On Parameterized Counting

Juan Andrés Montoya

Dissertation zur erlangung des Doktorgrades

der Fakultät für Mathematik und Physik

der Albert-Ludwigs-Universität Freiburg im Breisgau

July, 2008

Dekan:

Prof. Dr. Joerg Flum

Gutachter:

Prof. Dr. Joerg Flum

Prof. Dr. Yijia Chen

Datum der mündl. Prüfung: 05-09-2008

To my son

Preface

In this work we have tried to prove that parameterized exact counting is very hard and that parameterized approximate counting is almost tractable. To this end, we have looked for parameterized analogues of the theorems of Toda and Stockmeyer. By the way we have discovered that the main difficulty consists in designing efficient probability amplification algorithms. It is surprising because such a computational task is almost trivial from the classical point of view. This difficulty has led us to the following conclusion: The main limitation in parameterized structural complexity is that one is not allowed to perform product constructions. How can we overcome this limitation? We have made, in this work, some proposals which work well in some specific cases, but we believe that this question deserves further investigations.

Contents

Acknowledgments

I want to thank my advisor, Professor Joerg Flum, without his support I would have been unable to develop this work. Thanks go also to Moritz Muller because of his cultural support and our stimulating mathematical conversations. I apologize because of my little command of English. This work is dedicated to my son and my wife.

Chapter 1

Introduction

1.1 Parameterized Complexity

Parameterized complexity is a young but mature theory, [FG2], [DF]. Parameterized complexity theory provides a framework for a refined analysis of hard algorithmic problems. Let us quote Flum and Grohe ([FG2] preface, page v): "*Classical complexity theory, analyses problems by the amount of a resource, usually time or space, that is required by algorithms solving them. The amount of the resource required is measured as a function of the input size. Measuring complexity only in terms of the input size means ignoring any structural information about the input instances in the resulting complexity theory. Sometimes, this makes problems appear harder than they typically are. Parameterized complexity theory measures complexity not only in terms of the input size, but in addition in terms of a parameter, which is a numerical value that may depend on the input in an arbitrary way. The main intention of this theory is to address complexity issues in situations where we know that the parameter is comparatively small*".

A good example of such a situation is the problem of evaluating database queries. From the classical point of view this problem is tractable only in very restrictive cases, (the evaluation of conjunctive queries is already NP hard!). If one reviews the hardness proofs for the database evaluation problem, it is easy to note that it is necessary to consider instances for which the size of the query nontrivially depends on the size of the database. In real life, databases are huge and queries are small, which suggests that we can consider the size of the query as a parameter and measure the complexity of the problem in terms of two independent quantities, database size and query size. If we want to obtain something new we have to consider a new (parameterized) notion of tractability.

The central notion of parameterized complexity theory is *fixed parameter tractability.* This relaxes the classical notion of tractability, polynomial time solvability, by admitting algorithms whose nonpolynomial behavior is restricted only by the parameter, in addition the theory provides us with a *parameterized intractability theory,* which allow us to prove the intractability of certain problems, by classifying them into parameterized complexity classes by means of suitable parameterized reductions.

1.2 Counting Complexity

A typical class of computational problems is the class of counting problems. Counting problems are at least as hard as decision problems, because if we can count the number of solutions we can decide if there exists at least one solution. Counting complexity, the complexity analysis of counting problems, was developed by L. Valiant with a series of ground breaking articles published in 1979, [V1], [V2]. Valiant proved that some counting

problems are harder than expected. He proved that the problem of counting the number of perfect matchings in a graph is $\#P$ complete, this is surprising because the corresponding decision problem, the problem of deciding if a graph has at least one perfect matching belongs to P. The big surprise came next when S. Toda proved that every problem in the polynomial hierarchy can be reduced to any $\#P$ complete problem [T], that implies that every problem in the polynomial hierarchy can be reduced to the problem of counting the number of perfect matchings. Thus, we can conclude:

1. Hard counting problems are very much harder than the corresponding decision problems, (if the polynomial hierarchy does not collapse).

2. There are counting problems which are intractable, although the corresponding decision problems are tractable.

When we cope with counting problems we have the following alternative: We can try to compute approximate solutions instead of computing exact solutions. Approximating a counting problem is easier than computing exact solutions to the problem. Stockmeyer proved [S] that approximating the problem #**SAT** can be done in probabilistic polynomial time if oracle access to the decision problem **SAT** is provided, from this theorem Stockmeyer obtained, as a corollary, that probabilistic approximate counting belongs to the second level of the polynomial hierarchy.

Parameterized counting complexity is not yet a mature theory, there are few works on the topic [FG1], [M], and no structural theorems like the one of Toda or the one of Stockmeyer. It is important to remark that one of the main open problems in parameterized counting complexity is the proof of a parameterized analogue of Valiant's Theorem on the complexity

of counting matchings, however Flum and Grohe in [FG1] were able to prove that the counting of cycles and paths is hard from the parameterized point of view although the corresponding decision problems are fixed parameter tractable.

1.3 Parameterized probabilistic classes

Probabilistic computation is a major topic in complexity theory, there is a lot of work on the fine structure of probabilistic complexity classes. Parameterized probabilistic classes are introduced and studied, for the first time, in this research. We introduce parameterized probabilistic classes through a parameterized analogue of the classical BP operator.

We deal, in this thesis, with the following problem: Given a parameterized class $\mathcal{C}$, does $\mathcal{C}$ have the probability amplification property? Holding $\mathcal{C}$ the probability amplification property means that this class is robust, that is to say, given L a problem in $BP \cdot \mathcal{C}$ we can decrease its error probability by reducing L to some other problem in $BP \cdot \mathcal{C}$.

In chapter five we focus our research on the class $W[P]$. The problem considered in this chapter arises in the struggle for parameterized analogues of Toda's and Stockmeyer's theorems. This is the case because, if one try to mimic Toda's proof (or Stockmeyer's proof), one has to prove that the parameterized probabilistic classes defined using the BP operator have nice probability amplification properties. In this thesis we have tried to obtain parameterized analogues of some structural theorems of classical counting complexity. To this end, we have introduced and studied some parameterized operators analogous to the classical operators used in the definitions of the classes of the polynomial hierarchy and in the definitions of most of the classical probabilistic classes. The operator calculus that

arises from these definitions is related to the operator calculus of Downey and Fellows, (see [DF] chapter 16). We consider that our approach is more naive than Downey-Fellow's approach and because of this it has some technical advantages, but it is important to remark that, while our operator calculus works well only for parameterized classes above $W[P]$, the calculus of Downey and Fellows can be used to study the classes between FPT and $W[P]$, including the classes of the W hierarchy [DF].

1.4 A philosophical preface

Most readers should find that they can understand most of the dissertation on the basis of just the following mantra.

Given $n, k \in \mathbb{N}$, *one can identify the elements of the set* $\{0,1\}^{k\log(n)}$ *with the elements of the set* $\{s \in \{0,1\}^n : \|s\| \leq k\}$, *where* $\|s\|$ *denotes the Hamming weight of* s.

Such a mantra is only a way to express the $k\log(n)$ trick of Downey and Fellows, ([FG2] page 52). Our mantra yields the following point of view about parameterized classes: Parameterized classes are an special type of bounded nondeterministic classes, strongly related to the bounded nondeterministic classes of Kintala-Fisher [KF]. This point of view has some advantages that we have tried to exploit in this dissertation. A partial list of the *mantra applications* used in this work is the following one:

1. We could define some parameterized operators analogous to the operators studied by Toda in [T].

2. We could define a hierarchy of parameterized classes, the $PH[P]$ hierarchy, analogous to the polynomial hierarchy.

We have used these applications to develop a complexity theoretic analysis of parameterized counting. When we have to deal with counting problems we can consider two approaches: Exact counting and parameterized counting. In this dissertation we have tried to analyze the parameterized hardness of exact and approximate counting. In the classical framework there are two major theorems, which say a lot about the complexity of exact and approximate counting. Those theorems are Toda's theorem [T] and Stockmeyer's theorem [S].

Toda's theorem says that exact counting is very hard, it says that hard counting problems in $\#P$ are harder than any problem in the polynomial hierarchy.

Stockmeyer's theorem says that approximate counting is not very hard, it says that approximate counting of problems in $\#P$ belongs to the second level of the polynomial hierarchy.

The main goal of this dissertation was to obtain suitable parameterized versions of these theorems.

In chapter three we have tried to prove a parameterized analogue of Toda's theorem. If we are interested in proving a parameterized Toda's theorem, there are several possible statements that can be, (have to be), considered:

1. Is exact counting of problems in $\#W\,[1]$ harder than any problem in the W hierarchy?

2. Is exact counting of problems in $\#W\,[1]$ harder than any problem in the A hierarchy?

3. Is exact counting of problems in $\#W\,[P]$ harder than any problem in the A hierarchy?.

We have considered each one of the three possibilities listed above. After some attempts, we have realized that the third possibility was the most amenable to deal with. In this dissertation we prove the following theorem.

Theorem 1 *(Parameterized Toda's theorem)*

If $\oplus \cdot FPT$ *is maj-closed, then*

for all $L \in A$*, we have that* $L \preceq_R p\text{-}\#WSAT\,(CIRC)$.

Here $\preceq_R$ is a suitable notion of parameterized random reducibility. We claim that our parameterized Toda's theorem is a suitable parameterized version of Toda's theorem, since it says that exact counting of problems in $\#W\,[P]$ is harder than any problem in the A hierarchy. To prove our theorem we have tried to mimic the classical proof of Toda. By the way we had to solve many technical problems, but the structure of the proof resembles Toda's proof. The proof of our parameterized Toda's theorem can be divided in the following stages.

1. We proved a parameterized theorem of Valiant and Vazirani [VV] claiming that $W\,[P]$ is included in $BP \cdot \oplus \cdot FPT$.

2. We showed, using an inductive argument, that $PH\,[P]$ is included in $BP \cdot \oplus \cdot FPT$.

3. We observed that $A \subseteq PH\,[P]$ and $BP \cdot \oplus \cdot FPT \preceq_R p\text{-}\#WSAT\,(CIRC)$.

In chapter four we have tried to prove a parameterized analogue of Stockmeyer's theorem. We have obtained the following theorem.

Theorem 2 *(Parameterized Stockmeyer's theorem)*

If $W\,[P]$ *is* $\wedge$*-closed, then approximate counting of problems in* $\#W\,[P]$ *is included in the second level of the* $PH\,[P]$ *hierarchy.*

Remark 3 *We introduce below the notion of parameterized normal class and we prove that $W[P]$ is a parameterized normal class, it implies that $W[P]$ is closed under binary conjunctions. We say that $W[P]$ is $\wedge$-closed if it is closed under some special type of parameterized Turing reductions [DF], [FG2], (the exact definition will be introduced below), and this condition is stronger than being closed under binary conjunctions.*

We claim that our parameterized Stockmeyer's theorem is a suitable parameterized analogue of Stockmeyer's theorem. It says that approximate counting of problems in $\#W[P]$ is not very hard, i.e. the complexity of approximate counting for problems in $\#W[P]$ is close to the complexity of $p\text{-}WSAT(CIRC)$. The proof of our parameterized Stockmeyer's theorem is divided in the following stages:

1. We defined parameterized gap problems whose complexity is equivalent to the complexity of approximate counting.

2. We proved that those gap problems belong to $BP \cdot W[P]$.

3. We proved a parameterized analogue of the Theorem of Lautemann and Sipser [L] claiming that, if $W[P]$ is $\wedge$-closed, then $BP \cdot W[P]$ is included in the second level of the $PH[P]$ hierarchy.

It is important to remark that these two theorems can(should), be improved by:

1. Removing the hypothesis about the closure of $\oplus \cdot FPT$ under majority reductions.

2. Removing the hypothesis about the $\wedge$-closure of $W[P]$.

We deal with a related problem in chapter five, but we can say something in advance. It is easy to prove that if $\oplus \cdot FPT$ is closed under majority reductions, then $\oplus \cdot FPT$ is closed under $\wedge$-reductions. So, we can concentrate our efforts in analyzing the plausibility of the following hypothesis:

Hypothesis 4 *$W[P]$ and $\oplus \cdot FPT$ are $\wedge$-closed.*

Firs at all we note that:

Proposition 5 *1. If $W[P]$ is not $\wedge$-closed we have that $W[P] \neq FPT$.*

2. If $\oplus \cdot FPT$ is not $\wedge$-closed we have that $\oplus \cdot FPT \neq FPT$.

Proof. FPT is $\wedge$-closed ■

Corollary 6 *1. If $W[P]$ is not $\wedge$-closed we have $NP \neq P$.*

2. If $\oplus \cdot FPT$ is not $\wedge$-closed we have $\oplus \cdot P \neq P$.

It means that proving that our hypothesis are false, (if it is the case), is a very hard problem. We believe that $W[P]$ has more possibilities than $\oplus \cdot FPT$ of being $\wedge$-closed, because of this we focus our analysis on $W[P]$. We say some things about the implausibility of this hypothesis in chapter five and we consider that proving that $W[P]$ is $\wedge$-closed will represent a major breakthrough in parameterized complexity.

Toda's and Stockmeyer's theorems are the core of this dissertation. In the remaining chapters we have considered some technical matters. Let us finish this section with a remark about proof techniques. In this dissertation we have discovered that typical hashing arguments [G] can be easily adapted to the parameterized framework. Most of the main

technical arguments in this dissertation are based on hashing techniques. The applications of hashing in this dissertation are standard, we have mainly used the following two types of hashing applications:

1. Using hashing we can go from *There are at least one certificate* to *With high probability, there is exactly one certificate*, (see for example our proof of the parameterized theorem of Valiant and Vazirani).

2. Using hashing we can go from *Either there are many certificates or there are so few* to *The probability that there are at least one certificate is either very high or very small,* (see for example our proof of the theorem claiming that approximate counting belongs to $BP \cdot W[P]$).

1.5 Organization of the work

This dissertation is organized into seven chapters, including the introduction. In chapter two we introduce the basic concepts of parameterized complexity theory. The parameterized complexity of exact counting is studied in chapter three, this chapter is divided into five sections. In section 3.1 we introduce some parameterized operators, which are analogous to the classical operators, $\forall, \exists, \oplus$ and BP, these operators allow us to define new parameterized classes from old ones. Using the operators $\forall$ and $\exists$ we can define a hierarchy of parameterized classes analogous to the polynomial hierarchy. The BP operator allows us to define probabilistic parameterized classes. In section 3.2 we study the probability amplification properties of parameterized classes defined by means of the BP operator. We prove that a parameterized class $BP \cdot \mathcal{C}$ is well behaved, i.e. has good probability amplifica-

tion properties, if the ground class $\mathcal{C}$ is closed under parameterized majority reductions. To prove this fact we have to use the pseudorandom generator of Ajtai, Komlos and Szemeredi [G]. In section 3.3 we prove that the parameterized operator $BP \cdot \oplus$ is idempotent, i.e. $BP \cdot \oplus \cdot BP \cdot \oplus \cdot FPT$ is included in $BP \cdot \oplus \cdot FPT$. In section 3.4 we prove a parameterized analogue of the theorem of Valiant and Vazirani [VV], this theorem plays a major role in the proof of the main theorem in this chapter. To prove our parameterized version of the theorem of Valiant and Vazirani we have to use hashing techniques [G] adapted to the parameterized framework. In section 3.5 we prove the main theorem of the chapter, our parameterized Toda's theorem. Working on the results obtained in previous sections, (section 3.3, on probability amplification, and section 3.4, the parameterized version of the theorem of Valiant and Vazirani), the proof of the theorem is an easy inductive argument. Chapter four is about approximate counting, it is divided into three sections. In section 4.1 we study the complexity of approximating, within a constant range, counting problems in the class $\#W[P]$. To this end, we introduce and analyze the complexity of a family of suitable parameterized gap problems. In section 4.2 we introduce the parameterized Arthur-Merlin class $BP \cdot \exists \cdot FPT$, (it is important to remark that $BP \cdot \exists \cdot FPT$ is equal to $BP \cdot W[P]$), and we prove that the gap problems introduced before belong to this class. In subsection 4.3 we prove a parameterized analogue of the theorem of Lautemann and Sipser [L], that is, we prove that:

1. $BP \cdot FPT \subseteq \exists \cdot \forall \cdot FPT \cap \forall \cdot \exists \cdot FPT$.

2. If $W[P]$ is $\wedge$-closed, $BP \cdot W[P] \subseteq \forall \cdot \exists \cdot FPT$.

As a corollary we obtain a parameterized analogue of Stockmeyer's theorem, i.e. we prove that approximate counting of $\#W[P]$ problems belongs to the second level of the $PH[P]$ hierarchy. In chapter five we analyze the parameterized complexity of probability amplification, we prove that it is very unlikely that $W[P]$ has the probability amplification property. In chapter six we explore, in more depth, the $PH[P]$ hierarchy. Chapter six is divided into two sections. In section 6.1 some structural properties of the $PH[P]$ hierarchy are proved and machine characterizations for each one of the levels of the hierarchy are presented. In section 6.2 we study two different approaches to obtain descriptive characterizations for each one of the levels of the hierarchy. Finally in chapter seven some conclusions are presented.

Chapter 2

A technical Preface

In this chapter we introduce the basic definitions of parameterized complexity theory, much more information can be found in [DF] and in [FG2].

Notation 7 *$\{0,1\}^*$ is the set of finite 0-1 words. Given $n \in \mathbb{N}$, the symbol $\{0,1\}^n$ denotes the set of 0-1 words of length n.*

Definition 8 *A parameterized language (or parameterized problem) is a subset of $\{0,1\}^* \times \mathbb{N}$.*

Definition 9 *Given $s \in \{0,1\}^n$, the Hamming weight of s is equal to $|\{i \leq n : s(i) = 1\}|$ and is denoted by $\|s\|$.*

Definition 10 *p-$WSAT(CIRC)$ is the following parameterized problem:*

- *Instances: A circuit C and $k \in \mathbb{N}$.*
- *Parameter: k.*

- *Problem: Decide if there exists a satisfying assignment of C whose Hamming weight is less than or equal to k.*

Definition 11 *An fpt algorithm is an algorithm M such that, on input $(x,k) \in \{0,1\}^* \times \mathbb{N}$, the running time of M is upperbounded by $f(k)\, p(|x|)$, for some computable function f and some polynomial p.*

Definition 12 *The parameterized class FPT, is the class of parameterized problems which can be solved using an fpt algorithm.*

Given L, L^* two parameterized languages, L is *fpt many one reducible* to L^*, $(L \preceq_{fpt} L^*)$, if and only if there exists an *fpt* algorithm M such that, on input $(x,k) \in \Sigma^* \times \mathbb{N}$, the algorithm M computes a pair (x^*, k^*) which satisfies:

1. $(x,k) \in L$ if and only if $(x^*, k^*) \in L^*$.

2. $k^* \leq g(k)$, for some computable function g, which does not depend on (x,k).

A parameterized class is a set of parameterized languages closed under *fpt* many one reductions. Note that if $\mathcal{C}$ is a parameterized class, then $\mathcal{C} \subseteq FPT$. A typical parameterized language is a set $\{(x,k) : \varphi((x,k))\}$, where φ is some suitable predicate. If a parameterized language L is equal to $\{(x,k) : \varphi((x,k))\}$ we say that φ *defines* L. Given $\mathcal{C}$ a parameterized class and given $L = \{(x,k) : \varphi((x,k))\}$, we say that φ is a $\mathcal{C}$ predicate if and only if $L \in \mathcal{C}$. A parameterized class is a *parameterized normal class* if and only if given φ and α two $\mathcal{C}$ predicates, $\varphi \wedge \psi$ and $\varphi \vee \psi$ are $\mathcal{C}$ predicates. It is important to stress that FPT is a parameterized normal class. When we work with a parameterized normal classes we are

allowed to perform easy combinatorial manipulations, as for example boolean combinations, within the class. Typical parameterized classes are normal classes. We explicitly write parameterized normal class (or parameterized normal class) if it is important to stress this property of the parameterized class under consideration, but most of the time if we only write parameterized class we mean parameterized normal class.

Definition 13 *Given L, we have $\langle L\rangle_{fpt} := \{L^* : L^* \preceq_{fpt} L\}$.*

Given L a parameterized language, the class $\langle L\rangle_{fpt}$ is not necessarily a parameterized normal class, but for most of the parameterized languages that we will consider in this dissertation this is not the case.

Definition 14 $W[P] := \langle p\text{-}WSAT(CIRC)\rangle_{fpt}$.

It is easy to prove that $W[P]$ is a parameterized normal class, actually we will prove something more general (see proposition 30, chapter 3).

Definition 15 *A $W[P]$ restricted Turing machine $\mathbb{M}$ is a nondeterministic Turing machine such that:*

1. *There exist a computable function f and a polynomial p such that, on every run of $\mathbb{M}$ with input (x,k), the running time of $\mathbb{M}$ is upperbounded by $f(k)\,p(|x|)$.*
2. *There exists a computable function g such that, on every run of $\mathbb{M}$ with input (x,k), the machine $\mathbb{M}$ guesses at most $g(k)\log(|x|)$ nondeterministic bits.*

Theorem 16 *$L \in W[P]$ if and only if there exists a $W[P]$ restricted Turing machine $\mathbb{M}$ that decides L.*

A proof of this theorem can be found in ([FG1], theorem 3.9).

Definition 17 *A parameterized counting problem is a function* $h : \{0,1\}^* \times \mathbb{N} \to \mathbb{N}$.

Given h, h^* two parameterized counting problems, h is *parsimonious reducible* to h^*, (symbolically $h \preceq_{par} h^*$), if and only if there exists an *fpt* algorithm M such that, on input (x,k), M computes a pair (x^*, k^*) which satisfies:

1. $k^* \leq g(k)$ for some suitable computable function g.

2. $h((x,k)) = h^*((x^*, k^*))$.

Definition 18 *Given a parameterized counting problem* h, *we define*

$$\langle h \rangle_{par} := \{h^* : h^* \preceq_{par} h\}$$

Definition 19 *p-#WSAT (CIRC) is the following parameterized counting problem:*

- *Instances: A circuit* C *and* $k \in \mathbb{N}$.
- *Parameter:* k.
- *Problem: Compute the number of satisfying assignments of* C *whose Hamming weight is equal to* k.

Definition 20 $\#W[P] := \langle p\text{-}\#WSAT(CIRC) \rangle_{par}$.

To finish with this chapter we introduce the classes of the A hierarchy, which are defined as the closure under *fpt* reductions of some **m***odel* **c***hecking problems* related to fragments of first order logic [DF]. It is important to remark that all the classes in the A hierarchy are parameterized normal classes.

Definition 21 *Given* $t \in \mathbb{N}$, $p\text{-}MC\left(\Sigma_t\right)$ *is the following parameterized problem:*

- *Instances: A finite structure* $\mathcal{U}$ *and a* Σ_t *formula* α *in the vocabulary of* $\mathcal{U}$.
- *Parameter: The length of* α *denoted by* $|\alpha|$.
- *Problem: Decide if* $\mathcal{U} \vDash \alpha$.

Definition 22 *Given* $t \in \mathbb{N}$, $A\left[t\right] := \left\langle p\text{-}MC\left(\Sigma_t\right)\right\rangle_{fpt}$.

Chapter 3

On the parameterized complexity of exact counting

In this chapter we study the parameterized complexity of hard parameterized counting problems , specifically we study the parameterized complexity of the problem p-$\#WSAT\,(CIRC)$, the parameterized analogue of the $\#P$ complete problem $\#\mathbf{SAT}$. From the definition of $\#W\,[P]$ we have that the problem p-$\#WSAT\,(CIRC)$ is $\#W\,[P]$ complete. In this chapter we analyze the parameterized complexity of p-$\#WSAT\,(CIRC)$ by comparing it with a large family of parameterized decision problems, the parameterized problems located in the A hierarchy. This way of analyzing the complexity of a hard counting problem resembles Toda's analysis [T] of the complexity of $\#\mathbf{SAT}$. Actually in this paper we have tried to obtain, and we have partially obtained, a suitable parameterized analogue of Toda's theorem.

Toda's theorem [T] states that hard counting problems are harder than any problem in the

polynomial hierarchy, specifically Toda's theorem says that every problem in the polynomial hierarchy is Turing reducible to $\sharp$**SAT** . We prove that any parameterized problem in the A hierarchy is random reducible to p-$\sharp WSAT\,(CIRC)$.

Theorem 23 *(Parameterized Toda's theorem)*

For all $i \in \mathbb{N}$ *and for all* $L \in A\,[i]$

$L \preceq_R p$-$\sharp WSAT\,(CIRC)$.

Here, $\preceq_R$ is a notion of parameterized random reducibility that will be introduced at the end of the chapter.

