UNIVERSITY of CALIFORNIA
SANTA CRUZ

FIXED-POINT LOGICS, DESCRIPTIVE COMPLEXITY, AND RANDOM SATISFIABILITY

A dissertation submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Albert Atserias

December 2002

The dissertation of Albert Atserias is approved:

Professor Phokion G. Kolaitis, Chair

Professor Martín Abadi

Professor Luca de Alfaro

Frank Talamantes
Vice Provost and Dean of Graduate Studies

Contents

Abstract v

Acknowledgements vii

Chapter 1 Introduction 1

1.1 Finite Model Theory and Descriptive Complexity 1

1.2 Results of the Dissertation 3

1.2.1 Least Fixed-Point Logic in Finite Set Theory 4

1.2.2 Complexity of the Fixed-Points of Bounded Formulas 11

1.2.3 Datalog on Random CNF Formulas 17

1.2.4 Consequences for Proof Complexity and Constraint Satisfaction . . . 22

Chapter 2 Preliminaries 27

2.1 Languages and Structures 27

2.2 First-Order Logic 29

2.3 Least Fixed-Point Logic 33

2.4 Uniform Definability and Queries 39

2.5 Model of Computation 40

Chapter 3 Fixed-Point Logics in Finite Set Theory 43

3.1 Basic Definitions 43

3.2 On $\mathrm{LFP}(\Delta_0)$ vs. FO 44

3.3 Restricted $\mathrm{LFP}(\Delta_0)$ Collapses to FO 49

3.4 Unary $\mathrm{LFP}(\Delta_0)$ and Binary $\mathrm{LFP}(\Delta_0)$ Collapse to FO 59

3.5 On the Logical Complexity of the Formulas 66

Chapter 4 Descriptive Complexity of the Fixed-Points of Bounded Formulas 71

4.1 Key Lemma 71

4.2 Closure Functions of Positive Δ_0-Formulas 73

4.3 Complexity of Evaluating $\mathrm{LFP}(\Delta_0)$ 78

4.4 Circuit Uniformity and the Complexity Class $\mathbf{AC}^0$ 81

4.5 Capturing Results without Input Predicates 85
4.6 Capturing Results with Input Predicates 91
4.7 Consequences 95

Chapter 5 Datalog on Random CNF Formulas 97
5.1 About Datalog and Infinitary Logic 97
5.2 Formulas as Structures and Satisfiability as Homomorphism 102
5.3 Random Model of CNF Formulas 106
5.4 Main Result and Discussion of Proof Techniques 108
5.5 Extension Axioms and the Matching Game 111
5.6 Proof of Main Result 116

Chapter 6 Consequences for Proof Complexity 123
6.1 Easier Games 123
6.2 On the Complexity of Resolution 128

Chapter 7 Conclusions 137
7.1 General Discussion 137
7.2 About Finite Set Theory 138
7.3 About Uniformity of Circuits 140
7.4 About Inexpressibility Results and Proof Complexity 142

Bibliography 147

Abstract

Fixed-Point Logics, Descriptive Complexity,

and Random Satisfiability

by

Albert Atserias

We study the expressive power and the computational aspects of fixed-point logics on finite structures. Fixed-point logics is the generic name for the several known extensions of first-order logic with a mechanism to incorporate recursion through inductive definitions.

In the first part of the thesis, we compare the expressive power of least fixed-point logic LFP with that of first-order logic FO on structures that are equipped with the powerful built-in predicate known as BIT, or equivalently, the set-membership relation when the elements of the universe are interpreted as hereditarily finite sets. The power and importance of this built-in predicate stems from the fact that it provides logics with the ability to simulate low-level computations. Moreover, its set-theoretic interpretation allows us to use methods and techniques from set theory. We consider natural fragments of LFP syntactically defined in terms of the bounded formulas of set theory. The goal is to identify the boundary between the fragments of LFP that collapse to FO and those whose collapse is unlikely or will be hard to settle. First, we show that certain large natural fragments of LFP collapse to FO. The proofs of these collapsing results are based on the absoluteness properties that the bounded formulas of set theory enjoy. Second, we prove that the collapse of essentially the next fragment would imply unexpected results in complexity theory. For these fragments, we focus on their computational aspects and identify the complexity classes they capture.

In the second part of the thesis, we obtain inexpressibility results for Datalog and use them to derive consequences for computational complexity. Datalog is a pure fixed-point logic without built-in predicates capable of expressing many important properties, such as unsatisfiability of 2-CNF formulas, non 2-colorability of graphs, the monotone circuit value problem, and path systems, among others. We revisit the model for random 3-CNF formulas with a fixed density, and prove that every Datalog property that implies unsatisfiability of these formulas must have asymptotic probability zero, even when formulas are taken in the region in which most formulas are unsatisfiable. Then we observe that our negative result allows us re-derive the well-known lower bounds for Resolution refutations of random 3-CNF formulas, and the Pigeonhole Principle. Thus, our results establish precise connections between the areas of finite model theory and propositional proof complexity.

Acknowledgements

There are many people to whom I want to express my gratitude. The first in the list is my advisor Phokion Kolaitis for having taught me so wonderfully the finite model theory, among many other things. I hope and expect to keep learning from him in the near and later future. I also want to thank José Balcázar for having suggested UCSC as a good place for graduate studies at a stage when all I knew was that I wanted to learn about "logic and complexity".

In the early stage of my graduate studies I shared experiences with a number of great people I would like to thank: Ahmed Amer, Bao Lichun, Dan Ford, Mark Boyd, Nigel Duffy, Raman Muthukrishnan, Ray Wheeler, Tom Rafill, Wei Ma, Xiang Zheng, and others. In the late stage, Dean Bailey kindly accepted that we share an office as workplace. The experience was fun and fruitful. I am also grateful to him. I don't want to forget the staff of the CS/CSE department, and specially Carol Mullane, whose continuous help was of great value.

Finally, I want to thank my people back home in Barcelona. Their permanent encouragement was a crucial ingredient in the soup. Without them this dissertation would not had been possible, and that is why I dedicate it to them.

Chapter 1

Introduction

1.1 Finite Model Theory and Descriptive Complexity

The fields of finite model theory and descriptive complexity theory were launched by Fagin's Theorem [Fag74], which asserts that the complexity class $\mathbf{NP}$ is captured by the existential fragment of second-order logic when inputs are naturally encoded as finite structures. Later on, the complexity class $\mathbf{P}$ was also characterized by Immerman and Vardi [Imm86, Var82]: $\mathbf{P}$ is captured by least fixed-point logic when inputs are encoded as finite structures with a built-in linear order. There are two important concepts involved in the statement of the Immerman-Vardi Theorem on which we want to elaborate: (1) the fact that the logical formalism is *least fixed-point logic*, and (2) the fact that the result requires a *built-in linear order*.

Least fixed-point logic is the extension of first-order logic with the ability to form least fixed-points of positive first-order formulas. Inductive definability theory was developed in the 1960's as the consequence of a need to formalize computability on abstract domains, other than the natural numbers [Mos74]. Quite interestingly, the same concepts re-appeared over and over again in the context of computer science, partly because (least) fixed-points can be reached by iteration, the quintessence of computation. Perhaps more importantly, least fixed-points of positive first-order formulas on finite structures can be computed efficiently, that is, in time polynomial in the size of the structure. The main contribution of Immerman and Vardi was then in realizing that every such computation can also be encoded as the least fixed-point of a positive first-order formula, when the structures are equipped with a built-in linear order. And this leads us to the second aspect of the Immerman-Vardi Theorem on which we want to focus.

Intuitively, a built-in predicate is one whose interpretation is the same in every structure. For example, equality is commonly assumed as primitive in most logics and can itself be interpreted as a built-in predicate. The reader will have noticed that Fagin's Theorem is about arbitrary finite structures without any required built-in predicate, as opposed to the Immerman-Vardi Theorem that requires a built-in linear order. Could we have a similar result for the complexity class $\mathbf{P}$? That is, is there a logic that captures $\mathbf{P}$ on arbitrary finite structures [CH82, Gur88]? This fundamental question remains open after more than 20 years of research in the area. All attempts to capture $\mathbf{P}$ by different logics, such as the Immerman-Vardi Theorem or Grädel's Theorem [Grä92], require some form of built-in predicate. The general belief is that the answer to the question above is negative, when properly formalized [Gur88], and that this is due to a fundamental difference between $\mathbf{P}$ and $\mathbf{NP}$. Since it is known that if $\mathbf{P} = \mathbf{NP}$ then there is a logic that captures $\mathbf{P}$, a *proof* that the answer is negative would in fact prove that $\mathbf{P} \neq \mathbf{NP}$.

The research presented in this dissertation is very much motivated by the two aspects of the Immerman-Vardi Theorem that we just discussed. We study the expressive power of several fixed-point logics on finite structures, with and without built-in predicates. When built-in predicates are available, we characterize expressive power in terms of computational power through the hierarchy of complexity classes provided by complexity theory. When built-in predicates are waived, we prove strong non-expressibility results that shed light on the computational difficulty of certain problems such as the satisfiability problem for propositional logic SAT. The connection with SAT is deeper than simple intuition since we succeed in deriving proof complexity lower bounds from our inexpressibility results. This establishes, for the first time, a formal connection between finite model theory and propositional proof complexity that may turn out to be fruitful in the future.

1.2 Results of the Dissertation

This section describes the contents of each of the chapters of the dissertation. Chapter 2 contains the necessary preliminaries. The results of the dissertation are presented in Chapters 3, 4, 5, and 6. As a guide before we get into the technical content, let us say that the results of Chapters 3 and 4 are concerned with the issue of built-in predicates in least fixed-point logic, and the results of Chapters 5 and 6 are concerned with inexpressibility results, and their consequences. The last chapter is devoted to the conclusions and to discuss the open problems that our work suggests.

1.2.1 Least Fixed-Point Logic in Finite Set Theory

As stated already, the Immerman-Vardi Theorem establishes that on finite structures with a built-in linear order, least fixed-point logic LFP captures the complexity class **P**. But what about first-order logic FO? Is it able to capture **P** on some infinite class of finite structures? Let us note that there exist infinite classes of finite structures on which LFP collapses to FO, such as the class of all finite equivalence relations, and both fail to capture **P** on them. But the following question remains open: is it possible that **P** and LFP collapse to FO on some infinite class of finite structures with built-in linear order? It is known that FO fails to capture **P** on a number of important classes of structures even when a built-in linear order is available. For example, the property "the number of edges is even" is not first-order expressible on the class of all ordered finite graphs, or the property "the number of atoms is a perfect square" is not first-order expressible on the class of all lexicographically ordered finite Boolean algebras. Since such properties are clearly computable in polynomial-time, FO is strictly included in **P** or LFP on such classes. In fact, all empirical evidence suggests that LFP is always strictly more expressive than FO on structures with built-in linear order. Kolaitis and Vardi conjectured so:

Conjecture 1 (The Ordered Conjecture [KV92]) *On every infinite class of finite structures with built-in linear order, there is a query that is definable in least fixed-point logic but that is not definable in first-order logic.*

There are some natural classes of structures on which the relationship between LFP and FO remains unknown, and on which the conjecture becomes particularly interesting.

