution C' and the remained graph to be explored G'. The root is labeled with $\{\emptyset, G\}$. At the beginning, choose a $\{u,v\} \in E$, and make $C' \leftarrow C' \cup \{u\}$ and $G' \leftarrow (V \backslash \{u\}, E \backslash \{\{u,v'\} \mid \text{for all } \{u,v'\} \in E\})$. Call a recursive procedure searching a vertex cover of size $k-1$ in G'. Do similarly construction for v. Note that these constructions can be executed in $\mathscr{O}(||G||)$. Repeat this process until $k=0$.

A better algorithm is described in (CHEN *et al.*, 2010), and provides a solution to this problem $\mathscr{O}(1.2738^k + k \cdot n)$ time, such that n is the number of vertices, and k is the size of the solution.

The parameterization function κ is defined in a general way, and it comes out with two "pathological" parameterizations: $\kappa_{\text{one}}(x) := 1$, and $\kappa_{\text{size}}(x) := \max\{1, |x|\}$, which we call trivial parameterizations. In the first case, a parameterized problem (Q, κ_{one}) is fixed-parameter tractable if and only if Q is polynomial-time decidable. In the second case, all parameterized problems $(Q, \kappa_{\text{size}})$ are fixed-parameter tractable.

This property of κ_{size} can be generalized to all parameterizations that increases monotonically with the size of the input leading to the fixed-parameter tractability trivially, and it will be useful to set apart good parameterizations from anomalous ones. A function $f : \mathbb{N} \to \mathbb{N}$ is *non-decreasing* if for all $m,n \in \mathbb{N}$ with $m < n$ we have $f(m) \leq f(n)$. A function f is *unbounded* if for all $n \in \mathbb{N}$ there exists an $m \in \mathbb{N}$ such that $f(m) \geq n$.

Proposition 2.6. *(FLUM; GROHE, 2006) Let $g : \mathbb{N} \to \mathbb{N}$ be a computable non-decreasing and unbounded function, Σ a finite alphabet, and $\kappa : \Sigma^* \to \mathbb{N}$ a parameterization that $\kappa(x) \geq g(|x|)$ for all $x \in \Sigma^*$. Then for every decidable set $Q \subseteq \Sigma^*$, the problem (Q, κ) is fixed-parameter tractable.*

To construct a robust complexity theory for parameterized problems, a proper notion of reduction is considered avoiding any aspect of complexity confinement within the parameter. Next, it leads to the definitions of hardness and completeness. Moreover, some parameterized classes are constructed regarding the closure under the following parameterized reductions.

Definition 2.7 (Parameterized reduction). Given the parameterized problems (Q, κ) and (Q', κ') in the alphabets Σ and Σ', respectively, an *fpt-reduction from* (Q, κ) *to* (Q', κ') is a mapping $R : \Sigma^* \to (\Sigma')^*$ such that: (i) For all $x \in \Sigma^*$ we have $(x \in Q \Leftrightarrow R(x) \in Q')$. (ii) R is computable by an *fpt-algorithm* (with respect to κ). That is, there is a computable function f such that $R(x)$ is computable in time $f(\kappa(x)) \cdot |x|^c$ for some constant c. (iii) There is a computable function $g : \mathbb{N} \to \mathbb{N}$ such that $\kappa'(R(x)) \leq g(\kappa(x))$ for all $x \in \Sigma^*$.

Let C be a parameterized class. A parameterized problem (Q, κ) is *C-hard under fpt-reductions* if every problem in C is fpt-reductible to (Q, κ). A parameterized problem (Q, κ) is *C-complete under fpt-reductions* if (Q, κ) is C-hard and $(Q, \kappa) \in C$.

It can be shown that FPT is closed under fpt-reduction. Let $(Q, \kappa), (Q', \kappa')$ be parameterized problems and (Q', κ') in FPT. If there is an fpt-reduction from (Q, κ) to (Q', κ'), then (Q, κ) is in FPT too.

Lemma 2.8. *(FLUM; GROHE, 2006) FPT is closed under fpt-reduction.*

Proof. Let (Q, κ) and (Q', κ') be two parameterized problems. Assume that $(Q', \kappa') \in FPT$, and that there exists an fpt-reduction from (Q, κ) to (Q', κ'). We concatenate the fpt-reduction with the fpt-algorithm that solves (Q', κ') to decide (Q, κ) in a fpt-time.

Consider an instance $x \in Q$ such that $\kappa(x) = k$. Then, there is a fpt-reduction with running time $f(k) \cdot |x|^c$ generating an instance $x' \in Q'$ with $\kappa'(x') = k' \leq g(k)$ for some computable functions g, f, and a constant $c \in \mathbb{N}$. From the first assumption, there is an fpt-algorithm for (Q', κ') with running time in $h(k') \cdot |x'|^d$ for some computable function h, and constant d. Putting this two procedures together, we have a decision procedure for Q with running time $f(k) \cdot |x|^c + h(k') \cdot |x'|^d$. As $|x'| \leq f(k) \cdot |x|^c$ and $k' \leq g(k)$, the time is bounded by $f(k) \cdot |x|^c + h(g(k)) \cdot (f(k) \cdot |x|^c)^d \leq (f(k) + h(g(k)) \cdot f(k)^d) \cdot \mathcal{O}(|x|^{cd})$. □

It should be noted that, for every problem Q decidable in PTIME, the parameterized version (Q, κ) is in FPT for every parameterization κ, and that, for each problem $(Q, \kappa) \in$ FPT, its slices $(Q, \kappa)_\ell$ (Definition 2.2) are polynomial-time decidable.

Another important point is that not every polynomial-time reduction is a fixed-parameterized tractable reduction. The classical polynomial-time reduction from the INDEPENDENT-SET problem to the VERTEX-COVER problem does not serve as an fpt-reduction. An *independent set* in a graph $G = (V, E)$ is subset $I \subseteq V$ such that, for all $u, v \in I$, there is no edge $\{u, v\} \in E$. The polynomial-time reduction considers the complement of the independent set as the vertex cover in the same graph. The parameter in each case is the size of the solution, and the reduction produces a parameter $V - k$ for a vertex cover, while the independent set has size k. This case infringes the condition (iii) from Definition 2.7.

The framework of parameterized complexity is quite robust for the classification of parameterized problems within FPT. Namely, the methods of *bounded search trees* (DOWNEY; FELLOWS, 1995b), *kernelization* (BUSS; GOLDSMITH, 1993), *color coding* (ALON *et* *al.*, 1995), and *iterative compression* (REED *et al.*, 2004) are examples of techniques that were extensively applied to assert the fixed-parameter tractability for many different problems (CYGAN *et al.*, 2015).

In Sections 2.2 and 2.3, we will exhibit the question of fixed-parameter intractability, defining some parameterized intractable classes that have similar role as NP-complete problems. However, the fact that all slices of a fixed-parameter tractable problem are polynomial-time decidable leads to a simple fixed-parameter intractability result.

Example 2.9. Consider for example the p-COLORABILITY problem. Let $G = (V, E)$ be a graph. It is said to be *k-colorable* if there exists a mapping c from V to $[k]$ such that, for each $(u, v) \in E$, $c(u) \neq c(v)$.

p-COLORABILITY	
Instance:	A graph G, and a natural k.
Parameter:	k.
Problem:	Decide whether G k-colorable.

The 3-COLORABILITY is a well-known NP-complete problem (JOHNSON; GAREY, 1979). If p-COLORABILITY is in FPT, then there exists a polynomial-time algorithm for 3-COLORABILITY. Assuming that P $\neq$ NP, it is unlikely that p-COLORABILITY is in FPT.

2.2 Fixed-Parameter Intractability

Many computational problems with different parameterizations resist to the fixed-parameter tractability, and they were distributed over many parameterized intractable classes (para-C classes, W-Hierarchy, A-Hierarchy, ... (FLUM; GROHE, 2006)). Some parameterized classes are the analog form for some classical class. Others are artificially constructed but have their importance inside of the framework. In this section, we briefly introduce the elements for the fixed-parameter intractability.

For each complexity class C, we define the class XC for which all parameterized problem in XC have their slices in C. This definition is non-uniform with respect to the algorithmic realization of each slice, i.e., for some parameterized problem $(Q, \kappa) \in$ XC, each slice $(Q, \kappa)_\ell$ for $\ell > 0$ may be attested by a different algorithm with resources bounded considering C.

Definition 2.10 (Slicewise polynomial). The class *XP* (for *slicewise polynomial*) is the parameterized analog of the exponential time class. A parameterized problem (Q, κ) is in XP, if there is an algorithm that decides if $x \in Q$ in at most $f(\kappa(x)) \cdot |x|^{g(\kappa(x))}$ steps, for some computable functions $f, g : \mathbb{N} \rightarrow \mathbb{N}$.

As a consequence of the previous definition, all fixed-parameter problems are in XP, and then FPT $\subseteq$ XP. The XP class is also be closed under fpt-reductions. As stated in (FLUM; GROHE, 2006), XP will serve as a framework for the theory of parameterized complexity where almost all problems with some natural parameterization are within XP.

Another construction of parameterized classes, denoted by *para-C*, interpret a pre-computation procedure in terms of the parameterization was first defined in (FLUM; GROHE, 2003). After consuming an arbitrarily time in terms of the parameter k, produces an instance to be decided in some computational complexity class C. Here the working example is the para-NP class, and we define below.

Definition 2.11 (Para-NP). A parameterized problem (Q, κ) is in para-NP if there exist an alphabet Π, a computable function $\pi : \mathbb{N} \rightarrow \Pi^*$, and a problem $X \subseteq \Sigma^* \times \Pi^*$ such that $X \in$ NP and for all instances $x \in Q$ we have $x \in Q \Leftrightarrow (x, \pi(\kappa(x))) \in X$.

The following theorem appears in the context of many parameterized classes showing the equivalence between two concepts: a machine characterization of a parameterized class, and a definition that considers a pre-computation. Hence, para-NP can also be seen as the class of parameterized problems decided by a non-deterministic algorithm with "fpt-time".

Theorem 2.12. *(FLUM; GROHE, 2003; FLUM; GROHE, 2006) Let (Q, κ) be a parameterized problem in the alphabet Σ. The following statements are equivalent:*

1) *(Q, κ) is in para-NP*
2) *There is a non-deterministic algorithm that decides if $x \in (Q, \kappa)$ in at most $f(\kappa(x)) \cdot |x|^{\mathcal{O}(1)}$ steps, such that f is a computable function.*

Proof. $1) \Rightarrow 2)$. Assume that there exists a computable function $\pi : \mathbb{N} \rightarrow \Pi^*$ for some finite alphabet Π, and a problem $X \in$ NP such that, for all instances $x \in Q$, we have $x \in Q \Leftrightarrow (x, \pi(\kappa(x))) \in X$.

There exists a non-deterministic algorithm $\mathbb{A}$ that decides Q on the input (x, y) in time $\mathcal{O}((|x| + |y|)^d)$ where $y = \pi(k)$ and $k = \kappa(x)$. Then we construct another non-deterministic algorithm $\mathbb{A}_Q$ that, on the input x, computes the parameter k in polynomial-time, say $\mathcal{O}(|x|^c)$ for some natural c, and compute the string obtained by the function π in time bounded by $g(k)$ for some time constructible function g. After this, the algorithm $\mathbb{A}_Q$ simulates $\mathbb{A}$ on the input x and the generated string $\pi(\kappa(x)) = y$ with $|y| \leq g(k)$. This algorithm $\mathbb{A}_Q$ has running time bounded by $|x|^c + g(k) + (|x| + g(k))^d$. Since $|x| + g(k) \geq 1$, the algorithm has running time bounded by $g(k)^{d+1} \cdot |x|^{d+1}$.

2) $\Rightarrow$ 1). Let (Q, κ) a parameterized problem that is decidable by a non-deterministic algorithm $\mathbb{A}_Q$ in time $f(k) \cdot |x|^c$ with $k = \kappa(x)$. Let Π be the alphabet $\{1, \S\}$ and define $\pi : \mathbb{N} \to \Pi^*$ by $\pi(k) := k\S f(k)$ where k and $f(k)$ are written in unary. Let $X \subseteq \Sigma^* \times \Pi^*$ be the decision problem defined by the following algorithm $\mathbb{A}$.

Given $(x, y) \in \Sigma^* \times \Pi^*$, first $\mathbb{A}$ checks whether $y = \kappa(x)\S u$ for some $u \in \{1\}^*$. If this is not the case, $\mathbb{A}$ rejects, otherwise $\mathbb{A}$ simulates $|u| \cdot |x|^c$ steps of the computation of $\mathbb{A}_Q$ on input x. If $\mathbb{A}_Q$ stops in this time and accepts, then $\mathbb{A}$ accepts, otherwise $\mathbb{A}$ rejects.

Since $|u| \leq |y|$, one easily verifies that $\mathbb{A}$ runs in polynomial-time; moreover: $x \in Q \Leftrightarrow \mathbb{A}$ accepts $(x, \kappa(x)\S f(\kappa(x))) \Leftrightarrow (x, \pi(\kappa(x))) \in X$. □

Proposition 2.13. *(FLUM; GROHE, 2006) The class para-NP is closed under fpt-reductions.*

Proof. Analogous to the proof of Lemma 2.8. □

Proposition 2.14. *(FLUM; GROHE, 2006) Para-NP = FPT if and only if P = NP.*

Proof. Assume that FPT = para-NP. For every problem $Q \in$ NP, we have $(Q, \kappa_{\text{one}}) \in$ para-NP for the trivial parameterization κ_{one} with $\kappa_{\text{one}}(x) = 1$ for all $x \in \Sigma^*$. Then, we have also that $(Q, \kappa_{\text{one}}) \in$ FPT, and there exists a deterministic algorithm with running time $f(\kappa_{\text{one}}(x)) \cdot |x|^{\mathcal{O}(1)} = f(1) \cdot |x|^{\mathcal{O}(1)}$ for some computable function f. Hence $Q \in$ PTIME. □

The following theorem says that, when we find a finite set of NP-hard slices of a parameterized problem (Q, κ), then (Q, κ) is para-NP-hard. In Corollary 3.7, we apply the following result to conclude that the satisfiability problem for the monadic fragment of FO parameterized by the quantifier rank is para-NP-hard.

Theorem 2.15. *(FLUM; GROHE, 2006) Let (Q, κ) be a parameterized problem, and non-trivial i.e. $\emptyset \subsetneq Q \subsetneq \Sigma^*$. Then (Q, κ) is paraNP-hard under fpt-reductions if, and only if, a union of finitely many slices of (Q, κ) is NP-hard i.e. there are $\ell, m_1, \ldots, m_\ell$ such that*

$$(Q, \kappa)_{m_1} \cup (Q, \kappa)_{m_2} \cup \cdots \cup (Q, \kappa)_{m_\ell}$$

is NP-hard under polynomial-time reductions.

Proof. Assume that (Q,κ) is a para-NP-hard problem in the alphabet Σ. Let $Q' \subseteq (\Sigma')^*$ be an NP-hard problem. Then we have that (Q',κ_{one}) is in para-NP, and that there is an fpt-reduction $R : (\Sigma')^* \to (\Sigma)^*$ from (Q',κ_{one}) to (Q,κ). Let f,c,g be chosen according to Definition 2.7. Then, for all $x' \in (\Sigma')^*$, $R(x')$ can be computed in time $f(1)\cdot|x'|^c$, and $\kappa(R(x')) \leq g(1)$. Thus R is a polynomial-time reduction from Q' to $(Q,\kappa)_1 \cup (Q,\kappa)_2 \cup \cdots \cup (Q,\kappa)_{g(1)}$. Since Q' is NP-hard, this implies that $(Q,\kappa)_1 \cup (Q,\kappa)_2 \cup \cdots \cup (Q,\kappa)_{g(1)}$ is NP-hard.

In the opposite direction, assume that $(Q,\kappa)_{m_1} \cup (Q,\kappa)_{m_2} \cup \cdots \cup (Q,\kappa)_{m_\ell}$ is NP-hard. Consider a parameterized problem (Q',κ') in para-NP over the alphabet Σ'. We show that $(Q',\kappa') \leq^{fpt} (Q,\kappa)$. By definition, there exists an alphabet Π, a computable function $\pi : \mathbb{N} \to \Pi^*$ and a problem X in NP such that, for all $x \in (\Sigma')^*$ we have $s \in Q' \Leftrightarrow (x,\pi(\kappa'(x))) \in X$.