3.1 The Basic Operators

In this section we introduce some basic operators and list some of their basic properties. Each one of the operators and each one of the properties, (that we list and prove in this section), plays some role in the proof of our parameterized Toda's theorem. We introduce these operators to define some parameterized classes, these classes are analogous to the classical classes that are used in the work of Toda.

Notation 24

1. *Given* $n, m, s \in \mathbb{N}$*, the expression* $n \equiv m\,(s)$ *denotes that* n *is congruent with* m *modulus* s*.*

2. *Given* f *a computable function, we will use the symbol* $\{0,1\}^f$ *to denote the set* $\{0,1\}^{f(k)\log(|x|)}$*.*

3. *Given* $a \in \{0,1\}$ *and given* $n \in \mathbb{N}$*, the symbol* a^n *denotes the word* $\underbrace{a...a}_{n\ times}$ *.*

Definition 25 *Given L a parameterized language and given $\mathcal{C}$ a parameterized class*

1. *$L \in \exists \cdot \mathcal{C}$ if and only if there exist $\Omega \in \mathcal{C}$ and f a computable function such that*

   $(x,k) \in L \iff \exists y \in \{0,1\}^{f} \left((x,y,k) \in \Omega\right)$.

2. *$L \in \forall \cdot \mathcal{C}$ if and only if there exist $\Omega \in \mathcal{C}$ and f a computable function such that*

   $(x,k) \in L \iff \forall y \in \{0,1\}^{f} \left((x,y,k) \in \Omega\right)$.

3. *$L \in BP \cdot \mathcal{C}$ if and only if there exist $\Omega \in \mathcal{C}$ and f a computable function such that*

   - $(x,k) \in L \Rightarrow \Pr_{y \in \{0,1\}^{f}} \left[(x,y,k) \in \Omega\right] \geq \frac{3}{4}$.
   - $(x,k) \notin L \Rightarrow \Pr_{y \in \{0,1\}^{f}} \left[(x,y,k) \in \Omega\right] \leq \frac{1}{4}$.

4. *$L \in \oplus \cdot \mathcal{C}$ if and only if there exist $\Omega \in \mathcal{C}$ and f a computable function such that*

   $(x,k) \in L \iff \left|\left\{y \in \{0,1\}^{f} : \left((x,y,k) \in \Omega\right)\right\}\right| \equiv 1\,(2)$.

Remark 26 *(On notation) Most of the time we will be involved with parameterized classes obtained from FPT applying at least two of our operators. Consider for example the class $\forall \cdot \exists \cdot FPT$, from the definition we have that given $L \in \forall \cdot \exists \cdot FPT$ there exist $\Omega \in \exists \cdot FPT$ and a computable function f such that, given $(x,k) \in \{0,1\}^{*} \times \mathbb{N}$, the pair (x,k) belongs to L if and only if there exists $y \in \{0,1\}^{f}$ such that $(x,y,k) \in \Omega$. Now, we try to use our definition on Ω, a typical instance of Ω is a triple $(x,y,k) \in \{0,1\}^{*} \times \{0,1\}^{*} \times \mathbb{N}$, it looks problematic because in each one of our definitions we have considered languages whose instances are pairs, i.e. elements of $\{0,1\}^{*} \times \mathbb{N}$, but it is not really a problem because elements of $\{0,1\}^{*} \times \{0,1\}^{*} \times \mathbb{N}$, (and more generally, elements of $\left(\{0,1\}^{*}\right)^{n} \times \mathbb{N}$, with $n \geq 1$), can be effectively codified as elements of $\{0,1\}^{*} \times \mathbb{N}$ in such a way that the size*

of the code linearly depends on the size of the triple to be codified. So, we can apply our definition to Ω. *Thus, we have that there exist* $\Delta \in FPT$ *and a computable function* g *such that given* $(x, y, k) \in \{0,1\}^* \times \{0,1\}^* \times \mathbb{N}$, *the triple* $(x, y, k) \in \Omega$ *if and only if there exists* $z \in \{0,1\}^{g(k)\log(|(x,y)|)}$ *such that* $(x, y, z, k) \in \Delta$. *Now, we observe that there exists* $m \in \mathbb{N}$ *such that*

$$|(x,y)| \leq m\left(|x| + |y|\right) \leq m\left(|x| + f\left(k\right)\log\left(|x|\right)\right)$$

Thus, we have that

$$\log\left(|(x,y)|\right) \leq \log\left(m\left(|x| + f\left(k\right)\log\left(|x|\right)\right)\right) \leq h\left(k\right)\log\left(|x|\right)$$

for some computable function h. *So, we can suppose without loss of generality, that there exists a computable function* t *such that given* $(x, y, k) \in \{0,1\}^* \times \{0,1\}^* \times \mathbb{N}$, *we have that* $(x, y, k) \in \Omega$ *if and only if there exits* $z \in \{0,1\}^{t(k)\log(|x|)}$ *such that* $(x, y, z, k) \in \Delta$.

Probability amplification is a very important property of some of the classes of the form $BP \cdot \mathcal{C}$. This property plays a major role in most of the arguments in which these classes are involved. We will introduce below a formal notion of probability amplification.

Definition 27 *The Majority function is the boolean function* $\bigotimes$ *defined in the following way, given* $(a_1, ..., a_n)$ *a boolean vector, we have that*

$$\bigotimes_{i \leq n} a_i = 1 \text{ if and only if } |\{i \leq n : a_i = 1\}| \gneqq \frac{n}{2}$$

Lemma 28 *Let* A, B *be two sets such that* $A \subseteq B$. *For all* $n \geq 1$, *there exists* $m_n \in \mathbb{N}$ *such that*

1. *If* $\Pr_{a \in B}\left[a \in A\right] \geq \frac{3}{4}$, *then* $\Pr_{a_1,...,a_{m_n} \in B}\left[\bigotimes_{i \leq m_n} a_i \in A\right] \geq 1 - \frac{1}{n}$.

2. If $\Pr_{a\in B}[a\in A]\leq\frac{1}{4}$*, then* $\Pr_{a_1,\ldots,a_{m_n}\in B}\left[\bigotimes_{i\leq m_n} a_i\in A\right]\leq\frac{1}{n}$.

Proof. First we note that 2 is a consequence of 1. We suppose $A\subseteq B$ such that $\Pr_{a\in B}[a\in A]=\frac{3}{4}$. First, given $i,j\in\mathbb{N}$, with $i\leq j$ and j odd, we upperbounded $q_i=\Pr_{a_1,\ldots,a_j\in A}\left[|\{k\leq j:a_k\in A\}|=i\right]$. If $i\leq\frac{j}{2}$ we have

$$q_i=\left(\begin{array}{c} j \\ i \end{array}\right)\left(\frac{3}{4}\right)^i\left(\frac{1}{4}\right)^{j-i}\leq\left(\begin{array}{c} j \\ i \end{array}\right)\left(\frac{3}{4}\right)^i\left(\frac{1}{4}\right)^{j-i}3^{\frac{j}{2}-i}=\left(\begin{array}{c} j \\ i \end{array}\right)\left(\frac{3}{16}\right)^{\frac{j}{2}}$$

Let p_j be equal to $\Pr_{a_1,\ldots,a_j\in A}\left[\bigotimes_{k\leq j} a_k\in A\right]$, we have that

$$p_j=\sum_{k\geq\frac{j}{2}}^{j}q_k=1-\sum_{k=0}^{\frac{j-1}{2}}q_k\geq 1-\sum_{k=0}^{\frac{j-1}{2}}\left(\begin{array}{c} j \\ k \end{array}\right)\left(\frac{3}{16}\right)^{\frac{j}{2}}\geq 1-\frac{1}{2}\left(\frac{3}{8}\right)^{\frac{j}{2}}$$

Thus, we have that if $j\geq\frac{2\log(n)}{3-\log(3)}$ then $p_j\geq 1-\frac{1}{n}$. So, given $n\geq 1$ we can take m_n equal to the smallest odd integer bigger than $\frac{2\log(n)}{3-\log(3)}$ ■

Proposition 29 *(Weak probability amplification) Let* $\mathcal{C}$ *be a parametrized normal class, given* $L\in BP\cdot\mathcal{C}$ *and given* $n\in\mathbb{N}$*, there exist* $\Omega\in\mathcal{C}$ *and a computable function* f *such that*

- $(x,k)\in L\Rightarrow\Pr_{y\in\{0,1\}^f}[(x,y,k)\in\Omega]\geq 1-\frac{1}{n}$.
- $(x,k)\notin L\Rightarrow\Pr_{y\in\{0,1\}^f}[(x,y,k)\in\Omega]\leq\frac{1}{n}$.

Proof. Given $L\in BP\cdot\mathcal{C}$, there exist $\Phi\in\mathcal{C}$ and a computable function g such that

- $(x,k)\in L\Rightarrow\Pr_{y\in\{0,1\}^g}[(x,y,k)\in\Phi]\geq\frac{3}{4}$
- $(x,k)\notin L\Rightarrow\Pr_{y\in\{0,1\}^g}[(x,y,k)\in\Phi]\leq\frac{1}{4}$

We fix $n \in \mathbb{N}$ and we consider the language Ω defined in the following way $(x, y_1, ..., y_{m_n}, k) \in \Omega$ if and only if

$$\left((y_1, ..., y_{m_n} \in \{0,1\}^g) \ \& \ \left(\bigotimes_{i \leq m_n} (x, y_i, k) \in \Phi \right) \right)$$

where m_n is taken like in last lemma. Note that m_n does not depend on (x, k) and as a consequence we have that the predicate that we are using to define the language Ω is a boolean combination of $\mathcal{C}$ predicates, then a $\mathcal{C}$ predicate. Hence, $\Omega \in \mathcal{C}$. Moreover, if we take $A = \{y \in \{0,1\}^g : (x, y, k) \in \Phi\}$, we obtain from last lemma

- $(x, k) \in L \Rightarrow \Pr_{y \in \{0,1\}^{m_n g}} [(x, y, k) \in \Omega] \geq 1 - \frac{1}{n}$.
- $(x, k) \notin L \Rightarrow \Pr_{y \in \{0,1\}^{m_n g}} [(x, y, k) \in \Omega] \leq \frac{1}{n}$.

■

Next lemma says that all the parameterized classes considered in this work are parameterized normal classes, (since FPT is a parameterized normal class).

Proposition 30 *If $\mathcal{C}$ is a parameterized normal class, then $BP \cdot \mathcal{C}$, $\oplus \cdot \mathcal{C}$, $\forall \cdot \mathcal{C}$ and $\exists \cdot \mathcal{C}$ are parameterized normal classes.*

Proof. The verifications for the cases $\forall \cdot \mathcal{C}$ and $\exists \cdot \mathcal{C}$ are straightforward. We make the verifications for the operator BP, the proofs for the operator $\oplus$ are easy but tedious. If α and φ are $BP \cdot \mathcal{C}$ predicates, we can use weak probability amplification to claim that there exist Ω_α, $\Omega_\varphi \in \mathcal{C}$ and two computable functions f and g such that

- $(x, k) \in \{(z, l) : \alpha\} \Rightarrow \Pr_{y \in \{0,1\}^f} [(x, y, k) \in \Omega_\alpha] \geq \frac{7}{8}$.

- $(x,k) \notin \{(z,l) : \alpha\} \Rightarrow \Pr_{y \in \{0,1\}^f}[(x,y,k) \in \Omega_\alpha] \leq \frac{1}{8}$.

and

- $(x,k) \in \{(z,l) : \varphi\} \Rightarrow \Pr_{y \in \{0,1\}^g}[(x,y,k) \in \Omega_\varphi] \geq \frac{7}{8}$.
- $(x,k) \in \{(z,l) : \varphi\} \Rightarrow \Pr_{y \in \{0,1\}^g}[(x,y,k) \in \Omega_\varphi] \leq \frac{1}{8}$.

Since $\mathcal{C}$ is normal the language $\Omega_{\alpha \vee \varphi}$ defined by $(x,y,k) \in \Omega_{\alpha \vee \varphi}$ if and only if

$$\Big(y \in \{0,1\}^{\max\{f(k)\log(|x|), g(k)\log(|x|)\}} \ \ \& \ \ ((x,y_f,k) \in \Omega_\alpha \text{ or } (x,y_g,k) \in \Omega_\varphi)\Big)$$

belongs to $\mathcal{C}$, where y_f denotes the boolean vector of length $f(k)\log(|x|)$ whose entries are the first $f(k)\log(|x|)$ entries of y. It is clear that

- $(x,k) \in \{(z,l) : \alpha \vee \varphi\} \Rightarrow \Pr_y[(x,y,k) \in \Omega_{\alpha \vee \varphi}] \geq \frac{3}{4}$.
- $(x,k) \notin \{(z,l) : \alpha \vee \varphi\} \Rightarrow \Pr_y[(x,y,k) \in \Omega_{\alpha \vee \varphi}] \leq \frac{1}{4}$.

Thus, we have obtained a representation of $\{(z,l) : \alpha \vee \varphi\}$ as a member of $BP \cdot \mathcal{C}$ and we can conclude that $\alpha \vee \varphi$ is a $BP \cdot \mathcal{C}$ predicate.

Now we want to prove that $\{(z,l) : \alpha \wedge \varphi\} \in BP \cdot \mathcal{C}$. Since $\mathcal{C}$ is normal the language $\Omega_{\alpha \wedge \varphi}$ defined by $(x,y,k) \in \Omega_{\alpha \wedge \varphi}$ if and only if

$$\Big(y \in \{0,1\}^{\max\{f(k)\log(|x|), g(k)\log(|x|)\}} \ \ \& \ \ ((x,y_f,k) \in \Omega_\alpha \ \& \ (x,y_g,k) \in \Omega_\varphi)\Big)$$

belongs to $\mathcal{C}$. It is clear that

- $(x,k) \in \{(z,l) : \alpha \wedge \varphi\} \Rightarrow \Pr_y[(x,y,k) \in \Omega_{\alpha \wedge \varphi}] \geq \frac{3}{4}$.
- $(x,k) \notin \{(z,l) : \alpha \wedge \varphi\} \Rightarrow \Pr_y[(x,y,k) \in \Omega_{\alpha \wedge \varphi}] \leq \frac{1}{4}$.

So, we can conclude that $\alpha \wedge \varphi$ is a $BP \cdot \mathcal{C}$ predicate. ■

We say that an operator F is monotone if and only if given $\mathcal{C}$ and $\mathcal{C}^*$ two parameterized classes:

1. $\mathcal{C} \subseteq F \cdot \mathcal{C}$.
2. $\mathcal{C} \subseteq \mathcal{C}^*$ implies $F \cdot \mathcal{C} \subseteq F \cdot \mathcal{C}^*$.

Lemma 31 *Let $\mathcal{C}$ be a parameterized class*

1. *$\exists$, $\forall$, $\oplus$, and BP are monotone.*
2. *$\exists \cdot FPT = W[P]$.*
3. *$co - BP \cdot \mathcal{C} \subseteq BP \cdot (co - \mathcal{C})$.*
4. *If $co - \mathcal{C} = \mathcal{C}$, then $co - BP \cdot \mathcal{C} = BP \cdot \mathcal{C}$.*
5. *$co - \oplus \cdot \mathcal{C} = \oplus \cdot \mathcal{C}$.*
6. *$\oplus \cdot \oplus \cdot \mathcal{C} = \oplus \cdot \mathcal{C}$.*
7. *$BP \cdot BP \cdot \mathcal{C} = BP \cdot \mathcal{C}$.*

Proof. The proof of item 1 is straightforward, item 2 is an easy consequence of the machine characterization of $W[P]$, item 4 is an easy consequence of item 3. We prove items 3, 5, 6 and 7.

- (item 3) Given $L \in co - BP \cdot \mathcal{C}$, there exist $\Omega \in \mathcal{C}$ and f a computable function such that

1. $(x,k) \in L \Rightarrow \Pr_{y\in\{0,1\}^f}[(x,y,k) \in \Omega] \leq \frac{1}{5}$.
2. $(x,k) \notin L \Rightarrow \Pr_{y\in\{0,1\}^f}[(x,y,k) \in \Omega] \geq \frac{4}{5}$.

Hence

1. $(x,k) \in L \Rightarrow \Pr_{y\in\{0,1\}^f}[(x,y,k) \in \Omega^c] \geq \frac{3}{4}$.
2. $(x,k) \notin L \Rightarrow \Pr_{y\in\{0,1\}^f}[(x,y,k) \in \Omega^c] \leq \frac{1}{4}$.

We can conclude that $L \in BP\cdot(co-\mathcal{C})$, since $\Omega^c \in co-\mathcal{C}$.

- (item 5) We prove that $co-\oplus\cdot\mathcal{C}$ is included in $\oplus\cdot\mathcal{C}$, note that it is sufficient because given $\mathcal{C}$ a parameterized class, if $co-\mathcal{C}$ is included in $\mathcal{C}$, then $co-\mathcal{C}$ is equal to $\mathcal{C}$. So, let L be a language in $co-\oplus\cdot\mathcal{C}$, we have that there exist $\Omega \in \mathcal{C}$ and f a computable function such that

$$(x,k) \in L \Leftrightarrow \left|\left\{y \in \{0,1\}^f : (x,y,k) \in \Omega\right\}\right| \equiv 0\,(2)$$

What must we do? We only have to transform an even number in an odd number, the easiest way to perform such a task is by adding one, i.e. we have to add an additional trivial solution, but we have to ensure that this trivial solution is not already a solution, we can enforce this if we add some bits and use these bits as markers. So, we define a language Ω^* in the following way

$(x,y,k) \in \Omega^*$ if and only if

$y \in \{0,1\}^{2f}$ & $\left(y = 0^{2f(k)\log(|x|)}\right.$ or $\left.\left(y = 1^{f(k)\log(|x|)}z \text{ \& } (x,z,k) \in \Omega\right)\right)$.

It is clear that $\Omega^* \in \mathcal{C}$ (since $\mathcal{C}$ is a normal class) and

$$\left|\left\{y \in \{0,1\}^{2f} : (x,y,k) \in \Omega^*\right\}\right| = \left|\left\{y \in \{0,1\}^{f} : (x,y,k) \in \Omega\right\}\right| + 1.$$

Thus, we have proven that there exist $\Omega^* \in \mathcal{C}$ and a computable function g $(g = 2f)$ such that

$(x,k) \in L \Leftrightarrow |\{y \in \{0,1\}^g : (x,y,k) \in \Omega^*\}| \equiv 1\,(2)\,.$

So, we can conclude that $L \in \oplus \cdot \mathcal{C}$.

- (item 6) Since $\oplus$ is monotone $\oplus \cdot \mathcal{C} \subseteq \oplus \cdot \oplus \cdot \mathcal{C}$. Given $L \in \oplus \cdot \oplus \cdot \mathcal{C}$ there exist $\Phi \in \oplus \cdot \mathcal{C}$ and a computable function f such that

  $(x,k) \in L \Leftrightarrow \left|\left\{y \in \{0,1\}^f : (x,y,k) \in \Phi\right\}\right| \equiv 1\,(2)\,.$

  Furthermore, there exist $\Omega \in \mathcal{C}$ and a computable function g such that

  $(x,y,k) \in \Phi \Leftrightarrow |\{z \in \{0,1\}^g : (x,y,z,k) \in \Omega\}| \equiv 1\,(2)\,.$

  Suppose (x,k) is an instance of L, which is the size of the next set?

  $$\left\{(y,z) \in \{0,1\}^f \times \{0,1\}^g : (x,y,z,k) \in \Omega\right\}$$

  Let $S_{x,k}$ be the set $\left\{y \in \{0,1\}^f : (x,y,k) \in \Phi\right\}$, and given $y \in \{0,1\}^f$, let $S_{x,y,k}$ be the set $\{z \in \{0,1\}^g : (x,y,z,k) \in \Omega\}$. We have that

  $$\begin{aligned} & \left\{(y,z) \in \{0,1\}^f \times \{0,1\}^g : (x,y,z,k) \in \Omega\right\} \\ = & \bigcup_{y \in S_{x,k}} (\{y\} \times S_{x,y,k}) \cup \bigcup_{y \notin S_{x,k}} (\{y\} \times S_{x,y,k}) \end{aligned}$$

  Note that

  1. If $y \in S_{x,k}$, then $|S_{x,y,k}|$ is an odd number.
  2. If $y \notin S_{x,k}$, then $|S_{x,y,k}|$ is an even number.
  3. $(x,k) \in L$ if and only if $|S_{x,k}|$ is an odd number.

From these observations we can conclude that

$$(x,k) \in L \Leftrightarrow \left|\left\{(y,z) \in \{0,1\}^f \times \{0,1\}^g : (x,y,z,k) \in \Omega\right\}\right| \equiv 1 \,(2)$$

since, given $(x,k) \in L$ the solution set of (x,k) is the disjoint union of an odd number of sets of odd size (from 1 and 3) plus an union of sets of even size (from 2). Furthermore, given $(x,k) \notin L$, the solution set of (x,k) is the disjoint union of an even number of sets of odd size (again from 1 and 3) plus an union of sets of even size (from 2). Thus, we have obtained a representation of L as a member of $\oplus \cdot \mathcal{C}$.

- (item 7) Since BP is monotone $BP \cdot \mathcal{C} \subseteq BP \cdot BP \cdot \mathcal{C}$. Given $L \in BP \cdot BP \cdot \mathcal{C}$, there exist $\Phi \in BP \cdot \mathcal{C}$ and a computable function f such that

  1. $(x,k) \in L \Rightarrow \Pr_{y \in \{0,1\}^f}[(x,y,k) \in \Phi] \geq \frac{9}{10}$.
  2. $(x,k) \notin L \Rightarrow \Pr_{y \in \{0,1\}^f}[(x,y,k) \in \Phi] \leq \frac{1}{10}$.

  Furthermore, there exist $\Omega \in \mathcal{C}$ and a computable function g such that

  1. $(x,y,k) \in \Phi \Rightarrow \Pr_{z \in \{0,1\}^g}[(x,y,z,k) \in \Omega] \geq \frac{9}{10}$.
  2. $(x,y,k) \notin \Phi \Rightarrow \Pr_{z \in \{0,1\}^g}[(x,y,z,k) \in \Omega] \leq \frac{1}{10}$.

  Thus, we have

  1. $(x,k) \in L \Rightarrow \Pr_{(y,z) \in \{0,1\}^f \times \{0,1\}^g}[(x,y,z,k) \in \Omega] \geq \frac{81}{100} \geq \frac{3}{4}$.
  2. $(x,k) \notin L \Rightarrow \Pr_{(y,z) \in \{0,1\}^f \times \{0,1\}^g}[(x,y,z,k) \in \Omega] \leq \frac{19}{100} \leq \frac{1}{4}$.

  So, we can conclude that $L \in BP \cdot \mathcal{C}$

■

3.2 Majority Reductions and Probability Amplification

A probabilistic parameterized class $BP \cdot \mathcal{C}$ is well behaved if $BP \cdot \mathcal{C}$ has some type of probability amplification, i.e. $BP \cdot \mathcal{C}$ is well behaved if given $L \in BP \cdot \mathcal{C}$ we can decrease the error probability associated to L by reducing L to some other problem in the class $BP \cdot \mathcal{C}$. Here, we introduce a formal notion of *well behavedness* that we call the *pam property*.

For some of the arguments in the previous section we had to use weak probability amplification. For most of the main arguments in this dissertation we will need a stronger property, the *pam* property.

Definition 32 *Given $\mathcal{C}$ a parameterized class, $\mathcal{C}$ has the pam property if and only if for all $L \in BP \cdot \mathcal{C}$ and for all computable function g, there exist $\Omega \in \mathcal{C}$ and a computable function f such that*

- $(x,k) \in L \Rightarrow \Pr_{y \in \{0,1\}^f}[(x,y,k) \in \Omega] \geq 1 - 2^{-g(k)\log(|x|)}$.
- $(x,k) \notin L \Rightarrow \Pr_{y \in \{0,1\}^f}[(x,y,k) \in \Omega] \leq 2^{-g(k)\log(|x|)}$.

In this section we explore the relation between the closure of $\mathcal{C}$ under majority reductions and the probability amplification properties of $BP \cdot \mathcal{C}$. We prove that if $\mathcal{C}$ is maj-closed, then $\mathcal{C}$ has the *pam* property. The proof of this theorem is very similar to the classical analogue, but in addition we have to use in the proof the pseudorandom generator of Ajtai, Komlos and Szemeredi [AKS] in order to save random bits.

Definition 33 *L is majority reducible to L^* if and only if there exist an fpt algorithm M and two computable functions f, g such that, on input (x,k), the algorithm M computes a sequence $(x_1,k_1),...,\left(x_{f(k)\log(|x|)},k_{f(k)\log(|x|)}\right)$ which satisfies:*

1. $(x,k) \in L \Leftrightarrow \bigotimes_{j \leq f(k)\log(|x|)} (x_j, k_j) \in L^*$.

2. *For all* $i \leq f(k)\log(|x|)$, *we have that* $k_i \leq g(k)$.