These are the classes of structures that have strong built-in predicates in addition to the linear order. For example, one may consider classes of structures with built-in linear order and built-in arithmetic relations such as addition and multiplication. On such classes, establishing that LFP is strictly more expressive than FO is much more difficult. In particular, it is not known whether LFP is strictly more expressive than FO on the class of arithmetical finite structures

$$\mathcal{A} = \{(\{0, \ldots, n-1\}, <, +, \times) : n \in \mathbb{N}\}.$$

It turns out that this question is literally equivalent to the difficult complexity-theoretic question $\mathbf{LINH} \overset{?}{=} \mathbf{E}$. Indeed, LFP is strictly more expressive than FO on $\mathcal{A}$ if and only if $\mathbf{LINH} \neq \mathbf{E}$ [DLW96]. Here, $\mathbf{LINH}$ is the Linear-Time Hierarchy of Wrathall [Wra78], that is, the class of problems computable in time $O(n)$ by an alternating Turing machine making a constant number of alternations, and $\mathbf{E}$ is the class of problems computable in linear exponential time $2^{O(n)}$ by a deterministic Turing machine. The question $\mathbf{LINH} \overset{?}{=} \mathbf{E}$ is the linear time analogue of the more well-known question $\mathbf{PH} \overset{?}{=} \mathbf{EXP}$, where $\mathbf{PH}$ is the Polynomial-Time Hierarchy of Stockmeyer [Sto77], and $\mathbf{EXP}$ is the class of problems computable in exponential time $2^{n^{O(1)}}$. Thus, settling the Ordered Conjecture is at least as hard as a difficult question in complexity theory. It turns out that refuting it is also difficult since it would imply that $\mathbf{P} \neq \mathbf{PSPACE}$ [DH95].

Another well-known built-in relation to consider is the BIT predicate which consists of all pairs of natural numbers (i, m) such that the i-th bit of the binary representation of m is one. Formally,

$$\text{BIT} = \{(i, m) : \lfloor m/2^i \rfloor \equiv 1 \pmod 2\}.$$

Note that the least significant bit is the 0-th according to this definition. The class of structures that arises is

$$\mathcal{B} = \{(\{0, \ldots, n-1\}, <, \mathrm{BIT}) : n \in \mathbb{N}\}.$$

It turns out that FO is equally expressive on $\mathcal{A}$ and $\mathcal{B}$. That is, a query is first-order expressible in $\mathcal{A}$ if and only if it is first-order expressible in $\mathcal{B}$. This follows from the fact that BIT is first-order definable from $+$ and $\times$, and that $+$ and $\times$ are first-order definable from BIT and $<$ (see [Imm99]). For the same reason, LFP is equally expressive on $\mathcal{A}$ and $\mathcal{B}$. Thus, the question whether LFP collapses to FO on $\mathcal{B}$ is also difficult. We point out that this question was raised by Gurevich, Immerman, and Shelah [GIS94] as a "fascinating question in complexity theory and logic related to uniformity of circuits and logical descriptions". Indeed, it can be shown that LFP is strictly more expressive than FO on $\mathcal{B}$ if and only if $\mathbf{DLOGTIME}$-uniform $\mathbf{AC}^0$ is different than $\mathbf{P}$-uniform $\mathbf{AC}^0$, and by the facts above, if and only if $\mathbf{LINH} \neq \mathbf{E}$. We will return to the issue of the class $\mathbf{AC}^0$ and uniformity of circuits later on.

The results in Chapter 3 of this dissertation advance the state of knowledge about the ordered conjecture on $\mathcal{B}$ by seeking to delineate the boundary where this conjecture becomes hard to settle. In order to do that, we study certain natural fragments of LFP on $\mathcal{B}$. We identify a natural proper fragment of LFP for which the ordered conjecture cannot be settled without resolving open problems in complexity theory at the same time. We then establish that the ordered conjecture actually *fails* when further restricted to certain fairly expressive fragments of LFP on $\mathcal{B}$, which means that these fragments collapse to first-order logic on $\mathcal{B}$. To isolate these fragments of LFP, it is more convenient to work with an interesting alternative interpretation of the BIT relation that we discuss next.

While the definition of BIT appears to be highly arithmetical, there is a surprising re-interpretation of its meaning in terms of finite sets. To illustrate this, let $k = \lceil \log_2 n \rceil$ be the length of the binary representation of $n-1$, and observe that every subset of $\{0, \ldots, k-1\}$ can be represented as a "bit-array" of length k. Indeed, the set $S \subseteq \{0, \ldots, k-1\}$ is represented by $(a_{k-1}, \ldots, a_0) \in \{0,1\}^k$, where $a_i = 1$ if and only if $i \in S$. Conversely, every bit-array of length k corresponds to a subset of $\{0, \ldots, k-1\}$. Indeed, the bit-array $(a_{k-1}, \ldots, a_0) \in \{0,1\}^k$ corresponds to the set $\{i : a_i = 1\}$. Now, since every number in $\{0, \ldots, n-1\}$ is represented uniquely as a bit-array of length k through its binary representation, this proves that there is a one-to-one mapping from $\{0, \ldots, n-1\}$ to $\mathcal{P}(\{0, \ldots, k-1\})$, the set of all subsets of $\{0, \ldots, k-1\}$. If we iterate this one more step, we obtain a one-to-one mapping from $\{0, \ldots, n-1\}$ to $\mathcal{P}(\mathcal{P}(\{0, \ldots, k'-1\}))$, where $k' = \lceil \log_2 k \rceil$ is the length of the binary representation of $k-1$. Continuing this way we obtain a one-to-one mapping from $\{0, \ldots, n-1\}$ to $\mathcal{P}(\mathcal{P}(\cdots \mathcal{P}(\emptyset) \cdots))$. This idea gives rise to what is known as the Ackermann isomorphism defined by the recursion:

$$\begin{aligned} e(0) &= \emptyset, \\ e(m) &= \{e(i) : (i,m) \in \mathrm{BIT}\}. \end{aligned}$$

It is not hard to prove that e is an isomorphism between $(\mathbb{N}, \mathrm{BIT})$ and $(V_\omega, \in)$, where V_ω is the set of hereditarily finite sets defined by $V_\omega = \bigcup_{n \geq 0} V_n$, and $V_0 = \emptyset,\ V_{n+1} = \mathcal{P}(V_n)$ (see [Bar75]). The cumulative hierarchy of finite ranks V_n automatically suggests an interesting class of finite structures:

$$\mathcal{FR} = \{(V_n, \in) : n \in \mathbb{N}\}.$$

In fact, the Ackermann bijection suggests that we re-interpret the class $\mathcal{B}$ by

$$\mathcal{BFR} = \{(P_n, \in) : n \in \mathbb{N}\},$$

where $P_n = \{e(0), e(1), \ldots, e(n-1)\}$. Note that $\mathcal{BFR}$ does not come with a built-in linear order. However, a result of Dawar, Doets, Lindell, and Weinstein [DDLW98] implies that there is a first-order formula that defines a linear order on every structure from $\mathcal{BFR}$. Moreover, the linear order coincides with the image of the standard linear order $<$ under the Ackermann isomorphism. Thus, the special case of the ordered conjecture on $\mathcal{B}$ raised by [GIS94] becomes equivalent to the following question:

Question 1 *Is* LFP *contained in* FO *on* $\mathcal{BFR}$*?*

This framework makes it possible to consider set-theoretic concepts and techniques with the hope that these will provide new insights on the ordered conjecture on $\mathcal{B}$. In this work we will focus on the concept of Δ_0-formulas from set-theory. These are the first-order formulas that are obtained by forcing all quantifiers to be of the form $(\exists x \in y)$ and $(\forall x \in y)$. The collection of Δ_0-formulas has played a fundamental role in the development of set theory. The key point is that Δ_0-formulas are *absolute* and *expressive*. We will make strong use of these facts in our results.

Let $\mathrm{LFP}(\Delta_0)$ be the fragment of least fixed-point logic that consists of the least fixed-points $\mathrm{LFP}_{\overline{x},R}\varphi(\overline{x}, R)$ of all Δ_0-formulas $\varphi(\overline{x}, R)$ that are positive in the relation symbol R. We consider the following variant of the ordered conjecture:

Question 2 *Is* $\mathrm{LFP}(\Delta_0)$ *contained in* FO *on* $\mathcal{BFR}$*?*

Clearly, a negative answer would imply the ordered conjecture on $\mathcal{B}$ and thus $\mathbf{LINH} \neq \mathbf{E}$. On the other hand, it is conceivable that $\mathrm{LFP}(\Delta_0)$ could collapse to FO on $\mathcal{BFR}$ because the class of Δ_0-formulas is a well-behaved proper fragment of all formulas. The first result of Chapter 3 establishes that if $\mathrm{LFP}(\Delta_0) \subseteq \mathrm{FO}$ on $\mathcal{BFR}$, then $\mathbf{P} \subseteq \mathbf{LINH}$, which in turn implies that $\mathbf{P} \neq \mathbf{PSPACE}$. Thus, Question 2 cannot be settled without resolving important open problems in complexity theory at the same time.

This result motivates that we consider further restrictions of $\mathrm{LFP}(\Delta_0)$ and ask whether they collapse to FO on $\mathcal{BFR}$. Let us isolate a fairly expressive fragment of the Δ_0-formulas: A formula $\varphi(\overline{x}, R)$ is called *restricted* Δ_0 if it is Δ_0 and every occurrence of R involves bound variables of φ only. We show that if $\varphi(\overline{x}, R)$ is an arbitrary restricted Δ_0-formula that is positive in R, then its least fixed-point $\mathrm{LFP}_{\overline{x},R}\varphi(\overline{x}, R)$ is first-order definable on $\mathcal{BFR}$. To illustrate the power of least fixed-points of restricted Δ_0-formulas, let us consider the following example. Let $\varphi(x, y, R)$ be the formula

$$(\exists y' \in y)(\forall x' \in x)(y' \notin x \wedge (x' \notin y \rightarrow R(x', y'))).$$

From its very definition we see that $\varphi(x, y, R)$ is restricted Δ_0, and positive in R. It turns out that $\mathrm{LFP}_{x,y,R}\varphi(x, y, R)$ defines a linear order on $\mathcal{BFR}$. Indeed, suppose that all elements of x and y have already been ordered. Then, x is declared smaller than y if the largest element in the symmetric difference of x and y belongs to y. It is not hard to see that this defines a linear order, and that this is exactly what the least fixed-point formula $\mathrm{LFP}_{x,y,R}\varphi(x, y, R)$ expresses. Now, from our general result, we see that this linear order is first-order definable on $\mathcal{BFR}$. Let us point out that the concept of restricted Δ_0-formulas that we introduced is inspired by the abovementioned result of Dawar, Doets, Lindell, and Weinstein showing that the linear order is first-order definable on $\mathcal{BFR}$. Indeed, Dawar et al. introduced the formula $\mathrm{LFP}_{x,y,R}\varphi(x,y,R)$ and proved that this particular case is first-order definable by an adhoc argument. The new contribution of our result is in identifying the concept of restricted Δ_0-formula, and generalizing the techniques of Dawar et al. to work for all such formulas. This required some new ideas, such as using absoluteness arguments to prove that restricted Δ_0-formulas that are positive have unique fixed-points. We also illustrate the power of our result by showing that certain basic queries such as *ordinal addition* are definable by a restricted Δ_0-formula, and therefore definable by a first-order formula on $\mathcal{BFR}$.