Since $(Q,\kappa)_{m_1} \cup (Q,\kappa)_{m_2} \cup \cdots \cup (Q,\kappa)_{m_\ell}$ is NP-hard, there is a polynomial-time reduction from X to $(Q,\kappa)_{m_1} \cup (Q,\kappa)_{m_2} \cup \cdots \cup (Q,\kappa)_{m_\ell}$, that is, a polynomial-time computable mapping $R : (\Sigma')^* \times \Pi^* \to \Sigma^*$ such that for all $(x,y) \in (\Sigma') \times \Pi^*$ we have

$$(x,y) \in X \Leftrightarrow R(x,y) \in (Q,\kappa)_{m_1} \cup (Q,\kappa)_{m_2} \cup \cdots \cup (Q,\kappa)_{m_\ell}.$$

We construct an fpt-reduction S from (Q',κ') to (Q,κ). Fix an arbitrary instance $x_0 \in \Sigma^* \backslash Q$. Since (Q,κ) is non-trivial, there is an x_0 such that

$$S(x) := \begin{cases} R(x,\pi(\kappa'(x))), & \text{if } \kappa(R(x,\pi(\kappa'(x)))) \in \{m_1,\ldots,m_\ell\}, \\ x_0, & \text{otherwise.} \end{cases}$$

It is easy to check that $x \in Q' \Leftrightarrow S(x) \in Q$ and that S is computable by an fpt-algorithm. □

Using the notion of bounded non-determinism in terms of the parameter leads to another interesting parameterized class.

Definition 2.16 (The class W[P]). The W[P] class is the class of parameterized problems (Q,κ) for which there is a non-deterministic Turing machine $\mathbb{M}$, on alphabet Σ, that decides $x \in Q$ using at most $f(k)\cdot p(|x|)$ steps with at most $h(k)\cdot\log|x|$ non-deterministic ones for some computable functions f and h.

Example 2.17. An example of a problem in W[P] is the parameterized version of the *longest common subsequence problem*. Let $\bar{a} = a_1 \ldots a_n$ and $\bar{b} = b_1 \ldots b_s$ be strings over the alphabet Σ. We say that $\bar{b}$ is a *subsequence* of $\bar{a}$ if $s \leq n$ and $b_1 = a_{i_1},\ldots,b_s = a_{i_s}$ for some $i_1,\ldots,i_s$ with $1 \leq i_1 < \ldots < i_s \leq n$.

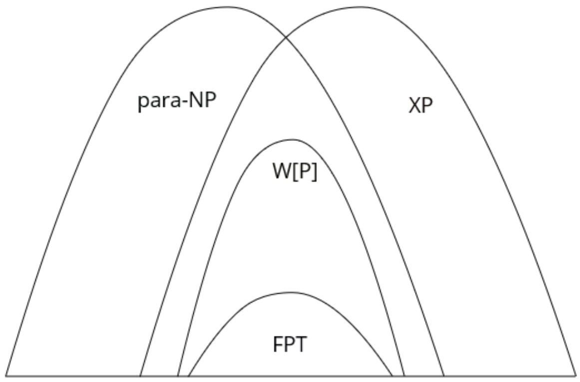
p-LCS	
Instance:	Strings $\bar{a}_1, \ldots \bar{a}_m \in \Sigma^*$ for some alphabet Σ and $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Decide whether there is a string of length k in Σ^* that is a subsequence of $\bar{a}_i$ for $i = 1, \ldots, m$.

To attest the membership of p-LCS in W[P], we consider a non-deterministic Turing machine that, on the input, guesses a string $\bar{b} \in \Sigma^*$ of length k in $\mathcal{O}(k \cdot \log |\Sigma|)$ steps and then deterministically verifies that $\bar{b}$ is a subsequence of every $\bar{a}_i$.

It is easy to see that W[P] is closed under fpt-reduction, and if we do not allow any non-deterministic step, then we can see that FPT is contained in W[P]. Hence we can sum up the relation between these classes by the following theorem.

Theorem 2.18. *(FLUM; GROHE, 2006) FPT $\subseteq$ W[P] $\subseteq$ XP $\cap$ para-NP.*

Figura 1 – Parameterized classes.



Fonte: Made by the author himself.

2.3 W-Hierarchy

W-hierarchy is the most representative group for the fixed-parameter intractability. The origin of the W-hierarchy is connected with the origin of parameterized complexity theory while some fpt-reduction from p-WSAT(Γ) classified many problems, the weighted satisfiability problem for different fragments of propositional logic Γ, and showing the completeness for some finite level of the hierarchy (DOWNEY; FELLOWS, 1992b; DOWNEY; FELLOWS, 1992a).

Recall the definition of the propositional satisfiability problem. For a propositional formula $\alpha \in \Gamma$, the *weight* is the number of propositional variables assigned to 1. Then we define the *weighted satisfiability prolem*.

p-WSAT(Γ)	
Instance:	$\gamma \in \Gamma$ and $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Is γ satisfiable by an assignment of weight k.

Each level of this hierarchy of parameterized classes, consider the satisfiability for different fragments of propositional logic characterized by an alternated use of *big conjunctions* (conjunction over a finite sequence of subformulas), $\bigwedge$ or *big disjunction* $\bigvee$ interpreted as new operators jointly with $\wedge$ (*small conjunction*), $\vee$ (*small disjunction*), and $\neg$.

We describe the fragments $\Gamma_{t,d}$ that will characterize each level t of the W-Hierarchy that corresponds to t nested levels of big connectives with at mots d literals on the lowest level. A *literal* is an atomic formula, or a negated atomic formula. For $t \geq 0$ and $d \geq 1$, we define the following class of propositional formulas $\Gamma_{t,d}$ and $\Delta_{t,d}$:

$$\Gamma_{0,d} := \{\lambda_1 \wedge \ldots \wedge \lambda_c \mid c \leq d, \text{ and literals } \lambda_1, \ldots, \lambda_c\},$$

$$\Delta_{0,d} := \{\lambda_1 \vee \ldots \vee \lambda_c \mid c \leq d, \text{ and literals } \lambda_1, \ldots, \lambda_c\},$$

$$\Gamma_{t+1,d} := \left\{ \bigwedge_{i \in I} \delta_i \mid \text{ a finite set } I, \text{ and, for all } i \in I, \delta_i \in \Delta_{t,d} \right\},$$

$$\Delta_{t+1,d} := \left\{ \bigvee_{i \in I} \gamma_i \mid \text{ a finite set } I, \text{ and, for all } i \in I, \gamma_i \in \Gamma_{t,d} \right\},$$

There are many combinatorial properties associated with the definition of these fragments of propositional logic, and we are reviewing in short. Chronologically, the original definition of W-Hierarchy is given below considering the closure under fpt-reduction:

Definition 2.19 (W-Hierarchy). For all $t \geq 1$, we define

$$W[t] := \bigcup_{d \geq 1} \left\{ (Q, \kappa) \mid (Q, \kappa) \leq^{\text{fpt}} p\text{-WSAT}(\Gamma_{t,d}) \right\}.$$

Again, many problems that could not be settled into FPT were fortunately classified in terms of reductions from p-WSAT($\Gamma_{t,d}$). For example, we can show that p-CLIQUE, p-INDEPENDENT-SET (DOWNEY; FELLOWS, 1995a) and p-LCS (longest commom subsequence when parameterize by the number of strings k and the size of the subsequence) are W[1]-complete (BODLAENDER *et al.*, 1995; DOWNEY; FELLOWS, 2012), p-DOMINATING-SET and p-DOMINATING-CLIQUE are W[2]-complete (DOWNEY; FELLOWS, 1992a). In Annex A, we present a different proof that the parameterized version of k-SUM is in W[1]. This result was open for some time, and was only proved in (ABBOUD *et al.*, 2014).

In (DOWNEY *et al.*, 1998; FLUM; GROHE, 2001), the connection between the p-WSAT(Γ) and a similar version of the model checking problem for some fragments of first-order logic was established in terms of parameterized reduction. Now we define the first-order logic machinery[1].

The set of quantifier-free formulas is denoted by Σ_0 and Π_0. For $t > 0$, we define Σ_{t+1} as the class of formulas in the form $\exists x_1 \ldots \exists x_k \varphi$, such that $\varphi \in \Pi_t$; and Π_{t+1} as the class of all formulas in the form $\forall x_1 \ldots \forall x_k \varphi$, where $\varphi \in \Sigma_t$.

We define $\varphi(X_1, \ldots X_l)$ as a first-order logic formula with free variables $X_1, \ldots X_l$ (as second-order variables). Given a vocabulary τ of φ, and for all $i \in [l]$, s_i corresponds to the arity of X_i. A solution for φ in $\mathscr{A}$ is a tuple $\bar{S} = (S_1, \ldots, S_l)$, where, for each $i \in [l]$, $S_i \subseteq A^{s_i}$, such that $\mathscr{A} \models \varphi(\bar{S})$. Thus, $\varphi(X)$ corresponds to a first-order formula with a unique relation variable X. We define *the weighted definability problem* for a formula φ.

p-WD$_\varphi$	
Instance:	A structure $\mathscr{A}$ and $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Decide if exists $S \subseteq A^s$ with $\lvert S \rvert = k$ such that $\mathscr{A} \models \varphi(S)$.

For a class of formulas $\Phi \subseteq$ FO, we define p-WD-Φ as the class of all problems p-WD$_\varphi$ such that $\varphi \in \Phi$. Then, we provide a different characterization of W hierarchy in terms of p-WD-Φ for fragments Π_t.

Theorem 2.20 (W Hierarchy). *(DOWNEY et al., 1998; FLUM; GROHE, 2001) For all $t \geq 1$, we define*

$$W[t] = [p\text{-}WD\text{-}\Pi_t]^{fpt}.$$

We say that the W-hierarchy is build by $W[t]$ with $t \geq 1$.

From our perspective, the previous result lifted the classification of W-hierarchy in terms of a more succinct representation represented by an FO expression. We close this section with some examples. Among other things, due to the closure of the classes of the W-hierarchy, to show the membership within some finite level, one has to produce a structure and a first-order formula that satisfies it. Here are some examples.

Example 2.21. p-CLIQUE is in W[1]:

[1] In the next chapter, we present a definition of the syntax of FO aiming the definition of structural parameters.

A clique in a graph $G = (V, E)$ is a subset of vertices $C \subseteq V$ such that there exists an edge for all pair of vertices in C. The parameterized version p-CLIQUE has the size of the clique k as the parameter.

p-CLIQUE	
Instance:	a graph G, $k \in \mathbb{N}$
Parameter:	k
Problem:	Decide if G has a clique with size k.

The following sentence is satisfiable in a graph G if, and only if, it contains a clique with size k. In the following sentence, X is a subset of V that forms a clique.

$$\text{clique}(X) := \forall x \forall y((Xx \wedge Xy \wedge x \neq y)) \rightarrow Exy).$$

Hence, p-CLIQUE is in p-WD-Π_1

Example 2.22. p-DOMINATING-SET is in W[2]:

A dominant set in a graph $G = (V, E)$ corresponds to a subset of vertices $D \subseteq V$ in which, for all vertex $v \in V$, $v \in D$ or there exists a vertex $u \in V$ such that $\{v, u\} \in E$.

p-DOMINATING-SET	
Instance:	a graph G, $k \in \mathbb{N}$
Parameter:	k
Problem:	Decide if G has a dominant set with size k.

The following sentence is satisfiable in a graph G if, and only if, it contains a dominating set of size k.

$$\text{dominant}(X) := \forall x \exists y(Xy \wedge (Exy \vee x = y)).$$

Thus, p-DOMINATING-SET is in p-WD-Π_2

To summarize, we presented the basic tools for parameterized analysis that will be used in the following chapters. In particular, in the next chapter, we will apply the notion of fixed-parameter tractability for first-order fragments defined by the quantifier prefix and the vocabulary used.

3 SATISFIABILITY OF PREFIX-VOCABULARY FRAGMENTS OF FO

In this chapter, we explore the parameterized complexity of the satisfiability problem for some decidable fragments of first-order logic classified by their prefix and vocabulary. Our study extends the computational complexity status of many decidable fragments of first-order logic compiled in the textbook “The Classical Decision Problem” (BÖRGER *et al.*, 2001, Chapter 6) within the framework of parameterized complexity. Using well-known concepts from mathematical logic, we obtain the fixed-parameter tractability for these fragments identifying the source of the computational difficulty and reducing to the propositional satisfiability and the satisfiability of these fragments.

In Section 3.1, we define the prefix-vocabulary classes, the parameterized versions of the satisfiability problem, and we summarize the decidability and complexity results for these classes. In Section 3.2, we provide some fixed-parameter reductions from satisfiability for the classical prefix-vocabulary classes to p-SAT. In Section 3.3, we show that all prefix classes with modest complexity are in FPT considering some parameterization. In Section 3.4, we extend the fixed-parameter tractability for two functional classes: $[\text{all}, \omega, \omega]$ and $[\exists, \text{all}, \text{all}]$. The results presented here were published in (BUSTAMANTE *et al.*, 2018) and (BUSTAMANTE *et al.*, 2019a).

3.1 Introduction

The *satisfiability problem* consists of deciding, given a formula φ, if there exists a model $\mathfrak{A}$ for the formula φ or not. In the general case, the first-order satisfiability SAT(FO) is undecidable (EBBINGHAUS *et al.*, 2013). In this work, we are interested in decidable prefix-vocabulary fragments characterized by their quantifiers pattern in the prenex normal form, and the use of relation, function symbols with different arities and the equality relation.

Recall that a *vocabulary* τ is a finite set of relation, function and constant symbols. Each symbol $\sigma \in \tau$ is associated with a natural number, its *arity*(σ). The *arity* of τ is the maximum arity of its symbols. A τ-*structure* $\mathfrak{A}$ is a tuple $(A, R_1^{\mathfrak{A}}, \ldots R_r^{\mathfrak{A}}, f_1^{\mathfrak{A}}, \ldots, f_s^{\mathfrak{A}}, c_1^{\mathfrak{A}}, \ldots c_t^{\mathfrak{A}})$ such that A is a non-empty set, called the *domain*, and each $R_i^{\mathfrak{A}}$ is a relation under $A^{\text{arity}(R_i)}$ interpreting the symbol $R_i \in \tau$, each $f_i^{\mathfrak{A}}$ is a function from $A^{\text{arity}(f_i)}$ to A interpreting the symbol $f_i \in \tau$, and each $c_i^{\mathfrak{A}}$ is an element of A interpreting the symbol c_i. We assume structures with finite domain, and, without loss of generality, we use a domain of naturals $\{1, \ldots, n\}$ denoted by

$[n]$.

A τ-term is a variable x, a constant c, or an m-ary function symbol f applied to τ-terms $t_1, t_2, \ldots, t_m$, $f(t_1, t_2, \ldots, t_m)$. If R is an m-ary relation symbol, and $t_1, t_2, \ldots t_m$ are τ-terms, then $R(t_1, t_2, \ldots, t_m)$, and $t_1 = t_2$ are *τ-formulas*, which we call *atomic formulas*. If φ and ψ are τ-formulas, then $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $\neg\varphi$ are *τ-formulas*. If x is a variable, and φ is a τ-formula, then $\forall x\varphi$ and $\exists x\varphi$ are *τ-formulas*. A *sentence* is a formula in which every variable in a subformula is in the scope of a corresponding quantifier. A formula is in *prenex normal form* if it is of the form $Q_1 x_1 \ldots Q_\ell x_\ell \psi$, such that ψ is a quantifier-free formula and $Q_1 \ldots Q_\ell \in \{\exists, \forall\}^*$ is the *prefix*. We define the *quantifier rank* $\mathrm{qr}(\varphi)$ as the maximum number of nested quantifiers occurring in φ. If φ is an atomic formula, then $\mathrm{qr}(\varphi) = 0$. If $\varphi := \neg\varphi'$, then $\mathrm{qr}(\varphi) = \mathrm{qr}(\varphi')$. If $\varphi := (\psi \,\square\, \theta)$, where $\square \in \{\wedge, \vee\}$, then $\mathrm{qr}(\varphi) = \max\{\mathrm{qr}(\psi), \mathrm{qr}(\theta)\}$. If $\varphi := Qx\psi$, where $Q \in \{\exists, \forall\}$, then $\mathrm{qr}(\varphi) = \mathrm{qr}(\psi) + 1$.