We will say that $\mathcal{C}$ is maj-closed if and only if $\mathcal{C}$ is closed under majority reductions. Next theorem says that in order to amplify the success probability (equivalently, to decrease the error probability), we can make a big saving of random bits if we use a suitable *pseudorandom generator*.

Theorem 34 *(AKS theorem)*

There exist an algorithm, namely AKS, and constants $N_1, N_2 \in \mathbb{N}$, *such that for every* $i, m \in \mathbb{N}$ *and for all* $a \in \{0,1\}^{N_1(m+i)}$, *on input* (a,i,m), *the algorithm AKS computes a sequence* $a_1, ..., a_{iN_2} \in \{0,1\}^m$ *such that for all* $A \subseteq \{0,1\}^m$

1. $|A| \geq \frac{3}{4}2^m \Rightarrow \Pr_{a \in \{0,1\}^{N_1(m+i)}} \left[\bigotimes_{j \leq iN_2} a_j \in A\right] \geq 1 - 2^{-i}$.

2. $|A| \leq \frac{1}{4}2^m \Rightarrow \Pr_{a \in \{0,1\}^{N_1(m+i)}} \left[\bigotimes_{j \leq iN_2} a_j \in A\right] \leq 2^{-i}$.

3. *The running time of AKS is bounded by a polynomial* $p(m,i)$.

Remark 35 *The algorithm AKS is the pseudorandom generator of Ajtai, Komlos and Szemeredi [AKS] which is based on expander graphs [G]. A proof of the theorem and some interesting remarks about its meaning can be found in [LW] pages 26-27 and 14-15.*

Using AKS theorem we can easily prove the following theorem which says us that there exists a deep relation between the closure under majority reductions and probability amplification.

Theorem 36 *If* $\mathcal{C}$ *is maj-closed, then* $\mathcal{C}$ *has the pam property.*

Proof. Let L, Ω be two languages such that $L \in BP \cdot \mathcal{C}$, $\Omega \in \mathcal{C}$ and

- $(x,k) \in L \Rightarrow \Pr_{y \in \{0,1\}^f}[(x,y,k) \in \Omega] \geq \frac{3}{4}$.
- $(x,k) \notin L \Rightarrow \Pr_{y \in \{0,1\}^f}[(x,y,k) \in \Omega] \leq \frac{1}{4}$.

where f is some computable function. Given g a computable function we define Ω^g in the following way

$$\Omega^g := \left\{ (x,y,k) : y \in \{0,1\}^{N_1(f+g)} \ \& \bigotimes_{j \leq N_2 g(k)\log(|x|)} (x, z_j, k) \in \Omega \right\}$$

where $z_1, \ldots, z_{N_2 g(k)\log(|x|)}$ is the output sequence of AKS on input

$$(y, g(k)\log(|x|), f(k)\log(|x|))$$

i.e. we are setting $a = y$; $i = g(k)\log(|x|)$; $m = f(k)\log(|x|)$ and

$A = \left\{ y \in \{0,1\}^f : (x,y,k) \in \Omega \right\}$.

Note that $\Omega^g \in \mathcal{C}$ because $\mathcal{C}$ is maj-closed and given (x,k) an instance of L

- $(x,k) \in L \Rightarrow \left| \left\{ y \in \{0,1\}^f : (x,y,k) \in \Omega \right\} \right| \geq \frac{3}{4} 2^{f(k)\log(|x|)}$.
- $(x,k) \notin L \Rightarrow \left| \left\{ y \in \{0,1\}^f : (x,y,k) \in \Omega \right\} \right| \leq \frac{1}{4} 2^{f(k)\log(|x|)}$.

It follows from the AKS Theorem that

- $(x,k) \in L \Rightarrow \Pr_{y \in \{0,1\}^{N_1(f+g)}}[(x,y,k) \in \Omega^g] \geq 1 - 2^{-g(k)\log(|x|)}$.
- $(x,k) \notin L \Rightarrow \Pr_{y \in \{0,1\}^{N_1(f+g)}}[(x,y,k) \in \Omega^g] \leq 2^{-g(k)\log(|x|)}$.

Therefore we can conclude that $\mathcal{C}$ has the *pam* property ■

Corollary 37 *FPT has the pam property.*

Proof. FPT is closed under majority reductions because FPT is closed under parameterized Turing reductions [DF] ■

3.3 $BP \cdot \oplus$ is idempotent

We say that an operator Δ is *idempotent* if and only if we have that $\Delta \cdot \Delta \cdot FPT = \Delta \cdot FPT$. The classical class $BP \cdot \oplus \cdot P$ plays a major role in the work of Toda. An important step in the proof of Toda is proving that the classical operator $BP \cdot \oplus$ is idempotent. If we suppose that the parameterized class $\oplus \cdot FPT$ is closed under majority reductions we can prove that the parameterized operator $BP \cdot \oplus$ is idempotent.

Theorem 38 *If $\oplus \cdot FPT$ is maj-closed*

1. *$\oplus \cdot FPT$ has the pam property.*
2. *$\oplus \cdot BP \cdot \oplus \cdot FPT \subseteq BP \cdot \oplus \cdot \oplus \cdot FPT$.*
3. *$\exists \cdot BP \cdot \oplus \cdot FPT \subseteq BP \cdot \exists \cdot \oplus \cdot FPT$.*
4. *$BP \cdot \oplus \cdot BP \cdot \oplus \cdot FPT = BP \cdot \oplus \cdot FPT$.*

Proof.

1. This is a corollary of last theorem.
2. We can prove something more general. Given $\mathcal{C}$ a parameterized class, if $\mathcal{C}$ has the *pam* property, then $\oplus \cdot BP \cdot \mathcal{C} \subseteq BP \cdot \oplus \cdot \mathcal{C}$.

Let L be a language in $\oplus \cdot BP \cdot \mathcal{C}$. We want to prove that $L \in BP \cdot \oplus \cdot \mathcal{C}$. There exist $\Delta \in BP \cdot \mathcal{C}$ and a computable function f such that $(x,k) \in L$ if and only if

$$\left| \left\{ z \in \{0,1\}^f : (x,z,k) \in \Delta \right\} \right| \equiv 1 \,(2)$$

Given h a computable function, since $\mathcal{C}$ has the *pam* property, there exist $\Phi_h \in \mathcal{C}$ and a computable function $g_h = N_1(f+h)$ such that

- $(x,z,k) \in \Delta \Rightarrow \Pr_{y \in \{0,1\}^{g_h}} [(x,y,z,k) \in \Phi_h] \geq 1 - 2^{-h(k)\log(|x|)}$.
- $(x,z,k) \notin \Delta \Rightarrow \Pr_{y \in \{0,1\}^{g_h}} [(x,y,z,k) \in \Phi_h] \leq 2^{-h(k)\log(|x|)}$.

Given x, z, k, we define two sets $S(x,k)$ and $H(x,y,k)$ in the following way

$$S(x,k) := \left\{ v \in \{0,1\}^f : \Pr_{y \in \{0,1\}^{g_{2f}}} [(x,y,v,k) \in \Phi_{2f}] \geq 1 - 2^{-2f(k)\log(|x|)} \right\}$$

and

$$H(x,z,k) := \{ y \in \{0,1\}^{g_{2f}} : (x,y,z,k) \in \Phi_{2f} \}.$$

From the definition of $S(x,k)$ we have:

(a) If $(x,k) \in L$, then $|S(x,k)|$ is an odd number.

(b) If $(x,k) \notin L$, then $|S(x,k)|$ is an even number

(c) If $z \notin S(x,k)$, then

$$\Pr_{y \in \{0,1\}^{g_{2f}}} [(x,y,z,k) \in \Phi_{2f}] = \Pr_y [y \in H(x,z,k)] \leq 2^{-2f(k)\log(|x|)}$$

(d) If $z \in S(x,k)$, then

$$\Pr_{y \in \{0,1\}^{g_{2f}}} [(x,y,z,k) \notin \Phi_{2f}] = \Pr_y [y \in H(x,z,k)^c] \leq 2^{-2f(k)\log(|x|)}$$

Note that

$$\Pr_y\left[y \in \left(\bigcap_{z\in S(x,k)} H(x,z,k) \cap \bigcap_{z\notin S(x,k)} (H(x,z,k))^c\right)\right] \geq 1 - 2^{-f(k)\log(|x|)} \qquad \text{(in1)}$$

Inequality (in1) follows from

$$\Pr_y\left[y \in \left(\bigcap_{z\in S(x,k)} H(x,z,k) \cap \bigcap_{z\notin S(x,k)} (H(x,z,k))^c\right)\right] \geq$$

$$1 - \Pr_y\left[y \in \left(\bigcup_{z\in S(x,k)} H(x,z,k)^c \cup \bigcup_{z\notin S(x,k)} H(x,z,k)\right)\right] \geq$$

$$1 - \left(\sum_{z\in S(x,k)} \Pr_y[y \in H(x,z,k)^c] + \sum_{z\notin S(x,k)} \Pr_y[y \in H(x,z,k)]\right) \geq$$

$$1 - 2^{f(k)\log(|x|)}\left(2^{-2f(k)\log(|x|)}\right) = 1 - 2^{-f(k)\log(|x|)}$$

And from $[in1]$ we obtain:

- If $(x,k) \in L$, then

  $\Pr_y\left[\left|\left\{z \in \{0,1\}^f : (x,y,z,k) \in \Phi_{2f}\right\}\right| \equiv 1\,(2)\right] \geq 1 - 2^{-f(k)\log(|x|)}$.

  This is the case because if $(x,k) \in L$, we have that

  $$\Pr_y\left[\left|\left\{z \in \{0,1\}^f : (x,y,z,k) \in \Phi_{2f}\right\}\right| \equiv 1\,(2)\right] \geq$$

  $$\Pr_y\left[y \in \left(\bigcap_{z\in S(x,k)} H(x,z,k) \cap \bigcap_{z\notin S(x,k)} (H(x,z,k))^c\right)\right]$$

  Last inequality follows from the following fact

  If $y \in \left(\bigcap_{z\in S(x,k)} H(x,z,k) \cap \bigcap_{z\notin S(x,k)} (H(x,z,k))^c\right)$, we have that

  $\left\{z \in \{0,1\}^f : (x,y,z,k) \in \Phi_{2f}\right\} = S_{x,k}$ and $|S_{x,k}|$ is an odd number, since $(x,k) \in L$ (it follows from a). Thus, we have

  $$\Pr_y\left[\left|\left\{z \in \{0,1\}^f : (x,y,z,k) \in \Phi_{2f}\right\}\right| \equiv 1\,(2)\right] \geq$$

$$\Pr_y\left[y \in \left(\bigcap_{z \in S(x,k)} H(x,z,k) \cap \bigcap_{z \notin S(x,k)} (H(x,z,k))^c\right)\right] \geq 1 - 2^{-f(k)\log(|x|)}$$

- If $(x,k) \notin L$, then

  $\Pr_y\left[\left|\left\{z \in \{0,1\}^f : (x,y,z,k) \in \Phi_{2f}\right\}\right| \equiv 1\,(2)\right] \leq 2^{-f(k)\log(|x|)}$.

  Here, the argument is very similar. If $(x,k) \notin L$, we have that

  $$\Pr_y\left[\left|\left\{z \in \{0,1\}^f : (x,y,z,k) \in \Phi_{2f}\right\}\right| \equiv 1\,(2)\right] \leq$$

  $$\Pr_y\left[y \in \left(\bigcap_{z \in S(x,k)} H(x,z,k) \cap \bigcap_{z \notin S(x,k)} (H(x,z,k))^c\right)^c\right] \leq 2^{-f(k)\log(|x|)}$$

3. The proof of item three is very similar to the proof of item two, we omit the details.

4. From item two and the monotonicity of the BP and $\oplus$ operators we have that

   $BP \cdot \oplus \cdot BP \cdot \oplus \cdot FPT \subseteq BP \cdot BP \cdot \oplus \cdot \oplus \cdot FPT \subseteq BP \cdot BP \cdot \oplus \cdot FPT$.

   Now, we use that $\oplus \cdot FPT$ is a normal class to claim that $BP \cdot BP \cdot \oplus \cdot FPT = BP \cdot \oplus \cdot FPT$. Thus, we have that $BP \cdot \oplus \cdot BP \cdot \oplus \cdot FPT \subseteq BP \cdot \oplus \cdot FPT$. The other inclusion is a consequence of monotonicity

■

3.4 The theorem of Valiant and Vazirani

Toda's Theorem is, in some sense, a generalization of the Theorem of Valiant and Vazirani. The Theorem of Valiant and Vazirani says that NP is random reducible to UP. A weaker version of this theorem says that NP is included in $BP \cdot \oplus \cdot P$, this weaker statement of the Theorem of Valiant and Vazirani plays a major role in the proof of Toda because it is the first step of the inductive argument that is used by Toda in his remarkable proof.

In this section we prove a parameterized analogue of the weak version of the theorem of Valiant and Vazirani, our parameterized version is the following theorem

Theorem 39 *Given $\mathcal{C}$ a parameterized class, if $\mathcal{C}$ is maj-closed, then $\exists \cdot \mathcal{C} \subseteq BP \cdot \oplus \cdot \mathcal{C}$.*

To prove our parameterized version of the theorem of Valiant and Vazirani we use hashing techniques similar to the techniques used in some proofs of the classical theorem [LW], [VV]. The proof of the theorem is divided in two stages. In the first stage using hashing techniques we prove that for every language $L \in \exists \cdot \mathcal{C}$ there exist a language $\Omega \in \mathcal{C}$ and two computable functions f, g such that

- $(x,k) \in L \Rightarrow \Pr_{z \in \{0,1\}^g} \left[\left| \left\{ y \in \{0,1\}^f : (x,y,z,k) \in \Omega \right\} \right| \equiv 1 \,(2) \right] \geq \frac{1}{16 f(k) \log(|x|)}$.
- $(x,k) \notin L \Rightarrow \Pr_{z \in \{0,1\}^g} \left[\left| \left\{ y \in \{0,1\}^f : (x,y,z,k) \in \Omega \right\} \right| \equiv 1 \,(2) \right] = 0$.

In the second stage we use the *pairwise independence sampling technique [LW],* (also called *two bit sampling*), to amplify probabilities and to obtain the following result:
for every language $L \in \exists \cdot \mathcal{C}$ there exist a language $\Omega \in \mathcal{C}$ and two computable functions f, g such that

- $(x,k) \in L \Rightarrow \Pr_{z \in \{0,1\}^g} \left[\left| \left\{ y \in \{0,1\}^f : (x,y,z,k) \in \Omega \right\} \right| \equiv 1 \,(2) \right] \geq \frac{2}{3}$.
- $(x,k) \notin L \Rightarrow \Pr_{z \in \{0,1\}^g} \left[\left| \left\{ y \in \{0,1\}^f : (x,y,z,k) \in \Omega \right\} \right| \equiv 1 \,(2) \right] = 0$.

3.4.1 The first stage, Valiant-Vazirani's hashing

The main theorem of this subsection is the following one

Theorem 40 *Given $L \in \exists \cdot \mathcal{C}$, there exist $\Omega_M \in \mathcal{C}$ and two computable functions f, g such that*

- $(x,k) \in L \Rightarrow \Pr_{z \in \{0,1\}^g}\left[\left|\left\{y \in \{0,1\}^f : (x,y,z,k) \in \Omega_M\right\}\right| \equiv 1\,(2)\right] \geq \frac{1}{16 f(k) \log(|x|)}$.
- $(x,k) \notin L \Rightarrow \Pr_{z \in \{0,1\}^g}\left[\left|\left\{y \in \{0,1\}^f : (x,y,z,k) \in \Omega_M\right\}\right| \equiv 1\,(2)\right] = 0$.

The core of the argument, that is used in the proof of the theorem, is an standard application of hashing. Hashing allow us to reduce a language for which every positive instance has a certificate into a language for which every positive instance has, with high probability, a unique certificate. Note that this is the type of the reduction that we are looking for.

Notation 41 *Given A and B two sets, B^A denotes the set of functions from A to B.*

Definition 42 *Given A, B two sets, a set $H_{A,B} \subseteq B^A$ is an U_2-hashing family if and only if for all $a, b \in A$, (with $a \neq b$), and for all $c, d \in B$*

$\Pr_{h \in H_{A,B}}[h(a) = c \;\&\; h(b) = d] = \frac{1}{|B|^2}$.

From here on, the letters A, B will denote two sets and $H_{A,B}$ will denote an U_2-universal family of hashing functions from A to B.

Proposition 43 *Given a and b two elements of A, if $a \neq b$ we have*

$\Pr_{h \in H_{A,B}}[h(a) = h(b)] = \frac{1}{|B|}$.

Proof. $\Pr_{h \in H_{A,B}}[h(a) = h(b)] = \sum_{c \in B} \Pr_{h \in H_{A,B}}[h(a) = c \;\&\; h(b) = c] = |B| \frac{1}{|B|^2} = \frac{1}{|B|}$ ■

Let S be a subset of A of size N, we will consider the random variable X_S, with domain $H_{A,B}$, defined by

$X_S(h) := \left|\left\{(a,b) \in S^2 : a \neq b \;\&\; h(a) = h(b)\right\}\right|$.

For the expected value $E[X_S]$ of X_S we have

Proposition 44 $E[X_S] = \begin{pmatrix} N \\ \\ 2 \end{pmatrix} \frac{1}{|B|}$.

Proof. Given $a, b \in S$, with $a \neq b$, we consider the indicator variable $X_{(a,b)}$, defined on $H_{A,B}$ and given by

$X_{(a,b)}(h) = 1$ if and only if $h(a) = h(b)$.

We note that

1. $X_S = \sum_{a \neq b} X_{(a,b)}$.
2. The random variables $\left\{X_{(a,b)} : a \neq b\right\}$ are independent.

From 1 and 2 we obtain

$$E[X_S] = \sum_{a \neq b} E\left[X_{(a,b)}\right] = \sum_{a \neq b} \Pr_{h \in H_{A,B}}[h(a) = h(b)] = \begin{pmatrix} N \\ \\ 2 \end{pmatrix} \frac{1}{|B|} \quad \blacksquare$$

Corollary 45 *If $|B| = \lambda N$, then $E[X_S] \leq \frac{N}{2\lambda}$.*

Proof. $E[X_S] = \begin{pmatrix} N \\ \\ 2 \end{pmatrix} \frac{1}{\lambda N} = \frac{N(N-1)}{2\lambda N} = \frac{N-1}{2\lambda} \leq \frac{N}{2\lambda}$ $\blacksquare$

Notation 46 *Given $n, r \in \mathbb{N}$ with $r \leq n$ and given $z \in \{0,1\}^n$, $(z) \restriction_r$ is the boolean vector of length r whose entries are the first r entries of y.*

The set $H^*_{n,r} := \{h_{a,b} : a, b \in \{0,1\}^n \ \& \ h_{a,b}(z) := (az + b) \restriction_r\}$ is a U_2-hashing family [G], (the arithmetical operations are computed in the field $GF(2^n)$). In this section we we will work, all the time, with these U_2-hashing families.

Remark 47 *It is important to stress that the number of bits required to specify a function $h \in H^*_{n,r}$ is $O(n)$, since in order to specify h, one only have to specify a pair $(a, b) \in \{0,1\}^{2n}$.*

In the following, the symbol $\mathcal{C}$ will denote a parameterized normal class and L will denote a language in $\exists \cdot \mathcal{C}$. We know, since $L \in \exists \cdot \mathcal{C}$, that there exist a language $\Omega \in \mathcal{C}$ and a computable function f such that

$(x,k) \in L$ if and only $\left\{ y \in \{0,1\}^{f} : (x,y,k) \in \Omega \right\} \neq \varnothing$

Notation 48 *We will use the symbol $S_{x,k}$ to denote the set $\left\{ y \in \{0,1\}^{f} : (x,y,k) \in \Omega \right\}$.*

We define a new parameterized language $\oplus \cdot L$ in the following way

Definition 49 *$\oplus \cdot L$ is the following parameterized language*

- *Input:* (x,k,h,r,α), *where* $r \leq f(k) \log(|x|)$; $h \in H^{*}_{f(k)\log(|x|),r+1}$ *and* $\alpha \in \{0,1\}^{r+1}$.
- *Parameter:* k.
- *Problem: Decide if* $\left| S_{x,k} \cap h^{-1}(\alpha) \right|$ *is an odd number.*

Proposition 50 *$\oplus \cdot L$ belongs to $\oplus \cdot \mathcal{C}$.*

Proof. First, we note that

$(x,k,h,r,\alpha) \in \oplus \cdot L$ if and only if $\left| \left\{ y \in \{0,1\}^{f} : (x,y,k) \in \Omega \ \& \ h(y) = \alpha \right\} \right| \equiv 1 \,(2)$.

Now, we note that the predicate $(h(y) = \alpha)$ is an FPT predicate, since it can be decided in time $(f(k) \log(|x|))^{m}$ for some natural number m. To finish with the proof we have to verify that the predicate $((x,y,k) \in \Omega \ \& \ h(y) = \alpha)$ is a $\mathcal{C}$ predicate. This is the case, since $(x,y,k) \in \Omega$ is a $\mathcal{C}$ predicate, the predicate $(h(y) = \alpha)$ is an FPT predicate and $\mathcal{C}$ is a parameterized normal class ■

Definition 51 *Given Δ and Φ two parameterized languages, Δ is RP-reducible to Φ if and only if there exists a probabilistic algorithm M such that, for some computable function g and for every instance (x,k) of Δ we have*

1. *The running time of every run of* M*, on input* (x,k)*, is upperbounded by* $g(k)\,p(|x|)$*, for some polynomial* p*.*

2. *On every run of* M *with input* (x,k)*, the output* $M(x,k)$ *is an instance* (x^*,k^*) *of* Φ *such that* $k^* \leq g(k)$*.*

3. *On every run of* M *with input* (x,k)*, the algorithm* M *uses* $g(k)\log(|x|)$ *random bits.*

4. *Finally we have*

   - *If* $(x,k) \in \Delta$*, then* $\Pr[M(x,k) \in \Phi] \geq \frac{1}{h(k)\log(|x|)}$*.*
   - *If* $(x,k) \notin \Delta$*, then* $\Pr[M(x,k) \in \Phi] = 0$*.*

   where h *is some computable function and the probability is computed on the random guesses of* M*.*

Before proving our parameterized (weak) theorem of Valiant and Vazirani, we have to remember the well known Markov's inequality.

Lemma 52 *(Markov's inequality)*

Suppose $\mathbb{P}$ *is a probability space. Given* X *a random variable on* $\mathbb{P}$*,* $E[X]$ *the expected value of* X *and* $\mu \gneqq 0$*, we have*

$\Pr_{a\in\mathbb{P}}[X(a) \geq \mu] \leq \frac{E[X]}{\mu}$.

Proof. Suppose that $\Pr_{a\in\mathbb{P}}[X(a) \geq \mu] \gneqq \frac{E[X]}{\mu}$, then we have that

$$E[X] \geq \sum_{a:X(a)\geq\mu} X(a)\,p(a) \geq \mu \Pr_{a\in\mathbb{P}}[X(a) \geq \mu] \gneqq \mu\frac{E[X]}{\mu} = E[X] \quad \blacksquare$$

Now, we can state and prove the parameterized (weak) theorem of Valiant and Vazirani.

Theorem 53 *(Parameterized weak Valiant-Vazirani's theorem) L is RP-reducible to $\oplus \cdot L$.*

Proof. First we suppose that $|S_{x,k}| \leq 2^{f(k)\log(|x|)-1}$. We can suppose this without loss of generality. This is the case, since given $L \in \exists \cdot \mathcal{C}$ and given $\Delta \in \mathcal{C}$ such that for some computable function g

$(x,k) \in L$ if and only if $\exists y \in \{0,1\}^{g} \left((x,y,k) \in \Delta\right)$

We can define a new language $\Delta^* \in \mathcal{C}$ (remember that $\mathcal{C}$ is a parameterized normal class), in the following way

$\Delta^* := \left\{(x,z,k) : z \in \{0,1\}^{2g} \ \& \ \left(x, z \restriction_{g(k)\log(|x|)}, k\right) \in \Delta \ \& \ \phi\right\}$

where ϕ is equal to $z_{g(k)\log(|x|)+1} ... z_{2g(k)\log(|x|)} = 0^{g(k)\log(|x|)}$. It is clear that

- $(x,k) \in L$ if and only if $\exists z \in \{0,1\}^{2g} \left((x,z,k) \in \Delta^*\right)$.
- $\left|\left\{z \in \{0,1\}^{2g} : \left((x,z,k) \in \Delta^*\right)\right\}\right| \leq 2^{g(k)\log(|x|)} \leq 2^{2g(k)\log(|x|)-1}$.