After this, we consider fragments of $\mathrm{LFP}(\Delta_0)$ obtained by restricting the number of free variables in the Δ_0-formulas under consideration. We observe that if R is a unary relation symbol and $\varphi(x,R)$ is a Δ_0-formula that is positive in R, then the least fixed-point of $\varphi(x,R)$ coincides with the least fixed-point of some restricted Δ_0-formula $\varphi^*(x,R)$. It follows that unary $\mathrm{LFP}(\Delta_0)$ collapses to FO on $\mathcal{BFR}$. This raises the question whether any similar collapses can be proved for $\mathrm{LFP}(\Delta_0)$-formulas of higher arities, while keeping in mind that we cannot hope to show that $\mathrm{LFP}(\Delta_0)$-formulas of arbitrary arities collapse to FO on $\mathcal{BFR}$ without simultaneously showing that $\mathbf{P} \neq \mathbf{PSPACE}$. Our main result along these lines is that binary $\mathrm{LFP}(\Delta_0)$ collapses to FO on $\mathcal{BFR}$. This is the most technically difficult result of Chapter 3 and its proof requires a number of new ideas to deal with the non-restricted occurrences of the recursive relation R. Moreover, the result has an interesting complexity-theoretic interpretation: while the least fixed-point of a binary Δ_0-formula is obviously a binary relation on the set involved elements, our theorem says that quantification over *sets* of these elements can simulate the fixed-point computation. The obvious statement would have been that quantification over *binary relations* of these elements can simulate the fixed-point computation. Thus, our theorem provides a non-obvious speed-up. Unfortunately, the proof technique of our theorem does not seem to extend to ternary $\mathrm{LFP}(\Delta_0)$.

The last section of Chapter 3 deals with the logical complexity of the first-order formulas that define the fixed-points. It turns out that the proofs of the theorems show that the fixed-points of restricted Δ_0-formulas are definable by a Σ-formula on $\mathcal{BFR}$. The class of Σ-formulas is the smallest class of formulas containing the Δ_0-formulas and closed under conjunction, disjunction, bounded quantification $(\exists y \in x)$, $(\forall y \in x)$, and existential quantification $(\exists y)$. We observe that, in fact, the formula can be chosen to be Σ_1, meaning that every unbounded existential quantifier precedes every bounded quantifier. We note that this need not be immediately obvious since the Σ-reflection principle, which means that every Σ-definable predicate is also Σ_1-definable, is not known to hold on $\mathcal{BFR}$. Also, by duality, the formulas may be chosen to be Π_1, the class that is obtained from the Δ_0-formulas by closing under universal quantification $(\forall y)$. Therefore, least fixed-points of positive restricted Δ_0-formulas are Δ_1-definable on $\mathcal{BFR}$.

1.2.2 Complexity of the Fixed-Points of Bounded Formulas

The isomorphism mapping BIT to the membership relation $\in$ constitutes a good source of inspiration to obtain results that explain the expressive power of first-order logic and fixed-point logic when strong built-in relations are available. Moreover, the set-theoretic framework provides new concepts to consider, such as Δ_0-formulas, and new techniques to apply, such as absoluteness arguments. The results in Chapter 3 are good examples of this. However, the complexity aspects of $\mathrm{LFP}(\Delta_0)$ were not completely studied in Chapter 3, and the goal of Chapter 4 is to complete this study.

The first result of the chapter concerns the *closure functions* of the least fixed-points corresponding to positive Δ_0 and restricted Δ_0-formulas. Informally, if $\varphi(\overline{x}, R)$ is a first-order formula that is positive in the relation symbol R, and $\mathbf{M}$ is a structure, the closure ordinal of $\varphi(\overline{x}, R)$ on $\mathbf{M}$ is the minimum number of iterations of φ that are needed to obtain its least fixed-point on $\mathbf{M}$. Then, the closure function of φ is the function that assigns, to each cardinality n, the maximum closure ordinal of φ over structures of size n. Thus, the closure function of a formula can be viewed as a measure of the complexity of evaluating its least fixed-point. In particular, establishing that certain formulas have slowly growing closure functions is a valuable result as far as the efficiency of evaluating their fixed-points is concerned. We note, by the way, that it is easy to see that on structures of cardinality n, every closure function of a positive first-order formula is bounded by a polynomial function $n^{O(1)}$, and that this upper bound is tight.

We already argued that the least fixed-points of positive Δ_0 and restricted Δ_0-formulas form an interesting and well-behaved subclass of LFP. The first part of Chapter 4 will provide more evidence of this by analyzing their closure functions.

We prove that the closure functions of positive restricted Δ_0-formulas are bounded by $O(\log^*(n))$. Here $\log^*(n)$ stands for the minimal r such that $\log^{(r)}(n) = 1$, where $\log^{(r)}(n)$ is the r-fold composition of $\lfloor \log_2(n) \rfloor$ with itself. Formally, $\log^{(0)}(n) = 1$ and $\log^{(r+1)}(n) = \lfloor \log_2(\log^{(r)}(n)) \rfloor$. In a sense then, the least fixed-points of positive restricted

Δ_0-formulas are well-behaved because their closure functions grow extremely slowly. We should stress, however, that if the closure functions of positive restricted Δ_0-formulas were actually bounded by some constant, then the fact that their least fixed-point is first-order definable would only be a trivial consequence. Therefore, it is interesting to observe, as we do, that the closure functions of certain (natural) positive restricted Δ_0-formulas meet the upper bound $O(\log^*(n))$ optimally up to constant factors.

As far as general positive Δ_0-formulas are concerned, we prove that their closure functions are bounded by polylogarithmic functions $(\log(n))^{O(1)}$. Again, this shows that the least fixed-points of positive Δ_0-formulas can be evaluated much faster than general first-order formulas. Let us point out that while polylogarithmic functions grow much slower than polynomial functions, they do grow much faster than $O(\log^*(n))$. This establishes a deep difference between least fixed-points of restricted Δ_0-formulas and general Δ_0-formulas which provides further intuition on why one fragment collapses to FO while the other possibly does not. Finally, we also prove the tightness of the upper bound by exhibiting positive Δ_0-formulas whose closure functions grow polylogarithmically.

The results that we just mentioned establish that the least fixed-points of positive Δ_0-formulas can be evaluated much faster than those of general first-order formulas. However, we aim for an exact complexity-theoretic classification of the properties that are expressible in $\mathrm{LFP}(\Delta_0)$. That is, we want to study the descriptive complexity of $\mathrm{LFP}(\Delta_0)$. Our first result in this respect is that every query expressible in $\mathrm{LFP}(\Delta_0)$ is computable in **DPOLYLOGTIME** on arbitrary finite structures with built-in membership relation. Here, **DPOLYLOGTIME** is the class of languages that are decidable in polylogarithmic time by a deterministic Turing machine. We note that when considering sublinear time-bounds, it is common to assume as we do that the machine model is equipped with a random access mechanism to the input.

But which complexity class is really underlying $\mathrm{LFP}(\Delta_0)$? Before we answer this question, let us recall the well-known results of Barrington, Immerman, and Straubing [BIS90] about the descriptive complexity of first-order logic. The Barrington-Immerman-Straubing Theorem states that FO captures $\mathbf{LH}$ on finite structures with built-in BIT or membership relation. Here, $\mathbf{LH}$ is the Logarithmic-Time Hierarchy of Sipser (see [Bus87]), that is, the class of problems that are computable in logarithmic-time $O(\log n)$ by alternating Turing machines making a constant number of alternations. We note that the Logarithmic-Time Hierarchy $\mathbf{LH}$ is the logarithmic-time analogue of the Linear-Time Hierarchy $\mathbf{LINH}$, which is the linear-time analogue of the most popular Polynomial-Time Hierarchy $\mathbf{PH}$ of Stockmeyer. Moreover, Barrington, Immerman, and Straubing also showed that $\mathbf{LH}$ coincides with $\mathbf{DLOGTIME}$-uniform $\mathbf{AC}^0$, which is the class of problems that are computable by a family of circuits of constant alternation depth and polynomial-size, and the family of circuits is $\mathbf{DLOGTIME}$-uniform in the sense that the circuit for structures of size n is recognizable in time $O(\log n)$ by a deterministic Turing machine.

Now, let us focus back to $\mathrm{LFP}(\Delta_0)$ on the class $\mathcal{BFR}$ without additional relations. We observe that on this particular class, $\mathrm{LFP}(\Delta_0)$ is even in non-uniform $\mathbf{AC}^0$ for some trivial reasons. This means that every query that is definable in $\mathrm{LFP}(\Delta_0)$ on $\mathcal{BFR}$ is already computable by circuits of constant alternation depth and polynomial-size, but the circuits are non-uniform in the sense that a different circuit is used for each input length. The inclusion of $\mathrm{LFP}(\Delta_0)$ into non-uniform $\mathbf{AC}^0$ is due to the trivial reason that queries on $\mathcal{BFR}$ depend only on the cardinality of the structure, and such queries are always in non-uniform $\mathbf{AC}^0$. Since it is known that first-order logic FO captures a very uniform version of $\mathbf{AC}^0$, and since $\mathrm{LFP}(\Delta_0)$ is certainly a uniform class, the interesting question becomes:

Question 3 *Which uniform version of* $\mathbf{AC}^0$ *is captured by* $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$*, the first-order closure of* $\mathrm{LFP}(\Delta_0)$*?*

Our second main result of Chapter 4 is the complete answer to this question: on $\mathcal{BFR}$, the logic $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ captures $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$. In words, what this means is that $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ captures the class of queries that are computable by a uniform family of circuits of constant alternation-depth and polynomial-size, and the family of circuits is uniform in the sense that the circuit for structures of size n is recognizable in time $(\log n)^{O(1)}$ by a deterministic Turing machine. We will make all these concepts precise in Chapter 4. It turns out that $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$ is a fairly natural complexity class since we also show that it coincides with $\mathbf{LH}^{\mathbf{P}}$, the logarithmic-time hierarchy of Sipser with an oracle to $\mathbf{P}$. In words, this is the class of problems that are computable in logarithmic-time and a constant number of alternations by an alternating Turing machine that has access to an arbitrary oracle in the complexity class $\mathbf{P}$. Thus, our result can be stated in the following circuit-free, clean form: $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ captures $\mathbf{LH}^{\mathbf{P}}$ on $\mathcal{BFR}$.

As a corollary to these capturing results, we obtain an exact characterization of the complexity-theoretic difficulties of showing $\mathrm{FO}+\mathrm{LFP}(\Delta_0) = \mathrm{FO}$ on $\mathcal{BFR}$. We show that the collapse is equivalent to $\mathbf{P} \subseteq \mathbf{LINH}$. This question is extremely interesting since, intuitively speaking, it amounts to asking whether alternation can replace loop nesting. We note that the complexity class $\mathbf{LINH}$ coincides with the rudimentary languages $\mathbf{RUD}$ [Wra78] introduced by Smullyan [Smu61] and has received a good deal of attention by researchers of different areas, from formal language theory [Jon69, BN78] to bounded arithmetic [Bus98, Woo81, Wil79].

We then consider the descriptive complexity of $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ on arbitrary finite structures with built-in membership relation. Unfortunately, we are able to provide an exact answer only in the case that the underlying vocabulary of the class of structures is unary (on classes of *words* with built-in membership relation). In that case, $\mathrm{FO}+\mathrm{LFP}(\Delta_0)$ still captures $\mathbf{LH}^{\mathbf{P}}$, or alternatively, $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$. This result is more subtle than it seems since the $\mathrm{LFP}(\Delta_0)$ subformulas may speak freely about the unary relations of the structure. Thus, it is not even immediately obvious that $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ is in non-uniform $\mathbf{AC}^0$ since those formulas may require $\mathbf{DPOLYLOGTIME}$ to evaluate. However, the crucial observation is that the relevant part of the unary relations on which the $\mathrm{LFP}(\Delta_0)$ subformulas depend turns out to be very small. We take advantage of this fact to prove our theorem.