We define six structural parameters for a first-order formula. For a fixed formula φ, we define τ_φ as the set of symbols occurring in the formula φ. Then we denote *the number of relation symbols* in τ_φ by $\#\mathrm{r}(\varphi)$, *the number of function symbols* in τ_φ by $\#\mathrm{f}(\varphi)$, *the maximum arity of a symbol in τ_φ* by $\mathrm{ar}(\varphi)$, *the number of terms* occurring in φ by $|T|$, where T is the set of terms in φ, and *the maximum size of a term* in φ by $|\varphi_{\mathrm{term}}|$. The last two parameters will be considered in Section 3.4.

Let $\mathfrak{A}$ be a τ-structure and $a_1, a_2, \ldots, a_m$ be elements of the domain. If $\varphi(x_1, x_2, \ldots x_m)$ is a τ-formula with free variables $x_1, x_2, \ldots x_m$, then we write $\mathfrak{A} \models \varphi(a_1, a_2, \ldots a_m)$ to denote that $\mathfrak{A}$ satisfies φ if $x_1, x_2, \ldots x_m$ are interpreted by $a_1, a_2, \ldots a_m$, respectively. If φ is a sentence, then we write $\mathfrak{A} \models \varphi$ to denote that $\mathfrak{A}$ satisfies φ, or that $\mathfrak{A}$ is a model of φ.

Definition 3.1 (Prefix-Vocabulary Classes (BÖRGER *et al.*, 2001)**).** A prefix-vocabulary fragment $[\Pi, \bar{p}, \bar{f}]$ is a set of first-order formulas in the prenex normal form, without equality, where Π is a quantifier pattern, *i.e.*, a string on $\{\exists, \forall, \exists^*, \forall^*\}$ denoting a set of quantifier prefixes, $\bar{p}$ is a relation arity sequence $(p_1, p_2, \ldots)$, and $\bar{f}$ is a function arity sequence $(f_1, f_2, \ldots)$ where $p_a, f_a \in \mathbb{N} \cup \{\omega\}$ are the number of relations and functions of arity a, respectively. Occasionally, we use **all** to denote an arbitrary sequence of arities, or an arbitrary prefix. We denote an empty sequence $(0, 0, \ldots)$ by (0). In case $\bar{f} = (0)$, we may write $[\Pi, \bar{p}]$ instead of $[\Pi, \bar{p}, (0)]$. The prefix-vocabulary fragment $[\Pi, \bar{p}, \bar{f}]_=$ is defined in the same way, but allowing formulas with the equality symbol $=$.

For example, $[\exists^*, \mathrm{all}, \mathrm{all}]_=$ is the set of all existential first-order sentences with an arbitrary vocabulary.

Parameterized Satisfiability for First-Order Fragments

Now, we are going to introduce the parameterized satisfiability problem for first-order logic.

Definition 3.2 (The Parameterized Satisfiability Problem). Let X be a fragment of first-order logic, and κ be some parameterization. We define p-κ-SAT(X) in the following way:

p-κ-SAT(X)	
Instance:	A first-order formula $\varphi \in X$.
Parameter:	κ, some parameterization.
Problem:	Decide whether φ is a satisfiable sentence.

If we consider a list of parameters L, we denote p-[L]-SAT(X) as the parameterized satisfiability problem with parameterization $\kappa(\varphi) := \sum_{\iota \in L} \iota(\varphi)$ such that ι is a parameterization function.

The parameters considered here are:

- $\mathrm{qr}(\varphi)$. the quantifier rank,
- $\#\mathrm{r}(\varphi)$, the *number of relation symbols* in τ_φ,
- $\#\mathrm{f}(\varphi)$, the *number of function symbols* in τ_φ,
- $\mathrm{ar}(\varphi)$ the arity of τ_φ,
- $|T|$ the *number of terms* occurring in φ where T is the set of terms in φ, and
- $|\varphi_{\mathrm{term}}|$, the maximum size of a term in φ.

A combination of these parameters leads to different parameterized problems. For example, we have p-qr-SAT(X) is the parameterized satisfiability for the class X when considering the quantifier rank as the parameter. Similarly, for p-#r-SAT(X) when considering the number of relation symbols as the parameter, and p-$[\mathrm{qr}, \#\mathrm{r}]$-SAT$(X)$ when considering both parameters.

Decidability and Computational Complexity

Here we reproduce the results of decidability and computational complexity for these prefix-vocabulary classes presented in (BÖRGER *et al.*, 2001). Let us consider the first-order satisfiability in its classical form. The *classification problem* for the satisfiability of first-order logic considers then, for each subset $X \subseteq FO$, if SAT(X) is decidable, or undecidable (BÖRGER *et al.*, 2001). In Chapter 6 of (BÖRGER *et al.*, 2001), the decidable cases are divided into

Tabela 3 – Maximal Prefix-Vocabulary classes.

Prefix-Vocabulary Class	**Reference**
(1) $[\exists^*\forall^*, \text{all}]_=$,	(Bernays, Schönfinkel 1928) (BERNAYS; SCHÖNFINKEL, 1928)
(2) $[\exists^*\forall^2\exists^*, \text{all}]$,	(Gödel 1932, Kalmár 1933, Schütte 1934)
	(GÖDEL, 1932; KALMÁR, 1933; SCHÜTTE, 1934)
(3) $[\text{all}, (\omega), (\omega)]$,	(Löb 1967, Gurevich 1969) (LÖB, 1967; GUREVICH, 1969)
(4) $[\exists^*\forall\exists^*, \text{all}, \text{all}]$,	(Gurevich 1973) (GUREVICH, 1973)
(5) $[\exists^*, \text{all}, \text{all}]_=$,	(Gurevich 1976) (GUREVICH, 1976)
(6) $[\text{all}, (\omega), (1)]_=$,	(Rabin 1969) (RABIN, 1969)
(7) $[\exists^*\forall\exists^*, \text{all}, (1)]_=$,	(Shelah 1977) (SHELAH, 1977)

Fonte: “The Classical Decision Problem” (BÖRGER *et al.*, 2001).

maximal (see Table 3), *classical* (see Table 4), and *modest complexity* classes (see Table 5). This division has didactic and chronological importance and it gives different complexity analysis.

We are mainly interested in the *classical* and *modest complexity* classes restricted to their relational cases while we observe that a general conversion of first-order formulas over finite domains to propositional formulas handles the parameterized complexity classification. Here, this procedure is understood as a transformation from a first-order sentence to a propositional formula considering some finite domain. The *classical* classes are depicted in Table 4 and we will explore them in Section 3.2.

Tabela 4 – Classical Prefix-Vocabulary classes.

Prefix-Vocabulary Class	**Reference**
(A) $[\text{all}, (\omega)]_{(=)}$	**(Löwenheim, 1915)** (LÖWENHEIM, 1915)
(B) $[\exists^*\forall^*, \text{all}]$	**(Bernays, Schönfinkel 1928)** (BERNAYS; SCHÖNFINKEL, 1928)
(C) $[\exists^*\forall\exists^*, \text{all}]$	**(Ackermann 1928)** (ACKERMANN, 1928)
(D) $[\exists^*\forall^2\exists^*, \text{all}]$	**(Gödel 1932, Kalmár 1933, Schütte 1934)**
	(GÖDEL, 1932; KALMÁR, 1933; SCHÜTTE, 1934)

Fonte: “The Classical Decision Problem” (BÖRGER *et al.*, 2001).

The strategy for decidability for most of these classes is carried out by the *finite model property*. For a class X with the finite model property, one can think of an algorithm that iterates over the structure size. For each possible structure $\mathfrak{A}$ with that fixed size, it verifies whether $\mathfrak{A} \models \varphi$ and, simultaneously, verifies if $\neg\varphi$ is a valid sentence. Moreover, it is possible to obtain an upper bound on the size of the structure. The following lemma specifies the size of the model for formulas in the classical classes, and we will use them in most of the results of

Section 3.2 and 3.3.

Lemma 3.3. *(BÖRGER et al., 2001; SCHÜTTE, 1934)*

(i) *Let ψ be a satisfiable sentence in $[all, \omega]$. Then ψ has a model with at most 2^m elements where ψ has m monadic predicates.*

(ii) *Let ψ be a satisfiable sentence in $[all, \omega]_=$. Then ψ has a model with at most $q \cdot 2^m$ elements where ψ has quantifier rank q and m monadic predicates.*

(iii) *Let $\psi := \exists x_1 \ldots \exists x_p \forall y_1 \ldots \forall y_m \varphi$ be a satisfiable sentence in $[\exists^* \forall^*, all]_=$. Then ψ has a model with at most $\max(1, p)$ elements.*

(iv) *Let $\psi := \exists x_1 \ldots \exists x_p \forall y_1 \forall y_2 \exists z_1 \ldots \exists z_m \varphi$ be a satisfiable sentence in $\exists^p \forall^2 \exists^m$ containing t predicates of maximal arity h. Then ψ has a model with cardinality at most*

$$4^{10tm^2 2^h (p+1)^{h+4}} + p.$$

In some cases, an upper bound on the running time of the satisfiability problem can be found. Using nondeterminism, we can guess a structure with a size less than or equal to the size provided by the finite model property, and then we evaluate the input formula on this structure. For example, the satisfiability problem for $[\text{all}, (\omega)]$ is in NTIME(2^n), where n is the size of the formula and, for the class $[\exists^* \forall^2 \exists^*, \text{all}]$, the same problem is in NTIME($2^{n/\log n}$). The complexity of the satisfiability for most of these classes is addressed in (LEWIS, 1980; FÜRER, 1981; GRÄDEL, 1989).

The second group of prefix-vocabulary classes that we are interested in this work are those on which the satisfiability problem is in P, NP, Co-NP, Σ_2^p, Π_2^p, and PSPACE designated as *modest classes* (BÖRGER *et al.*, 2001, Section 6.4). In Table 5, we summarize the description of modest complexity classes with their respective complexity result from (BÖRGER *et al.*, 2001).

We give an example of how this classification works for the monadic classes in Fig. 2. The class $[\text{all}, (\omega)]$ and $[\text{all}, (\omega)]_=$ called the *Löwenhein class* and the *Löwenhein class with equality*, or, alternatively, *relational monadic fragments*, and all classes below these classes are considered as classes of modest complexity, and, $[\text{all}, (\omega), (\omega)]$ the *full monadic class*, and $[\text{all}, (\omega), (1)]_=$ the *Rabin's class* are maximal with respect to decidability.

3.2 Parameterized Complexity of Classical Classes

Our strategy to prove that p-$[L]$-SAT(X) is fixed-parameter tractable, for some prefix-vocabulary class X and some list of parameters L, is to present an fpt-reduction to the

Tabela 5 – Prefix-vocabulary classes of modest complexity

Prefix-Vocabulary Class	Complexity classification
$[\exists\forall^*, \text{all}]_=$ $[\exists^*\forall^u, \text{all}]_=$ for $u \in \mathbb{N}$ $[\exists^p\forall^2\exists^*, \bar{s}]$ for $p \in \mathbb{N}$ and $\bar{s}$ finite $[\exists^p\forall\exists^*, \bar{s}]_=$ for $p \in \mathbb{N}$ and $\bar{s}$ finite $[\Pi_t, (m)]_=$ $t, m \in \mathbb{N}$, and Π_t containing at most t universal quantifiers	**NP**
$[\exists^*, (0)]_=$ $[\exists^*, (1)]$ $[\exists, (\omega)]$ $[\forall, (\omega)]$	**NP-complete**
$[\exists^p\forall^*, \bar{s}]_=$ for $p \in \mathbb{N}$ and $\bar{s}$ finite $[\Pi_t, (m)]_=$ for $t, m \in \mathbb{N}$,and Π_t containing at most t existential quantifiers	**Co-NP**
$[\exists^2\forall^*, (0)]_=$ $[\exists^2\forall^*, (1)]$ $[\forall^*\exists, (0)]_=$ $[\forall^*\exists, (1)]$ $[\forall\exists\forall^*, (0)]_=$ $[\forall\exists\forall^*, (1)]$	**Co-NP-complete**
$[\exists^*\forall^*, (0)]_=$ $[\exists^*\forall^*, (1)]$ $[\exists^2\forall^*, (\omega)]$	**Σ_2^p-complete**
$[\forall^*\exists^*, (0)]_=$ $[\forall^*\exists^*, (\omega)]$	**Π_2^p-complete**
$[\exists^*\forall\exists, (0,1)]$ $[\forall\exists, (\omega)]$	**PSPACE-complete**

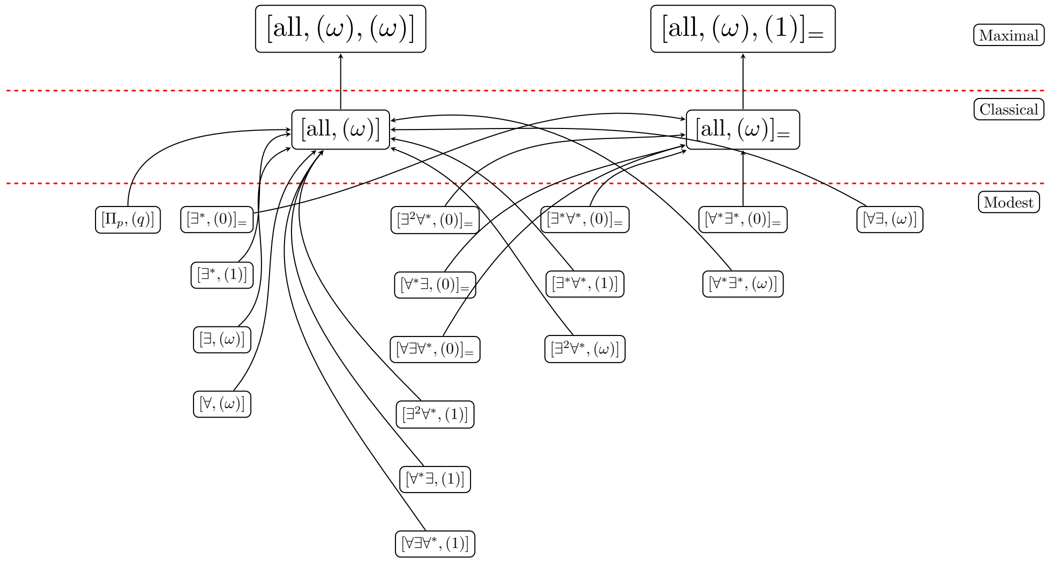
Fonte: "The Classical Decision Problem" (BÖRGER *et al.*, 2001).

propositional satisfiability problem p-SAT. As we know, p-SAT is in FPT and FPT is closed under fpt-reductions, then we obtain that p-$[L]$-SAT(X) is in FPT too[1]. The essential tools for these results are the finite model property and the conversion to propositional formulas that we outline below.

In the conversion of a first-order formula into a propositional one, the finite model

[1] Another way to see the fixed-parameter tractability for these problems can be obtained by cycling over the structures up to some domain size limits imposed by the finite model property.

Figura 2 – The inclusion relation for monadic classes with modest complexity on Löwenheim's classes, and the maximal classes $[\text{all}, (\omega), (\omega)]$, $[\text{all}, (\omega), (1)]_=$



Fonte: "The Classical Decision Problem" (BÖRGER *et al.*, 2001).

property provides a bound on the number of propositional variables since they represent the description of the finite structure. As we see in Theorem 3.4, the quantifier rank is intrinsically related to the conversion due to the replacement of the quantifiers by their corresponding connectives. In the next subsection, we present our first result, and it will act as a prototypical argument for the other proofs.

The Löwenheim Class and the Löwenheim Class with Equality

For the monadic fragments $[\text{all}, (\omega)]$ and $[\text{all}, (\omega)]_=$, we can show that the parameterized satisfiability problem is in FPT when parameterized by the quantifier rank and the number of relation symbols. Moreover, based on Proposition 3.6 given below, we observe evidence of intractability when parameterized by the quantifier rank only.