So, let us continue with the proof. Let M be the following probabilistic algorithm:

On input (x,k)

1. Choose $r \in \{1, ..., f(k)\log(|x|) - 1\}$.
2. Choose $h \in H^*_{f(k)\log(|x|),r+1}$.
3. Choose $\alpha \in \{0,1\}^{r+1}$.
4. Queries (x,k,h,r,α) to the oracle $\oplus \cdot L$. Print oracle's answer.

If $(x,k) \notin L$, then clearly M will respond NO. If $(x,k) \in L$, then for some

$$s \in \{1, ..., f(k)\log(|x|) - 1\}$$

we have that $2^{s-1} \leq |S_{x,k}| \leq 2^s$. It is clear that

$$\Pr_r [r = s] \geq \frac{1}{f(k)\log(|x|) - 1} \geq \frac{1}{f(k)\log(|x|)}$$

Assume $r = s$. If we set $A = \{0,1\}^f$, $B = \{0,1\}^{r+1}$, $S = S_{x,k}$, $N = |S_{x,k}|$, $H_{A,B} = H^*_{f(k)\log(|x|),r+1}$ and $X_S = X_{S_{x,k}}$, we obtain from corollary 45 that $E\left[X_{S_{x,k}}\right]$, the expected number of colliding pairs, is less than or equal to $\frac{|S_{x,k}|}{2\lambda}$, for some $\lambda \in [2,4]$. Thus, $E\left[X_{S_{x,k}}\right] \leq \frac{|S_{x,k}|}{4}$. Given $h \in H^*_{f(k)\log(|x|),r+1}$ we say that h is a *good* function if and only if h satisfies the following inequality

$X_{S_{x,k}}(h) = \left|\left\{(a,b) \in S^2_{x,k} : a \neq b \;\&\; h(a) = h(b)\right\}\right| \lneqq \frac{|S_{x,k}|}{2}$.

We can use Markov's inequality, taking $\mu = \frac{|S_{x,k}|}{2}$, to prove that

$$\Pr_{h \in H^*_{f(k)\log(|x|),r+1}} [h \text{ is good}] \geq \frac{1}{2}$$

Suppose h is good, consider the set C_h defined by

$\{a \in S_{x,k} : \forall b\,(a \neq b \Rightarrow h(a) \neq h(b))\}$

Which is the size of C_h? Let D_h be the set

$\left\{(a,b) \in S^2_{x,k} : a \neq b \;\&\; h(a) = h(b)\right\}$

Note that $C_h = (\pi_1(D_h))^c$, where π_1 is the projection from $S^2_{x,k}$ to $S_{x,k}$. It implies that $|C_h| = |S_{x,k}| - |\pi_1(D_h)| \gneqq |S_{x,k}| - \frac{|S_{x,k}|}{2} = \frac{|S_{x,k}|}{2}$. Hence, if h is a good function, we have that $|h(C_h)| \geq \frac{|\{0,1\}^{r+1}|}{8}$, (where $h(C_h)$ is the image of $S_{x,k}$ under h). Thus, we have

$$\text{if } h \text{ is good} \Rightarrow \Pr_{\alpha \in \{0,1\}^{r+1}} [\alpha \in h(C_h)] \geq \frac{1}{8}$$

To finish with the proof we only have to observe that: If (x,k) is a YES instance of L and $2^{s-1} \leq |S_{x,k}| \leq 2^s$ we have

$\Pr_{h,r,\alpha}\left[\left|\left\{y \in \{0,1\}^f : (x,y,k) \in \Omega \;\&\; h(y) = \alpha\right\}\right| \equiv 1\,(2)\right] \geq$

$\Pr_{h,r,\alpha}\left[r = s \;\&\; h \text{ is good } \&\; \alpha \in h\left(C_h\right)\right] \geq \left(\frac{1}{f(k)\log(|x|)}\right)\left(\frac{1}{2}\right)\left(\frac{1}{8}\right) = \frac{1}{16f(k)\log(|x|)}$ ■

Remark 54 *Note that the random words (h, r, α) are elements of $\{0,1\}^{4f}$ (if we add some dummy bits), i.e. the size of the random certificates for (x, k) is linearly related to the size of the nondeterministic certificates.*

From last theorem we obtain as an easy corollary the main result of this subsection.

Corollary 55 *Given $L \in \exists \cdot \mathcal{C}$, there exist $\Omega_M \in \mathcal{C}$ and a computable function f such that*

- $(x,k) \in L \Rightarrow \Pr_{z \in \{0,1\}^{4f}}\left[\left|\left\{y \in \{0,1\}^f : (x,y,z,k) \in \Omega_M\right\}\right| \equiv 1\,(2)\right] \geq \frac{1}{16f(k)\log(|x|)}$.
- $(x,k) \notin L \Rightarrow \Pr_{z \in \{0,1\}^{4f}}\left[\left|\left\{y \in \{0,1\}^f : (x,y,z,k) \in \Omega_M\right\}\right| \equiv 1\,(2)\right] = 0$.

Note that for the case $(x,k) \in L$ we have obtained the lower bound $\frac{1}{16f(k)\log(|x|)}$. Unfortunately this lower bound is still very small. In next subsection we will use the pairwise independence sampling technique in order to increase this bound (in order to amplify the success probability).

3.4.2 The second stage, improving the bounds

In this subsection we prove our parameterized theorem of Valiant and Vazirani. We work on the results obtained in last subsection. We use the parameterized weak Valiant-Vazirani's theorem and two bit sampling (i.e. the pairwise independence sampling technique) to obtain our theorem.

Given $i \in \mathbb{N}$ and $j \in GF\left(2^i\right)$, we consider the random variable Y_j^i, with domain $GF\left(2^i\right)^2$, defined by

$Y_j^i((a,b)) := aj + b$

where $a, b \in GF\left(2^i\right)$ and the arithmetical operations are the operations of the Galois field $GF\left(2^i\right)$.

Claim 56 *For all $i \geq 0$ and for all $j \in GF\left(2^i\right)$, the random variable Y_j^i is uniformly distributed on $GF\left(2^i\right)$, i.e. for all $c \in GF\left(2^i\right)$ we have that* $\Pr_{a,b}\left[Y_j^i((a,b)) = c\right] = \frac{1}{2^i}$.

Proof. Fix i, j, a and c; we have that $Y_j^i((a,b)) = c$ if and only if $b = c - aj$. Hence, we have

$\Pr_{a,b}\left[Y_j^i((a,b)) = c\right] = \frac{2^i}{2^{2i}} = \frac{1}{2^i}$ ■

Claim 57 *The sequence $Y_1^i, ..., Y_{2^i}^i$ is **pairwise independent**, i.e. for all $c, d \in GF\left(2^i\right)$ and for all $j_1, j_2 \in GF\left(2^i\right)$. If $j_1 \neq j_2$, then*

$\Pr_{a,b}\left[Y_{j_1}^i((a,b)) = c \;\&\; Y_{j_2}^i((a,b)) = d\right] = \frac{1}{2^{2i}}$.

Proof. Fix i, j_1, j_2 and c, with $j_1 \neq j_2$; we have that $Y_{j_1}^i((a,b)) = c$ and $Y_{j_2}^i((a,b)) = d$ if and only if (a,b) is a solution of the linear system

$$\begin{bmatrix} j_1 & 1 \\ j_2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix}$$

Note that this system has exactly one solution, namely (a_0, b_0), since

$$\det\left(\begin{bmatrix} j_1 & 1 \\ j_2 & 1 \end{bmatrix}\right) = j_1 - j_2 \neq 0$$

Thus, we have that

$\Pr_{a,b}\left[Y_{j_1}^i((a,b)) = c \;\&\; Y_{j_2}^i((a,b)) = d\right] = \Pr_{a,b}\left[a = a_0 \;\&\; b = b_0\right] = \frac{1}{2^{2i}}$ ■

Lemma 58 *Given $X_1, ..., X_m$ a sequence of pairwise independent random variables and $X = X_1 + ... + X_m$, we have that*

$\sigma_X^2 = \sum_{i \leq m} \sigma_{X_i}^2$.

Proof. $\sigma_X^2 = E\left[\left(\sum_{i \leq m} X_i\right)^2\right] - E\left[\sum_{i \leq m} X_i\right]^2 =$

$$\sum_{i \leq m} E\left[X_i^2\right] + \sum_{i \neq j} 2E\left[X_i X_j\right] - \left(\sum_{i \leq m} E\left[X_i\right]^2 + \sum_{i \neq j} 2E\left[X_i\right] E\left[X_j\right]\right)$$

If we use the pairwise independence of the sequence $X_1, ..., X_m$, we have that given $i, j \leq m$, if $i \neq j$, then $E\left[X_i X_j\right] = E\left[X_i\right] E\left[X_j\right]$. It implies that

$\sigma_X^2 = \sum_{i \leq m} E\left[X_i^2\right] - \sum_{i \leq m} E\left[X_i\right]^2 = \sum_{i \leq m} E\left[X_i^2\right] - E\left[X_i\right]^2 = \sum_{i \leq m} \sigma_{X_i}^2$ ■

Lemma 59 *(Chebyshev's Inequality)*

Given X a random variable with expected value ρ_X and standard deviation equal to σ_X, we have that

$\Pr\left[|X - \rho_X| \geq \lambda \sigma_X\right] \leq \frac{1}{\lambda^2}$.

Proof. Consider the random variable $Y = (X - \rho_X)^2$. Note that $E[Y] = \sigma_X^2$. Now we use Markov's Inequality, taking $\mu = (\lambda \sigma_X)^2$, to obtain

$\Pr\left[|X - \rho_X| \geq \lambda \sigma_X\right] \leq \Pr\left[Y \geq (\lambda \sigma_X)^2\right] \leq \frac{\sigma_X^2}{\lambda^2 \sigma_X^2} = \frac{1}{\lambda^2}$ ■

We are ready to use two bit sampling in order to increase the meager lower bound in the parameterized weak Valiant-Vazirani's theorem. First, we state, without proof, an easy technical lemma.

Lemma 60 *Given $i \geq 7$, $\varepsilon \in \left[0, 2^{-i}\right]$ and $M(i, \varepsilon) = 16i^2\epsilon\left(1 - \epsilon - \frac{1}{2i}\right)$ we have that*

1. $16i \lneqq 2^i$.

2. $\frac{M(i,\epsilon)}{12i} \lneqq \frac{1}{9}$.

3. $4i - 1 + M(i, \varepsilon) \neq 0$.

The following technical theorem is the core of our two bit sampling amplification algorithm.

Theorem 61 *(Two bit sampling amplification)*

Suppose $i \geq 7$ *and suppose that* A *is a subset of* $GF\left(2^i\right)$ *such that* $\Pr_y[y \in A] \geq \frac{1}{4i}$, *we have that*

$\Pr_{a,b}\left[\bigvee_{j \leq 16i}(aj + b) \in A\right] \geq \frac{5}{9}$

Proof. Pick $A_0 \subseteq A$ such that for some $\varepsilon \in \left[0, 2^{-i}\right]$ we have that $\Pr_y[y \in A_0] = \frac{1}{4i} + \varepsilon$. Given $j \leq 16i$, the random variable X_j is the bivalued random variable, with domain $GF\left(2^i\right)^2$, defined by

$X_j((a,b)) := 1$ if and only if $(aj + b) \in A_0$

For all $j \leq 16i$, we have $E(X_j) = \frac{1}{4i} + \varepsilon$ and

$\sigma^2_{X_j} = E\left[X_j^2\right] - E[X_j]^2 = \frac{1}{4i} + \varepsilon - \left(\frac{1}{4i} + \varepsilon\right)^2 = \frac{4i-1}{(4i)^2} + \varepsilon\left(1 - \varepsilon - \frac{1}{2i}\right)$.

Let X be the random variable $X_1 + ... + X_{16i}$. It is clear that $E(X) = 4 + 16i\varepsilon$. Moreover we have, as a consequence of the pairwise independence of the sequence $X_1, ..., X_{16i}$, that

$\sigma^2_X = 16i\left(\frac{4i-1}{(4i)^2} + \varepsilon\left(1 - \varepsilon - \frac{1}{2i}\right)\right) = \frac{4i-1+M(i,\varepsilon)}{i}$.

Given $\lambda := \frac{\sqrt{12}}{\sigma_X}$ we have

$\Pr[X = 0] \leq \Pr\left[|X - E(X)| \geq \sqrt{12}\right] = \Pr[|X - E(X)| \geq \lambda\sigma_X]$

$\leq \frac{1}{\lambda^2} = \frac{4i-1+M(i,\varepsilon)}{12i} = \frac{4i-1}{12i} + \frac{M(i,\varepsilon)}{12i} \lneqq \frac{4i}{12i} + \frac{1}{9} = \frac{4}{9}$.

Thus, $\Pr[X = 0] \lneqq \frac{4}{9}$ and

$\Pr_{a,b}\left[\bigvee_{j \leq 16i}(aj + b) \in A\right] \geq \Pr_{a,b}\left[\bigvee_{j \leq 16i}(aj + b) \in A_0\right] \geq 1 - \Pr[X = 0] \geq \frac{5}{9}$ ■

We want to use last theorem to increase the success probability obtained in our parameterized weak Valiant-Vazirani's theorem. First we have to introduce some definitions.

Definition 62 *Given two parameterized languages* Γ *and* Γ^**, the language* Γ *is disjunctive reducible to* Γ^* *if and only if there exist an fpt algorithm* M *and two computable functions* f, g *such that, on input* (x,k)*, the algorithm* M *computes a sequence*

$(x_1,k_1),...,\left(x_{f(k)\log(|x|)},k_{f(k)\log(|x|)}\right)$

which satisfies:

1. $(x,k)\in\Gamma\Longleftrightarrow\bigvee_{i\leq f(k)\log(|x|)}(x_i,k_i)\in\Gamma^*$.

2. *For all* $i\leq f(k)\log(|x|)$ *we have that* $k_i\leq g(k)$.

Given $\mathcal{C}$ a parameterized class, we say that $\mathcal{C}$ is $\vee$-*closed* if and only if $\mathcal{C}$ is closed under disjunctive reductions.

Proposition 63 *If* $\mathcal{C}$ *is maj-closed, then* $\mathcal{C}$ *is* $\vee$*-closed.*

Proof. Let L and Σ be two languages, we suppose that $\Sigma\in C$ is a non empty language and that L is disjunctive reducible to Σ. There exist an *fpt* algorithm M and two computable functions f, g such that, on input (x,k), the algorithm M computes a sequence $(x_1,k_1),...,\left(x_{f(k)\log(|x|)},k_{f(k)\log(|x|)}\right)$ that satisfies:

1. $(x,k)\in L\Longleftrightarrow\bigvee_{i\leq f(k)\log(|x|)}(x_i,k_i)\in\Sigma$.

2. For all $i\leq f(k)\log(|x|)$, we have that $k_i\leq g(k)$.

We pick $(x_0,k_0)\in\Sigma$ and we define an algorithm N in the following way

On input (x,k)

1. N computes a sequence $(x_1, k_1), ..., \left(x_{f(k)\log(|x|)}, k_{f(k)\log(|x|)}\right)$ by simulating the computation of M on input (x, k).

2. N computes the sequence $(y_1, s_1), ..., \left(y_{2f(k)\log(|x|)}, s_{2f(k)\log(|x|)}\right)$ defined by

   - If $i \leq f(k)\log(|x|)$, then (y_i, s_i) is equal to (x_i, k_i).
   - If $i \ngeq f(k)\log(|x|)$, then (y_i, s_i) is equal to (x_0, k_0).

It is clear that

1. $(x, k) \in L$ if and only if $\bigotimes_{i \leq 2f(k)\log(|x|)} (y_i, s_i) \in \Sigma$.
2. For all $i \leq 2f(k)\log(|x|)$, we have that $k_i \leq g(k) + k_0$.
3. N is an *fpt* algorithm.

Thus, we have proven that L is majority reducible to Σ. Since C is maj-closed we can claim that $L \in C$ and we can conclude that C is $\vee$-closed ■

From last proposition we obtain, as an easy corollary, the following result.

Corollary 64 *Given $L \in \exists \cdot \mathcal{C}$, if $\mathcal{C}$ is maj-closed, there exist $\Phi^* \in \oplus \cdot \mathcal{C}$ and a computable function f such that*

1. $(x, k) \in L \Rightarrow \Pr_{y \in \{0,1\}^{8f}} [(x, y, k) \in \Phi^*] \geq \frac{5}{9}$.

2. $(x, k) \notin L \Rightarrow \Pr_{y \in \{0,1\}^{8f}} [(x, y, k) \in \Phi^*] = 0$.

Proof. We suppose without loss of generality that $f(k)\log(|x|) \geq 7$. Since $\mathcal{C}$ is *maj*-closed, $\mathcal{C}$ is $\vee$-closed. We use the parameterized weak Valiant-Vazirani's theorem to claim that there exist $\Phi \in \mathcal{C}$ and a computable function f such that

1. For all $(x,k) \in L$

$$\Pr_{z \in \{0,1\}^{4f}} \left[\left| \left\{ y \in \{0,1\}^{f} : (x,y,z,k) \in \Phi \right\} \right| = 1 \right] \geq \frac{1}{16 f(k) \log(|x|)}.$$

2. For all $(x,k) \notin L$

$$\Pr_{z \in \{0,1\}^{4f}} \left[\left| \left\{ y \in \{0,1\}^{f} : (x,y,z,k) \in \Phi \right\} \right| = 1 \right] = 0.$$

Let Φ^* be the language

$$\left\{ (x,y,z_1,z_2,k) : z_1, z_2 \in \{0,1\}^{4f} \ \ \& \ \ \bigvee_{j \leq 64 f(k) \log(|x|)} (x,y,z_1 j + z_2, k) \in \Phi \right\}$$

$\Phi^* \in \mathcal{C}$ since $\mathcal{C}$ is $\vee$-closed. Furthermore it is easy to prove, (using theorem 61), that

1. For all $(x,k) \in L$

$$\Pr_{z_1, z_2 \in \{0,1\}^{4f}} \left[\left| \left\{ y \in \{0,1\}^{f} : (x,y,z_1,z_2,k) \in \Phi^* \right\} \right| = 1 \right] \geq \frac{5}{9}.$$

2. For all $(x,k) \notin L$

$$\Pr_{z_1, z_2 \in \{0,1\}^{4f}} \left[\left| \left\{ y \in \{0,1\}^{f} : (x,y,z_1,z_2,k) \in \Phi^* \right\} \right| = 1 \right] = 0.$$

Remark: In the definition of Φ^* we are considering the elements of the set $\{0,1\}^{4f}$ and the elements of the set $\{i : i \leq 64 f(k) \log(|x|)\}$ as elements of the Galois field $GF\left(2^{4f(k)\log(|x|)}\right)$. The arithmetical operations used in the definition of Φ^* are the arithmetical operations of $GF\left(2^{4f(k)\log(|x|)}\right)$ ■

Now we can state our parameterized theorem of Valiant and Vazirani, at this point the proof is straightforward

Corollary 65 *(The parameterized theorem of Valiant and Vazirani)*

If $\mathcal{C}$ is maj-closed, then

$\exists \cdot \mathcal{C} \subseteq BP \cdot \oplus \cdot \mathcal{C}$.

It is worth noting that, as a byproduct, we have obtained the following theorem.

Theorem 66 *(RP-amplification)*

Suppose that $\mathcal{C}$ is $\vee$-closed and suppose that L is a language for which there exist $\Omega \in \mathcal{C}$ and two computable functions f, g such that

- $(x,k) \in L \Rightarrow \Pr_{y \in \{0,1\}^f}[(x,y,k) \in \Omega] \geq \frac{1}{g(k)\log(|x|)}$.
- $(x,k) \notin L \Rightarrow \Pr_{y \in \{0,1\}^f}[(x,y,k) \in \Omega] = 0$.

Then, $L \in BP \cdot \mathcal{C}$.

3.5 Toda o nada: A parameterized Toda's theorem

In this section we state and prove our parameterized version of Toda's theorem. First we have to introduce a parameterized analogue of the polynomial hierarchy.

Definition 67 *Given $\mathcal{C}$ a parameterized class we define, for each $i \in \mathbb{N}$, a new class $\Sigma_i \cdot \mathcal{C}$.*

1. $\Sigma_0 \cdot \mathcal{C} := \Pi_0 \cdot \mathcal{C} := \mathcal{C}$.
2. $\Sigma_{i+1} \cdot \mathcal{C} := \exists \cdot \Pi_i \cdot \mathcal{C}$.
3. $\Pi_{i+1} \cdot \mathcal{C} := \forall \cdot \Sigma_i \cdot \mathcal{C}$.

Now we are ready to define our parameterized analogue of the polynomial hierarchy, the $PH[P]$ hierarchy.

$PH[P] := \bigcup_{i \in \mathbb{N}} (\Sigma_i \cdot FPT)$

Claim 68 1. *If $co - \mathcal{C} = \mathcal{C}$, then $\Pi_i \cdot \mathcal{C} \subseteq co - \Sigma_i \cdot \mathcal{C}$.*

2. *For all* $i \in \mathbb{N}$, *we have that* $A[i] \subseteq \Sigma_i \cdot FPT$.

The proof of the first item is an standard inductive argument. The second item of the claim is an easy consequence of the machine characterization of the classes $\Sigma_i \cdot FPT$ (see chapter 6). The role and structure of the following lemma is very similar to the role and structure of the inductive argument used by Toda at the end of his proof.

Lemma 69 *If* $\oplus \cdot \mathcal{C}$ *is maj-closed, then for all* $i \geq 1$

$\Sigma_i \cdot \mathcal{C} \subseteq BP \cdot \oplus \cdot \mathcal{C}$

Proof. We use induction on i

1. $\exists \cdot \mathcal{C} \subseteq \exists \cdot BP \cdot \oplus \cdot \mathcal{C}$ (Theorem 65).

2. We suppose that $\Sigma_i \cdot \mathcal{C} \subseteq BP \cdot \oplus \cdot \mathcal{C}$

   $\Sigma_{i+1} \cdot \mathcal{C} = \exists \cdot \Pi_i \cdot \mathcal{C} \subseteq \exists \cdot co - (BP \cdot \oplus \cdot \mathcal{C}) \subseteq \exists \cdot BP \cdot \oplus \cdot \mathcal{C} \subseteq BP \cdot \exists \cdot \oplus \cdot \mathcal{C}$

   $\subseteq BP \cdot (BP \cdot \oplus) \cdot \oplus \cdot \mathcal{C} = BP \cdot \oplus \cdot \mathcal{C}$

■

Corollary 70 *If* $\oplus \cdot FPT$ *is maj-closed, then*

1. *For all* $i \in \mathbb{N}$, *we have that* $\Sigma_i \cdot FPT \subseteq BP \cdot \oplus \cdot FPT$.

2. *For all* $i \in \mathbb{N}$, *we have that* $A[i] \subseteq BP \cdot \oplus \cdot FPT$.

The last step of our proof consists in proving that every parameterized problem in the class $BP \cdot \oplus \cdot FPT$ is random reducible to $p\text{-}\#WSAT(CIRC)$. First we have to introduce our notion of random reducibility.

Definition 71 *Given L and L^* two parameterized languages, L is random reducible to L^*, ($L \preceq_R L^*$), if and only if there exist a randomized fpt algorithm M and two computable functions h, g such that*

1. *On every run of M with input (x,k), the algorithm M computes an instance (x^*,k^*) of L^* such that $k^* \leq h(k)$.*

2. *On every run of M with input (x,k), the number of random guesses is equal to $g(k)\log(|x|)$.*

3. *For all instance (x,k) of L*

- $(x,k) \in L \Rightarrow \Pr[(M(x,k)) \in L^*] \geq \frac{3}{4}$.
- $(x,k) \notin L \Rightarrow \Pr[M(x,k) \in L^*] \leq \frac{1}{4}$.

where the probability is computed on the random guesses of M.

We say that $L \preceq_R p\text{-}\#WSAT(CIRC)$ if there exist a language Ω such that:

1. $L \preceq_R \Omega$.

2. $\Omega \in FPT(p\text{-}\#WSAT(CIRC))$, i.e. there exists a Turing machine $\mathbb{M}$ with oracle's access to $p\text{-}\#WSAT(CIRC)$ and such that $\mathbb{M}$ decides Ω in *fpt* time.

It is very easy to verify that given $L \in BP \cdot \oplus \cdot FPT$, the language L is random reducible to $p\text{-}\#WSAT(CIRC)$. Now we are ready to state the main theorem of the chapter, at this point the proof of the theorem is straightforward.

Theorem 72 *(Parameterized Toda's theorem)*

If $\oplus \cdot FPT$ *is maj-closed, then*

1. *For all* $i \in \mathbb{N}$ *and for all* $L \in \Sigma_i \cdot FPT$, *we have that* $L \preceq_R p\text{-}\#WSAT(CIRC)$.