For languages of higher arities, we are able only to relate the expressive power of $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ to a complexity-theoretic question. We show that if $\mathbf{P} \subseteq \mathbf{RUD}_{n^{1/r}}$, then $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ collapses to FO on finite structures with built-in membership relation over a vocabulary of arity at most r. The class $\mathbf{RUD}_{n^{1/r}}$ was introduced by Jones [Jon75] as a natural subclass of the rudimentary languages $\mathbf{RUD} = \mathbf{RUD}_n$. Moreover, Allender and Gore observed that $\mathbf{RUD}_{n^\epsilon}$ contains complete problems of each level of $\mathbf{PH}$.

1.2.3 Datalog on Random CNF Formulas

The results of Chapters 3 and 4 are concerned with the expressive power of least fixed-point logic when strong built-in predicates, such as the membership relation, are available. This brings several interesting connections to computational complexity, in the form of collapsing and capturing results. However, studying logics with built-in predicates is not the only way of obtaining connections with computational complexity. In Chapter 5 we will focus on Datalog, a pure fixed-point logic without built-in predicates. We obtain strong inexpressibility results for this logic, and then use them to prove complexity-theoretic lower bounds related to the the satisfiability problem for propositional logic. Moreover, our results will bring three different active areas of research in computational complexity together: the complexity of random 3SAT, the complexity of resolution proofs, and the complexity of constraint satisfaction problems. Let us start by introducing the framework of random 3SAT.

The *phase transition* phenomenon that certain combinatorial problems exhibit has inspired a new line of research within the field of computational complexity. The focus is no longer exclusively set to the worst-case complexity of the problem, but also in studying its structure for the typical instances when drawn from certain distributions of interest. This research is motivated by the need to understand the structure of hard-to-solve $\mathbf{NP}$-complete problems of practical importance. Moreover, the structural analysis of the typical instance is often followed by an analysis of the complexity of the known algorithms to solve the problem. As a matter of fact, the methods for these two tasks are very often related, and one task can be seen as complementary of the other. The ultimate, more ambitious goal of the program would be to understand the *average-case* complexity of the $\mathbf{NP}$-complete problem under consideration.

The satisfiability problem for 3-CNF formulas, the most popular $\mathbf{NP}$-complete problem, has been intensively studied from this perspective. The distribution of interest in this context is parameterized by the number of variables n and the number of clauses m of the formula. Thus, following the standard notation, we let $\mathcal{F}(n, m, 3)$ denote the probability space of random 3-CNF formulas with n variables and m clauses generated uniformly and independently from the set of all possible clauses on exactly three distinct variables. The ratio m/n is denoted by Δ and is called the *clause density*. It is conjectured that the probability that a random formula F is unsatisfiable exhibits a phase transition from asymptotically 0 to asymptotically 1 around a critical clause density Δ_c [CR92]. As far as partial results are concerned, it has been proven that if $\Delta < 3.42$ then F is almost surely satisfiable [KKL02] and if $\Delta > 4.571$ then F is almost surely unsatisfiable [KKS$^+$01]. Moreover, experimental results suggest that Δ_c is around 4.2 [SML96], but the existence of such a constant is only a conjecture for the moment.

This structural analysis of the typical instance of the 3-CNF satisfiability problem is complemented by an analysis of the complexity of the known algorithms to solve it. The first results in this respect were only experimental showing that the running time of the Davis-Putnam-Logemann-Loveland procedure (DPLL) for satisfiability experiments a sharp jump around $\Delta \approx 4.2$, and decays slowly as the ratio Δ increases. These empirical observations are confirmed by two deep theoretical results. First, Broder, Frieze, and Upfal [BFU93] proved that if $\Delta < 1.63$ then a *simple* variant of DPLL running in linear time succeeds in finding a satisfying assignment with high probability. Second, Chvátal and Szeméredi [CS88], as improved by [BKPS98], proved that every Resolution refutation of an unsatisfiable F almost surely requires size $2^{\Omega(n/\Delta^5)}$. Thus, since a DPLL run on F provides a Resolution refutation when F is unsatisfiable no matter which heuristic is used for the splitting rule [BKPS98], this provides a proof that all DPLL-based algorithms behave badly in the unsatisfiable region with small Δ. The results of Broder, Frieze, and Upfal were later improved to values of Δ closer to 4.2 (see the nice survey by Achlioptas [Ach01]), and the results of Chvátal and Szeméredi were extended to other proof systems such as the Polynomial Calculus and Res(2) [BSI99, ABE02].

We remark that, at the time of writing, the known lower bounds on the satisfiability threshold Δ_c are algorithmic, while the known upper bounds are not. What this means is that for every density Δ for which it has been proven that a random formula is almost surely satisfiable, there actually is a polynomial-time algorithm that certifies this with probability bounded away from zero. The situation at the upper-end is quite different. All known constant upper bounds on Δ_c follow by a counting argument that does not provide with a polynomial-time algorithm that certifies unsatisfiability. In fact, the theoretical results of Chvátal and Szeméredi about Resolution, and their extension to other proof systems such as the Polynomial Calculus and Res(2), prove that every algorithm that implicitly or explicitly produces a refutation in such systems will fail to succeed in polynomial-time or with non-zero probability. Hence, these results suggest a stronger form of hardness for satisfiability. Indeed, it seems that for constant values of Δ beyond the critical point, certifying unsatisfiability on a significant fraction of the inputs is computationally hard. A definitive argument in its favor would be proving the *non-existence* of a property of 3-CNF formulas that (1) is decidable in polynomial-time, (2) it implies unsatisfiability, and (3) holds with probability bounded away from 0 when $\Delta > 4.2$ is constant. Obviously, this hypothesis implies $\mathbf{P} \neq \mathbf{NP}$, and therefore its proof seems out of reach with current methods. In any case, it is possible to obtain some partial unconditional results of this type, and it is clear that they provide some valuable information on our problem.

In Chapter 5 we initiate a study of the descriptive complexity of properties that imply unsatisfiability with non-zero probability. We show the *non-existence* of a property of 3-CNF formulas that (1) is expressible in Datalog when 3-CNF formulas are encoded as finite relational structures, (2) it implies unsatisfiability, and (3) holds with probability bounded away from 0 when Δ is constant. Datalog is a well-known query language studied in the context of database theory (see Ullman [Ull89]), capable of expressing many interesting problems including Graph Reachability, Non-2-Colorability, Unsatisfiability of 2-CNFs, and some $\mathbf{P}$-complete problems such as the Monotone Circuit Value Problem and Path Systems among others (see [KV00a, Pap85, Ull89]). In a nutshell, Datalog may be viewed as the existential positive fragment of first-order logic augmented with a mechanism of recursion. In fact, Chandra and Harel [CH82] proved that Datalog coincides with the existential positive fragment $\exists\mathrm{LFP}$ of LFP. Let us add that Datalog captures a significant but proper fraction of polynomial-time (see [Ull89] again), and in the presence of a successor relation and negations on the given predicates, it captures polynomial-time exactly [Pap85].

We point out that not all $\mathbf{NP}$-complete problems and all distributions on the instances need to be hard to certify in the sense above. For example, it is know that a random graph taken from the uniform distribution on all graphs on n nodes is almost surely non-3-colorable. Nonetheless, the property that such a graph contains a complete subgraph on four nodes is certainly (1) expressible in Datalog, (2) it implies non-3-colorability, and (3) holds with non-zero probability over the uniform distribution. In fact, the property is expressible in positive existential first-order logic, and holds with asymptotic probability one, as it is easy to see.

The main ingredient of our proof is the design of a winning strategy for the Duplicator in the *existential k-pebble game* between the structure encoding a randomly generated formula, and the structure encoding a fixed template formula that we know is satisfiable. The existential k-pebble game is a powerful combinatorial tool introduced by Kolaitis and Vardi [KV95, KV00a] to analyze the expressive power of Datalog with k variables. In fact, the existential k-pebble game captures the expressive power of the stronger logic $\exists\mathcal{L}^k_{\infty\omega}$, the existential positive fragment of infinitary logic with k variables. Thus, our lower bound is much stronger than stated above since what we actually prove is that every property that implies unsatisfiability of formulas with n variables and is (non-uniformly) expressible in the infinitary logic $\exists\mathcal{L}^k_{\infty\omega}$ when $k \leq n/(\ln(n))^2$ must have asymptotic probability 0 when $n \rightarrow \infty$ and Δ is constant. Note that we allow the number of variables k to grow with the number of propositional variables n, a parameter that is not allowed to vary in a uniform Datalog program.

The design of the winning strategy for the Duplicator relies on the fact that the *incidence graph* of a random 3-CNF formula, the bipartite graph that relates clauses with the variables that occur in them, satisfies certain extension axioms that we introduce based on the expander properties of the graph. The use of extension axioms is a recurring theme in proving 0-1 laws for several logics and distributions on the instances [Fag76, Lyn80, SS88, KV90]. Moreover, the expander properties of these graphs have also been used in the context of propositional proof complexity to prove lower bounds in the size of Resolution refutations [BSW01, BSG01]. This establishes an interesting link between the fields of propositional proof complexity and finite model theory that we discuss next.

1.2.4 Consequences for Proof Complexity and Constraint Satisfaction

Propositional proof complexity is devoted to the study of the length of proofs in propositional proof systems. Unquestionably, certain tautologies are more obvious than others, and this is reflected by the length of their proofs in a fixed proof system. In addition, the study of the lengths of proofs yields significant insight into the combinatorial and computational content of the tautology. The field started with the pioneering work of Cook and Reckhow [CR79], and has been developing continuously since then.

One of the well-studied proof systems is propositional Resolution, introduced by Robinson [Rob65]. Strictly speaking, Resolution is a refutational system, which means that it provides proofs of contradiction of unsatisfiable CNF formulas rather than proofs of tautologies. However, it is well known that both frameworks are easily translatable one into the other without much effort. The first significant lower bounds in propositional proof complexity were for Resolution. Indeed, Haken [Hak85] showed that the propositional encoding of the Pigeonhole Principle proposed by Cook and Reckhow requires Resolution refutations of exponential size. Later on, Chvátal and Szeméredi [CS88] proved that a 3-CNF formula randomly generated from $\mathcal{F}(n, \Delta n, 3)$ requires exponential-size Resolution refutations almost surely.

Our results of Chapter 5 establish a link between finite model theory and propositional proof complexity. As a matter of fact, the idea of playing existential k-pebble games on random 3-CNF formulas came out of the following informal reasoning: if a certain formula is hard to refute in Resolution, shouldn't it be hard to tell apart from a satisfiable formula by a bounded player that confronts an adversary?

In order to make this argument more explicit, we elaborate on the idea of using the encoding of combinatorial principles as propositional tautologies to obtain finite structures on which playing pebble games might be easier than expected. For example, we consider the encoding of the Pigeonhole Principle as an unsatisfiable CNF formula in the spirit of Cook and Reckhow, and show that the Duplicator wins the existential pebble game on the finite structure encoding this formula and a template structure encoding a satisfiable formula. What this result is saying is that the Pigeonhole Principle is a hard example for all Datalog programs that attempt to solve unsatisfiability. This phrasing is inspired by the results of Kolaitis and Vardi relating the Constraint Satisfaction Problems that are solvable by k-Datalog programs, and the finite structures on which the Duplicator wins the k-pebble game [KV00b]. We will elaborate a little bit more on this later.