Theorem 3.4. *The satisfiability problem* p-[qr,#r]-SAT$([\text{all}, (\omega)])$ *is in FPT.*

Proof. We give an fpt-reduction to p-SAT. Let $\varphi \in [\text{all}, (\omega)]$ be a satisfiable formula with r monadic relation symbols and quantifier rank q. By Lemma 3.3-*(i)*, there is a model with at most 2^r elements. As we transform φ into a propositional formula φ^*, we represent each relation by 2^r propositional variables. More precisely, for each relation R_i with $1 \leq i \leq r$, and for each element $j \in [2^r]$, we use the variable p_{ij} to represent the truth value of $R_i(j)$. The translation works as follows. By structural induction on φ, apply the conversion of existential quantifiers into big disjunctions, and universal quantifiers into big conjunctions. We formally define this conversion as:

$$\varphi^* = \begin{cases} p_{ij} & \text{If } \varphi := R_i(j) \text{ is an atomic formula;} \\ \bigvee_{j\in[2^r]} (\psi[x/j])^* & \text{If } \varphi := \exists x\psi; \\ \bigwedge_{j\in[2^r]} (\psi[y/j])^* & \text{If } \varphi := \forall y\psi. \end{cases}$$

It is easy to see that φ has a model if and only if φ^* is a satisfiable formula. Each inductive step constructs a formula of size $2^r \cdot |\varphi|$, and the whole process takes $O((2^r)^q \cdot n)$ where n is the size and q is the quantifier rank of φ. As the number of variables is bounded by $k \cdot 2^r$, this leads to the desired fpt-reduction. □

The satisfiability problem for the Löwenheim class with equality $[\text{all}, (\omega)]_=$ is also in FPT when considering the quantifier rank and the number of monadic relations.

Theorem 3.5. *The satisfiability problem* p-[qr,#r]-SAT$([\text{all}, (\omega)]_=)$ *is in FPT.*

Proof. Using the same idea of Theorem 3.4, and the finite model property from Lemma 3.3.*(ii)*, we can describe an fpt-reduction from p-[qr,#r]-SAT$([\text{all}, (\omega)]_=)$ to p-SAT.

For a satisfiable formula $\varphi \in [\text{all}, (\omega)]_=$ with at most $r = \#\text{r}(\varphi)$ monadic relation symbols and quantifier rank $q = \text{qr}(\varphi)$, there is a model with at most $q \cdot 2^r$ elements by Lemma 3.3.*(ii)*. The number of steps on the conversion is bounded by $O((q \cdot 2^r)^q \cdot n)$ where n is the formula size. Each atomic formula (including those with equality symbol) is converted into a propositional variable, and this number is a function of the size of the domain and the size of the vocabulary, hence a function of q and r. Then, the whole process can be done in FPT. □

However, when we choose the quantifier rank as the parameter, it is unlikely to obtain an fpt-algorithm for the satisfiability of the Löwenheim's class.

Proposition 3.6. *Unless $P = NP$,* p-qr-SAT$([\text{all}, (\omega)])$ *is not in XP.*

Proof. Assume, by contradiction, that p-qr-SAT$([\text{all}, (\omega)])$ is in XP. Then there is an algorithm that solves the problem in time $f(q) \cdot n^{g(q)}$, where n is the size of the formula, q is the quantifier rank of the input formula, and f and g are computable functions. Hence, for the first slice of the problem, $[\exists, (\omega)]$ and $[\forall, (\omega)]$ (see Table 5), there is an algorithm that runs in $f(1) \cdot n^{g(1)} \in O(n)$. This is a contradiction with the fact that these problems are NP-complete and with the reasonable assumption that $\text{P} \neq \text{NP}$. □

As a consequence of Theorem 2.15 and that $[\exists, (\omega)]$ and $[\forall, (\omega)]$ are NP-complete problems, we have that p-qr-SAT$[\text{all}, (\omega)]$ is para-NP-hard under fpt-reductions.

Corollary 3.7. p-qr-SAT$([\text{all}, (\omega)])$ *is paraNP-hard.*

Proof. The result follows directly from Theorem 2.15. □

The Bernays-Schönfinkel-Ramsey Class

Taking the same idea from Theorem 3.4 for the monadic class, if we choose a parameter that bounds the number of propositional variables, and a conversion procedure that can be conducted in FPT, we can provide an fpt-reduction for the parameterized satisfiability of prefix-vocabulary classes to p-SAT. This is the case of the Bernays-Schönfinkel-Ramsey class $[\exists^*\forall^*, \text{all}]_=$ when parameterized by the quantifier rank, number of relations, and arity of τ_φ.

Theorem 3.8. p-[qr, #r, ar]-SAT$([\exists^*\forall^*, \text{all}])$ *and* p-[qr, #r, ar]-SAT$([\exists^*\forall^*, \text{all}]_=)$ *are in FPT.*

Proof. Let φ be a satisfiable formula in $[\exists^*\forall^*, \text{all}]$ in the form $\exists x_1 \ldots \exists x_k \forall y_1 \ldots \forall y_\ell \psi$. By Lemma 3.3.*(iii)*, φ has a model of size at most $k \leq \text{qr}(\varphi)$. Then it will be necessary at most $\#\text{r}(\varphi) \cdot k^{\text{ar}(\varphi)}$ propositional variables to represent the whole structure data. Applying the conversion described in Theorem 3.4 considering the finite domain, we will produce a satisfiable propositional formula with the number of variables bounded by $g(\text{qr}(\varphi), \#\text{r}(\varphi), \text{ar}(\varphi))$, for some computable function g. Clearly, this reduction can be done in FPT.

The same argument can be applied to the Ramsey's class $[\exists^*\forall^*, \text{all}]_=$. □

The Ackermann and Gödel-Kalmár-Shütte Classes

For the Ackermann class $[\exists^*\forall\exists^*, \text{all}]$ and Gödel-Kalmár-Shütte class $[\exists^*\forall^2\exists^*, \text{all}]$, by Lemma 3.3.*(iv)*, the finite model property provides a model with size related to the parameters of the class definition (the number of existential quantifiers, the number of relations, the maximum arity). If we consider the quantifier rank and the number of relation symbols as the parameters, we can show it is in FPT.

Theorem 3.9. p-[qr, #r, ar]-SAT$([\exists^*\forall\exists^*, \text{all}])$ *and* p-[qr, #r, ar]-SAT$([\exists^*\forall^2\exists^*, \text{all}])$ *are in FPT.*

Proof. For example, consider $[\exists^*\forall^2\exists^*, \text{all}]$. Let $\varphi := \exists x_1 \ldots \exists x_k \forall y_1 \forall y_2 \exists z_1 \ldots \exists z_\ell \psi$ be a first-order formula in a vocabulary with r relation symbols of maximum arity $\text{ar}(\varphi)$. By Lemma 3.3.*(iv)*[1],

[1] Unfortunately, the finite model property for the class $\exists^*\forall^2\exists^*$ from (SCHÜTTE, 1934) is hard to follow.

there is a model of size bounded by $s := 4^{10\cdot r\cdot \ell^2\cdot 2^{\mathrm{ar}(\varphi)}\cdot (k+1)^{\mathrm{ar}(\varphi)+4}} + k$ that satisfies φ. There is a bound for a satisfiable structure of φ. Considering the conversion to propositional formula, the number of propositional variables will be bounded by $r \cdot s^{\mathrm{ar}(\varphi)}$. By structural induction on φ, apply the conversion of existential quantifiers into big disjunctions, and universal quantifiers into big conjunctions. Then it introduces one propositional variable to each possible assignment of tuples and relation symbols. This conversion is clearly a function of $s, r, \mathrm{ar}(\varphi)$ and n, the size of φ. This lead to an fpt-reduction to p-SAT.

The same argument can be applied to the Ackermann's class $[\exists^*\forall\exists^*, \mathrm{all}]$. □

The results presented in this section are summarized in Table 6.

Tabela 6 – The parameterized complexity of the classical solvable cases.

Problem	**Result**
p-[qr, #r]-SAT$([\mathrm{all}, (\omega)])$	FPT (Theorem 3.4)
p-qr-SAT$([\mathrm{all}, (\omega)])$	paraNP-hard (Corollary 3.7)
p-[qr, #r]-SAT$([\mathrm{all}, (\omega)]_=)$	FPT (Theorem 3.5)
p-[qr, #r, ar]-SAT$([\exists^*\forall^*, \mathrm{all}]_{(=)})$	FPT (Theorem 3.8)
p-[qr, #r, ar]-SAT$([\exists^*\forall\exists^*, \mathrm{all}])$	FPT (Theorem 3.9)
p-[qr, #r, ar]-SAT$([\exists^*\forall^2\exists^*, \mathrm{all}])$	FPT (Theorem 3.9)

Fonte: Made by the author himself.

3.3 Parameterized Complexity of Modest Complexity Classes

In this section, we analyze the parameterized complexity of prefix-vocabulary classes with modest complexity. We summarize our results on Table 7. For these classes, it is possible to point out a parameter that put the problem in FPT. First, for some of these classes, the inclusion of the relational monadic class leads to an fpt result for the same parameters chosen. Then, we begin with a corollary of Theorem 3.4.

Corollary 3.10.

(i) p-#r-SAT(X) *is in FPT for* $X \in \{[\exists, (\omega)], [\forall, (\omega)], [\forall\exists, (\omega)]\}$.

(ii) p-[qr, #r]-SAT(X) *is in FPT for* $X \in \{[\exists^2\forall^*, (\omega)], [\forall^*\exists^*, (\omega)]\}$.

Proof. Let $\varphi \in X$ and $r = \#\mathrm{r}(\varphi)$. We already know that the finite model property gives us, for a satisfiable formula, a model of size at most 2^r. Then the same conversion procedure used in Theorem 3.4 leads to an fpt-reduction to p-SAT. In these cases, the quantifier rank is a constant.

The claim *(ii)* follows directly from Theorem 3.4. □

Again, we can apply Lemma 3.3.(i) to give an fpt-reduction for the satisfiability problem of some classes with modest complexity.

Theorem 3.11. *p-qr-SAT(X) is in FPT for* $X \in \{[\forall^*\exists,(1)],[\forall\exists\forall^*,(1)],[\exists^*\forall^*,(1)]\}$.

Proof. Consider the satisfiability problem for the class $[\forall^*\exists,(1)]$. By Lemma 3.3.(i), a satisfiable formula $\varphi := \forall x_1 \ldots \forall x_k \exists y \psi$ with one monadic relation and quantifier rank $q = \mathrm{qr}(\varphi)$ has a model with size at most 2. Applying the same conversion of Theorem 3.4, the number of steps will be bounded by $2^q \cdot n$ where n is the size of φ. This will lead to an fpt-reduction to a propositional formula with two propositional variables.

The same reduction works for $[\forall\exists\forall^*,(1)]$ and $[\exists^*\forall^*,(1)]$. □

In the sequence, we can use Lemma 3.3.(ii) to obtain a reduction from the satisfiability of some classes with modest complexity to p-SAT when we use the quantifier rank as a parameter.

Theorem 3.12. *p-qr-SAT(X) is in FPT for* $X \in \{\,[\Pi_t,(m)]_=,\ [\Pi_t,(m)]_=,$
$[\forall^*\exists,(0)]_=,[\forall\exists\forall^*,(0)]_=,[\forall^*\exists^*,(0)]_=\}$.

Proof. Consider the satisfiability problem for the class $[\Pi_t,(m)]_=$. Let φ be a satisfiable formula in $[\Pi_t,(m)]_=$ with at most m monadic relations and quantifier rank q. Then, by Lemma 3.3.*(ii)*, φ has a model with size at most $q \cdot 2^m$. Each monadic relation can be represented by $q \cdot 2^m$ propositional variables. So, in order to transform φ into a propositional formula, $q \cdot m \cdot 2^m$ propositional variables will be necessary to represent all relations. Applying the same conversion of Theorem 3.5, it will lead to an fpt-reduction to p-SAT.

The same reduction holds for $[\Pi_t,(q)]_=$ with at most t existential quantifiers. For $[\forall^*\exists,(0)]_=$, $[\forall\exists\forall^*,(0)]_=$, and $[\forall^*\exists^*,(0)]_=$, the size of the structure is bounded by $qr(\varphi)$ and the reduction follows in the same way. □

Formulas with a leading block of existential quantifiers can be handled with the finite model property by Lemma 3.3. (iii).

Theorem 3.13. *p-[qr, #r, ar]-SAT(X) is in FPT for* $X \in \{[\exists\forall^*,all]_=,[\exists^*\forall^u,all]\}$.

Proof. Consider the satisfiability problem for the class $[\exists^*\forall^u, \text{all}]$. Let φ be a satisfiable formula in $[\exists^*\forall^u, \text{all}]$ with fixed natural u. So, $\varphi := \exists x_1 \ldots \exists x_k \forall y_1 \ldots \forall y_u \psi$. By Lemma 3.3.*(iii)*, φ has a model with size at most $k \leq \text{qr}(\varphi)$. Let τ_φ the vocabulary of φ with maximum arity $\text{ar}(\varphi)$ and size $\#\text{r}(\varphi)$. Applying the same conversion presented in Theorem 3.4, it will return a propositional formula with at most $\#\text{r}(\varphi) \cdot \text{qr}(\varphi)^{\text{ar}(\varphi)}$ propositional variables, and the whole process can be carried out by an fpt-algorithm.

For the class $[\exists\forall^*, \text{all}]_=$, the finite model property will provide a model of size 1, and each universally quantified variable could be handled as a dummy variable. □

Theorem 3.14. *p-qr-SAT(X) is in FPT for* $X \in \{[\exists^*, (0)]_=, [\exists^*, (1)], [\exists^p\forall^*, \bar{s}]_=,$ $[\exists^2\forall^*, (0)]_=, [\exists^2\forall^*, (1)], [\exists^*\forall^*, (0)]_=\}$.

Proof. Consider the satisfiability problem for $[\exists^*\forall^*, (0)]_=$. By Lemma 3.3.*(*iii), for a satisfiable formula $\varphi := \exists x_1 \ldots \exists x_k \forall y_1 \ldots \forall y_l \psi$ there is a model with size $k \leq \text{qr}(\varphi)$. Applying the same translation of Theorem 3.4, and considering that only the equality symbol is in τ_φ, the number of propositional variables obtained in the reduction is bounded by a function of $\text{qr}(\varphi)$. This will lead to an fpt-reduction to p-SAT. □

Theorem 3.15. *p-qr-SAT(X) is in FPT for* $X \in \{\, [\exists^p\forall^2\exists^*, \bar{s}],\ [\exists^p\forall\exists^*, \bar{s}]_=,$ $[\exists^*\forall\exists, (0,1)]\}$.

Proof. We will consider the satisfiability problem of $[\exists^p\forall^2\exists^*, s]$ as an example. Let $\varphi :=$ $\exists x_1 \ldots \exists x_p \forall y_1 \forall y_2 \exists z_1 \ldots \exists z_k \psi$ be a first-order formula with a fixed vocabulary with r relation symbols of arity $\text{ar}(\varphi)$. By Lemma 3.3.(*iv*), there is a model of size bounded by $\ell := 4^{10r \cdot k^2 2^{\text{ar}(\varphi)} (p+1)^{\text{ar}(\varphi)+4}} +$ p that satisfies φ. As $p, r, \text{ar}(\varphi)$ are constants, the size of this model is a function of k.

Then we can describe each relation with at most $\ell^{\text{ar}(\varphi)}$ propositional variables. All relations are described by a binary string with length $r \cdot \ell^{\text{ar}(\varphi)}$. By structural induction on φ, apply the conversion procedure of Theorem 3.4. Introduce one propositional variable to each possible assignment of a tuple to a relation symbol, which is a function of k. This conversion is a function of k and n, the size of φ. This process leads to an fpt reduction to p-SAT. □

We summarize the results of this section in Table 7.

Tabela 7 – Prefix-vocabulary classes of modest complexity in which their satisfiability problem is in FPT with respect to some parameter.