2. *For all* $i \in \mathbb{N}$ *and for all* $L \in A[i]$, *we have that* $L \preceq_R p\text{-}\#WSAT(CIRC)$.

Open Problem 73 *Is* $\oplus \cdot FPT$ *maj-closed?*

Note that the closure of $\oplus \cdot P$ under majority reductions is a consequence of the Papadimitriou-Zachos's theorem [PZ] which says that $\oplus \cdot P^{\oplus \cdot P}$ is equal to $\oplus \cdot P$. Unfortunately it is very unlikely that the class $\oplus \cdot FPT$ is closed under parameterized Turing reductions (i.e. It is very unlikely that we can prove a parameterized Papadimitriou-Zachos's theorem).

Open Problem 74 *Derandomize the reduction from* $BP \cdot \oplus \cdot FPT$ *into* $\#W[P]$.

Chapter 4

On the parameterized complexity of approximate counting

In this chapter we study the parameterized complexity of approximating some hard parameterized counting problems, specifically we study the parameterized complexity of approximating any problem in the class $\#W[P]$, the parameterized analogue of the class $\#P$. We make such an analysis by comparing the complexity of approximating the counting problems in the class $\#W[P]$ with the parameterized complexity of a large family of parameterized decision problems, the parameterized problems located in the $PH[P]$ hierarchy. This way of analyzing the complexity of approximating hard counting problems resembles Stockmeyer's analysis [S] of the complexity of approximating $\#\mathbf{SAT}$. Actually in this dissertation we have tried to obtain, and we have obtained, a suitable parameterized analogue of Stockmeyer's theorem. While exact counting of problems in $\#W[P]$ is very hard, (is harder than any problem in the $PH[P]$ hierarchy!), approximate counting seems to be not

very hard. In the following sections we will prove that approximate counting belongs to the $PH\,[P]$ hierarchy. The proof is divided in the following stages:

1. Given $\chi \in \#W\,[P]$, we prove that the complexity of approximating χ within any given constant range is equal to the complexity of a parameterized gap problem that we call $p\text{-}approx(\chi)$.

2. Using hashing techniques we prove that for every $\chi \in \#W\,[P]$, the problem $p\text{-}approx(\chi)$ belongs to $BP \cdot \exists \cdot FPT$.

3. We prove a parameterized analogue of the theorem of Lautemann and Sipser. We prove that, if $W\,[P]$ is $\wedge$-closed, then $BP \cdot \exists \cdot FPT$ is included in the second level of the $PH\,[P]$ hierarchy. The proof of this fact is based on Lautemann's proof but in addition we have to use the AKS algorithm to reduce the number of random bits used in the probabilistic arguments in the proof. As a corollary we obtain a parameterized analogue of Stockmeyer's theorem claiming that approximate counting belongs to the second level of the $PH\,[P]$ hierarchy.

4.1 Approximate counting and gap problems

In this section we show that approximate counting of problems in $\#W\,[P]$ can be codified by means of suitable gap decision problems. We begin with some definitions.

Definition 75 *Given a parameterized counting problem χ and given $c \geq 1$, an fpt algorithm M approximates χ within the range c if and only if for all instance (x,k) of χ the algorithm M outputs a computable number $M\,(x,k)$ such that*

$\chi(x,k)\frac{1}{c} \leq M(x,k) \leq \chi(x,k)\,c.$

In this section we will investigate the hardness of approximating, within a constant range, a given problem $\chi \in \#W[P]$. First, we have to state and prove a technical lemma.

Lemma 76 *Given $\chi \in \#W[P]$ and given $c \in \mathbb{N}$, the function $\chi_c : \Sigma^* \times \mathbb{N} \to \mathbb{N}$ defined by*

$\chi_c(x,k) = (\chi(x,k))^c$

belongs to $\#W[P]$.

Proof. If $\chi \in \#W[P]$, we know that there exist a $W[P]$ restricted Turing machine $\mathbb{M}$ and a computable function g such that

1. For every run of $\mathbb{M}$ with input (x,k), the machine $\mathbb{M}$ makes $g(k)\log(|x|)$ nondeterministic guesses.

2. The running time of every run of $\mathbb{M}$, with input (x,k), is bounded by $g(k)\,p(|x|)$.

3. $\chi(x,k) = \#acc(\mathbb{M}(x,k))$.

   Let $\mathbb{M}_c$ be a $W[P]$ restricted Turing machine such that

- For every run of $\mathbb{M}_c$ with input (x,k), the machine $\mathbb{M}_c$ nondeterministically guesses $r_1, ..., r_c \in \{0,1\}^g$.

- For all $i \leq c$, the algorithm $\mathbb{M}_c$ simulates the run of $\mathbb{M}$ on input (x,k), when $\mathbb{M}$ uses the nondeterministic choices codified by r_i.

- $\mathbb{M}_c$ accepts (x,k) if and only if for all $i \leq c$, the simulation of the deterministic computation of $\mathbb{M}$, on input (x, r_i, k), ends in an accepting state.

It is clear that $\chi_c(x,k) = (\#acc(M(x,k)))^c$ ■

Next lemma says that if we can compute approximations within the range 2, we can compute approximations within every constant range $c \gneqq 1$. Because of this, we will only investigate the complexity of computing approximations within the range 2.

Lemma 77 *Given χ a $\#W[P]$ complete problem, (complete under parsimonious reductions), if there exists an algorithm M that approximates χ within the range 2, then for all $c \gneqq 1$, there exists an algorithm M_c that approximates χ within the range c.*

Proof. Let d be a natural number such that $(2)^{\frac{1}{d}} \leq c$. The algorithm M_c is the following one:

On input (x,k)

1. M_c computes a pair (x^*,k^*) such that $\chi(x^*,k^*) = (\chi(x,k))^d$.

2. M_c simulates M on input (x^*,k^*).

3. Given t the output of M on input $\left(x^*,k^*\right)$, the algorithm M_c outputs $(t)^{\frac{1}{d}}$.

The computation in step 1 can be realized in *fpt* time, since M_c only has to compute a pair (x^*,k^*) such that $\chi(x^*,k^*) = \chi_c(x,k)$, remember that $\chi_c \in \#W[P]$ and χ is $\#W[P]$ complete. In step 2, the algorithm M_c computes a number t such that

$\Pr\left[\frac{1}{2}(\chi(x,k))^d \leq t \leq 2(\chi(x,k))^d\right] \geq \frac{2}{3}$.

Therefore

$\frac{2}{3} \leq \Pr\left[\left(\frac{1}{2}\right)^{\frac{1}{d}} \chi(x,k) \leq (t)^{\frac{1}{d}} \leq (2)^{\frac{1}{d}} \chi(x,k)\right] \leq \Pr\left[\frac{1}{c}\chi(x,k) \leq (t)^{\frac{1}{d}} \leq c\chi(x,k)\right]$.

Since $(t)^{\frac{1}{d}}$ is the output of M_c, we can conclude that

$\Pr\left[\frac{1}{c}\chi(x,k) \leq M_c(x,k) \leq c\chi(x,k)\right] \geq \frac{2}{3}$ ■

Remark 78 *Note that, if χ is a $\#W[P]$ complete problem and M is an fpt algorithm which computes approximations to χ within a given constant range, one can use M to compute approximations to any $\#W[P]$ problem, within the given range, in fpt time.*

Now, given $\chi \in \#W[P]$, we will define a parameterized *gap problem* that corresponds in some sense to the problem of approximating χ within the range 2. Let $A = \{n \in \mathbb{N} : n \geq 1\} \cup \left\{\frac{1}{2}\right\}$

Definition 79 *p-approx (χ) is the parameterized gap problem defined by*

Instances: (x, k, c)*, where* (x, k) *is an instance of* χ *and* $c \in A$.

Parameter: k.

Yes-instances: (x, k, c) *is a Yes-instance if and only if* $\chi(x, k) \geq 2c$.

No-instances: (x, k, c) *is a No-instance if and only if* $\chi(x, k) \leq c$.

Remark 80 *If we suppose that for any instance (x, k) of χ*

$\chi(x, k) = \left|\left\{y \in \{0,1\}^h : (x, y, k) \in \Omega\right\}\right|$

where $\Omega \in FPT$ and h is some computable function, we can restrict the definition of p-approx (χ) to the set of triples (x, k, c) that satisfy the inequality $c \leq 2^{h(k)\log(|x|)}$.

Remark 81 *Note that $\left(x, k, \frac{1}{2}\right)$ is a Yes instance of p-approx (χ) if and only if $\chi(x, k) \geq 1$ and $\left(x, k, \frac{1}{2}\right)$ is a No-instance if and only if $\chi(x, k) = 0$.*

Now we prove that we can approximate χ within the range 2, (in *fpt* time), if oracle access to *p-approx*(χ) is provided.

Let (x, k) be a positive instance of χ, i.e. $\chi(x, k) \geq 1$, we can suppose, without loss of generality, that there exist a computable function g and a natural number $m \leq |x|^{g(k)}$ such that

$2^m \leq \chi(x,k) \leq 2^{m+1}$

It follows from the definition of m that

1. $\frac{1}{2}\chi(x,k) \leq 2^m \leq 2(\chi(x,k))$.

2. $\frac{1}{2}\chi(x,k) \leq 2^{m+1} \leq 2\chi(x,k)$.

So, in order to approximate $\chi(x,k)$ within the range 2 it is sufficient to compute a number $n \in \{m, m+1\}$. The proof of the following theorem is based on this fact.

Theorem 82 *Given $\chi \in \#W[P]$, there exists an fpt algorithm M with access to the oracle p-approx(χ) and such that*

1. *M approximates χ within the range 2.*

2. *There exists a computable function g such that M queries the oracle at most $g(k)\log(|x|)$ times.*

Proof. Suppose (x,k) is a positive instance of χ. On input (x,k) the algorithm M computes a number $m \in \mathbb{N}$ such that $2^m \leq \chi(x,k) \leq 2^{m+1}$, after that M outputs 2^m. If (x,k) is a negative instance M outputs 0.

First two easy facts.

- If $i \leq m-1$, then $(x,k,2^i)$ is a Yes-instance of *p-approx*(χ).

- If $i \geq m+1$, then $(x,k,2^i)$ is a No-instance of *p-approx*(χ).

The algorithm M works in the following way:

On input (x,k)

1. For all $i \in \{-1, 1, ..., g(k)\log(|x|)\}$, the algorithm M computes $v_i \in \{0, 1\}$ such that: $v_i = 0$ if the answer to the oracle query $(x, k, 2^i)$ is No, otherwise $v_i = 1$.

2. M computes $m := \min\{i : v_i = 0\}$.

3. If $m = -1$ the algorithm M outputs 0, otherwise M outputs 2^m.

Fact: If (x, k) is a negative instance M outputs 0.

Fact: If (x, k) is a positive instance and M outputs 2^m. Then,

either $2^m \leq \chi(x, k) \leq 2^{m+1}$ or $2^{m-1} \leq \chi(x, k) \leq 2^m$.

This is the case because, given n if $2^n \leq \chi(x, k) \leq 2^{n+1}$ we have that

1. If $i \leq n - 1$, then $v_i = 1$.

2. If $i \geq n + 1$, then $v_i = 0$.

Then, it is clear that $M(x, k) \in \left\{2^m, 2^{m+1}\right\}$. Hence, we have

$\frac{1}{2}\chi(x, k) \leq M(x, k) \leq 2\chi(x, k)$.

Thus, we have proven that the output of M is an approximation of $\chi(x, k)$ within the range 2 ■

Last theorem allows us to identify the problem of computing approximations to χ within the range 2 with the gap problem *p-approx*(χ).

In the following we will analyze the complexity of the parameterized gap problems *p-approx*(χ), with $\chi \in \#W[P]$. We want to prove that for all $\chi \in \#W[P]$, the problem *p-approx*(χ) belongs to some level of the $PH[P]$ hierarchy. To this end we will prove the following facts:

1. For all $\chi \in \#W[P]$, the language $p\text{-}approx(\chi) \in BP \cdot \exists \cdot FPT$.

2. A parameterized analogue of the theorem of Lautemann and Sipser [L].

4.2 Approximate counting belongs to $BP \cdot \exists \cdot FPT$

In this section we prove that $p\text{-}approx(\chi) \in BP \cdot \exists \cdot FPT$. The core of the argument, in the proof of this theorem, is an standard hashing argument. Hashing allow us to transform a dicothomy of the form:

Either there are so many certificates, (more than $2c$*) or there are so few, (less than* c*).*

Into a dicothomy of the form:

The probability that there are at least one certificate is either very high (bigger then $\frac{3}{4}$*) or very small, (less than* $\frac{1}{4}$*).*

Note that this is the type of transformation that we need to prove that $p\text{-}approx(\chi) \in BP \cdot \exists \cdot FPT$. The proof resembles the final part of the proof showing that the problem *non-graphisomorphism* belongs to $BP \cdot \exists \cdot P$.

Remark 83 *Remember that* $BP \cdot \exists \cdot FPT = BP \cdot W[P]$.

We begin remembering some things about U_2-hashing families. Suppose that A, B are two sets and suppose that $H_{A,B} \subseteq B^A$ is an U_2-hashing family. Given $\alpha \in B$ and $S \subseteq A$, we consider the random variable $Y_{S,\alpha}$, with domain $H_{A,B}$, defined by

$Y_{S,\alpha}(h) = \left|S \cap h^{-1}(\alpha)\right|$.

Lemma 84 $E[Y_{S,\alpha}]$*, the expected value of* $Y_{S,\alpha}$*, is equal to* $\frac{|S|}{|B|}$.

Proof. Given $a \in S$, we consider the indicator variable Y_a, with domain $H_{A,B}$, defined by $Y_a(h) = 1$ if and only if $h(a) = \alpha$.

It is clear that $Y_{S,\alpha} = \sum_{a \in S} Y_a$ and $E[Y_{S,\alpha}] = \sum_{a \in S} E[Y_a]$. So, we only have to compute $E[Y_a]$ for each $a \in S$. Fixing $a \in S$, we have that $\Pr_h[h(a) = \alpha] = \frac{1}{|B|}$, (since $H_{A,B}$ is a U_2-hashing family). Hence, $E[Y_a] = \frac{1}{|B|}$ and $E[Y_{S,\alpha}] = \frac{|S|}{|B|}$ ■

Lemma 85 *(Letfover hashing lemma)*

Given $A, B, S, \alpha, H_{A,B}$ and $Y_{S,\alpha}$ as above, we have that

$\Pr_h\left[\left|Y_{S,\alpha} - \frac{|S|}{|B|}\right| \geqq \epsilon \frac{|S|}{|B|}\right] \leqq \frac{|B|}{\epsilon^2 |S|}$.

Proof. We note that

1. The indicator variables $(Y_a)_{a \in S}$ are pairwise independent.

2. For all $a \in S$ we have that

   $\sigma^2_{Y_a} = E\left[Y_a^2\right] - E[Y_a]^2 = E[Y_a](1 - E[Y_a]) \leq E[Y_a] = \frac{1}{|B|}$.

3. $\sigma^2_{Y_{S,\alpha}} = \sum_{a \in S} \sigma^2_{Y_a} \leq \frac{|S|}{|B|}$.

We can use Chebyshev's inequality and facts 1-3 to claim that

$\Pr_h\left[\left|Y_{S,\alpha} - \frac{|S|}{|B|}\right| \geqq \epsilon \frac{|S|}{|B|}\right] \leqq \frac{\sigma^2_{Y_{S,\alpha}}}{\left(\epsilon \frac{|S|}{|B|}\right)^2} \leq \frac{|B|}{\epsilon^2 |S|}$ ■

Remark 86 *Remember that $H^*_{n,m}$ is the U_2-hashing family*

$\{h_{a,b} : a, b \in \{0,1\}^n \ \& \ h_{a,b}(z) := (az + b) \restriction_m\}$.

Note that if we fix $S \subseteq \{0,1\}^n$, and we consider the random variable $Y_{S,0^m}$ with domain $H^*_{n,m}$, we have that $E[Y_{S,0^m}] = \frac{|S|}{2^m}$.

Let us begin with the proof. First we have to define the meaning of a gap problem belonging to $BP \cdot \exists \cdot FPT$.

We will say that $p\text{-}approx(\chi) \in BP \cdot \exists \cdot FPT$ if and only if there exist $\Omega \in FPT$ and two computable functions h, g such that:

- If (x,k,c) is a Yes-instance of $p\text{-}approx\,(\chi)$, then

  $\Pr_{z\in\{0,1\}^g}\left[\exists y \in \{0,1\}^h ((x,y,z,c,k) \in \Omega)\right] \geq \frac{3}{4}$

- If (x,k,c) is a No-instance of $p\text{-}approx(\chi)$, then

  $\Pr_{z\in\{0,1\}^g}\left[\exists y \in \{0,1\}^h ((x,y,z,c,k) \in \Omega)\right] \leq \frac{1}{4}$

Now we prove that given $\chi \in \#W\,[P]$, the parameterized gap problem $p\text{-}approx\,(\chi)$ belongs to $BP \cdot \exists \cdot FPT$. The proof relies on the leftover hashing lemma (lemma 85).

Given χ a problem in $\#W\,[P]$, there exists $\Omega \in FPT$ such that $\chi(x,k) = |S_{x,k}|$, where $S_{x,k}$ is the set $\left\{y \in \{0,1\}^h : (x,y,k) \in \Omega\right\}$. We consider the language $\Omega^6 \in FPT$ defined by

$\Omega^6 := \left\{(x,y_1,...,y_6,k) : y_1,...,y_6 \in \{0,1\}^h \ \&\ (x,y_1,k) \in \Omega, ..., (x,y_6,k) \in \Omega\right\}$.

Let $S^6_{x,k} = \left\{(y_1,...,y_6) : (x,y_1,...,y_6,k) \in \Omega^6\right\} = (S_{x,k})^6$.

Claim 87 *Given (x,k,c) an instance of p-approx (χ)*

1. *If (x,k,c) is a Yes-instance of p-approx(χ), then $|S^6_{x,k}| \geq 2^6 c^6$.*

2. *If (x,k,c) is a No-instance of p-approx(χ), then $|S^6_{x,k}| \leq c^6$.*

Lemma 88 *If (x,k,c) is a Yes-instance of p-approx(χ), then*

$\Pr_{r\in H^*_{n,m}}\left[\exists y \in \{0,1\}^{6\cdot h}\left(y \in S^6_{x,k}\ \&\ r\,(y) = 0^m\right)\right] \geq \frac{3}{4}$.

Proof. Given (x,k,c) a Yes-instance of *p-approx*(χ), we set $n = 6h(k)\log(|x|)$ and $m = \log\left(4c^6\right)$, (we can suppose, without loss of generality, that $m \leq n$). Let Y_m be the random variable, with domain $H^*_{n,m}$, defined by

$Y_m(r) := \left|S^6_{x,k} \cap r^{-1}(0^m)\right|$

That is, we are setting $A = \{0,1\}^n$, $B = \{0,1\}^m$, $H_{A,B} = H^*_{n,m}$, $S = S^6_{x,k}$, $\alpha = 0^m$ and $Y_{S,\alpha} = Y_m$. For ρ_m, the expected value of Y_m, we have that $\rho_m \geq 16$. Now, if we use the leftover hashing lemma, choosing $\epsilon = \frac{1}{2}$, we obtain

$\Pr_{r\in H^*_{n,m}}[Y_m(r) = 0] \leq \Pr_{r\in H^*_{n,m}}\left[|Y_m(r) - \rho_m| \gneqq \frac{1}{2}\rho_m\right] \leq \frac{1}{4}$.

Hence, we have

$\Pr_{r\in H^*_{n,m}}\left[\exists y \in \{0,1\}^{6\cdot h}\left(y \in S^6_{x,k}\ \&\ r(y) = 0^m\right)\right] \geq 1 - \frac{1}{4} = \frac{3}{4}$ ■

Lemma 89 *If (x,k,c) is a Not-instance of p-approx(χ), then*

$\Pr_{r\in H^*_{n,m}}\left[\exists y \in \{0,1\}^{6\cdot h}\left(y \in S^6_{x,k}\ \&\ r(y) = 0^m\right)\right] \leq \frac{1}{4}$.

Proof. Given (x,k,c) a No-instance of *p-approx*(χ), we set again $n = 6h(k)\log(|x|)$ and $m = \log\left(4c^6\right)$. First we note that for all $y \in \{0,1\}^{6\cdot h}$

$\Pr_{r\in H^*_{n,m}}[r(y) = 0^m] \leq 2^{-m}$

since $H^*_{n,m}$ is a U_2-hashing family. Therefore

$\Pr_{r\in H^*_{n,m}}\left[\exists y \in S^6_{x,k}\ (\ r(y) = 0^m)\right] \leq \sum_{y\in S^6_{x,k}} \Pr_{r\in H^*_{n,m}}[r(y) = 0^m]$

$\leq \left|S^6_{x,k}\right| 2^{-m} \leq c^6 2^{-m} \leq \frac{1}{4}$ ■

Theorem 90 *Given $\chi \in \#W[P]$, the gap language p-approx(χ) belongs to $BP\cdot\exists\cdot FPT$*

Proof. We consider the language Ω^* defined by

$\Omega^* := \{(x, y_1, ..., y_6, c, r, k) : \varphi\ \&\ \psi_c\}$

where

1. $\varphi := y_1, ..., y_6 \in \{0,1\}^h$ & $(x, y_1, k), ..., (x, y_6, k) \in \Omega$.

2. $\psi_c := r \in H^*_{n,m}$ & $r(y_1, ..., y_6) = 0^i$ & $m = \log\left(4c^6\right)$ & $n = 6h(k)\log(|x|)$.

3. h is a computable function.

4. $\Omega \in FPT$ is a language such that for every instance (x, k) of χ we have

   $$\chi(x,k) = \left|\left\{y \in \{0,1\}^h : (x,y,k) \in \Omega\right\}\right|.$$

Note that $\Omega^* \in FPT$. From last two lemmas we have that

- If (x, k, c) is a Yes-instance of *p-approx*(χ), then

  $$\Pr_{r \in H^*_{n,m}}\left[\exists y_1, ..., y_6 \in \{0,1\}^h ((x, y_1, ..., y_6, c, r, k,) \in \Omega^*)\right] \geq \tfrac{3}{4}.$$

- If (x, k, c) is a No-instance of *p-approx*(χ), then

  $$\Pr_{r \in H^*_{n,m}}\left[\exists y_1, ..., y_6 \in \{0,1\}^h ((x, y_1, ..., y_6, c, r, k,) \in \Omega^*)\right] \leq \tfrac{1}{4}.$$

So, we have proven that *p-approx*(χ) belongs to $BP \cdot \exists \cdot FPT$ ■

In order to prove that approximate counting belongs to the $PH[P]$ hierarchy, we are trying to prove that for all $\chi \in \#W[P]$, the problem *p-approx*(χ) belongs to $PH[P]$. Up to the moment we know that for every $\chi \in \#W[P]$, the gap language *p-approx*(χ) belongs to $BP \cdot \exists \cdot FPT$. In next subsection we will prove that, under certain complexity theoretic hypothesis, the parameterized class $BP \cdot \exists \cdot FPT$ is included in $\forall \cdot \exists \cdot FPT$.

4.3 The theorem of Lautemann and Sipser

In this subsection we prove a parameterized analogue of the theorem of Lautemann and Sipser [L]. Specifically we prove that:

1. $BP \cdot FPT \subseteq \exists \cdot \forall \cdot FPT \cap \forall \cdot \exists \cdot FPT$
2. If $W[P]$ is $\wedge$-closed, then $BP \cdot \exists \cdot FPT$ is included in $\forall \cdot \exists \cdot FPT$

As a corollary we obtain that, if $W[P]$ is $\wedge$-closed, approximate counting of problems in $\#W[P]$ belongs to $\forall \cdot \exists \cdot FPT$, i.e. parameterized approximate counting belongs to the second level of the $PH[P]$ hierarchy.

Remark 91 *Remember that $\exists \cdot FPT$ is equal to $W[P]$, (and $BP \cdot \exists \cdot FPT = BP \cdot W[P]$).*

Our proof is very close to Lautemann's proof, though we have to use the AKS algorithm to save random bits and we have to take into account some technical details.

4.3.1 The general case

Let $\mathcal{C}$ be a parameterized normal class such that $\mathcal{C}$ has the *pam* property. Along this subsection we will fix $L \in BP \cdot \mathcal{C}$, $\Omega \in \mathcal{C}$ and a computable function f such that

- $(x,k) \in L \Longrightarrow |S_{x,k}| \geq 2^{f(k)\log(|x|)}\left(1 - 2^{-k\log(|x|)}\right)$.
- $(x,k) \notin L \Longrightarrow |S_{x,k}| \leq 2^{f(k)\log(|x|)}2^{-k\log(|x|)}$.

where $S_{x,k} := \left\{ y \in \{0,1\}^f : (x,y,k) \in \Omega \right\}$.