More importantly perhaps, we also observe that our main inexpressibility result for $\exists\mathcal{L}^k_{\infty\omega}$ is a generalization of the result of Chvátal and Szeméredi. Before we explain why this is so, we need to digress shortly to introduce the concept of width in Resolution. In a Resolution refutation, the width is the maximal number of literals in a clause of the refutation. Ben-Sasson and Wigderson [BSW01] proved the following non-trivial result about the width of Resolution: If a 3-CNF formula F has a Resolution refutation of size S, then F has a Resolution refutation of width $O\left(\sqrt{n \log S}\right)$, where n is the number of propositional variables. This interesting result relates the width with the size in a form that is suitable to prove size lower bounds.

Indeed, if every Resolution refutation of F requires width w, then every Resolution refutation of F requires size $2^{\Omega(w^2/n)}$. Thus, lower bounds on width imply lower bounds on size. It is worth noting that a conjecture about a trade-off of this type was formulated already in 1981 [KM81]. What we realize is that there is a tight connection between the width of Resolution, and the number of pebbles needed to win the existential k-pebble game. We show the existence of a Datalog program P with $2k$ variables such that (1) if F has a Resolution refutation of width k, then P evaluates true on the structure encoding F, and (2) if P evaluates true on the structure encoding F, then F has a Resolution refutation of width $2k$. Thus, if the Duplicator wins the existential k-pebble game on the structure encoding F and the template structure, then every Resolution refutation of F requires width $k/2$. An immediate corollary of our main result about the Duplicator winning the $n/(\ln n)^2$-pebble game on a random 3-CNF formula F, and the size-width trade-off of [BSW01], is that every Resolution refutation of F requires size $2^{\Omega(n/(\ln n)^4)}$. The same argument can be used to obtain Haken's exponential lower bound for the Pigeonhole Principle. It should be pointed out that Ben-Sasson and Wigderson prove similar lower bounds by means of the width method in a more direct way; nonetheless, it is quite interesting that non-expressibility results in finite model theory yield lower bounds in proof complexity.

We turn next to a discussion about the implications of our results for the complexity of algorithms for constraint satisfaction problems. It was first pointed out by Feder and Vardi [FV98] that every constraint satisfaction problem can be cast as a homomorphism problem: given two finite structures $\mathbf{A}$ and $\mathbf{B}$ over the same relational vocabulary, is there a homomorphism from $\mathbf{A}$ to $\mathbf{B}$? Intuitively, $\mathbf{A}$ represents the variables and the tuples of constraint variables, $\mathbf{B}$ represents the values and the constraints, and the homomorphisms between $\mathbf{A}$ and $\mathbf{B}$ are the solutions to the constraint satisfaction problem. Note that satisfiability problems fit well in this framework by letting the universe of $\mathbf{B}$ be $\{0, 1\}$ and the relations of $\mathbf{B}$ be the truth-tables of the involved connectives.

Several concepts of *consistency* have been popularized by the AI community as powerful algorithmic tools to deal with the intractability of constraint satisfaction problems. The underlying common idea of all these concepts is that of identifying some structural property on the instances that guarantees success of a fixed constraint propagation algorithm in reasonable time. All these algorithms start with the initial constraints of the problem, and propagate them by inferring new and possibly tighter constraints, until the problem is either solved, or proved unsatisfiable.

Kolaitis and Vardi [KV00b] shed light on the connections between the concepts of consistency and the concept of winning strategy for the Duplicator in the existential k-pebble game. These seemingly unrelated concepts turned out to be tightly connected through a theorem stating that a constraint satisfaction problem given by the structures $\mathbf{A}$ and $\mathbf{B}$ can be solved by *establishing strong k-consistency* if and only if the Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$. As a matter of fact, what Kolaitis and Vardi pointed out is that solvability by the usual constraint propagation algorithms can be cast, in one form or another, as conditions on winning strategies for the Duplicator. With this interesting re-interpretation, our result about the Duplicator winning the existential k-pebble game on a random 3-CNF formula admits the following intuitive interpretation: any usual constraint propagation algorithm that attempts to solve satisfiability with small constraints will succeed on a negligible fraction of all 3-CNF formulas only. We note that resolution is only one very particular form of constraint propagation.

Chapter 2

Preliminaries

2.1 Languages and Structures

A finite relational *vocabulary* with constants, or *language*, is a finite set of relation and constant symbols $L = \{R_1, \ldots, R_s, c_1, \ldots, c_r\}$. Each relation symbol R in L has an associated natural number called its *arity*. Since our vocabularies will always be finite, sometimes we omit this term. If the vocabulary has constants, we usually indicate that explicitly.

Let $L = \{R_1, \ldots, R_s, c_1, \ldots, c_r\}$ be a relational vocabulary with constants. A *structure* over L, or an L*-structure*, is a tuple

$$\mathbf{M} = (M, R_1^{\mathbf{M}}, \ldots, R_s^{\mathbf{M}}, c_1^{\mathbf{M}}, \ldots, c_r^{\mathbf{M}}),$$

where M is a non-empty set called the *universe*, each $R_i^{\mathbf{M}}$ is a relation over M of the arity r_i of R_i, and each $c_i^{\mathbf{M}}$ is an element of M. We say that $R_i^{\mathbf{M}}$ and $c_i^{\mathbf{M}}$ are the *interpretations* of the symbols R_i and c_i respectively.

Example 1 Essentially all mathematical objects can be represented as structures for an appropriate vocabulary. We see two examples: graphs and groups. Let $L = \{E\}$ be the relational vocabulary of a single binary relation symbol E, also known as the vocabulary of graphs. An L-structure $\mathbf{M} = (M, E^{\mathbf{M}})$ is called a *graph* if $E^{\mathbf{M}}$ is a symmetric, non-reflexive binary relation over M. We say that M is the set of vertices of the graph, and $E^{\mathbf{M}}$ is the set of edges. Let $L = \{P, e\}$ be the relational vocabulary of one relation symbol P of arity three, and one constant symbol e, also known as the relational vocabulary of groups. An L-structure $\mathbf{M} = (M, P^{\mathbf{M}}, e^{\mathbf{M}})$ is called a *group* if $P^{\mathbf{M}}$ is the graph of a binary operation $\circ : M \times M \to M$ that is associative, has $e^{\mathbf{M}}$ as its neutral element, and every element of M has an inverse with respect to $\circ$.

The universe of a structure $\mathbf{M}$ is always denoted by M. If the structure is called $\mathbf{N}$, its universe is denoted by N, and similarly for other names. The interpretation of R_i or c_i in $\mathbf{M}$ is always denoted by $R_i^{\mathbf{M}}$ or $c_i^{\mathbf{M}}$, respectively. We say that $\mathbf{M}$ is a substructure of $\mathbf{N}$, denoted by $\mathbf{M} \subseteq \mathbf{N}$, if (1) $M \subseteq N$, (2) $R_i^{\mathbf{M}} = R_i^{\mathbf{N}} \cap M^{r_i}$ for every R_i in L, and (3) $c_i^{\mathbf{M}} = c_i^{\mathbf{N}}$ for every c_i in L.

Two structures $\mathbf{M}$ and $\mathbf{N}$ over the same vocabulary that are identical up to renaming of the elements of the universe are called *isomorphic*. Formally:

Definition 1 *Let L be a relational vocabulary with constants. We say that two L-structures $\mathbf{M}$ and $\mathbf{N}$ are isomorphic, denoted by $\mathbf{M} \cong \mathbf{N}$, if there exists a mapping $f : M \to N$ that is one-to-one and onto, and such that $(a_1, \ldots, a_{r_i}) \in R_i^{\mathbf{M}}$ if and only if $(f(a_1), \ldots, f(a_{r_i})) \in R_i^{\mathbf{N}}$ for every $a_1, \ldots, a_{r_i} \in M$ and $R_i \in L$, and $f(c_i^{\mathbf{M}}) = c_i^{\mathbf{N}}$ for every $c_i \in L$.*

2.2 First-Order Logic

We need a precise language to express properties of structures, such as those expressed in English in Example 1. First-order logic is a convenient formalism for that.

Let $\mathcal{V} = \{v_1, v_2, \ldots\}$ be an infinite but countable set of *first-order variables*, and let L be a relational vocabulary with constants. The set of *atomic formulas of* L is the set of formulas of the form $R(t_1, \ldots, t_r)$ or $t_1 = t_2$, where R is an r-ary relation symbol in L, and each t_i is a first-order variable or a constant symbol c in L. The set of *first-order formulas of* L is defined by closing the set of atomic formulas under negations, conjunctions, and universal quantification.

Definition 2 *Let L be a relational language with constants. The set of first-order formulas of L is the smallest set of formulas that is closed under the following rules:*

(i) if φ is an atomic formula of L, then φ is a formula,

(ii) if φ is a formula, then $\neg\varphi$ is also a formula,

(iii) if ψ and θ are formulas, then $(\psi \wedge \theta)$ is also a formula,

(iv) if φ is a formula and x is a first-order variable, then $(\forall x)(\varphi)$ is a formula.

We use $(\psi \vee \theta)$ as an abbreviation for $\neg(\neg\psi \wedge \neg\theta)$. Also, we use $(\psi \rightarrow \theta)$ as an abbreviation for $\neg(\psi \wedge \neg\theta)$, and $(\exists x)(\varphi)$ as an abbreviation for $\neg(\forall x)(\neg\varphi)$. It will be convenient to avoid unnecessary parenthesis. In order to do that, we view $\wedge$ and $\vee$ as associative symbols, and give them more precedence that $\rightarrow$. Also, the nesting of quantifiers will be discharged of redundant parenthesis. The following examples will help:

Example 2 We write $\varphi \wedge \psi \wedge \theta$ instead of $((\varphi \wedge \psi) \wedge \theta)$ or $(\varphi \wedge (\psi \wedge \theta))$. We write $\varphi \wedge \psi \rightarrow \theta$ instead of $((\varphi \wedge \psi) \rightarrow \theta)$. Similarly, we write $\varphi \rightarrow \psi \wedge \theta$ instead of $(\varphi \rightarrow (\psi \wedge \theta))$. Also, we write $(\forall x)(\exists y)(\varphi \wedge \psi)$ instead of $(\forall x)((\exists y)((\varphi \wedge \psi)))$. Let us note that the occurrences of negation $\neg$ are never ambiguous: they always apply to the next formula. Thus, $\neg\varphi \vee \psi$ is never thought as $\neg(\varphi \vee \psi)$. If we want to write that, we need to use the parenthetical form.

The set of *free variables* of a first-order formula φ, denoted by $F(\varphi)$, is defined inductively as follows: if φ is atomic of the form $R(t_1, \ldots, t_r)$ or $t_1 = t_2$, then $F(\varphi) = \{t_i : t_i \text{ is a first-order variable}\}$. If $\varphi = \neg\psi$, then $F(\varphi) = F(\psi)$. If $\varphi = (\psi \wedge \theta)$, then $F(\varphi) = F(\psi) \cup F(\theta)$. If $\varphi = (\forall x)(\psi)$, then $F(\varphi) = F(\psi) - \{x\}$. If $x \in F(\varphi)$, we say that x *occurs free* in φ. If $F(\varphi) = \emptyset$, we say that φ is a *sentence*. The notation $\varphi(x_1, \ldots, x_n)$ will mean that all free variables of φ are contained in $\{x_1, \ldots, x_n\}$. Note that we do not insist that every variable in $\{x_1, \ldots, x_n\}$ is actually a free variable of φ. However, when we write $\varphi(x_1, \ldots, x_n)$, that is usually the case.