Prefix-vocabulary Class	**Complex. SAT**	**Complex. p-κ-SAT**	
		Param. κ	Result
$[\exists\forall^*, \text{all}]_=$	**NP**	(qr+vs+ar)	FPT(Theo. 3.13)
$[\exists^*\forall^u, \text{all}]_=$ for $u \in \mathbb{N}$		(qr+vs+ar)	FPT (Theo. 3.13)
$[\exists^p\forall^2\exists^*, \bar{s}]$ for $p \in \mathbb{N}$ and $\bar{s}$ finite		qr	FPT (Theo. 3.15)
$[\exists^p\forall\exists^*, \bar{s}]_=$ for $p \in \mathbb{N}$ and $\bar{s}$ finite		qr	FPT (Theo. 3.15)
$[\Pi_t, (m)]_=$ $t, m \in \mathbb{N}$, Π_t containing at most t universal quantifiers		qr	FPT (Theo. 3.12)
$[\exists^*, (0)]_=$	**NP-complete**	qr	FPT (Theo. 3.14)
$[\exists^*, (1)]$		qr	FPT (Theo. 3.14)
$[\exists, (\omega)]$		vs	FPT (Cor. 3.10)
$[\forall, (\omega)]$		vs	FPT (Cor. 3.10)
$[\exists^p\forall^*, \bar{s}]_=$ for $p \in \mathbb{N}$ and $\bar{s}$ finite $\lvert\tau\rvert$	**Co-NP**	qr	FPT (Theo. 3.14)
$[\Pi_t, (m)]_=$ for $t, m \in \mathbb{N}$, and Π_t containing at most t existential quantifiers		qr	FPT (Theo. 3.12)
$[\exists^2\forall^*, (0)]_=$	**Co-NP-complete**	qr	FPT (Theo. 3.14)
$[\exists^2\forall^*, (1)]$		qr	FPT (Theo. 3.14)
$[\forall^*\exists, (0)]_=$		qr	FPT (Theo. 3.12)
$[\forall^*\exists, (1)]$		qr	FPT (Theo. 3.11)
$[\forall\exists\forall^*, (0)]_=$		qr	FPT (Theo. 3.12)
$[\forall\exists\forall^*, (1)]$		qr	FPT (Theo. 3.11)
$[\exists^*\forall^*, (0)]_=$	**Σ_p^2-complete**	qr	FPT (Theo. 3.14)
$[\exists^*\forall^*, (1)]$		qr	FPT (Theo. 3.11)
$[\exists^2\forall^*, (\omega)]$		(qr+vs)	FPT (Cor. 3.10)
$[\forall^*\exists^*, (0)]_=$	**Π_p^2-complete**	qr	FPT (Theo. 3.12)
$[\forall^*\exists^*, (\omega)]$		(qr+vs)	FPT (Cor. 3.10)
$[\exists^*\forall\exists, (0,1)]$	**PSPACE-complete**	qr	FPT (Theo. 3.15)
$[\forall\exists, (\omega)]$		vs	FPT (Cor. 3.10)

Fonte: Made by the author himself.

3.4 Parameterized Complexity of $[\mathbf{all}, (\omega), (\omega)]$ and $[\exists^*, \mathbf{all}, \mathbf{all}]_=$

Now we consider the functional classes $[\text{all}, (\omega), (\omega)]$ and $[\exists^*, \text{all}, \text{all}]_=$. If we look at those classes that are maximal concerning the finite model property (Table 2), it will remain the analysis of those with function symbols. Among them, we can apply the same argument of the previous sections for the classes that have an upper bound on the size of the structure. For others, we cannot attach the finite model property with some bound on the size of the structure. We only have an existential condition for a structure of a satisfiable formula within the class.

To handle prefix-vocabulary classes with function symbols, someone has to replace the function symbols to reduce to a relational class with finite model property or find a finite model based on possible ground terms constructed by the formula. Then, we can attest the fixed-parameter tractability of these functional classes when parameterized by the parameters that we already considered (quantifier rank, number of relations, and the arity of the vocabulary) with the number of functions, the number of terms, and the maximum size of a term.

The Löb-Gurevich class $[\textbf{\textit{all}}, (\omega), (\omega)]$

For the class $[\text{all}, (\omega), (\omega)]$, we can show that the parameterized satisfiability problem is in FPT when parameterized by the quantifier rank, the number of monadic relation symbols, the number of unary function symbols, and the maximum size of the terms. We need to modify a Lemma from (GRÄDEL, 1989) removing the unary function symbols. The lemma shows that every monadic formula φ of length n can be converted into a formula $\psi \in [\text{all}, (n), (0)]$ satisfiable over the same domains as ψ and with some bound on quantifier rank.

Lemma 3.16. *(GRÄDEL, 1989) Let φ be a formula in the prenex normal form with size n, quantifier rank q, r monadic relations, f unary function symbols, and with terms of the form $f^i x_j$ such that $i < t$ for some constant $t < n$. Then, there is an equivalent formula ψ without functions.*

Proof. Consider a first-order sentence φ as in the claim. For each $R_i, f_j \in \tau_\varphi$, we introduce a new monadic relation $Q_{\overline{ij}}$ for a term in the form $R_i f_j t$ for some arbitrary term t. Then φ is satisfiable over the same domains as

$$\varphi[R_i f_j t / Q_{\overline{ij}} t] \wedge \forall x (R_i f_j x \leftrightarrow Q_{\overline{ij}} x).$$

To each possible sequence of a relation and function symbol, apply the previous process.

Occasionally, we reach $Q_{\bar{a}} f_i t$ for some index $\bar{a}$. Thus, a new monadic relation $Q_{\overline{ai}}$ should be added. Taking a nested sequence of functions with size at most t, this will lead to a maximum of $O(r \cdot s^t)$ new monadic relation symbols.

Then we arrive at $\alpha \wedge \forall x \beta$ where α is a formula without function symbols and β a conjunction of formulas of the form $R_i f_j x \leftrightarrow Q_{\overline{ij}} x$ and $Q_{\bar{a}} f_j x \leftrightarrow Q_{\overline{aj}} x$. Let $f_1, \ldots, f_s$ be function symbols in β. Then, $\forall x \beta$ is the Skolem normal form of $\forall \exists y_1 \ldots \exists y_s \beta[f_i x / y_i]$ which is a relational formula, and

$$\psi := \alpha \wedge \forall x \exists y_1 \ldots \exists y_s \beta[f_i x / y_i]$$

is satisfiable over the same domains as φ. So ψ have at most $O(r + r \cdot s^t)$ monadic relation symbols with the quantifier rank bounded by $q + s + 1$. Thus ψ is in $[\text{all}, (r + r \cdot s^t), (0)]$. $\square$

Using the previous lemma, we can achieve the fixed-parameter tractability.

Theorem 3.17. *The satisfiability problem p-*[qr, #r, #f, $|\varphi_{\text{term}}|$]-SAT($[\text{all}, (\omega), (\omega)]$ *is in FPT.*

Proof. We give an fpt-reduction to p-SAT. Let $\varphi \in [\text{all}, (\omega), (\omega)]$ be a satisfiable formula with size n, quantifier rank q, r monadic relation symbols, s function symbols, and terms with size at most t. Using the Lemma 3.16, we obtain a first-order formula $\psi \in [\text{all}, (r + r \cdot s^t), (0)]$. By Lemma 3.3-*(i)*, there is a model with at most $2^{(r + r \cdot s^t)}$ elements with quantifier rank bounded by $q + s + 1$. The next step follow similarly to the relational case. As we will transform ψ into a propositional formula ψ^*, we will represent each relation by $2^{(r + r \cdot s^t)}$ propositional variables. More precisely, for each relation R_i with $1 \leq i \leq (r + r \cdot s^t)$, and for each element $j \in [2^{(r + r \cdot s^t)}]$, we use the variable p_{ij} to represent the truth value of $R_i(j)$. The conversion follows in the same way as in Theorem 3.4. Once again, it is easy to see that φ has a model if and only if ψ^* is a satisfiable formula. Each inductive step constructs a formula of size $2^{(r + r \cdot s^t)} \cdot |\varphi|$, and the whole process takes $O((2^{(r + r \cdot s^t)})^{q + s + 1} \cdot n)$ where n is the size of φ. As the number of variables is bounded by $(r + r \cdot s^t) \cdot 2^{(r + r \cdot (s)^t)}$, this leads to the desired fpt-reduction. $\square$

The Existential class $[\exists^*, \textit{all}, \textit{all}]_=$

The existential fragment with equality is one of the decidable cases that are maximal concerning the finite model property, and its satisfiability is NP-complete (BÖRGER *et al.*, 2001, pg. 304). The finite model property, then, is obtained through the size of the set of terms occurring in a given existential formula. In this case, we add the number of terms in the parameterization function. For all terms in the form $s = f s_1 \ldots s_r \in T$, we also consider $s_1 \ldots s_r$ and their sub-terms in T.

Lemma 3.18. *(BÖRGER et al., 2001) Let φ a first-order sentence in $[\exists^*, all, all]_=$ with quantifier rank q, and let be T the set of terms occurring in φ with $t = |T|$. Then φ has a model with size $q + t$.*

Theorem 3.19. *p-[qr, #r, #f, ar, $|T|$]-SAT$([\exists^*, \text{all}, \text{all}]_=)$ is in FPT.*

Proof. Let φ be in the form $\exists x_1 \dots \exists x_q \psi$, and let T be the set of terms occurring in φ with $|T| = t$. Also consider that the number of relation and function symbols are r and s, respectively, and that the arity of τ_φ is a. By the Lemma 3.18, there is a model for φ with size at most $q+t$.

All atoms are in the form $Pz_1 \dots z_\ell,\ z_1 = z_2,\ fz_1 \dots z_\ell = z$. It is obvious that the number of atomic formulas are bounded by a computable function in r, s, a, t. For each relation in τ_φ, we will need at most $(q+t)^a$ propositional variables to represent it. So, for all relation symbols, we need $r \cdot (q+t)^a$. For atomic formulas with equality symbol, we need a number of propositional variables bounded by a function in s, a, t.

Applying the conversion from Theorem 3.4, we will achieve an fpt-reduction to p-SAT. □

To summarize, we presented a strategy to provide fixed-parameter tractability for the satisfiability of many prefix-vocabulary classes. Further investigation on the maximal classes without finite model property ($[\text{all}, (\omega), (1)]_=$ and $[\exists^*\forall\exists^*, \text{all}, (1)]_=$) and the remaining classes with functional symbols ($[\forall^*, \text{all}, (1)]_=$, $[\exists^*\forall, \text{all}, (1)]_=$, and $[\exists^*\forall^*, (0), (1)]_=$) is still missing.

In the next chapter, we investigate the parameterized analysis to the matching problem with respect to associative, commutative, and associative-commutative theories.

4 MATCHING PROBLEMS

In this chapter, we consider the parameterized complexity analysis of the matching problem for first-order terms with associative (A), commutative (C), and associative-commutative (AC) function symbols. We extend the results presented in (AKUTSU *et al.*, 2017) considering the membership in W[P] and fixed-parameter tractability concerning different parameters. Particularly, we consider **the number of variables**, **the size of the substitution**, and **the size of the vocabulary**. We also consider the standard of equational unification (BENANAV *et al.*, 1987; BAADER; SNYDER, 2001).

In Sections 4.1 and 4.2, we briefly introduce the unification problem and present some known results related to complexity issues. In Section 4.3, we introduce the parameterized version of associative, commutative, and associative-commutative matching problem, p-κ-E-MATCHING for E $\in$ {A, C, AC}. In Section 4.4, we show the membership in W[P] of p-$|\text{var}(\varphi)|$-C-MATCHING when parameterized by **the number of variables**, and p-$|\theta|$-E-MATCHING for **the size of the substitution** when restricted to E $\in$ {A, AC}. In Section 4.5, we present a brute-force algorithm that guarantees the fixed-parameter tractability of these matching problems when parameterized by the **size of the vocabulary** and **the size of the substitution**. The results contained in this chapter have been presented in (BUSTAMANTE *et al.*, 2019b).

4.1 Introduction

The unification problem consists in verifying the existence of a substitution of variables that turns two first-order terms equal. Let f, g be arbitrary function symbols, a, b constants, and x, y variables. We would like to determine whether $f(x, g(a, b))$ and $f(g(y, b), x)$ have a substitution of the variables that makes them equal. One can check that $\theta := \{x \mapsto g(a, b), y \mapsto a\}$ is a solution for the previous example. Thus, we say that θ *unifies* the terms $f(x, g(a, b))$ and $f(g(y, b), x)$.

Unification has an essential place in many areas as automated reasoning, program verification, logic programming, and term rewriting (BAADER; SNYDER, 2001). Historically, the problem of unification already appeared in the PhD Thesis of Herbrand (HERBRAND, 1930). However, unification has only been proposed explicitly in theorem-proving context for first-order logic in the seminal paper of Robinson (ROBINSON, 1965). He describes a unification algorithm to be applied in the context of first-order logic to most general unifier as opposed to all possible instantiations. The proposed algorithm has a high complexity. It has exponential time and space complexity.

After Robinson's paper, many complexity results were obtained to the unification problem. First, a quadratic algorithm was proposed (ZILLI, 1975). Then a linear algorithm was presented in (PATERSON; WEGMAN, 1978). Finally, the polynomial completeness under log-space reductions was achieved (DWORK *et al.*, 1984). This kind of unification problem is called *Syntactic Unification*, and, once it is decidable in PTIME, it is not interesting from the perspective of parameterized complexity.

On the other hand, if someone may consider a certain axiomatization for which a function symbol must be interpreted, there are more solutions for the unification problem. For example, consider a commutative function f such that $f(x,y) \equiv f(y,x)$, and the previous equality example modulo commutativity of f: $f(x,g(a,b)) =_C f(g(y,b),x)$, then there are many solutions with respect to the substitution of x while y is substituted by a.

In the case presented before, unification is characterized as *Equational Unification*, and, for this version of the problem, many techniques were proposed like *paramodulation* (NIEUWENHUIS; RUBIO, 2001) and *narrowing* (HULLOT, 1980). In the seminal paper "Building-in Equational Theories" of Plotkin (PLOTKIN, 1972), the author showed how to build these tricky axioms into the context of an automated theorem prover without losing completeness.

In this chapter, we are mainly concerned with the matching problem, a restricted version of the unification problem, in which only one term has variables, and the other is a term free of variables. In other words, for a first-order term s and t, the problem asks for a substitution θ such that $s\theta =_E t$ for some equational theory E. Mnemonically and from now on, we denote s ("*source*") to be the term with variables and t ("*target*"), the ground term. They are said to *match* if there is a substitution θ such that $s\theta = t$. In (BENANAV *et al.*, 1987), the associative, commutative, and associative-commutative matching problems are shown to be NP-complete. In the next section, we review these complexity results.

4.2 Complexity of Matching Problem

Let $\mathscr{F}$ be a countable set of function symbols with some arity, and $\mathscr{V}$ a countable set of variables. A *term* t is inductively defined from variables in $\mathscr{V}$ closed under functions $f \in \mathscr{F}$. A function symbol with arity 0 is called a *constant*. We denote by $T(\mathscr{F},\mathscr{V})$ the set of terms built up from $\mathscr{F}$ and $\mathscr{V}$. A *ground term* is a term without variables, and the set of ground terms is denoted by $T(\mathscr{F})$. For a term s, $\mathscr{F}_s$ is the *set of function symbols occurring* in s, $\mathrm{var}(s)$ is the *set of variables occurring* in s, the *size* $|s|$ is the number of symbols in s, and $|s|_{\mathrm{var}}$ is the *maximum number of occurrences of a variable* in s. A function f is associative if it satisfies

$$f(f(x,y),z) = f(x, f(y,z)),$$

and it is commutative if it satisfies

$$f(x,y) = f(y,x).$$

A *substitution* θ is a mapping from the set of variables $\mathscr{V}$ to the set of terms $T(\mathscr{F})$. We are interested in finite substitutions, and we explicitly represent them by $\{x_1 \mapsto t_1, \dots, x_k \mapsto t_k\}$. The *size of a substitution* is defined as $k + \sum_{i=1}^{k} |t_i|$, and we denote by $|\theta|$. The domain of a substitution θ is extended to the set of all terms by inductively defining $\theta(f(t_1,\dots,t_n))$ to be $f(\theta(t_1),\dots,\theta(t_n))$. A substitution θ is said to *match* a term s with a term t if and only if $s\theta = t$. We can extend the notion of matching considering a set of equations E, an equational theory, taking into account the congruence classes of the relation generated by E.

Let s,t be two first-order terms. We say that s and t are *E-equal* for some equational theory $E \in \{A, C, AC\}$ if and only if they are equivalent under the axioms of the equational theory E. For example, if f is an associative and commutative symbol, $f(f(a,b),c) =_{AC} f(c, f(b,a))$.