Let AKS be the algorithm used in the AKS theorem, (theorem 34), and let N_1, N_2 be natural numbers such that, if $(x,k) \in L$, then

$\Pr_{v\in\{0,1\}^{N_1 f}}\left[\left\{v_1,...,v_{N_2 f(k)\log(|x|)}\right\}\cap S_{x,k}\neq\varnothing\right]\geq 1-2^{-2f(k)\log(|x|)}$.

Notation 92 *In the following if* $v\in\{0,1\}^{N_1 f}$*, the symbol* $AKS(v)$ *will denote the set* $\left\{v_1,...,v_{N_2 f(k)\log(|x|)}\right\}$*, where* $v_1,...,v_{N_2 f(k)\log(|x|)}$ *is the output sequence of* AKS *on input* $(v,2f(k)\log(|x|),f(k)\log(|x|))$.

Definition 93 *Given* $S\subseteq\{0,1\}^m$ *and* $v\in\{0,1\}^m$, $S+v:=\{s+v:s\in S\}$, *where* $+$ *is the addition operation in the vector space* $GF(2)^m$.

The next two lemmas are our parameterized version of the core of Lautemann's probabilistic argument.

Lemma 94 *Given* $S\subseteq\{0,1\}^f$*, if* $|S|\geq 2^{f(k)\log(|x|)}\left(1-2^{-k\log(|x|)}\right)$ *we have that* $\exists v\in\{0,1\}^{N_1\cdot f}\,\forall z\in\{0,1\}^f((AKS(v)+z)\cap S\neq\emptyset)$.

Proof. We prove that

$Pr_{v\in\{0,1\}^{N_1 f}}[\forall z\in\{0,1\}^f\,((AKS(v)+z)\cap S\neq\emptyset)]>0$.

If we fix $z\in\{0,1\}^f$, we have

$Pr_{v\in\{0,1\}^{N_1 f}}[AKS(v)+z\subseteq(S)^c]\leq 2^{-2f(k)\log(|x|)}$.

Therefore

$Pr_{v\in\{0,1\}^{N_1 f}}[\exists z\in\{0,1\}^f\,(AKS(v)+z\subseteq(S)^c)]$

$\leq\sum_{z\in\{0,1\}^f}Pr_{v\in\{0,1\}^{N_1 f}}[AKS(v)+z\subseteq(S)^c]\leq 2^{f(k)\log(|x|)}2^{-2f(k)\log(|x|)}\lneqq 1$.

Thus, we have

$Pr_{v\in\{0,1\}^{N_1 f}}[\forall z\in\{0,1\}^f\,(AKS(v)+z\not\subseteq(S)^c)]\gneqq 0$.

So, we have proven that

$\exists v\in\{0,1\}^{N_1\cdot f}\,\forall z\in\{0,1\}^f((AKS(v)+z)\cap S\neq\emptyset)$ ■

Claim 95 *For all $k \in \mathbb{N}$, there exists $N_k \in \mathbb{N}$ such that if $|x| \geq N_k$, then*

$2^{k \log(|x|)} \gneqq N_2 f(k) \log(|x|)$.

Lemma 96 *Given $S \subseteq \{0,1\}^f$ such that $|S| \geq 2^{f(k)\log(|x|)}\left(1 - 2^{-k\log(|x|)}\right)$ we have that for all $H \subseteq \{0,1\}^f$ if $|H| \leq N_2 f(k)\log(|x|) \lneqq 2^{k\log(|x|)}$, then there exists $z \in \{0,1\}^f$ such that $H + z \subseteq S$.*

Proof. Suppose that for all $z \in \{0,1\}^f$, we have that $H + z \nsubseteq S$, i.e. for all $z \in \{0,1\}^f$, $H \nsubseteq S + z$. Then, for all $z \in \{0,1\}^f$, there exists $s_z \in S$ such that $s_z \notin S + z$. It implies that there exists $s \in H$ such that $s \notin S + z$ for at least $\frac{2^{f(k)\log(|x|)}}{N_2 f(k)\log(|x|)}$ of the z's, i.e. there exists $s \in H$ such that $|\{u : s + u \notin S\}| \geq \frac{2^{f(k)\log(|x|)}}{N_2 f(k)\log(|x|)}$. Fix such an s and let $H_s = \{u : s + u \notin S\}$, we have that

- $H_s + s \subseteq (S)^c$.
- $|H_s + s| \geq \frac{2^{f(k)\log(|x|)}}{N_2 f(k)\log(|x|)}$.

Hence, we have that

$|(S)^c| \geq |H_s + s| \geq \frac{2^{f(k)\log(|x|)}}{N_2 f(k)\log(|x|)}$.

But this is a contradiction, since

$|(S)^c| \leq \frac{2^{f(k)\log(|x|)}}{2^{k\log(|x|)}}$ and $N_2 f(k)\log(|x|) \lneqq 2^{k\log(|x|)}$ ■

Corollary 97 *Given (x,k) an instance of L we have that*

1. *If $(x,k) \in L$, then*

   $\exists v \in \{0,1\}^{N_1 \cdot f}\, \forall z \in \{0,1\}^f((AKS(v) + z) \cap S_{x,k} \neq \emptyset)$.

2. *If* $(x,k) \in L$ *and* $|x| \geq N_k$, *then*

   for all $H \subseteq \{0,1\}^f$ *such that* $|H| \leq N_2 f(k) \log(|x|)$, *there exists* $z \in \{0,1\}^f$ *such that* $H + z \subseteq S_{x,k}$.

3. *If* $(x,k) \notin L$, *then*

   $\exists v \in \{0,1\}^{N_1 \cdot f} \, \forall z \in \{0,1\}^f ((AKS(v) + z) \cap (S_{x,k})^c \neq \emptyset)$.

4. *If* $(x,k) \notin L$ *and* $|x| \geq N_k$, *then*

   for all $H \subseteq \{0,1\}^f$ *such that* $|H| \leq N_2 f(k) \log(|x|)$, *there exists* $z \in \{0,1\}^f$ *such that* $H + z \subseteq (S_{x,k})^c$.

Notation 98 *Given* $v \in \{0,1\}^{N_1 f}$ *and* $v_1, ..., v_{N_2 f(k) \log(|x|)}$ *the output sequence of* AKS *on input* $(v, 2f(k)\log(|x|), f(k)\log(|x|))$, *the symbol* $AKS(v,i)$ *denotes* v_i, *the* i*th element of the output sequence.*

Let Ω_1, Ω_2 be the following pair of parameterized languages

$\Omega_1 := \{(x,v,z,k) : v \in \{0,1\}^{N_1 f} \ \& \ z \in \{0,1\}^f \ \& \ \bigwedge_{i \leq N_2 f(k) \log(|x|)} (x, AKS(v,i) + z, k) \in \Omega\}$.

$\Omega_2 := \{(x,v,z,k) : v \in \{0,1\}^{N_1 f} \ \& \ z \in \{0,1\}^f \ \& \ \bigvee_{i \leq N_2 f(k) \log(|x|)} (x, AKS(v,i) + z, k) \in \Omega\}$.

From last corollary we obtain the following result.

Corollary 99 *If* $(x,k) \in L$ *and* $|x| \geq N_k$. *Then*

1. $\forall v \in \{0,1\}^{N_1 f} \exists \ z \in \{0,1\}^f ((x,v,z,k) \in \Omega_1)$

2. $\exists v \in \{0,1\}^{N_1 f} \forall \ z \in \{0,1\}^f ((x,v,z,k) \in \Omega_2)$

We have almost obtained representations of L as $\forall \cdot \exists \cdot \mathcal{C}$ and $\exists \cdot \forall \cdot \mathcal{C}$ languages. It remains to be verified that the languages Ω_1 and Ω_2 are elements of the class $\mathcal{C}$. Unfortunately, to prove this fact, we have to suppose that $\mathcal{C}$ is closed under some specific type of reductions.

Definition 100 *Σ is conjunctive reducible to Σ^* if and only if there exist an fpt algorithm M and two computable functions h, g such that, on input (x, k), the algorithm M computes a sequence $(x_1, k_1), ..., \left(x_{h(k)\log(|x|)}, k_{h(k)\log(|x|)}\right)$ which satisfies*

1. $(x, k) \in \Sigma \Leftrightarrow \bigwedge_{i \leq h(k)\log(|x|)} (x_i, k_i) \in \Sigma^*$.
2. *For all $i \leq h(k)log(|x|)$, we have that $k_i \leq g(k)$.*

Definition 101 *Σ is disjunctive reducible to Σ^* if and only if there exist an fpt algorithm M and two computable functions h, g such that, on input (x, k), the algorithm M computes a sequence $(x_1, k_1), ..., \left(x_{h(k)\log(|x|)}, k_{h(k)\log(|x|)}\right)$ which satisfies*

1. $(x, k) \in \Sigma \Leftrightarrow \bigvee_{i \leq h(k)\log(|x|)} (x_i, k_i) \in \Sigma^*$.
2. *For all $i \leq h(k)log(|x|)$, we have that $k_i \leq g(k)$.*

We will say that $\mathcal{C}$ is $\wedge$-*closed* if and only if for all Σ, if there exists $\Sigma^* \in \mathcal{C}$ such that Σ is conjunctive reducible to Σ^*, then $\Sigma \in \mathcal{C}$. We define $\vee$-closed analogously.

Claim 102 *1. If $\mathcal{C}$ is $\wedge$-closed, then $\Omega_1 \in \mathcal{C}$.*

2. If $\mathcal{C}$ is $\vee$-closed, then $\Omega_2 \in \mathcal{C}$.

Let g be a computable function such that for all $k \in \mathbb{N}$, we have that $g(k) \geq N_k$. We consider the parameterized languages:

1. $L_{\geq} := \{(x,k) : |x| \geq g(k) \ \& \ (x,k) \in L\}$.

2. $L_{\leq} := \{(x,k) : |x| \leq g(k) \ \& \ (x,k) \in L\}$.

Claim 103 $L_{\leq} \in FPT$.

Proof. First, we note that the function $h : \mathbb{N} \to \mathbb{N}$ defined by

$h(k) := \min\left\{m \in \mathbb{N} : 2^{k\log(m)} \geq N_2 f(k)\log(m) \ \right\}$

is a computable function. Furthermore, there exists a computable function $g : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ such that any instance (x,k) of L can be decided in time $g(|x|,k)$. To finish with the proof it is sufficient to observe that any instance (x,k) of $L_{\leq}$ can be decided in time $p(|x|,k) + g(h(k),k)$, where $p(|x|,k)$ is the time required to decide if $|x| \leq h(k)$. Hence, we can suppose that p is a polynomial ■

Theorem 104 *If $\mathcal{C}$ is $\vee$-closed, $\wedge$-closed and $\mathcal{C}$ has the pam property, then $L_{\geq} \in \forall\cdot\exists\cdot\mathcal{C} \cap \exists\cdot\forall\cdot\mathcal{C}$.*

Proof. We prove that $L_{\geq} \in \exists\cdot\forall\cdot\mathcal{C}$, the proof of $L_{\geq} \in \forall\cdot\exists\cdot\mathcal{C}$ is very similar.

We know that

$(x,k) \in L_{\geq} \Rightarrow \exists v \in \{0,1\}^{N_1 f}\forall z \in \{0,1\}^{f}((x,v,z,k) \in \Omega_2)$. (corollary 97)

If $(x,k) \notin L_{\geq}$ we have that

$\forall v \in \{0,1\}^{N_1 f}\exists z \in \{0,1\}^{f}(AKS(v) + z \subseteq (S_{x,k})^c)$. (corollary 97)

Then, $(x,k) \notin L_{\geq}$ implies

$\neg\exists v \in \{0,1\}^{N_1 f}\forall z \in \{0,1\}^{f}((x,v,z,k) \in \Omega_2)$.

Thus, we have

$(x,k) \in L_{\geq} \Leftrightarrow \exists v \in \{0,1\}^{N_1 f}\forall z \in \{0,1\}^{f}((x,v,z,k) \in \Omega_2)$.

So, we can conclude that $L_{\geq} \in \exists \cdot \forall \cdot \mathcal{C}$, (since $\Omega_2 \in \mathcal{C}$) ■

Corollary 105 *(Parameterized abstract Lautemann-Sipser's theorem)*

1. *If $\mathcal{C}$ is $\wedge$-closed and $\mathcal{C}$ has the pam property, then $L \in \forall \cdot \exists \cdot \mathcal{C}$.*

2. *If $\mathcal{C}$ is $\vee$-closed and $\mathcal{C}$ has the pam property, then $L \in \exists \cdot \forall \cdot \mathcal{C}$.*

3. *If $\mathcal{C}$ is $\wedge$-closed and $\mathcal{C}$ has the pam property, then $BP \cdot \mathcal{C} \subseteq \forall \cdot \exists \cdot \mathcal{C}$.*

4. *If $\mathcal{C}$ is $\vee$-closed and $\mathcal{C}$ has the pam property, then $BP \cdot \mathcal{C} \subseteq \exists \cdot \forall \cdot \mathcal{C}$.*

4.3.2 Some specific cases: $BP \cdot \exists \cdot FPT \subseteq \forall \cdot \exists \cdot FPT$

In this subsection we consider the cases $\mathcal{C} = FPT$ and $\mathcal{C} = W[P]$.

Claim 106 *FPT is $\wedge$-closed, $\vee$-closed and FPT has the pam property.*

From the claim we obtain, as a corollary, our first parameterized analogue of the theorem of Lautemann and Sipser.

Corollary 107 *(Parameterized analogue of the theorem of Lautemann and Sipser)*

$BP \cdot FPT \subseteq \forall \cdot \exists \cdot FPT \cap \exists \cdot \forall \cdot FPT$.

Now we consider the case $\mathcal{C} = W[P]$.

Proposition 108 *$W[P]$ is $\vee$-closed.*

Proof. Given $L \in W[P]$ there exist a $W[P]$ restricted Turing machine $\mathbb{M}$ and a computable function f such that for all x, k

1. $(x,k) \in L$ if and only if $\mathbb{M}$ accepts (x,k).

2. For every run of $\mathbb{M}$, on input (x,k), the machine $\mathbb{M}$ makes $f(k)log(|x|)$ nondeterministic guesses at the beginning of the computation.

Let L^* be a language such that L^* is disjunctive reducible to L and let M, g, h be the algorithm and the functions in the definition of disjunctive reduction. We will define a $W[P]$ restricted Turing machine $\mathbb{M}^*$ that decides the language L^*. $\mathbb{M}^*$ is the following machine

On input (x,k)

1. $\mathbb{M}^*$ guesses $i \in \{1, ..., h(k)log(|x|)\}$.

2. $\mathbb{M}^*$ computes (x_i, k_i) the ith element of the output sequence of M on input (x,k).

3. $\mathbb{M}^*$ guesses $y \in \{0,1\}^{f(k_i)\log(|x_i|)}$.

4. $\mathbb{M}^*$ simulates the deterministic part of the computation of $\mathbb{M}$, on input (x_i, k_i), when $\mathbb{M}$ uses the nondeterministic guesses codified by y.

It is clear that $(x,k) \in L^*$ if and only if $\mathbb{M}^*$ accepts (x,k) ■

Proposition 109 *If $W[P]$ is $\wedge$-closed, then $W[P]$ is maj-closed.*

Proof. Let L be a language in $W[P]$ and let Σ be a language which is maj-reducible to L. There exist an *fpt* algorithm M and two computable functions f and g such that for all (x,k)

1. On input (x,k), the algorithm M computes a sequence

$$(x_1, k_1), ..., \left(x_{f(k)\log(|x|)}, k_{f(k)\log(|x|)}\right).$$

2. $(x,k) \in \Sigma \Leftrightarrow \bigotimes_{i \leq f(k)\log(|x|)} (x_i, k_i) \in L.$

3. For all $i \leq f(k)\log(|x|)$, we have that $k_i \leq g(k)$.

Let $\mathbb{M}$ be the following nondeterministic $W[P]$ restricted Turing machine :

On input (x,k)

1. $\mathbb{M}$ simulates the computation of M on input (x,k).

2. Given $(x_1,k_1), ..., \left(x_{f(k)\log(|x|)}, k_{f(k)\log(|x|)}\right)$, the output of M on input (x,k), the machine $\mathbb{M}$ guesses $v \in \{0,1\}^{f(k)\log(|x|)}$ such that the Hamming weight of v is larger than $\frac{f(k)\log(|x|)}{2}$.

3. $\mathbb{M}$ computes the sequence $\{(x_i,k_i) : v_i = 1\}$.

4. $\mathbb{M}$ verifies that $\bigwedge_{i \leq f(k)\log(|x|):v_i=1} (x_i,k_i) \in L$.

We are allowed to suppose that $\mathbb{M}$ is a nondeterministic $W[P]$ restricted Turing machine, since we are supposing that $W[P]$ is $\wedge$-closed, and it implies that the verification in step four can be carried out for a machine of this type. Furthermore, it is easy to verify that $(x,k) \in \Sigma$ if and only if $\mathbb{M}$ accepts (x,k). Then, we have that $\Sigma \in W[P]$. Thus, we have proven that $W[P]$ is *maj*-closed ■

Corollary 110 *If $W[P]$ is $\wedge$-closed, then $W[P]$ has the pam property.*

Furthermore we have obtained as a corollary the following parameterized analogue of the theorem of Lautemann and Sipser.

Corollary 111 *(Parameterized strong Lautemann-Sipser theorem)*

1. *If* $W[P]$ *is* $\wedge$*-closed, then* $BP \cdot \exists \cdot FPT \subseteq \forall \cdot \exists \cdot FPT$.

2. *If* $W[P]$ *has the pam property, then* $BP \cdot \exists \cdot FPT \subseteq \exists \cdot \forall \cdot \exists \cdot FPT$.

Finally we can state our parameterized Stockmeyer's theorem. At this point the proof is straightforward.

Corollary 112 *(Parameterized Stockmeyer's theorem)*

1. *If* $W[P]$ *is* $\wedge$*-closed, then for every* $\chi \in \#W[P]$*, the problem p-approx* $(\chi) \in \forall \cdot \exists \cdot FPT$*, i.e. approximate counting belongs to the second level of the* $PH[P]$ *hierarchy.*

2. *If* $W[P]$ *has the pam property, then for every* $\chi \in \#W[P]$*, the problem p-approx* $(\chi) \in \exists \cdot \forall \cdot \exists \cdot FPT$*, i.e. approximate counting belongs to the third level of the* $PH[P]$ *hierarchy.*

Unfortunately we had to use an additional hypothesis. Can we remove this hypothesis?

Open Problem 113 *Is* $W[P]$ $\wedge$*-closed?*

We say some things concerning this open problem in next chapter.

Chapter 5

The hardness of probability amplification

In this chapter we study the parameterized complexity of probability amplification, specifically we study the parameterized complexity of probability amplification for the parameterized class $BP \cdot \exists \cdot FPT$, the parameterized analogue of the Arthur-Merlin class $BP \cdot \exists \cdot P$. It was shown, in last chapter, that it is possible to prove a parameterized analogue of a theorem of Stockmeyer [S] claiming that probabilistic approximate counting belongs to the polynomial hierarchy, if the class $W[P]$ has the *pam* property. In this chapter we analyze the plausibility of such an Hypothesis.

While probability amplification is an easy task in the classical world, it is not the case in the parameterized world. It is an open problem if we can make amplification of probabilities for most of the parameterized probabilistic classes defined using the parameterized BP operator. Furthermore, in the few positive cases amplification of probabilities requires

sophisticated techniques based on pseudorandom generators.

5.1 The parameterized Arthur-Merlin class and probability amplification

It was proved, in last chapter, that approximate counting belongs to the third level of the $PH\,[P]$ hierarchy, if the class $W\,[P]$ has some strong form of probability amplification, i.e. the *pam* property. We will show that it is very unlikely that the class $W\,[P]$ has the *pam* property.

Remark 114 *Remember that if $W\,[P]$ is $\wedge$-closed, $W\,[P]$ has the pam property. In this chapter we prove a partial converse, we prove that if $W\,[P]$ is not closed under conjunctive reductions, then there exist languages in $BP\cdot\exists\cdot FPT$ which can not be amplified using some special type of algorithms.*

We begin with some definitions. Let N_1, N_2 be two natural numbers.

Definition 115 *A (N_1, N_2)-sampling algorithm is an algorithm M such that for all $n, k \geq 1$ and for every $y \in \{0,1\}^{N_1 k \log(n)}$, on input (y, k), the algorithm M computes a sequence $y_1, ..., y_{N_2 k \log(n)}$ of elements of $\{0,1\}^{k \log(n)}$.*

Remark 116 *The notion of sampling algorithm is a very general notion. The AKS algorithm and the two bit sampling algorithm used in previous chapters are sampling algorithms. From a naive point of view a sampling algorithm corresponds to the first half of typical probability amplification algorithms, which are designed according to the following schema:*

1. *Do sampling.*

2. *Use majority voting, (or use some other type of voting schema), to take a decision.*

We begin considering the following parameterized problem.

Definition 117 *p-CLOGSAT is the following parameterized problem*

- *Instances:* (C,k)*, where* $k \in \mathbb{N}$ *and* C *is boolean circuit whose number of input gates is upperbounded by* $k\log(|C|)$.

- *Parameter:* k.

- *Problem: Decide if* (C,k) *is satisfiable.*

Lemma 118 *p-CLOGSAT is* $W[P]$ *complete.*

Proof. It is straightforward to verify that *p-CLOGSAT* belongs to $W[P]$. We only have to prove that *p-CLOGSAT* is $W[P]$ hard. To this end we will show that *p-WSAT (CIRC)* is *fpt* many one reducible to *p-CLOGSAT*. The proof is an easy application of the $k\log(n)$ trick of Downey and Fellows, ([FG2] page 52).

Let (C,k) be an instance of *p-WSAT (CIRC)* and let m be the number of input gates of C. We can compute in *fpt* time a circuit $D_{C,k}$ which maps $\{0,1\}^{k\log(m)}$ onto

$$\{s \in \{0,1\}^m : \text{the hamming weight of } s \text{ is less than or equal to } k\}$$

If we hardwire the output gates of $D_{C,k}$ and the input gates of C we obtain a circuit $H_{C,k}$ such that

1. $H_{C,k}$ is satisfiable if and only if $(C,k) \in$ *p-WSAT (CIRC)*.

2. The size of $H_{C,k}$ is bigger than the size of C.

3. The number of input gates of $H_{C,k}$ is equal to $k \log(m)$ and $k \log(m) \leq k \log(|H_{C,k}|)$.

Thus, $(H_{C,k}, k)$ is an instance of *p-CLOGSAT* such that

1. $(C, k) \in$ *p-WSAT* $(CIRC)$ if and only if $(H_{C,k}, k) \in$ *p-CLOGSAT*.

2. $(H_{C,k}, k)$ can be computed from (C, k) in *fpt* time.

So, we have proven that *p-WSAT* $(CIRC)$ is *fpt* many one reducible to *p-CLOGSAT* ■

Now, we will define a parameterized gap problem in $BP \cdot \exists \cdot FPT$, the parameterized gap problem *gap-p-CLOGSAT*. We will show that, if $W[P]$ is not closed under some special type of conjunctive reductions, then *gap-p-CLOGSAT* can not be amplified using a sampling algorithm.

Definition 119 *The parameterized gap problem gap-p-CLOGSAT is the following problem*

- *Instances:* $(C(X,Y), k)$, *where* $k \in \mathbb{N}$ *and* $C(X,Y)$ *is a boolean circuit whose input gates are partitioned in two blocks* X, Y *such that* $|X|, |Y| \leq k \log(|C|)$.

- *Parameter:* k.

- *Yes-instances:* $(C(X,Y), k)$ *such that*

  $\Pr_{y \in \{0,1\}^{k \log(|C|)}} [(C(X,y), k) \in$ *p-CLOGSAT*$] \geq \frac{3}{4}$.

- *No-instances:* $(C(X,Y), k)$ *such that*

  $\Pr_{y \in \{0,1\}^{k \log(|C|)}} [(C(X,y), k) \in$ *p-CLOGSAT*$] \leq \frac{1}{4}$.

It is straightforward to verify that $gap\text{-}p\text{-}CLOGSAT \in BP \cdot \exists \cdot FPT$.