We are ready to define the semantics of first-order formulas on structures. Let L be a relational vocabulary with constants, and let $\mathbf{M}$ be an L-structure. A *valuation function* is a mapping f from the set of first-order variables $\mathcal{V}$ to the universe of $\mathbf{M}$. It is convenient to extend f to the constant symbols of L by setting $f(c) = c^{\mathbf{M}}$ for every constant symbol c in L. We say that f is a valuation function for $\mathbf{M}$.

Definition 3 *Let L be a relational language with constants, let $\mathbf{M}$ be an L-structure, and let f be a valuation function for $\mathbf{M}$. Let φ be a first-order formula of L with free variable $x_1, \ldots, x_n$. We say that $\mathbf{M}$ and f satisfy φ, denoted by $\mathbf{M} \models \varphi[f]$, if one of the following* *cases holds:*

(i) $\varphi = R(t_1, \ldots, t_r)$ *for some* $R \in L$ *and* $(f(t_1), \ldots, f(t_r)) \in R^{\mathbf{M}}$*,*

(ii) $\varphi = t_1 = t_2$ *and* $f(t_1) = f(t_2)$*,*

(iii) $\varphi = \neg\psi$ *and not* $\mathbf{M} \models \psi[f]$*,*

(iv) $\varphi = \psi \wedge \theta$ *and* $\mathbf{M} \models \psi[f]$ *and* $\mathbf{M} \models \theta[f]$*,*

(v) $\varphi = (\forall x)(\psi)$ *and* $\mathbf{M} \models \psi[f_a^x]$ *for every* $a \in M$*,*

where f_a^x *is the valuation function that sets* $f_a^x(x) = a$*, and* $f_a^x(t) = f(t)$ *for every* $t \neq x$*.*

Let $\varphi(x_1, \ldots, x_n)$ be a first-order formula of L, let $\mathbf{M}$ be an L-structure, and let $a_1, \ldots, a_n$ be elements of M. The notation

$$\mathbf{M} \models \varphi[a_1, \ldots, a_n]$$

will be used to mean that φ is satisfied by $\mathbf{M}$ and any valuation function f for $\mathbf{M}$ such that $f(x_i) = a_i$ for every $i \in \{1, \ldots, n\}$. In particular, if φ is a sentence, we say that φ holds in $\mathbf{M}$, denoted by $\mathbf{M} \models \varphi$, if $\mathbf{M} \models \varphi[f]$ for every valuation function f for $\mathbf{M}$.

Example 3 We revisit the examples of graphs and groups of Example 1. Our goal is to express the axioms of graphs and groups in first-order logic. Let $L = \{E\}$ be the vocabulary of graphs, and let $\mathbf{M}$ be an L-structure. Let

$$\begin{aligned} \varphi_1 &= (\forall x)(\neg E(x, x)) \\ \varphi_2 &= (\forall x)(\forall y)(E(x, y) \rightarrow E(y, x)). \end{aligned}$$

If the sentences φ_1 and φ_2 hold in $\mathbf{M}$, then $\mathbf{M}$ is a graph. Indeed, φ_1 expresses the fact that E is anti-reflexive, and φ_2 expresses the fact that E is symmetric. Let $L = \{P, e\}$ be the relational vocabulary of groups, and let $\mathbf{M}$ be an L-structure. Let

$$\begin{aligned} \psi_1 &= (\forall x_1)(\forall x_2)(\exists x_3)(P(x_1, x_2, x_3)) \\ \psi_2 &= (\forall x_1)(\forall x_2)(\forall x_3)(\forall x_4)(P(x_1, x_2, x_3) \wedge P(x_1, x_2, x_4) \rightarrow x_3 = x_4). \end{aligned}$$

If ψ_1 and ψ_2 hold in $\mathbf{M}$, then $P^{\mathbf{M}}$ is the graph of a binary operation $\circ : M \times M \rightarrow M$. Indeed, ψ_1 expresses the fact that every pair (x_1, x_2) has an image x_3, and ψ_2 expresses the fact that this image is unique. Let

$$\begin{aligned} \psi_3 = {} & (\forall x_1)(\forall x_2)(\forall x_3)(\forall x_4)(\forall x_5)(\forall x_6)(\forall x_7)(P(x_1, x_2, x_4) \wedge P(x_2, x_3, x_5) \wedge \\ & \wedge P(x_1, x_5, x_6) \wedge P(x_4, x_3, x_7) \rightarrow x_6 = x_7). \end{aligned}$$

If ψ_3 also holds in $\mathbf{M}$, then $\circ$ is an associative operation. Indeed, ψ_3 expresses the fact that $(x_1 \circ x_2) \circ x_3 = x_1 \circ (x_2 \circ x_3)$ for every x_1, x_2 and x_3. Let

$$\begin{aligned} \psi_4 &= (\forall x)(P(x, e, x) \wedge P(e, x, x)) \\ \psi_5 &= (\forall x)(\exists y)(P(x, y, e) \wedge P(y, x, e)). \end{aligned}$$

If ψ_4 and ψ_5 also hold in $\mathbf{M}$, then $\mathbf{M}$ is a group. Indeed, ψ_4 expresses the fact that $e^{\mathbf{M}}$ is the neutral element of $\circ$, and ψ_5 expresses the fact that every element has an inverse with respect to $\circ$. Thus, $\psi_1, \ldots, \psi_5$ are the axioms of group theory (for the relational vocabulary of groups).

A first-order sentence φ expresses a property that may, or may not hold in a structure $\mathbf{M}$. A first-order formula with free variables $\varphi(x_1, \ldots, x_n)$, instead, defines a relation over the universe of the structure. Namely, it defines the set of tuples of the universe that satisfy the formula when they interpret the free variables of the formula. More formally, we define

$$\varphi^{\mathbf{M}} = \{(a_1, \ldots, a_n) \in M^n : \mathbf{M} \models \varphi[a_1, \ldots, a_n]\}.$$

In particular, if $\varphi(x)$ has a unique free variable x, then $\varphi^{\mathbf{M}}$ is a subset of M.

Example 4 Let $L = \{P, e\}$ be the relational vocabulary of groups, and let $\mathbf{M}$ be a group, that is, an L-structure that satisfies the axioms $\psi_1, \ldots, \psi_5$ in Example 3. Let

$$\varphi(x) \quad = \quad (\forall y)(\forall z_1)(\forall z_2)(P(x, y, z_1) \wedge P(y, x, z_2) \rightarrow z_1 = z_2).$$

Then, $\varphi^{\mathbf{M}}$ is the set of elements of the group that commute with every other element. Indeed, the formula says that $x \circ y = y \circ x$ for every y. Although we will not need this, let us say that the set $\varphi^{\mathbf{M}}$ is usually called the *center* of the group. Thus, we say that the formula $\varphi(x)$ defines the center of the group. Observe that if the center coincides with the whole group M, then all elements commute, which means that $\mathbf{M}$ is an *Abelian group*. Thus, if we add the sentence $(\forall x)(\varphi(x))$ to $\psi_1, \ldots, \psi_5$ we obtain the axioms of Abelian groups.

2.3 Least Fixed-Point Logic

We introduced first-order logic as a convenient language to express properties of structures. In this section we introduce least fixed-point logic, which is an extension of first-order logic to incorporate recursion through inductive definitions.

Let $\mathcal{S} = \{V_1, V_2, \ldots\}$ be an infinite but countable set of *second-order variables*. Each second-order variable has an associated number called its arity. Moreover, we assume that $\mathcal{S}$ contains infinitely many second-order variables of each arity. Let L be a relational vocabulary with constants, and let X be a second-order variable. We treat X as a relation symbol for the purposes of defining formulas. We define the class of *first-order formulas of* $L \cup \{X\}$ *that are positive in* X:

Definition 4 *Let L be a relational vocabulary with constants and let X be a second-order variable of arity r. The class of first-order formulas of $L \cup \{X\}$ that are positive in X, or positive formulas, is the smallest class of formulas that is closed under the following rules:*

(i) each atomic formula of the form $X(t_1, \ldots, t_r)$ is a positive formula,

(ii) if φ is a first-order formula of L, then φ is also a positive formula,

(iii) if ψ and θ are positive formulas, then $\psi \wedge \theta$ and $\psi \vee \theta$ are also positive formulas,

(iv) if φ is a positive formula, then $(\forall x)(\varphi)$ and $(\exists x)(\varphi)$ are also positive formulas.

We are ready to define the class of *least fixed-point formulas of* L:

Definition 5 *Let L be a relational vocabulary with constants. The class of least fixed-point formulas of L is the set of formulas of the form*

$$(\mathrm{LFP}_{x_1,\ldots,x_r,X}\varphi)(t_1, \ldots, t_r),$$

where $\varphi(x_1, \ldots, x_r, X)$ is a first-order formula of $L \cup \{X\}$ that is positive in r-ary second-order variable X, and each t_i is either a first-order variable or a constant symbol of L.

The set of free variables of a least fixed-point formula $(\mathrm{LFP}_{x_1,\ldots,x_r,X}\varphi)(t_1, \ldots, t_r)$ is defined to be the set of first-order variables in $\{t_1, \ldots, t_r\}$. Note that the only free variables of φ are among $\{x_1, \ldots, x_r\}$. Note also that the least fixed-point formulas, the way we defined them, do not allow nesting of fixed-point operators nor first-order parameters. It is known that this forms a robust class of formulas (see [Mos74]).

We turn next to the semantics. Let L be a relational vocabulary with constants, let $\mathbf{M}$ be an L-structure, and let X be a second-order variable of arity r. Observe that every first-order formula $\varphi(x_1, \ldots, x_r, X)$ of $L \cup \{X\}$ defines an operator $F_\varphi^{\mathbf{M}} : M^r \to M^r$ defined as follows

$$F_\varphi^{\mathbf{M}}(A) = \{(a_1, \ldots, a_r) \in M^r : \mathbf{M} \models \varphi[a_1, \ldots, a_r, A]\}.$$

Here, the notation $\mathbf{M} \models \varphi[a_1, \ldots, a_r, A]$ means that $(\mathbf{M}, A) \models \varphi[a_1, \ldots, a_r]$, where $(\mathbf{M}, A)$ denotes the $L \cup \{X\}$-structure that expands $\mathbf{M}$ by interpreting X by A.