We represent the matching problem for some equational theory E in the following way:

E-MATCHING	
Instance:	A first-order term $s \in T(\mathscr{F},\mathscr{V})$, a first-order term $t \in T(\mathscr{F})$.
Problem:	Does there exist a θ such that $s\theta =_E t$?

It is well known that E-MATCHING is NP-complete for $E \in \{A, C, AC\}$[1] (BENANAV *et al.*, 1987). It is easy to see that the problem is in NP. Considering two terms s,t and a substitution θ such that $s\theta =_E t$. The size of θ cannot be greater than $|s| + |t|$. Then, for any input terms s and t, someone has to guess θ such that $|\theta| \leq |s| + |t|$, apply θ to s, and check if $s\theta =_E t$. The last two steps can be obviously computed in polynomial time, and then, we achieve an algorithm in NP for the problem by non-deterministically guessing. To show the NP-hardness, the authors exhibit a polynomial reduction from 3SAT to E-MATCHING for each equational theory.

[1] When considering different equational problems at the same time, we abuse the notation denoting $\{A, AC\}$-MATCHING, for example, as the associative and associative-commutative matching problems.

Theorem 4.1. *(BENANAV et al., 1987) The* $\{A, C, AC\}$*-MATCHING problems are NP-complete.*

We define the equality problem for some equational theory E in the following way:

E-EQUALITY	
Instance:	A first-order term $s \in T(\mathscr{F}, \mathscr{V})$, a first-order term $t \in T(\mathscr{F})$.
Problem:	Decide whether $s\theta =_E t$?

The equality of terms under associative-commutative theories can be done in polynomial time (BENANAV *et al.*, 1987, Corollary 3). This result is obtained as a consequence of the matching problem for terms with distinct occurrences (DO-AC-MATCHING) whose runtime complexity is bounded by $\mathscr{O}(|s|^3 \times |t|)$.

Lemma 4.2. *(BENANAV et al., 1987) Associative-commutative equality can be done in polynomial time.*

4.3 Parameterized Complexity of {A, C, AC}-MATCHING

In this section, we summarize the relevant results contained in (AKUTSU *et al.*, 2017). We consider the parameterized version of the matching problem, p-κ-E-MATCHING. Again, given two terms $s \in T(\mathscr{F}, \mathscr{V})$ and $t \in T(\mathscr{F})$, the problem asks if there exists a substitution θ such that $s\theta =_E t$ for some equational theory $E \in$ {A, C, AC} for some parameterization κ. We already defined all the parameters that we consider for the matching problems.

- **the number of variables** of s: $\text{var}(s)$;
- **the size of the substitution** θ: $|\theta| = \text{var}(s) + \sum_{i=1}^{\text{var}(s)} |t_i|$;
- **the number of symbols** in $\mathscr{F}_t$: $|\mathscr{F}_t|$;
- **the number of occurrences of variables:** $|s|_{\textbf{var}}$.

We may also combine these parameters. For a list of parameters P, we define p-$[P]$-E-MATCHING as parameterized E-Matching for $E \in$ {A, C, AC}, where the parameterization is the sum of the parameters in P. We represent this parameterized problem in the following way:

p-$[P]$-E-MATCHING	
Instance:	A first-order term $s \in T(\mathscr{F}, \mathscr{V})$, and first-order term $t \in T(\mathscr{F})$, and a natural number k.
Parameter:	k such that $k = \sum_{\kappa \in P} \kappa(x)$.
Problem:	Does there exist a θ such that $s\theta =_E t$?

For example, p-$[|\mathscr{F}_t|, |\theta|]$-A-MATCHING is associative matching with parameterization $\kappa(s,t) = |\mathscr{F}_t| + |\theta|$.

For matching and unification problems, there are three different types of unification problems: *elementary*, *with constants*, and *general*. Given an equational theory E, an *elementary term* is a term containing only functional symbols of E. We call an E-matching problem *elementary* if the terms being matched are elementary. If we allow constants in the term, the E-matching is called *with constants* and, if we allow non-constant function symbols that not occurs within E, then the E-matching is called *general*. In this work, we are dealing with the general matching problem.

The parameterized version of the unification and matching problem was initially studied in (AKUTSU *et al.*, 2017) using as parameter **the number of variables**. The problem was shown to be W[1]-hard for associative and associative-commutative cases, and the proofs rely on an fpt-reduction from p-LCS, longest common subsequence, for both cases (see Example 2.17).

Theorem 4.3. *(AKUTSU et al., 2017) The p-$|var(s)|$-$\{A,AC\}$-MATCHING problems are W[1]-hard.*

They also exhibit a linear algorithm for the "unification of associative and commutative ground terms" which we recognize it as identical to the equality problem defined here.

Theorem 4.4. *(AKUTSU et al., 2017) $\{A,C\}$-EQUALITY can be done in linear time.*

For the commutative case, they conjectured that C-MATCHING is fixed-parameter tractable describing an algorithm using dynamic programming (AKUTSU *et al.*, 2017, Theorem 3). Considering terms in dag form, they constructed a 0-1 table by a bottom-up dynamic programming that steps on all pair of nodes. The algorithm computes all pair of nodes that can be matched and, finally, checks whether the roots of s and t match.

In the following sections, we will consider the membership in W[P] and in FPT for matching with respect to different parameters.

4.4 W[P] membership

Recall that the class W[P] is defined using algorithms with bounded non-determinism (see Definition 2.16) and that C-MATCHING was conjectured to be in FPT in (AKUTSU *et al.*, 2017). Let us consider our first result relevant to equational matching.

***p*-$|\mathbf{var}(s)|$-C-MATCHING**

Here, we settle a parameterized complexity result for C-MATCHING. We show the membership in W[P] providing a non-deterministic algorithm with fpt-time and a limited number of non-deterministic steps.

Theorem 4.5. *p-$|var(s)|$-C-MATCHING is in W[P].*

Proof. Given two terms $s \in T(\mathscr{F},\mathscr{V})$ and $t \in T(\mathscr{F})$ with $|\text{var}(s)| = k$. We design a Turing machine receiving s,t and k as inputs with running time $f(k) \cdot |(s+t)|^{\mathscr{O}(1)}$ and at most $h(k) \cdot \log|(s+t)|$ non-deterministic steps for some computable functions f and h.

For each variable x_i, it guesses a position v_i in t. To guess these positions, it needs $k \cdot \log|t|$ non-deterministic steps. The machine applies the substitution θ to s producing $s\theta$, and then checks if $s\theta =_C t$. For every variable and for each occurrence, the application process could be seem as a detection of the variable in s and a replacement by the sub-term t_i from t related to the position v_i. The previous application can be computed in $k \cdot (|s|+|t|)^{\mathscr{O}(1)}$ and the equality modulo commutativity is decided in linear time with respect to the size of $(s\theta,t)$ (Lemma 2.12). As the size of $s\theta$ is bounded by $k \cdot (|s|+|t|)^{\mathscr{O}(1)}$, the whole process leads to a running time bounded by $f(k) \cdot (|s|+|t|)^{O(1)}$. Then, we can conclude the membership in W[P] for $|\text{var}(s)|$-C-MATCHING. □

If we consider a parameter that is greater than the number of variables, membership in W[P] remains for the commutativity case.

***p*-$|\theta|$-{A, AC}-MATCHING**

One step further, considering *the size of the substitution*

$$|\theta| = |\text{var}(s)| + \sum_{i=1}^{|\text{var}(s)|} |t_i|,$$

we can verify membership in W[P] for the p-$|\theta|$-{A, AC}-MATCHING problems. Considering $|\theta|$ as the parameter, we can build up a non-deterministic Turing machine with similar behavior. It guesses a substitution θ and then checks if the equivalence holds.

Theorem 4.6. *The $|\theta|$-{A, AC}-MATCHING problems are in W[P].*

Proof. This proof is similar to the proof of Theorem 4.5. Given two terms $s \in T(\mathscr{F},\mathscr{V})$ and $t \in T(\mathscr{F})$, and some natural number k. Consider $|\theta| \leq k$ and $|\text{var}(s)| = \ell$. The algorithm guesses

ℓ terms $t_i \in T(\mathscr{F})$ with size bounded by $|\theta| \leq k$, instantiates them in s and checks if $s\,\theta =_E t$. Let $m = \max\{|ti| : 1 \leq i \leq \ell\}$. Again, the substitution and the equality modulo E are made in polynomial time observing that the process is similar in the proof of Theorem 4.5. In both cases, we obtain an algorithm in W[P] for $|\theta|$-{A, AC}-MATCHING. □

If we consider $|s|_{\text{var}}$, the number of occurrences of variables, as a parameter, it is unlikely that E-MATCHING is in FPT. Moreover, it is unlikely to be within XP assuming $P \neq NP$.

Theorem 4.7. *Unless $P = NP$, $|s|_{var}$-E-MATCHING is not in XP.*

Proof. Assume that $|s|_{\text{var}}$-E-MATCHING is in XP. By definition, there is an algorithm that solves the problem in time $f(k) \cdot n^{g(k)}$ for some $k = |s|_{\text{var}}$, and f and g are computable functions. Then, for $k = 2$, E-MATCHING is solved in time $\mathscr{O}(n^c)$, for some constant c. Assuming that $P \neq NP$, we reach a contradiction with the fact that E-MATCHING when $|s|_{\text{var}} = 2$ is already NP-complete (VERMA; RAMAKRISHNAN, 1992). □

The size of the vocabulary $\mathscr{F}$ is not a good parameter for the same reasons. The {A, C, AC}-MATCHING problems are NP-complete with fixed vocabulary (BENANAV *et al.*, 1987).

Theorem 4.8. *Unless $P = NP$, $|\mathscr{F}_t|$-E-MATCHING is not in XP.*

Proof. The proof is similar to the previous one. Assume that $|\mathscr{F}_t|$-E-MATCHING is in XP. Then, there exists an algorithm that on the input s,t, decides time $(|s| + |t|)^{g(|F_t|)}$ if s matches t. In this case, there is a polynomial time algorithm when the problem has 6 function symbols. Unfortunately, the problem is already NP-complete with at least 6 symbols (BENANAV *et al.*, 1987). □

4.5 Fixed-Parameter Tractability

From the perspective of parameterized complexity theory, the parameter is expected to be smaller than the input size. If we consider, for example, the size of the ground term $|t|$, it will lead to the case where the parameterized complexity is uninteresting, or trivially fixed-parameter tractable. In such conditions where the parameter increases monotonically with the size of the input, the problem is trivially in FPT (FLUM; GROHE, 2006, Proposition 2.6).

However, this is not the case for the parameters $|\mathscr{F}_t|$ and $|\theta|$, and we will describe an algorithm in FPT for the matching problems considered here.

p-$[|\mathscr{F}_t|, |\theta|]$-***{A, C, AC}-MATCHING***

We show an FPT brute-force algorithm that solves these matching problems when parameterized by $|\mathscr{F}_t| + |\theta|$. The algorithm enumerates of all possible substitutions checking whether it corresponds to a match. The idea of Algorithm 1 was inspired by the work of FERNAU *et al.*.

First, it constructs the set $T(\mathscr{F}_t)$ of ground terms with size at most k, for some natural number k. Then, for every tuple of size $|\text{var}(s)|$ of terms in $T(\mathscr{F}_t)$, we build a substitution θ, apply it into s, i.e., for every occurrence of a variable v_i, we remove its encoding from s inserting the encoding of the term $\theta(v_i)$, and evaluate whether $s\theta$ is equal to t modulo E $\in$ {A, C, AC}. The equality of terms with respect to associative, commutative, and associative-commutative terms implemented in Step 5 can be computed in polynomial time (see Lemma 4.2).

Algorithm 1 {A, C, AC}-MATCHING via brute force

INPUT: A term s in $T(\mathscr{F}, \mathscr{V})$, a term t in $T(\mathscr{F})$, and a natural number k.
OUTPUT: Yes iff there exists a substitution θ s.t. $s\theta = t$, and $|\theta| + |\mathscr{F}_t| \leq k$.

1: $T(\mathscr{F}_t) \leftarrow$ GENERATE(t,k) ▷ It constructs all terms in $\mathscr{F}_t$ with size bounded by k.
2: **for** `every tuple of terms` $(t_1, \ldots t_{|\texttt{var}(s)|})$ `in` $T(\mathscr{F}_t)$ **do**
3:     **for** $i = 1$ to $|\text{var}(s)|$ **do**
4:         $\theta \leftarrow \theta \cup \{x_i \mapsto t_i\}$
5:     **if** $s\theta =_E t$ **then return** Yes;
return No;

Proposition 4.9. *The running time of Algorithm 1 is bounded by*

$$k^{k^2+2+c} \cdot (|s| + |t|)^{\mathscr{O}(1)}$$

for some fixed natural c and $|\mathscr{F}_t| + |\theta| \leq k$.

Proof. Let $|\mathscr{F}_t|$ be the number of symbols in t. It is clear that $|\mathscr{F}_t| \leq k$. Then, the number of terms in $T(\mathscr{F}_t)$ with size at most k is in $\mathscr{O}(k^{k+1})$. Then, the main loop will take at most $\mathscr{O}((k^{k+1})^{|\text{var}(s)|})$ iterations. The construction of θ in Step 4 is bounded by the function k^2. For each variable x_i, it writes a term t_i with size bounded by k. The application of θ on s can be done in time polynomial in $k^{\mathscr{O}(1)} \cdot |s|^{\mathscr{O}(1)}$. For each variable x_i and for each occurrence, we need to find and erase the encoding of x_i changing to t_i, being careful to save the remained part of the string. The equality modulo E can be done in time $(|s\theta| + |t|)^{\mathscr{O}(1)}$ by Theorems 4.2 and 4.4. Then, the whole computational complexity of the algorithm is $(k^{k+1})^{|\text{var}(s)|} \cdot (|s| + |t)^{\mathscr{O}(1)}$. □

Theorem 4.10. *The* $[|\mathscr{F}_t|, |\theta|]$*-E-MATCHING problem is in FPT for* $E \in$ *{A, C, AC}.*

Proof. Algorithm 1 solves the matching problem for all equational theory considered here in time $f(|\mathscr{F}_t|, |\theta|) \cdot (|s| + |t|)^{\mathscr{O}(1)}$ for some computable function f. Then, we can conclude that they are in FPT. □

As we can see, a specific choice of parameters allows us to recognized the matching problems that are fixed-parameter tractable. We summarize these results in the following table.

Tabela 8 – Parameters for associative, commutative, and associative-commutative matching.

Problem / Parameter	A-MATCHING	C-MATCHING	AC-MATCHING
$\lvert\mathscr{F}_t\rvert$ or $\lvert s\rvert_{\text{var}}$	**not in XP**	**not in XP**	**not in XP**
$\lvert\text{var}(s)\rvert$	? / W[1]-hard[†]	W[P] / FPT[†]	? / W[1]-hard[†]
$\lvert\theta\rvert$	W[P]	W[P]	W[P]
$\lvert\theta\rvert + \lvert\mathscr{F}_t\rvert$	FPT	FPT	FPT

Fonte: Made by the author himself.

In the next chapter, we summarize the thesis and all future directions that we can see in a short distance.

[†] Results from (AKUTSU *et al.*, 2017)

5 CONCLUSION

In this thesis, we addressed the parameterized analysis of two logical problems within the setting of first-order logic: the satisfiability of some decidable classes of formulas, and the matching problem for associative, commutative, and associative-commutative terms.

The satisfiability problem for many decidable fragments of first-order logic was investigated in Chapter 3. To our knowledge, this was a first time that the satisfiability problem of first-order fragments has received a parameterized complexity analysis. For some prefix-vocabulary classes, satisfiability was shown to be fixed-parameter tractable concerning some parameters. We evaluated different parameters that appear in Definition 3.1. Combining the choice of the parameterization with a proper finite model property, we could construct a fixed-parameter reduction to the propositional satisfiability.

We have seen that in Section 3.2, for all classical classes, the satisfiability problem parameterized by **the size of the vocabulary**, **the quantifier rank**, and **the maximum arity** is fixed-parameter tractable, and, for all relational classes of modest complexity, satisfiability is also in FPT considering different parameters. For example, we observed that the satisfiability problem of Lowenheim's class $[\text{all}, \omega]$ is within FPT when parameterized by **the number of monadic relations** and **the quantifier rank** (Theorem 3.4). However, when just the quantifier rank is considered, the problem is not in XP, unless P$\neq$ NP (Proposition 3.6).