We will say that $gap\text{-}p\text{-}CLOPGSAT$ is amplified by means of the (N_1, N_2)-sampling algorithm M, if and only if:

1. For any Yes-instance $(C(X,Y),k)$

$$\Pr_{y\in\{0,1\}^{N_1 k\log(|C|)}}\left[\bigotimes_{i\leq N_2 k\log(|C|)}((C(X,y_i),k)\in p\text{-}CLOGSAT)\right]\geq 1-2^{-\log(|C|)}.$$

2. For any No-instance $(C(X,Y),k)$

$$\Pr_{y\in\{0,1\}^{N_1 k\log(|C|)}}\left[\bigotimes_{i\leq N_2 k\log(|C|)}((C(X,y_i),k)\in p\text{-}CLOGSAT)\right]\leq 2^{-\log(|C|)}.$$

3. The language $\{(C_y(X,Y),k):\varphi\}$ belongs to $W[P]$, where $y\in\{0,1\}^{N_1 k\log(|C|)}$; $(C_y(X,Y),k)$ is equal to $\left\langle (C(X,y_1),k),...,\left(C\left(X,y_{N_2 k\log(|C|)}\right),k\right)\right\rangle$ and φ is equal to:

$$(C(X,Y),k) \text{ is an instance of } gap\text{-}p\text{-}CLOGSAT, \text{ (either a Yes or a No instance)}$$

$$\&$$

$$\bigotimes_{i\leq N_2 k\log(|C|)}(C(X,y_i),k)\in p\text{-}CLOGSAT$$

Item 3 will be very important in the proof of the main theorem in this chapter. Note that items 1 and 2 can be fulfilled using a sampling algorithm like the one of Ajtai, Komlos and Szemeredi. When we try to amplify a language in $BP \cdot W[P]$, (or in $BP \cdot \mathcal{C}$, where $\mathcal{C}$ is some nondeterministic parameterized class), the problems arise during the voting stage, because in this stage we are demanded to certify long sequences, and long sequences seem to require long certificates, (we could say that conditions 1 and 2 do not matter because there exist good pseudorandom generators).

Note that in the definition of sampling algorithm there are not constraints concerning the running time. It is clear that if one wants to use a sampling algorithm to amplify probabilities in *fpt* time, one has to choose, to this end, an *fpt* sampling algorithm. We have not explicitly included, in the definition of sampling algorithm, a running time condition, because the main result of this chapter rules out the possibility of using a sampling algorithm to amplify *gap-p-CLOGSAT*, even if the sampling algorithm is time consuming.

Given D a natural number, we define a last parameterized language $p\text{-}CLOGSAT_D$.

Definition 120 *($p\text{-}CLOGSAT_D$)*

- *Instances:* $\left(C_1, ..., C_{Dk\log(|C_1|)}, k, M\right)$*, where* $k, M \in \mathbb{N}$ *and for all* $i \leq Dk\log(|C_1|)$*, we have that* C_i *is a circuit of size* M *whose number of input gates is less than or equal to* $k\log M$*.*

- *Parameter:* $k \in \mathbb{N}$*.*

- *Problem: Decide if for all* $i \leq Dk\log(M)$ *we have that* $(C_i, k) \in p\text{-}CLOGSAT$*.*

Note that, if $\left(C_1, ..., C_{Dk\log(M)}, k, M\right) \in p\text{-}CLOGSAT_D$, the size of their natural certificates, i.e. sequences $\left(v_1, ..., v_{Dk\log(M)}\right)$ of satisfying assignments, is equal to $Dk^2\log^2(M)$, which is very big because of the term $\log^2(M)$. Remember that, if a parameterized language L belongs to $W[P]$, there exists a computable function f such that for all instance (x, k) of L, if $(x, k) \in L$, the pair (x, k) can be certified using $f(k)\log(|x|)$ nondeterministic bits. Given $D \geq 1$, it is very unlikely that $p\text{-}CLOGSAT_D$ belongs to $W[P]$, since it is very unlikely that there exists a computable function f such that we can certify a random

positive instance $\left(C_1, ..., C_{Dk\log(M)}, k, M\right)$ of $p\text{-}CLOGSAT_D$ using $f(k)\log(M)$ nondeterministic bits.

Remark 121 *Note that if $W[P]$ is $\wedge$-closed, then for all $D \in \mathbb{N}$ the language $p\text{-}CLOGSAT_D$ belongs to $W[P]$*

Remark 122 *Note that given $N, D \geq 1$, the language $p\text{-}CLOGSAT_N \in W[P]$ if and only if $p\text{-}CLOGSAT_D \in W[P]$.*

We will say that $W[P]$ has the *democratic-pam* property, (probability amplification by sampling and majority voting), if and only if every language in $BP \cdot \exists \cdot FPT$ can be amplified by means of some sampling algorithm.

We are ready to prove the major result in this chapter. We will prove that $W[P]$ has the democratic-*pam* property if and only if $W[P]$ is $\wedge$-closed.

Theorem 123 *If gap-p-$CLOGSAT$ can be amplified by means of a sampling-algorithm, then there exist $D \in \mathbb{N}$ such that $p\text{-}CLOGSAT_D \in W[P]$.*

Proof. If we suppose that *gap-p-$CLOGSAT$* can be amplified by means of a sampling-algorithm, we can suppose without loss of generality that there exists $N_1, N_2 \in \mathbb{N}$ such that *gap-p-$CLOGSAT$* is amplified by a $(N_1, 2N_2)$-sampling algorithm. We will prove that $p\text{-}CLOGSAT_{N_2} \in W[P]$, (i.e. we set $D = N_2$).

Let M be such an algorithm and let $S := (C_1, k), ..., \left(C_{N_2 k\log(|C_1|)}, k\right)$ be a sequence, such that for all $i \leq N_2\log(|C_1|)$, we have that C_i is a circuit of size $|C_1|$ whose number of input gates is less than or equal to $k\log(|C_1|)$, that is $\left(C_1, ..., C_{N_2 k\log(|C_1|)}, k, |C_1|\right)$ is an instance of $p\text{-}CLOGSAT_{N_2}$. Given $y \in \{0,1\}^{N_1 k\log(|C_1|)}$ the algorithm M, on input (y, k),

computes a sequence $y_1, ..., y_{2N_2 k \log(|C_1|)}$. From the pair (S, y) we can define a No-instance of *gap-p-CLOGSAT* in the following way:

Let $C_{y,S}(X,Y) = \bigvee_{i \leq 2N_2 k \log(|C_1|)} (G_i(X) \wedge Y = y_i)$

With:

- $G_i(X) := C_i(X)$ if $i \leq N_2 k \log(|C_1|)$.
- $G_i(X) := (X = X)$ if $i = N_2 k \log(|C_1|) + 1$.
- $G_i(X) := (X \neq X)$ if $i \geqq N_2 k \log(|C_1|) + 1$.

It is clear that $|C_{y,S}(X,Y)| \geq |C_1|$. Note that

$|X|, |Y| \leq k \log(|C_1|) \leq k \log(|C_{y,S}(X,Y)|)$

that is, the pair $(C_{y,S}(X,Y), k)$ is an instance of *gap-p-LOGSAT*. We will prove that $(C_{y,S}(X,Y), k)$ is a No-instance

We have that, if $|C_1|^k \geq 8N_2 k \log(|C_1|)$,then

$\Pr_{v \in \{0,1\}^{k \log(|C_1|)}} [(C_{y,S}(X,v), k) \in p\text{-}CLOGSAT] \leq$

$\Pr_{v \in \{0,1\}^{k \log(|C_1|)}} \left[v \in \left\{y_1, ..., y_{2N_2 k \log(|C_1|)}\right\}\right] \leq \frac{1}{4}$.

Thus, if $|C_1|^k \geq 8N_2 k \log(|C_1|)$, the pair $(C_{y,S}(X,Y), k)$ is a No-instance of the language *gap-p-CLOGSAT*.

Now we note that given $y \in \{0,1\}^{N_1 k \log(|C_1|)}$

$\bigotimes_{i \leq 2N_2 k \log(|C_1|)} ((C_{y,S}(X, y_i), k) \in p\text{-}CLOGSAT)$ if and only if
$\bigwedge_{i \leq N_2 k \log(|C_1|)} (C_i, k) \in p\text{-}CLOGSAT$ if and only if
$\left(C_1, ..., C_{N_2 k \log(|C_1|)}, k, |C_1|\right) \in p\text{-}CLOGSAT_{N_2}$.

Remember that, if *gap-p-CLOGSAT* is amplified by M, we can decide if

$$\bigotimes_{i\leq 2N_2k\log(|C_1|)} \left(\left(C_{y,S}\left(X,y_i\right),k\right)\in p\text{-}CLOGSAT\right)$$

using a $W\left[P\right]$ restricted Turing machine. Hence, to verify if

$$\left(C_1,...,C_{N_2k\log(|C_1|)},k,|C_1|\right)\in p\text{-}CLOGSAT_{N_2}$$

we can use a $W\left[P\right]$ restricted Turing machine that verifies if

$$\bigotimes_{i\leq 2N_2k\log(|C_1|)} \left(\left(C_{y,S}\left(X,y_i\right),k\right)\in p\text{-}CLOGSAT\right)$$

and uses this information to give us the correct answer. We can conclude that, if we could amplify *gap-p-CLOGSAT* by means of the sampling-algorithm M, then $p\text{-}CLOGSAT_{N_2}\in W\left[P\right]$ ■

Theorem 124 *$W\left[P\right]$ is $\wedge$-closed if and only if $p\text{-}CLOGSAT_1\in W\left[P\right]$.*

Proof. It is clear that, if $W\left[P\right]$ is $\wedge$-closed, then $p\text{-}CLOGSAT_1\in W\left[P\right]$, since $p\text{-}CLOGSAT_1$ is conjunctive reducible to $p\text{-}CLOGSAT$.

Let L and Σ be two languages such that $L\in W\left[P\right]$ and Σ is conjunctive reducible to L. There exist an *fpt* algorithm M and two computable functions f and g such that

1. For all (x,k), the pair $(x,k)\in\Sigma$ if and only if $\bigwedge_{i\leq f(k)\log(|x|)}(x_i,k_i)\in L$.
2. For all $i\leq f\left(k\right)\log\left(|x|\right)$, we have that $k_i\leq g\left(k\right)$.

Without loss of generality we can suppose that L is equal to $p\text{-}CLOGSAT$, since $p\text{-}CLOGSAT$ is $W\left[P\right]$ complete. It is easy to show, using an standard padding argument, that there exist a computable function h and an *fpt* algorithm M^* such that, on input

$$\left(x_1,k_1\right),...,\left(x_{f(k)\log(|x|)},k_{f(k)\log(|x|)}\right)$$

the algorithm M^* computes a sequence

$$\left(\left(C_1,h\left(k\right)\right),...,\left(C_{h(k)\log(|C_1|)},h\left(k\right)\right)\right)$$

of boolean circuits, which satisfies:

1. $\bigwedge_{i \leq f(k)\log(|x|)} (x_i, k_i) \in p\text{-}CLOGSAT$ if and only if
   $\bigwedge_{i \leq h(k)\log(|C_1|)} (C_i, h(k)) \in p\text{-}CLOGSAT.$
2. For all $i \leq h(k)\log(|C_1|)$, we have that C_i is a circuit of size $|C_1|$ whose number of input gates is upperbounded by $h(k)\log(|C_1|)$.

It is clear that, the composition of the algorithms M and M^* computes an *fpt* many one reduction of Σ into $p\text{-}CLOGSAT_1$. We have that , if $p\text{-}CLOGSAT_1 \in W[P]$, then $\Sigma \in W[P]$. Thus, we have proven that $W[P]$ is $\wedge$-closed ■

Corollary 125 *$W[P]$ has the democratic-pam property if and only if $W[P]$ is $\wedge$-closed.*

Open Problem 126 *Is it true, that $\oplus \cdot FPT$ has the democratic-pam property if and only if $\oplus \cdot FPT$ is $\wedge$-closed?*

5.1.1 A more general framework: Sample and voting

In this chapter we have studied the power of amplification algorithms which are designed according to the following scheme:

1. Do sampling.
2. Do majority voting.

We have proven that these algorithms can not be used to amplify *gap-p-CLOGSAT*, unless $W[P]$ is closed under some special type of unbounded conjunctive reductions. The

argument seems to depend on some specific features of the majority function. In this subsection we show that it is not the case, we show that any amplification algorithm designed according to the following, more general scheme

1. Do sampling

2. Do voting, (according to some prefixed symmetric boolean voting scheme).

can not be used to amplify the language *gap-p-CLOGSAT*.

Suppose that $\mathcal{F} = \{f_n\}_{n\geq 1}$ is a sequence of boolean functions such that for any $n \geq 1$ the domain of f_n is the set $\{0,1\}^n$. An $\mathcal{F}$-amplification algorithm for *gap-p-CLOGSAT* is an algorithm M_A such that, for some (N_1, N_2)-sampling algorithm M, the algorithm M_A works, on input $(C(X,Y),k)$, in the following way

1. M_A chooses $y \in \{0,1\}^{N_1 k \log(|C|)}$.

2. M_A simulates the computation of M on input (y,k).

3. Given $y_1, ..., y_{N_2 k \log(|C|)}$ the output sequence of M, the algorithm M_A computes $v = f_{N_2 k \log(|C|)}\left(z_1^y, ..., z_{N_2 k \log(|C|)}^y\right)$, where for any $i \leq N_2 k \log(|C|)$ we have that $z_i^y = 1$ if and only if $(C(X, y_i), k) \in$ *p-CLOGSAT*.

4. M_A uses v to take a decision.

Note that, the only difference with the amplification algorithms studied before, is that in the third step (the voting stage), we are doing $\mathcal{F}$-voting instead of majority voting.

First at all, we consider the following question: Which are the sequences that we can use to amplify *gap-p-CLOGSAT*?

Consider the following situation. Suppose that M is a sampling algorithm and suppose that $\{\sigma_n\}_{n\geq 1}$ is a sequence of permutations such that, for any $i \geq 1$, we have that $\sigma_i \in \mathcal{S}_i$. Consider the sampling algorithm M^σ defined in the following way, on input (y,k)

1. M^σ simulates the computation of M on input (y,k).

2. Given $y_1, ..., y_{N_2 k \log(|C|)}$ the output sequence of M, the algorithm M^σ outputs

   $y_{\sigma_{N_2 k \log(|C|)}(1)}, ..., y_{\sigma_{N_2 k \log(|C|)}(N_2 k \log(|C|))}$.

It is natural to impose on $\mathcal{F}$, the following condition: The sequence $\mathcal{F}$ can not distinguish the algorithms M and M^σ.

This condition says that, for any $n \geq 1$, the output value of f_n does not change if we permute the inputs, i.e. we are demanding that, for any $n \geq 1$, the function f_n is a symmetric boolean function. In the following, we will restrict our attention to sequences of symmetric-monotone boolean functions. Remember that given $f_n : \{0,1\}^n \to \{0,1\}$ a symmetric-monotone boolean function, there exists $T(f_n) \leq n$ such that

$f_n(X_1, ..., X_n) = 1$ if and only if $|\{i \leq n : X_i = 1\}| \geq T(f_n)$.

The number $T(f_n)$ is called the threshold of f_n. So, given $\mathcal{F} = \{f_n\}_{n\geq 1}$, a sequence of symmetric-monotone functions, we can associate to $\mathcal{F}$ the threshold function $T(\mathcal{F}) : \mathbb{N} \to \mathbb{N}$ defined by $T(\mathcal{F})(n) = T(f_n)$. We consider two cases

1. The function $T(\mathcal{F})$ is computable and unbounded.

2. The function $T(\mathcal{F})$ is bounded.

Note that, the only thing that we have used of the sequence $\{\otimes_n\}_{n\geq 1}$, in the proof of theorem 123, is that the corresponding threshold function $\lceil \frac{n}{2} \rceil + 1$ is computable and

unbounded. It is easy to check that if the function $T(\mathcal{F})$ is computable and unbounded, we can mimic the proof of theorem 123 to show that:

If *gap-p-CLOGSAT* can be amplified using an $\mathcal{F}$-amplification algorithm. Then, $W[P]$ is closed under some special type of unbounded conjunctive reductions, (a possibility that seems to be very unlikely).

Now, we show that if $T(\mathcal{F})$ is bounded, the language *gap-p-CLOGSAT* can not be amplified using an $\mathcal{F}$-amplification algorithm. Our results rule out the possibility of amplifying the language *gap-p-CLOGSAT* using a sample and voting algorithm. Can someone figure out an alternative amplification procedure? We believe that we are providing strong evidence against that $W[P]$ has the *pam* property.

Suppose that for all $n \geq 1$, we have that $T(\mathcal{F})(n) \leq D$. Let M be a (N_1, N_2)-sampling algorithm. We show that we can no amplify *gap-p-CLOGSAT* using the pair $(M, \mathcal{F})$.

Theorem 127 *If $T(\mathcal{F})$ is bounded, the language gap-p-CLOGSAT can not be amplified using $\mathcal{F}$-voting.*

Proof. We suppose that $D \geq 2$, the case $D = 1$ is easy to handle. Suppose that $(C(X), k)$ is a positive instance of $p - CLOGSAT$. Given $\overrightarrow{a} \in \{0,1\}^D$ we define a circuit $G_{\overrightarrow{a}}(X, Y)$ in the following way

$G_{\overrightarrow{a}}(X, Y) := C(X) \wedge F_{\overrightarrow{a}}(Y)$

where $Y = \left\{Y_1, ..., Y_{k\log(|C|)}\right\}$ is a set of input gates such that $X \cap Y = \varnothing$ and $F_{\overrightarrow{a}}(Y)$ is the circuit $\bigwedge_{i \leq D} Y_i = \overrightarrow{a}(i)$. Note that, for any $\overrightarrow{a} \in \{0,1\}^D$ the pair $(G_{\overrightarrow{a}}(X, Y), k)$ satisfies

1. $|X|, |Y| \leq k\log(|G_{\overrightarrow{a}}|)$.

2. $\Pr_{y\in\{0,1\}^{k\log(|C|)}}\left[\left(G_{\overrightarrow{a}}\left(X,y\right),k\right)\in p\text{-}CLOGSAT\right]=$

   $\Pr_{y\in\{0,1\}^{k\log(|C|)}}\left[F_{\overrightarrow{a}}\left(y\right)=1\right]=2^{-D}\leq\frac{1}{4}.$

From 1 and 2 we have that $\left(G_{\overrightarrow{a}}\left(X,Y\right),k\right)$ is a No-instance of *gap-p-CLOGSAT*. Hence, we have

$$\Pr_{y\in\{0,1\}^{N_1k\log(|C|)}}\left[\left|\left\{i\leq N_2k\log\left(|C|\right):F_{\overrightarrow{a}}\left(y_i\right)=1\right\}\right|\geq D\right]\leq 2^{-\log(|C|)}\qquad[eq.1]$$

Note that, for any $y\in\{0,1\}^{N_1k\log(|C|)}$ and for any $i\leq N_2k\log\left(|C|\right)$, we have that $\bigvee_{\overrightarrow{a}\in\{0,1\}^D}F_{\overrightarrow{a}}\left(y_i\right)$. Now, if we suppose that $\log\left(|C|\right)\gneqq D2^D$, we have

$$\Pr_{y\in\{0,1\}^{N_1k\log(|C|)}}\left[\bigvee_{\overrightarrow{a}\in\{0,1\}^D}\left|\left\{i\leq N_2k\log\left(|C|\right):F_{\overrightarrow{a}}\left(y_i\right)=1\right\}\right|\geq D\right]=1$$

But, from $[eq.1]$ we have

$$\begin{aligned}\Pr_{y\in\{0,1\}^{N_1k\log(|C|)}}\left[\bigvee_{\overrightarrow{a}\in\{0,1\}^D}\left|\left\{i\leq N_2k\log\left(|C|\right):F_{\overrightarrow{a}}\left(y_i\right)=1\right\}\right|\geq D\right] &\leq\\ \sum_{\overrightarrow{a}\in\{0,1\}^D}\Pr_{y\in\{0,1\}^{N_1k\log(|C|)}}\left[\left|\left\{i\leq N_2k\log\left(|C|\right):F_{\overrightarrow{a}}\left(y_i\right)=1\right\}\right|\geq D\right] &\leq\\ 2^D2^{-\log(|C|)} &\lneqq \frac{2^D}{D2^D}\\ &\lneqq 1\end{aligned}$$

■

Which is the meaning of our results? We have proven that, if one wants to amplify probabilities, of languages in $BP\cdot W\left[P\right]$, using a sampling and voting algorithm, there are two possibilities.

1. One can choose as sampling algorithm a good pseudorandom generator. In this case,

if $W[P]$ is not $\wedge$-closed, one will be unable to fulfill condition 3 in our definition of probability amplification by means of a sampling algorithm.

2. One can choose a sampling algorithm, which only outputs unlikely sequences that can be certified using few bits. In this case, one will be unable to cheat *gap-p-CLOGSAT*, i.e. one will be unable to amplify *gap-p-CLOGSAT*.

We finish this chapter with a positive result concerning amplification of probabilities for languages in $BP \cdot W[P]$.

Definition 128 *(RP-amplification)*

Suppose that $\mathcal{C}$ is a parameterized class. If for any L for which there exist $\Omega \in \mathcal{C}$ and two computable functions f,g such that

- $(x,k) \in L \Rightarrow \Pr_{y \in \{0,1\}^f}[(x,y,k) \in \Omega] \geq \frac{1}{g(k)\log(|x|)}$.
- $(x,k) \notin L \Rightarrow \Pr_{y \in \{0,1\}^f}[(x,y,k) \in \Omega] = 0$.

we have that $L \in BP \cdot \mathcal{C}$. Then, we say that $\mathcal{C}$ has the RP-amplification property.

Theorem 129 *(RP-amplification property)*

1. *$W[P]$ has the RP-amplification property.*
2. *FPT has the RP-amplification property.*

Proof. In chapter 3, (theorem 66), we have proven that, if $\mathcal{C}$ is $\vee$-closed, then $\mathcal{C}$ has the RP-amplification property. FPT is $\vee$-closed, and $W[P]$ is $\vee$-closed (theorem 108). Hence, FPT and $W[P]$ have the RP-amplification property ■

Chapter 6

On $PH\,[P]$

In this chapter we analyze the structure of the $PH\,[P]$ hierarchy, a hierarchy of parameterized classes analogous to the polynomial hierarchy. The $PH\,[P]$ hierarchy was used in previous chapters (as an auxiliary construction) to prove suitable parameterized versions of some classical theorems.

In this chapter we focus our research on the following topics:

1. Machine characterizations of the classes in the $PH\,[P]$ hierarchy.

2. Descriptive characterizations of the classes in the hierarchy.

This chapter should be best understood as an appendix, where we have written down some boring but necessary facts concerning the $PH\,[P]$ hierarchy.

6.1 Basic facts

In previous chapters we have introduced, for all $i \in \mathbb{N}$, the classes $\Sigma_i \cdot FPT$ and $\Pi_i \cdot FPT$. Additionally we have defined the $PH[P]$ hierarchy as the union of these classes. In this chapter we investigate the $PH[P]$ hierarchy.

In this subsection we present machine characterizations of these classes. Machine characterizations are an standard way to define complexity classes. Machine characterizations of complexity classes provide us with robust definitions and useful proof techniques.

First we present complete problems for each one of the classes in the $PH[P]$ hierarchy. The parameterized problems that are presented in this section are analogous to the quantified versions of the classical problem **SAT**.

First we enumerate some basic facts about the $PH[P]$ hierarchy.

Fact 130 *1.* $\Pi_i \cdot FPT \subseteq \Sigma_{i+1} \cdot FPT$.

2. $\Sigma_i \cdot FPT \subseteq \Pi_{i+1} \cdot FPT$.

3. $\Sigma_i \cdot FPT = co - \Pi_i \cdot FPT$.

4. $\Pi_i \cdot FPT = co - \Sigma_i \cdot FPT$.

The proof of each one of the facts is straightforward and very similar to the proof of the corresponding classical facts.

Definition 131 *Given $i \in \mathbb{N}$, the problem p-$\Sigma_i CSAT$ is the following one:*

- *Instances:* $(C(X_1, ..., X_i), \langle k_1, ..., k_i \rangle)$, *where* $k_1, ..., k_i \in \mathbb{N}$ *and* C *is a boolean circuit whose input gates are partitioned in* i *blocks* $X_1, ..., X_i$.

- *Parameter:* $\langle k_1, ..., k_i \rangle$.
- *Problem: Decide if* $\exists^{k_1} Z_1 ... Q_i^{k_i} Z_i C\left(Z_1, ..., Z_i\right)$, *where* $\exists^k Z_j$ *is interpreted to mean "does there exists a weight* k *assignment to the block* X_j*" and* $\forall^k Z_j$ *is interpreted to mean "does for all weight* k *assignment to the block* X_j*". For all* $j \leq i$, *we have* $Q_j \in \{\exists, \forall\}$ *and* $Q_j = \exists$ *if and only if* j *is an odd number.*

Remark 132 *In last definition, we could take as parameter the number* $k_1 + ... + k_i$.

Given $i \in \mathbb{N}$, we define $p\text{-}\Pi_i CSAT$ accordingly.

Theorem 133 $p\text{-}\Sigma_i CSAT$ *is complete for* $\Sigma_i \cdot FPT$.