Proposition 1 [Mos74] *Let L be a relational vocabulary with constants, and let X be a second-order variable. If φ is a first-order formula of $L \cup \{X\}$ that is positive in X, then $F_\varphi^{\mathbf{M}}$ is a monotone operator. That is, if $A \subseteq B \subseteq M^r$, then $F_\varphi^{\mathbf{M}}(A) \subseteq F_\varphi^{\mathbf{M}}(B)$.*

Proof: The proof is by induction on the construction of φ. If φ is atomic of the form $X(t_1, \ldots, t_r)$, then $\mathbf{M} \models \varphi[a_1, \ldots, a_r, A]$ implies $\mathbf{M} \models \varphi[a_1, \ldots, a_r, B]$ whenever $A \subseteq B$. Thus, $F_\varphi^{\mathbf{M}}(A) \subseteq F_\varphi^{\mathbf{M}}(B)$. The inductive cases are as easy. □

Let $\varphi(x_1, \ldots, x_r, X)$ be a first-order formula of $L \cup \{X\}$ that is positive in X. For every ordinal α, let

$$I_\varphi^\alpha(\mathbf{M}) = F_\varphi^{\mathbf{M}}\left(\bigcup_{\beta<\alpha} I_\varphi^\beta(\mathbf{M})\right).$$

Since $F_\varphi^{\mathbf{M}}$ is a monotone operator, we have that $I_\varphi^\beta(\mathbf{M}) \subseteq I_\varphi^\alpha(\mathbf{M})$ whenever $\beta \leq \alpha$. By a cardinality argument, there must exist a minimal ordinal γ such that $I_\varphi^\gamma(\mathbf{M}) = I_\varphi^{\gamma+1}(\mathbf{M})$.

Such an ordinal is called *the closure ordinal* of φ on $\mathbf{M}$, and is denoted by $\mathrm{cl}_{\varphi}(\mathbf{M})$. Moreover, we let $I_{\varphi}(\mathbf{M}) = I_{\varphi}^{\gamma}(\mathbf{M})$. We note that

$$F_{\varphi}^{\mathbf{M}}(I_{\varphi}(\mathbf{M})) = I_{\varphi}(\mathbf{M}).$$

Thus, $I_{\varphi}(\mathbf{M})$ is a fixed-point of the operator defined by φ. Moreover, it can be seen that $I_{\varphi}(\mathbf{M})$ is the *least fixed-point* of $F_{\varphi}^{\mathbf{M}}$. This means that if A is another fixed-point of $F_{\varphi}^{\mathbf{M}}$, that is, if $F_{\varphi}^{\mathbf{M}}(A) = A$, then $I_{\varphi}(\mathbf{M}) \subseteq A$ (see [Mos74]).

Definition 6 *Let L be a relational vocabulary with constants, let $\mathbf{M}$ be an L-structure, and let f be a valuation function for $\mathbf{M}$. Let $\varphi = (\mathrm{LFP}_{x_1,\ldots,x_r,X}\psi)(t_1,\ldots,t_r)$ be a least fixed-point formula of L. We say that $\mathbf{M}$ and f satisfy φ, denoted by $\mathbf{M} \models \varphi[f]$, if the tuple $(f(t_1),\ldots,f(t_r))$ belongs to the least fixed-point $I_{\psi}(\mathbf{M})$ of the monotone operator defined by ψ.*

The following example borrowed from [KL00] illustrates the expressive power of least fixed-point formulas.

Example 5 Let $L = \{P, e\}$ be the relational vocabulary of groups, and let $\mathbf{M}$ be a finite Abelian group. That is, $\mathbf{M}$ is a finite L-structure that satisfies the axioms of Abelian groups presented in Examples 3 and 4. Let $\varphi(x, y)$ be the following least fixed-point formula with two variables:

$$(\mathrm{LFP}_{x,y,X}(y = e \vee (\exists z)(X(x,z) \wedge P(z,x,y))))(x,y).$$

It is not hard to see that, on $\mathbf{M}$, the formula $\varphi(x, y)$ defines the set of pairs of the form

$$\{(a,b) \in M : b \text{ is a power of } a\}.$$

Indeed, the i-th stage $I^i_\varphi(\mathbf{M})$ is exactly the set $\{(a, a^i) : a \in M\}$. Now, it is known from basic group theory, that an Abelian finite group is *simple* if and only if every non-trivial element generates the whole group. Thus, the sentence

$$(\forall x)(x \neq e \to (\forall y)(\varphi(x, y)))$$

expresses the property that $\mathbf{M}$ is a simple Abelian group. Note that this sentence is not a least fixed-point sentence in the sense of Definition 5. However, its meaning is clear when we interpret least fixed-points formulas as a mechanism of building up new relations.

Just as monotone operators have least fixed-points, they also have greatest fixed-points. For every ordinal φ, let

$$J^\alpha_\varphi(\mathbf{M}) = F^{\mathbf{M}}_\varphi\left(\bigcap_{\beta<\alpha} J^\beta_\varphi(\mathbf{M})\right).$$

Here, the empty intersection is the universe of $\mathbf{M}$ by convention. Since $F^{\mathbf{M}}_\varphi$ is a monotone operator, we have $J^\beta_\varphi(\mathbf{M}) \supseteq J^\alpha_\varphi(\mathbf{M})$ whenever $\beta \leq \alpha$. Note that the inclusion symbol is reversed. By a cardinality argument again, there must exist a γ such that $J^\gamma_\varphi(\mathbf{M}) = J^{\gamma+1}_\varphi(\mathbf{M})$. We let $J_\varphi(\mathbf{M}) = J^\gamma_\varphi(\mathbf{M})$ and observe that

$$F^{\mathbf{M}}_\varphi(J_\varphi(\mathbf{M})) = J_\varphi(\mathbf{M}).$$

Thus, $J_\varphi(\mathbf{M})$ is a fixed-point of $F^{\mathbf{M}}_\varphi$, and in fact, it is the greatest fixed-point. That is, if A is another fixed-point $F^{\mathbf{M}}(A) = A$, then $A \subseteq J_\varphi(\mathbf{M})$.

It is known that the least fixed-point and the greatest fixed-point of a monotone operator are related. The next proposition states how. We include its proof because we will use this fact in Chapter 3, and because we think that it is instructive.

Proposition 2 [Mos74] *Let L be a relational vocabulary with constants, let $\mathbf{M}$ be an L-structure, and let $\varphi(x_1, \ldots, x_r, X)$ be a first-order formula of $L \cup \{X\}$ that is positive in the r-ary second-order variable X. Then, $I_\varphi(\mathbf{M}) = M^r - J_\psi(\mathbf{M})$, where ψ is the dual first-order formula of φ, that is, $\psi = \neg\varphi(x_1, \ldots, x_r, X/\neg X)$. Observe that ψ is again positive in X.*

Proof: We prove, by induction on the ordinals, that $I^\alpha_\varphi(\mathbf{M}) = M^r - J^\alpha_\psi(\mathbf{M})$. By definition, $(a_1, \ldots, a_r) \in I^\alpha_\varphi(\mathbf{M})$ if and only if

$$\mathbf{M} \models \varphi[a_1, \ldots, a_r, \bigcup_{\beta<\alpha} I^\beta_\varphi(\mathbf{M})].$$

By induction hypothesis on β, this is equivalent to

$$\mathbf{M} \models \varphi[a_1, \ldots, a_r, \bigcup_{\beta<\alpha} (M^r - J^\beta_\psi(\mathbf{M}))].$$

Recall that $\psi = \neg\varphi(x_1, \ldots, x_r, X/\neg X)$. Therefore, this is equivalent to

$$\mathbf{M} \not\models \psi[a_1, \ldots, a_r, \bigcap_{\beta<\alpha} J^\beta_\psi(\mathbf{M})].$$

By definition, this is equivalent to $(a_1, \ldots, a_r) \notin J^\alpha_\psi(\mathbf{M})$, which means that $(a_1, \ldots, a_r) \in M^r - J^\alpha_\psi(\mathbf{M})$. This completes the proof by induction that $I^\alpha_\varphi(\mathbf{M}) = M^r - J^\alpha_\varphi(\mathbf{M})$. Now, for the conclusion, observe that if $(a_1, \ldots, a_r) \in I_\varphi(\mathbf{M})$, then $(a_1, \ldots, a_r) \in I^\alpha_\varphi(\mathbf{M})$ for some ordinal α. Therefore, $(a_1, \ldots, a_r) \notin J^\alpha_\psi(\mathbf{M})$, and so $(a_1, \ldots, a_r) \notin J_\psi(\mathbf{M})$ since $J^\alpha_\psi(\mathbf{M}) \supseteq J_\psi(\mathbf{M})$. Conversely, if $(a_1, \ldots, a_r) \notin J_\psi(\mathbf{M})$, then $(a_1, \ldots, a_r) \notin J^\alpha_\psi(\mathbf{M})$ for some α, which means that $(a_1, \ldots, a_r) \in I^\alpha_\varphi(\mathbf{M})$, and so $(a_1, \ldots, a_r) \in I_\varphi(\mathbf{M})$ since $I^\alpha_\varphi(\mathbf{M}) \subseteq I_\varphi(\mathbf{M})$. This concludes the proof. □

Let us conclude this section with the following well-known observation. Let $\mathbf{M}$ be a finite L-structure, and let $\varphi(x_1, \ldots, x_r, X)$ be a first-order formula of $L \cup \{X\}$ that is positive in the r-ary relation symbol X. Then the closure ordinal $\mathrm{cl}_{\varphi}(\mathbf{M})$ of φ on $\mathbf{M}$ is bounded by $|M|^r$. Indeed, the sequence

$$I_{\varphi}^{0}(\mathbf{M}) \subseteq I_{\varphi}^{1}(\mathbf{M}) \subseteq \cdots \subseteq I_{\varphi}^{\gamma}(\mathbf{M})$$

cannot last more than $|M|^r$ steps without reaching a fixed-point because each $I_{\varphi}^{\alpha}(\mathbf{M})$ is a subset of M^r. Thus, on finite structures, the closure ordinals of positive formulas are bounded by a polynomial in the size of the structure.

2.4 Uniform Definability and Queries

We mentioned already that formulas with free-variables define relations on structures. We will need the more general concept of *query* defined next:

Definition 7 *Let L be a relational vocabulary with constants, let C be a class of L-structures, let r be a natural number, and for every $\mathbf{M} \in C$ let $Q(\mathbf{M})$ be a subset of M^r. We say that $Q = \{(\mathbf{M}, Q(\mathbf{M})) : \mathbf{M} \in C\}$ is an r-ary query on C if for every $\mathbf{M}, \mathbf{N} \in C$ such that $\mathbf{M} \cong \mathbf{N}$ we have that $(\mathbf{M}, Q(\mathbf{M})) \cong (\mathbf{N}, Q(\mathbf{N}))$. If $r = 0$, we say that Q is a Boolean query.*

Boolean queries are identified with subsets of C in the obvious way; namely, if Q is a Boolean query on C, we write $\mathbf{M} \in Q$ or $\mathbf{M} \notin Q$ to mean that $Q(\mathbf{M}) = \{()\}$ or $Q(\mathbf{M}) = \emptyset$ respectively. Sometimes we say that a query is a *uniform relation*. In words, an r-ary query is a map from structures to r-ary relations on those structures, that is closed under isomorphisms. This closure property is required for queries to be definable by logics in the following sense:

Definition 8 *Let L be a relational vocabulary with constants, let C be a class of L-structures, let Q be an r-ary query on C, and let F be a class of formulas of L. We say that Q is F-definable on C if and only if there exists a formula $\varphi \in F$ with r free variables such that $Q(\mathbf{M}) = \varphi^{\mathbf{M}}$ for every $\mathbf{M} \in C$.*

We illustrate this definition with our running example of groups.

Example 6 Let $L = \{P, e\}$ be the relational vocabulary of groups, and let C be the class of all finite groups, that is, the set of all finite L-structures that satisfy all axioms of group theory $\psi_1, \ldots, \psi_5$ in Example 3. Let Q be the unary query that assigns, to each group in C, its center (recall Example 4). Then Q is first-order definable on C by the formula $\varphi(x)$ in Example 4. Let A be the set of all finite Abelian groups. Then A is a Boolean query, and it is first-order definable on C by the sentence $(\forall x)(\varphi(x))$. Finally, let S be the set of all finite Abelian simple groups. Then S is again a Boolean query, and it is definable on A by the least fixed-point sentence in Example 5.