We also expanded the idea of fixed-parameter tractability from relational classes to functional ones. In Section 3.4, we achieved two fixed-parameter results for the satisfiability of $[\text{all}, (\omega), (\omega)]$ and $[\exists^*, \text{all}, \text{all}]_=$. For both cases, the paremeterization included elements related the functional side of the vocabulary.

Now we outline some directions in order to improve the understanding of the parameterized complexity of the satisfiability of prefix-vocabulary classes.

1. Reducing the parameterization. For $[\exists^*\forall^*, \text{all}]_=$, $[\exists^*\forall\exists^*, \text{all}]$, and $[\exists^*\forall^2\exists^*, \text{all}]$, the fixed-parameter tractability is obtained when we consider $\text{qr}(\varphi) + \#\text{r}(\varphi) + \text{ar}(\varphi)$ as our parameterization function in Theorems 3.8 and 3.9. One question remains: What is the parameterized complexity of the satisfiability for these classes when some terms $(\text{qr}(\varphi), \#\text{r}(\varphi),$ and $\text{ar}(\varphi))$ are not considered in the parameterization function? The same question arises in the context of functional classes developed in Section 3.4.

2. Functional classes. Functional classes were not completely investigated. As future work, we indicate a further investigation of the parameterized complexity of the remaining maximal classes concerning finite model theory (Table 2), and some functional classes with modest complexity (Table 10).

Tabela 10 – Modest complexity prefix-vocabulary classes with functions.

Prefix-Vocabulary Class	**Complexity classification**
$[\exists^*\forall^m, (0), (1)]_=$	NP-complete
$[\exists^*\forall\exists^*, (0), (1)]_=$	NP-complete
$[\exists^k\forall^*, (0), (1)]_=$	Co-NP-complete
$[\exists^*\forall^*, (0), (1)]_=$	Σ_2^p-complete

Fonte: Made by the author himself.

3. Classes without finite model property. Finally, the finite model property does not hold for all decidable prefix-vocabulary classes (see (BÖRGER *et al.*, 2001, Chapter 7)). The methods applied here cannot be extended to $[\text{all}, (\omega), (1)]_=$ (RABIN, 1969) and $[\exists^*\forall\exists^*, \text{all}, (1)]_=$ (SHELAH, 1977). The decibility for these classes is obtained as a consequence of Rabin's Theorem that the monadic theory of the infinite binary tree is decidable.

For the second problem, the matching problem for associative (A), commutative (C), and associative-commutative (AC) functions, we provided parameterized complexity results connected to the membership in W[P] and fixed-parameter tractability in Chapter 4. The main questions for {A, AC}-MATCHING is their membership in W[1] when parameterized by **the number of variables** since they were proved to be W[1]-hard (AKUTSU *et al.*, 2017), and the fixed-parameter tractability of C-MATCHING for the same parameterization. More than this, we expand the analysis to other parameters.

Restricted to **the number of variables**, we showed that the C-MATCHING problem is in W[P] witnessed by an algorithm with a limited number of non-deterministic steps in terms of the number of variables. Increasing the parameterization, we consider **the size of the substitution**, we show that the $|\theta|$-{A, AC}-MATCHING problems are in W[P]. Combining **the size of the substitution** with **the number of functions**, we achieved the fixed-parameter tractability in all cases providing a brute-force algorithm that runs over all substitutions.

Now, we depicted some open problems and similar problems within unification theory that could be explored in the future.

1. Open problems. In this work, we could not solve the conjecture from (AKUTSU *et al.*, 2017), that C-MATCHING is in FPT when the number of variables is the parameter. We could not improve the dynamic programming strategy developed there nor could we apply any method to fpt algorithms. A proof of membership in W[1] would be interesting. For the other two problems, $|\text{var}(s)|$-{A, AC}-MATCHING, we cannot say anything better than the membership in para-NP, and we wonder if it is the case that these problems are in W[1].

Considering the size of the substitution, we would like to set the matching problems within some finite level of W-Hierarchy (W[1], W[2], W[3], $\ldots$). Also, we could not extend the W[1]-hardness of $|\text{var}(s)|$-{A, AC}-MATCHING for $|\theta|$-{A, AC}-MATCHING.

2. Other similar problems. In the future, we plan to reach the parameterized analysis to other unification/matching problems like high-order matching (STIRLING, 2006), and context unification (JEŻ, 2014). Unfortunately, these problems seem to fall in the same techniques used to prove the decidability for classes without finite model property. We do not know how to detach the non-elementary complexity of the existing algorithms.

BIBLIOGRAPHY

ABBOUD, A.; LEWI, K.; WILLIAMS, R. Losing weight by gaining edges. In: SCHULZ, A. S.; WAGNER, D. (Ed.). **Algorithms - (ESA 2014)**. Berlin, Heidelberg: Springer Berlin Heidelberg", 2014. v. 8737, p. 1–12.

ABU-KHZAM, F. N.; COLLINS, R. L.; FELLOWS, M. R.; LANGSTON, M. A.; SUTERS, W. H.; SYMONS, C. T. Kernelization algorithms for the vertex cover problem: Theory and experiments. **ALENEX/ANALC**, v. 69, 2004.

ACHILLEOS, A.; LAMPIS, M.; MITSOU, V. Parameterized modal satisfiability. **Algorithmica**, Springer-Verlag, v. 64, n. 1, p. 38–55, 2012.

ACKERMANN, W. Zum hilbertschen aufbau der reellen zahlen. **Mathematische Annalen**, Springer, v. 99, n. 1, p. 118–133, 1928.

AKUTSU, T.; JANSSON, J.; TAKASU, A.; TAMURA, T. On the parameterized complexity of associative and commutative unification. **Theoretical Computer Science**, Elsevier, v. 660, p. 57–74, 2017.

ALON, N.; YUSTER, R.; ZWICK, U. Color-coding. **J. ACM**, Citeseer, v. 42, n. 4, p. 844–856, 1995.

ASH, C. J. Sentences with finite models. **Mathematical Logic Quarterly**, Wiley Online Library, v. 21, n. 1, p. 401–404, 1975.

BAADER, F.; SNYDER, W. Unification theory. **Handbook of automated reasoning**, v. 1, p. 445–532, 2001.

BENANAV, D.; KAPUR, D.; NARENDRAN, P. Complexity of matching problems. **Journal of symbolic computation**, Elsevier, v. 3, n. 1-2, p. 203–216, 1987.

BERNAYS, P.; SCHÖNFINKEL, M. Zum entscheidungsproblem der mathematischen logik. **Mathematische Annalen**, Springer, v. 99, n. 1, p. 342–372, 1928.

BODLAENDER, H. L.; DOWNEY, R. G.; FELLOWS, M. R.; WAREHAM, H. T. The parameterized complexity of sequence alignment and consensus. **Theoretical Computer Science**, Elsevier, v. 147, n. 1-2, p. 31–54, 1995.

BÖRGER, E.; GRÄDEL, E.; GUREVICH, Y. **The classical decision problem**. Berlin Heidelberg: Springer-Verlag Berlin Heidelberg, 2001.

BUSS, J. F.; GOLDSMITH, J. Nondeterminism within p^. **SIAM Journal on Computing**, SIAM, v. 22, n. 3, p. 560–572, 1993.

BUSTAMANTE, L. H.; MARTINS, A. T.; FERREIRA, F. M. The parameterized complexity of some prefix-vocabulary fragments of first-order logic. In: MOSS, L. S.; QUEIROZ, R. de; MARTINEZ, M. (Ed.). **Logic, Language, Information, and Computation**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. p. 163–178. ISBN 978-3-662-57669-4.

BUSTAMANTE, L. H.; MARTINS, A. T.; FERREIRA, F. M. **Fixed-Parameter Tractability of Some Classes of Formulas in FO with Functions**. 2019. 19^{th} Brazilian Logic Conference (EBL2019).

BUSTAMANTE, L. H.; MARTINS, A. T.; FERREIRA, F. M. **Parameters for Associative and Commutative Matching**. 2019. 33^{rd} International Workshop on Unification (UNIF2019).

CESATI, M. Compendium of parameterized problems. **Dept. Computer Science, Systems, and Industrial Eng., University of Rome Tor Vergata**, 2006.

CHEN, J.; KANJ, I. A.; XIA, G. Improved upper bounds for vertex cover. **Theoretical Computer Science**, Elsevier, v. 411, n. 40-42, p. 3736–3756, 2010.

COURCELLE, B. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. **Information and Computation**, v. 85, n. 1, p. 12–75, 1990.

CYGAN, M.; FOMIN, F. V.; KOWALIK, Ł.; LOKSHTANOV, D.; MARX, D.; PILIPCZUK, M.; PILIPCZUK, M.; SAURABH, S. **Parameterized algorithms**. Cham: Springer, 2015. v. 4.

DOWNEY, R. G.; FELLOWS, M. R. Fixed-parameter intractability. In: IEEE. **[1992] Proceedings of the Seventh Annual Structure in Complexity Theory Conference**. Boston, 1992. p. 36–49.

DOWNEY, R. G.; FELLOWS, M. R. Fixed-parameter tractability and completeness. **Congressus Numerantium**, UTILITAS MATHEMATICA PUBLISHING INC, p. 161–161, 1992.

DOWNEY, R. G.; FELLOWS, M. R. Fixed-parameter tractability and completeness ii: On completeness for w[1]. **Theoretical Computer Science**, Elsevier, v. 141, n. 1-2, p. 109–131, 1995.

DOWNEY, R. G.; FELLOWS, M. R. Parameterized computational feasibility. In: **Feasible mathematics II**. Boston: Springer, 1995. p. 219–244.

DOWNEY, R. G.; FELLOWS, M. R. **Parameterized complexity**. New York: Springer Science & Business Media, 2012.

DOWNEY, R. G.; FELLOWS, M. R.; REGAN, K. Descriptive complexity and the w hierarchy. In: . US: AMS, 1998. v. 39, p. 119–134.

DWORK, C.; KANELLAKIS, P. C.; MITCHELL, J. C. On the sequential nature of unification. **The Journal of Logic Programming**, Elsevier, v. 1, n. 1, p. 35–50, 1984.

EBBINGHAUS, H.-D.; FLUM, J.; THOMAS, W. **Mathematical logic**. Berlin ; New York: Springer Science & Business Media, 2013.

FERNAU, H.; SCHMID, M. L.; VILLANGER, Y. On the parameterised complexity of string morphism problems. **Theory of Computing Systems**, Springer, v. 59, n. 1, p. 24–51, 2016.

FLUM, J.; GROHE, M. Fixed-parameter tractability, definability, and model-checking. **SIAM Journal on Computing**, SIAM, v. 31, n. 1, p. 113–145, 2001.

FLUM, J.; GROHE, M. Describing parameterized complexity classes. **Information and Computation**, Elsevier, v. 187, n. 2, p. 291–319, 2003.

FLUM, J.; GROHE, M. **Parameterized complexity theory**. Berlin, Heidelberg: Springer Science & Business Media, 2006.

FÜRER, M. Alternation and the ackermann case of the decision problem. **L'Enseignement Math**, v. 27, p. 137–162, 1981.

GALL, F. L. Powers of tensors and fast matrix multiplication. In: ACM. **Proceedings of the 39th international symposium on symbolic and algebraic computation**. New York, 2014. p. 296–303.

GÖDEL, K. Zum intuitionistischen aussagenkalkül. **Anzeiger der Akademie der Wissenschaften in Wien**, v. 69, p. 65–66, 1932.

GÖDEL, K. Zum entscheidungsproblem des logischen funktionenkalküls. **Monatshefte für Mathematik**, Springer, v. 40, n. 1, p. 433–443, 1933.

GOTTLOB, G.; SCARCELLO, F.; SIDERI, M. Fixed-parameter complexity in ai and nonmonotonic reasoning. **Artificial Intelligence**, Elsevier, v. 138, n. 1-2, p. 55–86, 2002.

GRÄDEL, E. Complexity of formula classes in first order logic with functions. In: SPRINGER. **International Conference on Fundamentals of Computation Theory**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989. p. 224–233.

GUREVICH, Y. The decision problem for the logic of predicates and of operations. **Algebra and Logic**, Springer, v. 8, n. 3, p. 160–174, 1969.

GUREVICH, Y. Formulas with one $\forall$. **Selected Questions in Algebra and Logic; in memory of A. Mal cev**, p. 97–110, 1973.

GUREVICH, Y. The decision problem for standard classes. **The Journal of Symbolic Logic**, Cambridge University Press, v. 41, n. 2, p. 460–464, 1976.

HAAN, R. de; SZEIDER, S. Parameterized complexity results for symbolic model checking of temporal logics. In: AAAI PRESS. **Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning**. Cape Town, South Africa, 2016. p. 453–462.

HAAN, R. de; SZEIDER, S. Parameterized complexity classes beyond para-np. **Journal of Computer and System Sciences**, Elsevier, v. 87, p. 16–57, 2017.

HERBRAND, J. **Recherches sur la théorie de la démonstration**. Tese (Doutorado) — Université de Paris, 1930.

HILBERT, D.; ACKERMAN, W. **Theoretische Logik**. Berlin Heidelberg: Springer, 1928.

HULLOT, J.-M. Canonical forms and unification. In: **International Conference on Automated Deduction**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1980. p. 318–334.

JEŻ, A. Context unification is in pspace. In: **International Colloquium on Automata, Languages, and Programming**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 244–255.

JOHNSON, D. S.; GAREY, M. R. **Computers and intractability: A guide to the theory of NP-completeness**. San Francisco: W. H. Freeman and Company, 1979. v. 1.

KALMÁR, L. Über die erfüllbarkeit derjenigen zählausdrücke, welche in der normalform zwei benachbarte allzeichen enthalten. **Mathematische Annalen**, Springer, v. 108, n. 1, p. 466–484, 1933.

LEWIS, H. R. Complexity results for classes of quantificational formulas. **Journal of Computer and System Sciences**, Elsevier, v. 21, n. 3, p. 317–353, 1980.

LÖB, M. Decidability of the monadic predicate calculus with unary function symbols. In: ASSN SYMBOLIC LOGIC INC 1325. **Journal of Symbolic Logic**. SOUTH OAK ST, CHAMPAIGN, IL 61820, 1967. v. 32, n. 4, p. 563.

LÖWENHEIM, L. Über möglichkeiten im relativkalkül. **Mathematische Annalen**, Springer, v. 76, n. 4, p. 447–470, 1915.

LÜCK, M.; MEIER, A.; SCHINDLER, I. Parametrised complexity of satisfiability in temporal logic. **ACM Transactions on Computational Logic (TOCL)**, ACM, v. 18, n. 1, p. 1, 2017.

MAJDODDIN, R. Uniform csp parameterized by solution size is in w[1]. In: **International Computer Science Symposium in Russia**. Cham: Springer International Publishing, 2019. p. 275–285.

MEIER, A.; ORDYNIAK, S.; RAMANUJAN, M.; SCHINDLER, I. Backdoors for linear temporal logic. **Algorithmica**, Springer, v. 81, n. 2, p. 476–496, 2019.

MOL, L. D. Closing the circle: An analysis of emil post's early work. **Bulletin of Symbolic Logic**, Cambridge University Press, v. 12, n. 2, p. 267–289, 2006.

NEŠETŘIL, J.; POLJAK, S. On the complexity of the subgraph problem. **Commentationes Mathematicae Universitatis Carolinae**, Charles University in Prague, Faculty of Mathematics and Physics, v. 26, n. 2, p. 415–419, 1985.

NIEUWENHUIS, R.; RUBIO, A. Paramodulation-based theorem proving. **Handbook of automated reasoning**, v. 1, p. 371–443, 2001.

PAPADIMITRIOU, C. H. **Computational complexity**. USA: John Wiley and Sons Ltd., 2003.

PATERSON, M. S.; WEGMAN, M. N. Linear unification. **Journal of Computer and System Sciences**, Academic Press, v. 16, n. 2, p. 158–167, 1978.

PĂTRAŞCU, M.; WILLIAMS, R. On the possibility of faster sat algorithms. In: **Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms**. Austin, Texas: SIAM, 2010. p. 1065–1075.