Proof. It should be clear for the reader that $p\text{-}\Sigma_i CSAT \in \Sigma_i \cdot FPT$. We prove that $p\text{-}\Sigma_i CSAT$ is $\Sigma_i \cdot FPT$ hard.

Using induction on $i \in \mathbb{N}$, we prove that $p\text{-}\Sigma_i CSAT$ is $\Sigma_i \cdot FPT$ hard and $p\text{-}\Pi_i CSAT$ is $\Pi_i \cdot FPT$ hard.

1. $(i = 1)$ $p\text{-}\Sigma_1 CSAT$ is $\Sigma_1 \cdot FPT$ hard since $p\text{-}\Sigma_1 CSAT = p\text{-}WSAT(CIRC)$ and $\Sigma_1 \cdot FPT = W[P]$. It is easy to verify that $p\text{-}\Pi_1 CSAT$ is $\Pi_1 \cdot FPT$ hard.

2. $(i = n)$ We suppose that $p\text{-}\Sigma_n CSAT$ is $\Sigma_n \cdot FPT$ hard and we suppose that $p\text{-}\Pi_n CSAT$ is $\Pi_n \cdot FPT$ hard.

3. $(i = n + 1)$ We prove that $p\text{-}\Sigma_{n+1} CSAT$ is $\Sigma_{n+1} \cdot FPT$ hard, the proof of the $\Pi_{n+1} \cdot FPT$ hardness of $p\text{-}\Pi_{n+1} CSAT$ is very similar.

   Let L be a language in $\Sigma_{n+1} \cdot FPT$, there exist $\Omega \in \Pi_n \cdot FPT$ and a computable function f such that

$(x,k) \in L \Leftrightarrow \exists y \in \{0,1\}^f \left((x,y,k) \in \Omega\right).$

We use the inductive hypothesis to claim that there exists an *fpt* algorithm M such that on input (x,y,k), an instance of Ω, the algorithm M computes $M\left(x,y,k\right)$, an instance of $p\text{-}\Pi_n CSAT$, such that

$(x,y,k) \in \Omega \Leftrightarrow M\left(x,y,k\right) \in p\text{-}\Pi_n CSAT.$

Given M we can define a nondeterministic *fpt* algorithm N in the following way

On input (x,k)

(a) N guesses $y \in \{0,1\}^f$.

(b) N simulates the computation of M on input (x,y,k).

First, we have that $(x,k) \in L$ if and only if there exists a run of N with input (x,k) such that the output is an element of $p\text{-}\Pi_n CSAT$

Second, we can use Cook's reduction [P] to claim that given (x,k) an instance of L, we can compute in *fpt* time a circuit $C_{x,k}\left(w\right)$ such that

$(x,k) \in L \Leftrightarrow \exists y \in \{0,1\}^f \left(C_{x,k}\left(y\right) \in p\text{-}\Pi_n CSAT\right)$

Hence, we have that

$(x,k) \in L \Leftrightarrow C_{x,k} \in p\text{-}\Sigma_{n+1} CSAT.$

So, we have proven that L is *fpt* many one reducible to $p\text{-}\Sigma_{n+1} CSAT$

■

Corollary 134 $PH\left[P\right] \subseteq AW\left[P\right]$.

Last corollary says that $PH[P]$ is not a very large class. For a definition of the class $AW[P]$ see [DF].

It is easy to obtain a machine characterization of the classes $\Sigma_i \cdot FPT$, (for all $i \in \mathbb{N}$), similar to the one obtained for $W[P]$ in [FG2].

Definition 135 *Given $i \in \mathbb{N}$, a $\Sigma_i[P]$ machine is an alternating Turing machine $\mathbb{M}$ such that on every run of $\mathbb{M}$ with input $(x, \langle k_1, ..., k_i \rangle)$:*

- *(Stage 1) $\mathbb{M}$ existentially guesses $f_1(k_1)\log(|x|)$ nondeterministic bits, where f_1 is a computable function.*
- *(Stage 2) $\mathbb{M}$ universally guesses $f_2(k_2)\log(|x|)$ nondeterministic bits, where f_2 is a computable function.*
- *(Stage $j \leq i$) If j is odd, $\mathbb{M}$ existentially guesses $f_j(k_j)\log(|x|)$ nondeterministic bits. If j is even, $\mathbb{M}$ universally guesses $f_j(k_j)\log(|x|)$ nondeterministic bits, (in both cases f_j is a computable function).*
- *(Stage $i+1$) In this stage, the last stage of the run, the machine works deterministically.*
- *For every run of $\mathbb{M}$, on input $(x, \langle k_1, ..., k_i \rangle)$, the running time is upperbounded by $f(k_1 + ... + k_i)\, p(|x|)$ for some computable function f and some polynomial p.*

Theorem 136 *$L \in \Sigma_i \cdot FPT$ if and only if there exists a $\Sigma_i[P]$ machine that decides L.*

Proof. It should be clear for the reader that given $L \in \Sigma_i \cdot FPT$, the language L can be decided using a $\Sigma_i[P]$ machine.

Let L be a parameterized language which can be decided using a $\Sigma_i[P]$ machine. We can use Cook's reduction to claim that L is *fpt* many one reducible to p-$\Sigma_i CSAT$. Hence, we have that $L \in \Sigma_i \cdot FPT$ ■

Remark 137 *Note that* $\mathbb{M}$ *is a* $\Sigma_1[P]$ *machine if and only if* $\mathbb{M}$ *is a* $W[P]$ *restricted Turing machine.[P].*

6.1.1 Descriptive characterizations

In this section we present two descriptive characterizations of the classes in the $PH[P]$ hierarchy. This section is strongly based on [FG3]. The main results in this section are straightforward generalizations of the descriptive characterizations of $W[P]$ obtained in [FG3].

In descriptive complexity theory, algorithmic problems are considered as classes of ordered structures in some vocabulary rather than languages over some alphabet. Consequently, parameterized problems are considered as subsets $L \subseteq Ord[\tau] \times \mathbb{N}^i$ for some vocabulary τ and some $i \in \mathbb{N}$. It is required that for each $\langle k_1, ..., k_i \rangle \in \mathbb{N}^i$ the $\langle k_1, ..., k_i \rangle$-slice, $L_{\langle k_1,...,k_i \rangle} := \{\mathcal{A} \in Ord[\tau] : (\mathcal{A}, \langle k_1, ..., k_i \rangle) \in L\}$, is closed under isomorphisms.

6.1.2 Slicewise definability

Let $\mathcal{L}$ be a logic. A parameterized problem $L \subseteq Ord[\tau] \times \mathbb{N}^i$ is $\mathcal{L}$ slicewise-definable, if there exists a computable function $\delta : \mathbb{N}^i \to \mathcal{L}$ such that for all $\mathcal{A} \in Ord[\tau]$ and for all $\langle k_1, ..., k_i \rangle \in \mathbb{N}^i$ we have:

$(\mathcal{A}, \langle k_1, ..., k_i \rangle) \in L$ if and only if $\mathcal{A} \models \delta(\langle k_1, ..., k_i \rangle)$

A family of logics $(\mathcal{L}_s)_{s\in\mathbb{N}}$ captures a parameterized complexity class $\mathcal{C}$ if for every vocabulary τ, every $i \in \mathbb{N}$ and every parameterized problem $L \subseteq Ord[\tau] \times \mathbb{N}^i$ we have

$L \in \mathcal{C}$ if and only if there is an $s \in \mathbb{N}$ such that L is slicewise $\mathcal{L}_s$ definable.

If this is the case we write $\mathcal{C} = \bigcup_{s\in\mathbb{N}} \text{slicewise-}\mathcal{L}_s$.

For $s \geq 1$, let $LFP^{[s]}$ consists of all formula of least fixed point logic of the form $[LFP_{\overline{x},X}\varphi]\,\overline{z}$, where X is of arity less than or equal to s and $\varphi \in FO^{[s]}$, that is, φ is a first order formula, and in φ at most s individual variables are quantified. Finally, given $i \in \mathbb{N}$, the symbol $\Sigma_i \cdot LFP^{[s]}$ will denote the class of all formulas of the form $Q_1\overrightarrow{x_1}...Q_i\overrightarrow{x_i}\varphi$ where $\varphi \in LFP^{[s]}$; for all $j \leq i$ we have that $Q_j \in \{\exists, \forall\}$ and for all $j \leq i$ we have $Q_j = \forall$ if and only if j is an even number.

Theorem 138 $\Sigma_i \cdot FPT = \bigcup_{s\in\mathbb{N}} \textit{slicewise-}\Sigma_i \cdot LFP^{[s]}$.

Proof. If we suppose that $L \subseteq Ord[\tau] \times \mathbb{N}^i$ is $\Sigma_i \cdot LFP^{[s]}$ definable, it is very easy to verify that there exists a $\Sigma_i[P]$ Turing machine that decides L.

In the following we will suppose that i is an even number, the case when i is an odd number is very similar.

For the other direction suppose that $L \subseteq Ord[\tau] \times \mathbb{N}^i$ belongs to $\Sigma_i \cdot FPT$. Choose a $\Sigma_i[P]$ Turing machine $\mathbb{M}$ accepting L. For every tuple $(\mathcal{A}, \langle k_1, ..., k_i\rangle) \in L$ there exists an accepting computation of $\mathbb{M}$, on input $(\mathcal{A}, \langle k_1, ..., k_i\rangle)$, whose running time is equal to $f(\langle k_1, ..., k_i\rangle)\,|A|^s$. At the beginning of the computation, in the first i steps, $\mathbb{M}$ existentially guesses $r_1 \in \{0,1\}^{k_1 \log(|A|)}$, $\mathbb{M}$ universally guesses $r_2 \in \{0,1\}^{k_2 \log(|A|)}$,..., $\mathbb{M}$ universally guesses $r_i \in \{0,1\}^{k_i \log(|A|)}$. We can arrange each one of the 0-1 strings r_j, (where $j \leq i$), in k_j blocks of $\log(|A|)$ bits, each such block corresponding to the binary representation

of a number less than or equal to $|A|$ and hence to an element of A. Let $\overrightarrow{a_j}$ be the tuple of k_j elements of A. Now, if we view the deterministic part of the computation of $\mathbb{M}$ as a $f\left(\langle k_1,...,k_i\rangle\right)|A|^s$ bounded deterministic algorithm M_d applied on $\left(\mathcal{A},\overrightarrow{a_1},...,\overrightarrow{a_i}\right)$. Then, the theorem of Immerman and Vardy says, [FG3], that for every $\langle k_1,...,k_i\rangle$, there exists an $LFP^{[s]}$ formula $\varphi_{\langle k_1,...,k_i\rangle}\left(\overrightarrow{y_1},...,\overrightarrow{y_i}\right)$ such that for all $\mathcal{A}\in Ord\left[\tau\right]$ and for every $\overrightarrow{a_1}\in A^{k_1},...,\overrightarrow{a_i}\in A^{k_i}$

$\left(\mathcal{A},\overrightarrow{a_1},...,\overrightarrow{a_i}\right)\models\varphi\left(\overrightarrow{y_1},...,\overrightarrow{y_i}\right)$ if and only if M_d accepts $\left(\mathcal{A},\overrightarrow{a_1},...,\overrightarrow{a_i}\right)$.

Thus, we have

$\mathcal{A}\models\exists\overrightarrow{y_1}\forall\overrightarrow{y_2}...\forall\overrightarrow{y_i}\varphi_{\langle k_1,...,k_i\rangle}\left(\overrightarrow{y_1},...,\overrightarrow{y_i}\right)$ if and only if $\mathbb{M}$ accepts $\left(\mathcal{A},\langle k_1,...,k_i\rangle\right)$ if and only if $\left(\mathcal{A},\langle k_1,...,k_i\rangle\right)\in L$.

This gives the desired slicewise definition of L in $\Sigma_i\cdot LFP^{[s]}$ ■

6.1.3 Fagin definability

In this subsection we obtain a second descriptive characterization of the classes $\Sigma_i\cdot FPT$, this new approach is usually called Fagin definability [FG3].

In this subsection we will consider vocabularies without function symbols.

Definition 139 *Given $k\in\mathbb{N}$, a vocabulary τ, a τ-structure $\mathcal{A}$ and a $\tau\cup\{X\}$-formula φ, (X is a relational variable of arity r).*

1. *$\mathcal{A}\models\exists^k X\varphi$ if and only if there exists $B\subseteq A^r$ such that $|B|=k$ and $(\mathcal{A},B)\models\varphi$.*

2. *$\mathcal{A}\models\forall^k X\varphi$ if and only if for all $B\subseteq A^r$, if $|B|=k$, then $(\mathcal{A},B)\models\varphi$.*

Let φ be a formula in the vocabulary $\tau\cup\{X_1,...,X_i\}$, where for each $j\leq i$, the variable

X_i is an r_i-ary relational symbol not contained in τ. The formula φ Σ_i**Fagin-defines** a parameterized problem p-$\Sigma_i FD_\varphi$.

Definition 140 *(p-$\Sigma_i FD_\varphi$)*

- *Instances:* *A τ-structure $\mathcal{A}$ and $\langle k_1, ..., k_i \rangle \in \mathbb{N}^i$.*
- *Parameter:* $\langle k_1, ..., k_i \rangle$.
- *Problem:* *Decide if $\mathcal{A} \models \exists^{k_1} X_1 \forall^{k_2} X_2 ... Q_i^{k_i} X_i \varphi$, where (for any $j \leq i$) Q_j is equal to $\forall$ if and only if j is an even number.*

For a Logic $\mathcal{L}$ we let $\Sigma_i FD(\mathcal{L})$ denote the class of all problems that are Σ_iFagin defined by a formula in $\mathcal{L}$. Remember that, given $\mathcal{C}$ a set of parameterized problems, $\langle \mathcal{C} \rangle_{fpt}$ denotes the closure of this class under *fpt* reductions. Downey, Fellows and Regan [DFR] proved that for each $t \geq 1$, we have $\langle \Sigma_1 FD(\Pi_t) \rangle_{fpt} = W[t]$, with Π_t the set of first order formulas of the form $\forall x_{11} ... \forall x_{1k_1} \exists x_{21} ... \exists x_{2k_2} ... Q x_{t1} ... Q x_{tk_t} \varphi$, where φ is quantifier free, $Q = \exists$ if t is even and $Q = \forall$ if t is odd. Flum and Grohe [FG3] proved that $\langle \Sigma_1 FD(FO(LFP)) \rangle_{fpt} = W[P]$, where $FO(LFP)$ is the Least fixed point Logic. In this subsection we prove that $\langle \Sigma_i FD(FO(LFP)) \rangle_{fpt} = \Sigma_i \cdot FPT$.

A boolean circuit C is in **negation normal form** if and only if for every not-gate s, the input of s is an input gate. It is well known that every circuit C can be transformed in *fpt* time in a equivalent circuit C^* in negation normal form [FG3].

Given $i \in \mathbb{N}$, we work with a vocabulary $\tau_i = \{I_1, ..., I_i, G_\wedge, G_\vee, G_\neg, E, o\}$, where $I_1, ..., I_i, G_\wedge, G_\vee$ and $G_\neg$ are relational symbols of arity 1, E is a relational symbol of arity 2 and o is a symbol of arity zero.

Given $C(X_1, ..., X_i)$, an instance of p-$\Sigma_i CSAT$, we define a τ_i structure $\mathcal{A}_C$ in the following way:

1. First, we compute in *fpt* time a circuit $C^*(X_1, ..., X_i)$ equivalent to $C(X_1, ..., X_i)$ and such that $C^*(X_1, ..., X_i)$ is in negation normal form.

2. From $C^*(X_1, ..., X_i)$ we define $\mathcal{A}_C$:

   - $A_C := \{s : s \text{ is a gate of } C^*\}$.
   - For all $j \leq i$: $I_j^{\mathcal{A}_C} := \{s \in A_C : s \text{ is an input gate in the block } X_j\}$.
   - $G_{\wedge}^{\mathcal{A}_C} := \{s \in A_C : s \text{ is an and-gate}\}$.
   - $G_{\vee}^{\mathcal{A}_C} := \{s \in A_C : s \text{ is an or-gate}\}$.
   - $G_{\neg}^{\mathcal{A}_C} := \{s \in A_C : s \text{ is a not-gate}\}$.
   - $E^{\mathcal{A}_C} := \{(s, s^*) \in A_C \times A_C : s^* \text{ is an input of } s\}$.
   - $o^{\mathcal{A}_C}$ is equal to the output gate of C^*.

Remark 141 *It is important to stress that the computation of $\mathcal{A}_C$ can be carried out in fpt time.*

Lemma 142 *Given $i \in \mathbb{N}$, there exists a FO formula $\varphi_i(x)$ in the vocabulary τ_i union $\{Y_1, ..., Y_i, X\}$ such that for every circuit $C(X_1, ..., X_i)$ and for every i-tuple $(T_1, ..., T_i)$, (where $T_1 \subseteq X_1, ..., T_i \subseteq X_i$), we have*

$(\mathcal{A}_C, T_1, ..., T_i) \models [LFP_{x,X}\varphi_i](o)$ if and only if the assignment $T_1 \cup ... \cup T_i$ satisfies the circuit C.

Proof. Let φ_i be the formula

$(Y_1(x) \vee ... \vee Y_i(x))$

$\vee (G_\wedge(x) \wedge \forall y (E(x,y) \rightarrow X(y)))$

$\vee (G_\vee(x) \wedge \exists y (E(x,y) \wedge X(y)))$

$\vee (G_\neg(x) \wedge \exists y (E(x,y) \wedge \neg Y_1(y) \wedge ... \wedge \neg Y_i(y)))$

It is very easy to verify that φ_i satisfies the conditions in the statement of the lemma ■

Corollary 143 *1. Given $(C(X_1,...,X_i), \langle k_1,...,k_i\rangle)$ an instance of p-$\Sigma_i CSAT$, it belongs to p-$\Sigma_i CSAT$ if and only if $\mathcal{A}_C \models \exists^{k_1} Y_1 \forall^{k_2} Z_2 ... Q^{k_i} Y_i [LFP_{x,X}\varphi_i](o)$.*

2. For all $i \geq 1$ we have p-$\Sigma_i CSAT \preceq_{fpt}$ p-$\Sigma_i FD_{\varphi_i}$.

The proof of the corollary is straightforward.

Proposition 144 *Given $\varphi(Y_1,...,Y_i)$ a $FO(LFP)$ formula, we have that p-$\Sigma_i FD_\varphi \in \Sigma_i \cdot FPT$.*

Proof. It is very easy to verify that given $\varphi(Y_1,...,Y_i)$, there exists a $\Sigma_i[P]$ Turing machine M_φ that decides p-$\Sigma_i FD_\varphi$ ■

From the corollary above and last proposition we obtain as an easy corollary the following theorem

Theorem 145 $\langle \Sigma_i FD(FO(LFP))\rangle_{fpt} = \Sigma_i \cdot FPT$.

Chapter 7

Conclusions

Proving parameterized versions of classical theorems is often hard work. We have tried to prove parameterized versions of Toda's theorem and Stockmeyer's theorem. We have partially achieved our goal. The success was partial because we had to use additional complexity theoretic hypothesis.

It is usual that arguments and techniques that work in the classical setting do not work in the parameterized setting, in the remaining cases the arguments have to be modified and many technical details have to be considered, this has been the case with our adaptation of Toda's and Stockmeyer's arguments. For example, while probability amplification is an easy task in the classical setting, for most of the probabilistic classes considered in this work, it is an open problem if we can amplify them. Furthermore, when, in the parameterized world, we can amplify we have to use sophisticated techniques based on pseudorandom generators.

In this dissertation we have proven the following theorems

1. (parameterized Toda's theorem) If $\oplus \cdot FPT$ is maj-closed, then

   $A \subseteq PH\left[P\right] \subseteq \langle p\text{-}\#WSAT\left(CIRC\right)\rangle_R$

   where A is the A hierarchy and $\langle p\text{-}\#WSAT\left(CIRC\right)\rangle_R$ denotes the closure of p-$\#WSAT\left(CIRC\right)$ under parameterized random reductions.

2. (parameterized Valiant-Vazirani theorem) If $\oplus \cdot FPT$ is $\vee$-closed, then $W\left[P\right] \subseteq BP \cdot \oplus \cdot FPT$.

3. (parameterized Stockmeyer's theorem) If $W\left[P\right]$ is $\wedge$-closed, then approximate counting of problems in $\#W\left[P\right]$ belongs to the second level of the $PH\left[P\right]$ hierarchy.

Note that in each one of the theorems listed above we have used an additional hypothesis concerning the closure of some parameterized class with respect to some special type of true table reductions. Why have we needed these closure assumptions? We have used these hypothesis to amplify probabilities. A natural question is the following one: To what extent does probability amplification depend on these closure assumptions? Chapter five is a first step into this line of research, we strongly believe that this question deserves further investigation.

A final interesting and positive conclusion is that hashing techniques seem to work well in the parameterized framework requiring only minor modifications.

Bibliography

[ADF] K.A. Abrahamson, R.G. Downey and M.R. Fellows. Fixed parameter tractability and completeness IV: On completeness for $W[P]$ and $PSPACE$ analogs. *Annals of pure and applied Logic,* 73:235-276, 1995.

[AKS] M. Ajtai, J. Komlos and E. Szemeredi. Deterministic simulation in $LOGSPACE$. *Proceedings* 19^{th} *STOC*, pages 132-140, 1987.

[ASK] A. Agrawal, Saxena and Kayal. $PRIMES$ is in P. *Annals of Mathematics,* 160(2): 781-793, 2004.

[AR] V. Arvind and V. Raman. Approximation algorithms for some parameterized counting problems. In P. Bose and P. Morin, editors, *Proceedings of the 13th Annual International Symposium on Algorithms and Computation,* Volume 2518 of *Lecture Notes in Computer Science,* pages 453-464. Springer-Verlag, 2002.

[ChFG] Y. Chen, J. Flum and M. Grohe. Machine characterizations of the classes of the W-hierarchy. In M. Baaz and J. Makowsky, editors, *Proceedings of the 17th International Workshop on Computer Science Logic,* Volume 2803 of *Lecture Notes in Computer Science,* pages 114-127. Springer-Verlag, 2003.

[ChFG2] Y. Chen, J. Flum and M. Grohe. Bounded nondeterminism and alternation in parameterized complexity theory. In *Proceedings of the 18th IEEE Conference on Computational Complexity,* pages 13-29. IEEE Computer Society, 2003.

[DJ] V. Dalmau and P. Jonsson. The complexity of counting homomorphism seen from the other side. *Theoretical Computer Science,* 329:315-323, 2004.

[DF] R.G. Downey and M.R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.

[DFR] R.G. Downey, M.R. Fellows and K. Regan. Descriptive complexity and the $W-$hierarchy. In P. Beame and S. Buss, editors, *Proof complexity and Feasible Arithmetic,* volume 39 of *AMS-DIMACS Volume Series,* pages 119-134. AMS, 1998.

[FG1] J. Flum and M. Grohe. The parameterized complexity of counting problems. *SIAM Journal of Computing,* 33(4):892-922, 2004.

[FG2] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer-Verlag, 2006.

[FG3] J. Flum and M. Grohe. Describing parameterized complexity classes. *Information and Computation,* 187(2):291-319, 2003.

[G] O. Goldreich. Randomized methods in Computation. Manuscript, 2001, http//www.wisdom.weizmann.ac.il/˜oded/rnd.html.

[KF] Ch. Kintala and P. Fisher. Computations with a restricted number of nondeter-

ministic steps. *Proceedings 9th ACM Symposium on Theory of Computing,* pp 178-185, 1977.

[L] C. Lautemann. BPP and the Polynomial Hierarchy. *Information processing Letters,* 17(4): 215-217, 1983.

[LW] M. Luby and A. Wigderson. Pairwise Independence and Derandomization. Manuscript, July 1995. http://www.icsi.berkeley.edu/~luby/pair_sur.html.

[M] C. McCartin. Parameterized counting problems. In K. Diks and W. Rytter, editors, *Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science Logic,* Volume 2420 of *Lecture Notes in Computer Science,* pages 556-567. Springer-Verlag, 2002.

[P] C.H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[PZ] C.H. Papadimitriou, S. Zachos. Two remarks on the power of counting. *Proceedings* 6^{th} *GI Conference on Theoretical Computer Sciences*, pages 269-276, 1983.

[SU] M. Schaefer and C. Umans. Completeness in the polynomial-time hierarchy: A compendium. *SIGACT News,* September 2002.

[S] L. Stockmeyer. On approximation Algorithms for $\#P$. *SIAM Journal on Computing,* 14(4): 849-861, 1985.

[T] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing,* 20(5):865-877, 1991.

[V1] L.G. Valiant. The complexity of computing the permanent. *Theoretical computer Science,* 8:189-201, 1979.

[V2] L.G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing,* 8(3):410-421, 1979.

[VV] L.G. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical computer Science,* 47:85-93, 1986.