PFANDLER, A.; RÜMMELE, S.; WALLNER, J. P.; WOLTRAN, S. On the parameterized complexity of belief revision. In: **Proceedings of the 24th International Conference on Artificial Intelligence**. Buenos Aires, Argentina: AAAI Press, 2015. (IJCAI'15), p. 3149–3155.

PLOTKIN, G. Building-in equational theories. **Machine intelligence**, v. 7, p. 73–90, 1972.

RABIN, M. O. Decidability of second-order theories and automata on infinite trees. **Transactions of the american Mathematical Society**, JSTOR, v. 141, p. 1–35, 1969.

RAMSEY, F. P. On a problem of formal logic. In: **Classic Papers in Combinatorics**. Boston, MA: Birkhäuser Boston, 1987. p. 1–24.

REED, B.; SMITH, K.; VETTA, A. Finding odd cycle transversals. **Operations Research Letters**, Elsevier, v. 32, n. 4, p. 299–301, 2004.

ROBINSON, J. A. A machine-oriented logic based on the resolution principle. **Journal of the ACM**, ACM, New York, NY, USA, v. 12, n. 1, p. 23–41, 1965.

SCHÜTTE, K. Untersuchungen zum entscheidungsproblem der mathematischen logik. **Mathematische Annalen**, Springer, v. 109, n. 1, p. 572–603, 1934.

SEESE, D. Linear time computable problems and first-order descriptions. **Math. Struct. in Comp. Science**, v. 6, p. 505–526, 1996.

SHELAH, S. Decidability of a portion of the predicate calculus. **Israel Journal of Mathematics**, Springer, v. 28, n. 1-2, p. 32–44, 1977.

SKOLEM, T. A. **Logisch-kombinatorische Untersuchungen über die Erfüllbarkeit oder Beweisbarkeit mathematischer Sätze, nebst einem Theoreme über dichte Mengen, von Th. Skolem**. as reprinted in Skolem 1970: J. Dybwad, 1920. 103–136 p.

STIRLING, C. A game-theoretic approach to deciding higher-order matching. In: SPRINGER. **International Colloquium on Automata, Languages, and Programming**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 348–359.

SZEIDER, S. On fixed-parameter tractable parameterizations of sat. In: SPRINGER SCIENCE & BUSINESS MEDIA. **Theory and Applications of Satisfiability Testing**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. v. 2919, p. 188–202.

URQUHART, A. Emil post. **Post."Logic from Russell to Church 5 (2009)**, Citeseer, p. 617–666, 2009.

VARDI, M. Y. The complexity of relational query languages (extended abstract). In: ACM. **Proceedings of the fourteenth annual ACM symposium on Theory of computing**. New York, NY, USA: ACM, 1982. p. 137–146.

VERMA, R. M.; RAMAKRISHNAN, I. Tight complexity bounds for term matching problems. **Information and Computation**, Elsevier, v. 101, n. 1, p. 33–69, 1992.

ZILLI, M. V. Complexity of the unification algorithm for first-order expressions. **Calcolo**, Springer, v. 12, n. 4, p. 361–371, 1975.

APÊNDICE A – CONSIDERING FO^2

In this appendix, we develop some ideas about a parameterized analysis of FO^2. The fragments of first-order logic with restricted use of variables have many computational applications. We denote FO^k as the class of first-order formulas with at most k variables. For $k \geq 3$, the satisfiability problem for FO^3 is undecidable.

Let us consider a solution for the satisfiability of FO^2 observing the finite model property and a polynomial time algorithm to verify $\mathscr{A} \models \varphi$.

p-SAT(FO^2)	
Instance:	A sentence φ in FO^2
Parameter:	k, the number of relations τ_φ
Problem:	Decide if φ is satisfiatible.

Theorem A.1. *The model checking problem for FO can be solved in time $O(|\varphi| \cdot |\mathscr{A}|^w \cdot w)$ such that w it the width of φ.*

Proof. The recursive definition of $\varphi(\mathscr{A})$ directly give us a recursive algorithm. Observe that for a formula $\varphi(x_1, \dots, x_k)$, compute $\varphi(\mathscr{A})$ from the immediate subformulas of φ takes $O(w \cdot |\mathscr{A}|^w)$ time. For example, assume that

$$\varphi(x_1, \dots, x_k) := \psi(x_{i_1}, \dots, x_{i_r}) \wedge \theta(x_{j_1}, \dots, x_{j_s}),$$

where $\{i_1, \dots, i_r\} \cup \{j_1, \dots, j_s\} = [k]$. Assume that $\{i_1, \dots, i_r\} \cap \{j_1, \dots, j_s\} = \{\ell_1, \dots, \ell_t\}$. $\square$

Corollary A.2. *For $k = 2$, MC(FO^2) can be solved in polynomial time. More precisely, there is an algorithm with running time bounded by $O(n \cdot |\mathscr{A}|^2)$.*

Consider the finite model property of FO^2. We can obtain, for satisfiable formulas in FO^2, an upper bound on the size of their models.

Theorem A.3. *For all satisfiable formulas of FO^2 with size n, there is a model of cardinality at most $2^{O(n)}$.*

This result is obtained from the satisfiability analysis of a formula in the Scott normal form

$$(\forall x)(\exists y)\alpha(x,y) \wedge \bigwedge_{i=1}^{m} (\forall x)(\exists y)\beta_i(x,y),$$

where α and β_i, with $1 \leq i \leq m$, are quantifier-free formulas.

Theorem A.4. *All first-order formula in FO^2 can be converted to the Scott normal form.*

Theorem A.5. *All satisfiable formula φ in FO^2 in the Scott normal form has a model of cardinality at most $3n \cdot 2^r$, where r is the number of relation symbols of φ.*

Then, we can decide the satisfiability of $\varphi \in \mathrm{FO}^2$ is given by the following procedure:

Algorithm 2 Satisfiability for FO^2.

INPUT: φ, with $|\varphi| = n$ and $|\tau_\varphi| = r$.
OUTPUT: Yes iff ...

```
for i = 2 to n·2^r do
    for all structure 𝒜 with |𝒜| = i do
        if 𝒜 ⊨ φ then return Yes;
return No;
```

With the finite model property, there is an upper bound on the size of the universe of the structure that satisfies a formula φ in FO^2. For a structure in the vocabulary

ANEXO A – DESCRIPTIVE COMPLEXITY OF K-SUM

Here we provide a different proof for the membership of the k-SUM into the W[1] class. This work was presented on XVIII Brazilian Logic Conference (EBL2017)

Introduction

Among many problems that were showed to be W[1]-hard, the k-SUM problem resist to the classified within W[1].

In (ABBOUD *et al.*, 2014), k-SUM (see Definition A.2) was shown to be $W[1]$-complete. They considered a version of the problem using integers of the domain of $[-n^{2k}, n^{2k}]$ such that, from a set of n integers, and k number sum zero. This problem is equivalent over arbitrary integers by linear-time randomized reductions.

Then, for this version of k-SUM, they obtained two different reductions showing that the problem is W[1]-hard and establishing that the problem is within $W[1]$.

Here, we proof that the problem is in W[1] reducing it to the model checking problem for W[1] class.

Background

Let us recall the definitions of fpt-reduction and the first level of W-hierarchy.

Given the instances (x,k) and (x',k') of parameterized problems P and P' respectively. The problem P is fpt-reductible to P', if there is an algorithm that computes an instance $(x',k') \in P'$ in time $f(k)\ |x|^c$, and there is an computable function g such that $k' \leq g(k)$.

Let $\tau = \{R_1, \dots, R_r, c_1, \dots, c_s\}$ be a set of relation symbols and constants symbols, and $\{x_1, x_2, \dots\}$ a countable set of variables. The language of First-order Logic is defined inductively, $x = y$ and $Rx_1, \dots x_r$ are formulas in First-order Logic. If φ , ψ are First-order formulas, $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$ are also First-order formulas.

The class W[1] is the first level of W-Hierarchy. For a definition of W[1], we consider the model checking problem for Σ_1 ($\varphi_k := \exists x_1, \dots, x_{f(k)}\ \psi$, such that ψ is quantifier-free formula), the existential fragment of FO, in some vocabulary τ.

MC (Σ_1)	
Instance:	A structure $\mathscr{A}$, a sentence $\varphi \in \Sigma_1$
Parameter:	$\|\varphi\|$
Problem:	Decide whether $\mathscr{A} \models \varphi$.

Theorem A.1. *(DOWNEY et al., 1998; FLUM; GROHE, 2003) For $t \geq 1, W[t] = [MC(STR[\tau], \Sigma_{t,l}^{FO})]^{FPT}$.*

To provide the result that k-SUM is in W[1], we need to encode a structure $\mathscr{A}$, and a formula φ from an positive instance of K-SUM.

Coding issues and k-SUM problem

The definition of k-SUM is given as it was formulated in (ABBOUD *et al.*, 2014).

Definition A.2. The (k,M)*-SUM problem* is to determine, given n integers $x_1, \cdots, x_n \in [0,M]$ and an integer $s \in [0,M]$, if there exists a subset $S \subseteq [n]$ of size $|S| = k$ such that $\sum_{i \in S} x_i = s$. The *k-SUM problem* is defined as $(k, n^{f(k)})$-SUM for some computable function f.

For the k-SUM problem, the parameter k will control at the same time the size of the solutions and the upper bound for the integer values assigned to x's.

In this work we will identify the k-SUM problem with the set of binary strings which represent positive solutions for the problem. Since k can be computed in polynomial time from the input of k-SUM, we could also codify k in this string. In this case,

$$k\text{-SUM} := \Big\{ \langle x_1 \rangle \ \dots \ \langle x_n \rangle \ \langle s \rangle \ \langle k \rangle \in \{0,1\}^* \mid \text{ there exists } S \subseteq [n],$$

$$\text{with } |S| = k \text{ such that} \sum_{i \in S} x_i = s \Big\}$$

These positive instances described in k-SUM have length equals to

$$|w| = f(k) \cdot (n+1) \cdot \log n + \log k \in \mathrm{O}(f(k) \cdot n \cdot logn),$$

and k can be obtained in time limited by $|w|$

The reduction to a model Checking Problem

The structure's encoding:

$$\langle \mathscr{A} \rangle \ := \underbrace{\langle 0 \rangle \dots \langle n \rangle \langle \langle R \rangle \langle \langle 1 \rangle \langle x_1 \rangle \rangle \dots \langle \langle n \rangle \langle x_n \rangle \rangle \rangle \langle 0 \rangle \dots \langle n+1 \rangle \langle s_1 \rangle \dots \langle s_{f(k)} \rangle}_{|\langle \mathscr{A} \rangle| \in O(n \cdot f(k) \cdot \log n)}$$

We need to encode the input for the model checking problem: A structure $\mathscr{A}$ and a formula $\varphi_k \in \Sigma_1$ such that $\mathscr{A} \vDash \varphi_k$

Theorem A.3. *There is a fpt-redution from the k-SUM problem to a model checking problem in* Σ_1

We can immediately represent an instance of k-SUM by a finite structure $\mathscr{A}$ over the vocabulary $\{R^{2k+1}, =, \leq, PLUS^3, 0, 1, n+1, s_1, \cdots, s_{2k}\}$ with domain $\{0, \cdots, n+1\}$. Then k-SUM can be expressed by a family of formulas $\{\phi_k\} \in \Sigma_1^{FO}$ in the form:

$$(\exists u_1, \cdots, u_k, \overline{v}_1, \cdots, \overline{v}_k) \bigwedge_{1 \leq i,j \leq k} \Big((u_i \neq u_j) \wedge (\overline{v}_i \neq^{4k} \overline{v}_j)\Big) \wedge \bigwedge_{i=1}^{k} R(u_i, \overline{v}_i) \wedge \left(\sum^{k} \overline{v}_i = \overline{s}\right),$$

where $R(i, \overline{j})$ is true when $x_i = \overline{j}$, with the constants $0, 1, n, s_1, \cdots s_{2k} \in |\mathscr{A}|$, $\overline{s} := \langle s_1, \cdots s_{2k} \rangle$, a fixed k from the problem, and the sum that can be expressed by

$$(\exists \overline{r}_1 \overline{r}_2 \cdots \overline{r}_{k-2}) PLUS^{6k}(\overline{v}_1, \overline{r}_1, \overline{s}) \wedge \bigwedge_{i=2}^{k-2} PLUS^{6k}(\overline{v}_i, \overline{r}_i, \overline{r}_{i-1}) \wedge PLUS^{6k}(\overline{v}_{k-1}, \overline{v}_k, \overline{r}_{k-2}). \tag{A.1}$$

k-SUM can be expressed by $\{\varphi_k\}$ formulas in Σ_1 in the form:

$$(\exists u_1, \ldots, u_{f(k)}, \overline{v}_1, \ldots, \overline{v}_{f(k)})$$
$$\bigwedge_{1 \leq i,j \leq k} \Big((u_i \neq u_j) \wedge (\overline{v}_i \neq^{2f(k)} \overline{v}_j)\Big) \wedge \bigwedge_{i=1}^{k} R(u_i, \overline{v}_i) \wedge \left(\sum^{k} \overline{v}_i =^{2f(k)} \overline{s}\right),$$

where $\overline{s} := \langle s_1, \ldots s_{f(k)} \rangle$. The sum above can be expressed by

$$(\exists \overline{r}_1 \overline{r}_2 \ldots \overline{r}_{f(k)-2}) \quad PLUS^{3f(k)}(\overline{v}_1, \overline{r}_1, \overline{s})$$
$$\wedge \bigwedge_{i=2}^{f(k)-2} PLUS^{3f(k)}(\overline{v}_i, \overline{r}_i, \overline{r}_{i-1}) \wedge PLUS^{3f(k)}(\overline{v}_{k-1}, \overline{v}_k, \overline{r}_{f(k)-2}).$$

As we can see, it is possible to define a $3f(k)$-ary PLUS relation for numbers in the interval $[n^{f(k)}]$ in the base n. With just one existential block of quantifiers, the length of the formula is a function of k. These variables represent the carries and the intermediate values.

$$\mathrm{PLUS}^{3f(k)}(x_1,\dots,x_{f(k)},y_1,\dots,y_{f(k)},z_1,\dots z_{f(k)}) :=$$

$$\exists u_1 \dots u_{f(k)}\ v_1 \dots v_{f(k)}$$

$$(\mathrm{PLUS}(x_{f(k)},y_{f(k)},z_{f(k)}) \wedge (v_{f(k)}=0) \wedge (z_{f(k)} \neq n+1))$$

$$\vee (\mathrm{PLUS}(x_{f(k)},u_{f(k)},n+1) \wedge \mathrm{PLUS}(u_{f(k)},z_{f(k)},y_{f(k)}) \wedge (v_{f(k)}=1))$$

$$\vee (\mathrm{PLUS}(u_{f(k)},y_{f(k)},n+1) \wedge \mathrm{PLUS}(u_{f(k)},z_{f(k)},x_{f(k)}) \wedge (v_{f(k)}=1))$$

$$\bigwedge_{i=f(k)-1}^{1} (\ ((v_{i+1}=0) \to (\mathrm{PLUS}(x_i,y_i,z_i) \wedge (v_i=0) \wedge (z_i \neq n+1))$$

$$\vee (\mathrm{PLUS}(x_i,u_i,n+1) \wedge \mathrm{PLUS}(u_i,z_i,y_i) \wedge (v_i=1))$$

$$\vee (\mathrm{PLUS}(u_i,y_i,n+1) \wedge \mathrm{PLUS}(u_i,z_i,x_i) \wedge (v_i=1)))$$

$$\wedge ((v_{i+1}=1) \to (\mathrm{PLUS}(x_i,y_i,z_i) \wedge (v_i=0) \wedge (z_i \neq n+1))$$

$$\vee (\mathrm{PLUS}(x_i,u_i,n+1) \wedge \mathrm{PLUS}(u_i,z_i,y_i) \wedge (v_i=1))$$

$$\vee (\mathrm{PLUS}(u_i,y_i,n+1) \wedge \mathrm{PLUS}(u_i,z_i,x_i) \wedge (v_i=1))))$$

Hence, in the form of Theorem (A.1) for the first level. This implies that k-SUM $\in W[1]$.

This question was not developed further. After some time the K-SUM was proved to be in W[1] by the machine characterization of the class in (MAJDODDIN, 2019).