2.5 Model of Computation

Our model of computation is the oracle alternating multi-tape Turing machine with random access to the input. This model, originally defined by Ruzzo [Ruz81] and used by Barrington, Immerman, and Straubing [BIS90], Buss [Bus87], and Sipser [Sip83] among others, is a modification of the model of oracle alternating multi-tape Turing machine of Chandra, Kozen, and Stockmeyer [CKS81] to allow sublinear time-bounds. These machines are equipped with an address tape on which to write a number in binary. When the machine enters a distinguished state with a number p written on its address tape, the head of the input tape jumps, in one step, to the p-th leftmost cell of the tape. Strictly speaking, the definition of Ruzzo [Ruz81] is slightly different from ours, but standard simulation arguments show that both models have the same computing power with only a constant factor loss in time or number of alternations (see [BIS90] and [Bus87] for example). In the case of deterministic machines, our model is slightly more robust, but this will not affect the generality of the results.

The fact that our Turing machines have random access to the input will allow us impose sublinear time-bounds on the machines. We note that a Turing machine with the standard sequential access to the input would not be able to read the whole input in that time.

Example 7 This example is not only useful to illustrate the operation of random access machines, but it will also be used later in the text. This result is attributed to Dowd by Buss [Bus87].

Proposition 3 *There exists a deterministic Turing machine with random access input, that on input w of length n, halts in exactly $4\lceil\log_2(n+1)\rceil$ steps with the binary representation of n written on the address tape.*

Proof: First the general idea. The machine determines the last position occupied by w. To do that, it queries the symbols at positions $2^0, 2^1, \ldots, 2^i$ until the blank symbol is found. Then it performs a binary search between positions 2^{i-1} and $2^i - 1$. The technicalities next. Observe that addresses $2^0, 2^1, \ldots, 2^i$ can be written on the address tape by adding 0's to an initial 1. Observe too that the binary search between 2^{i-1} and $2^i - 1$ consists in filling the zeros in 10^{i-2} with zeros or ones, from left to right, and according to a simple rule: zero if the symbol at the current address with a one is blank, and one otherwise. Thus, the running time is $\lceil \log_2(n+1) \rceil$ steps for the first phase, $\lceil \log_2(n+1) \rceil$ steps to let the head return to the first zero in 10^{i-2}, and $2\lceil \log_2(n+1) \rceil$ steps to complete the binary search; overall, $4\lceil \log_2(n+1) \rceil$ steps. □

Chapter 3

Fixed-Point Logics in Finite Set Theory

3.1 Basic Definitions

Let L be a relational vocabulary. A Δ_0-formula of the vocabulary $L \cup \{\in\}$ is a first-order formula such that all occurrences of quantifiers are of the form $(\forall x \in y)$ and $(\exists x \in y)$. More formally:

Definition 9 *Let L be a relational vocabulary. The class of Δ_0-formulas of $L \cup \{\in\}$ is the smallest class of formulas that contains all atomic formulas of $L \cup \{\in\}$, and is closed under negation, conjunction, disjunction, and bounded quantification: If φ is a Δ_0-formula, then $(\exists x \in y)(\varphi)$ and $(\forall x \in y)(\varphi)$ are also Δ_0-formulas. Actually, $(\exists x \in y)(\varphi)$ stands for $(\exists x)(x \in y \wedge \varphi)$, and $(\forall x \in y)(\varphi)$ stands for $(\forall x)(x \in y \rightarrow \varphi)$.*

We let $\mathrm{LFP}(\Delta_0)$ denote the fragment of LFP that consists of the least fixed-points of positive Δ_0-formulas. This means that every $\mathrm{LFP}(\Delta_0)$-formula is of the form

$$\mathrm{LFP}_{x_1,\ldots,x_k,R}\varphi(x_1,\ldots,x_k,R),$$

where $\varphi(x_1,\ldots,x_k,R)$ is a Δ_0-formula that is positive in the k-ary relation symbol R and has $x_1,\ldots,x_k$ as free variables. Note that every $\mathrm{LFP}(\Delta_0)$-formula involves a single application of the least fixed-point operator, that is, it contains no nested or iterated least fixed-points. Furthermore, no additional first-order parameters are allowed in $\mathrm{LFP}(\Delta_0)$-formulas.

In a series of papers, Sazonov has studied definability in a variant of $\mathrm{LFP}(\Delta_0)$ on the infinite structure $(V_\omega, \in)$ of all hereditarily finite sets (see [Saz97] for a survey). Here, we study uniform definability in $\mathrm{LFP}(\Delta_0)$ on the collection $\mathcal{BFR}$ of all finite structures that are images of the BIT-structures $\mathbf{BIT}_n = (\{0,1,\ldots,n-1\},\mathrm{BIT}_n)$ under the Ackermann isomorphism $e : \mathbb{N} \to V_\omega$. In particular, we investigate the question: does $\mathrm{LFP}(\Delta_0)$ collapse to FO on $\mathcal{BFR}$?

3.2 On $\mathrm{LFP}(\Delta_0)$ vs. FO

Although the complexity-theoretic aspects of $\mathrm{LFP}(\Delta_0)$ will be studied in depth in a later chapter of the dissertation, we start its study by observing that certain questions about its expressibility are tightly related to difficult questions in complexity theory. As explained in the introduction, separating FO from LFP on $\mathcal{BFR}$ is literally equivalent to separating $\mathbf{LINH}$ from **E**, an open problem in complexity theory. In this section, we show that even the collapse of $\mathrm{LFP}(\Delta_0)$ to FO on $\mathcal{BFR}$ would yield important results in complexity theory. Hence, it is difficult to either refute or confirm that $\mathrm{LFP}(\Delta_0)$ collapses to FO on $\mathcal{BFR}$. Indeed, if $\mathrm{LFP}(\Delta_0)$ does not collapse to FO, then LFP does not either, and so $\mathbf{LINH} \neq \mathbf{E}$. While if it does, then we show that $\mathbf{P} \subseteq \mathbf{LINH}$ which implies $\mathbf{P} \neq \mathbf{PSPACE}$. Both separations $\mathbf{LINH} \neq \mathbf{E}$ and $\mathbf{P} \neq \mathbf{PSPACE}$ are considered highly likely but very difficult to prove.

Recall that our model of computation is the oracle alternating multi-tape Turing machine with random access to the input. The only technical difference with the standard oracle alternating multi-tape Turing machine defined by Chandra, Kozen, and Stockmeyer [CKS81] is the ability to random-access the input. This means that the machine is equipped with a special address tape on which to write a number in binary, and whenever the machine enters a special query state with the number p written in the address tape, the input head jumps, in one step, to the p-th leftmost position of the tape. This will allow our machines to access any position of the input even when sublinear time bounds are imposed. Moreover, it is not hard to see that when the running time is as least linear, both models simulate each other within constant factor losses in time and number of alternations.

Given a time bound $t(n) \geq \log n$, and an alternation bound $a(n) \geq 0$, we define $\mathrm{ATIME}(t(n), a(n))$ to be the class of languages that are accepted by alternating multi-tape Turing machines with random access input within $t(n)$ steps of computation and $a(n)$ alternations. The polynomial-time hierarchy $\mathbf{PH}$ is known to coincide with $\bigcup_{c>0} \mathrm{ATIME}(n^c, c)$. Similarly, the linear-time hierarchy $\mathbf{LINH}$ is known to coincide with $\bigcup_{c>0} \mathrm{ATIME}(cn, c)$. Following the analogy, the logarithmic-time hierarchy $\mathbf{LH}$ is defined by Sipser (see [Bus87]) as $\bigcup_{c>0} \mathrm{ATIME}(c \log n, c)$.

Barrington, Immerman, and Straubing [BIS90] showed that first-order logic captures the logarithmic-time hierarchy $\mathbf{LH}$, on the class of words with BIT. Here, *words with BIT* are finite structures over the vocabulary $\{P, <, \mathrm{BIT}\}$, where P is a unary relation and $<, \mathrm{BIT}$ are binary relations, that have the form $(\{0, \ldots, n-1\}, P, <, \mathrm{BIT})$, where $<$ and BIT are the standard linear order on $\{0, \ldots, n-1\}$ and the BIT predicate respectively. Let $\mathcal{W}$ be the class of words with BIT. Technically, the result of Barrington, Immerman and Straubing means that a query on $\mathcal{W}$ is first-order definable if and only if a suitable encoding of it is computable in $\mathbf{LH}$. Here, *a suitable encoding* is defined as follows. For every $\mathbf{M} \in \mathcal{W}$ of the form $(\{0, \ldots, n-1\}, P, <, \mathrm{BIT})$, let $\chi(P)$ be the word $a_0 \ldots a_{n-1} \in \{0,1\}^n$, where $a_i = 1$ if and only if $i \in P$. For every $i \in \{0, \ldots, n-1\}$, let $\mathrm{b}_n(i)$ be the binary representation of i, with leading zeros if necessary so that the length of $\mathrm{b}_n(i)$ is $\lceil \log_2(n) \rceil$. Then, given a k-ary query Q on $\mathcal{W}$, the encoding of Q is denoted by $L(Q)$ and is defined to be the language over the alphabet $\{0, 1, \#\}$ that consists of all words of the form

$$1^n \# \chi(P) \# \mathrm{b}_n(a_1) \# \ldots \# \mathrm{b}_n(a_k)$$

where $\mathbf{M} \in \mathcal{W}$, $n = |M|$, and $(a_1, \ldots, a_k) \in Q(\mathbf{M})$. The precise result in [BIS90] is that Q is first-order definable on $\mathcal{W}$ if and only if $L(Q) \in \mathbf{LH}$. We have now all the necessary material to prove the first result about the expressive power of $\mathrm{LFP}(\Delta_0)$.

Theorem 1 *If* $\mathrm{LFP}(\Delta_0) \subseteq \mathrm{FO}$ *on* $\mathcal{BFR}$*, then* $\mathbf{P} \subseteq \mathbf{LINH}$*, which in turn implies that* $\mathbf{P} \neq \mathbf{PSPACE}$.

Proof: Let L be a language in $\mathbf{P}$ over the alphabet $\{0, 1\}$. We will show that the language $L' = \{1^{2^n} \# 0^{2^n} \# w \# \tilde{w} : w \in L,\ |w| = n\}$ over $\{0, 1, \#\}$ is in $\mathbf{LH}$; then a de-padding argument puts L in $\mathbf{LINH}$. Here, $\tilde{w}$ stands for the dual word of w obtained by interchanging

0's and 1's. By the Immerman-Vardi Theorem, there is a least fixed-point sentence φ of the vocabulary $\{P, <, \mathrm{BIT}, 0, \max\}$, where P is a unary relation symbol, that defines L on the class of words with BIT. We can assume that $\varphi = (\mathrm{LFP}_{\overline{x},R}\psi(\overline{x}, R))(\overline{0})$ with ψ first-order by the normal-form theorem for least fixed-point formulas. Moreover, $<$ need not occur in ψ, since by [DDLW98] it is first-order definable from BIT. Similarly, 0 and $\max$ need not occur in ψ, because they are first-order definable as well. We turn $\psi(x_1, \ldots, x_k, R)$ into a Δ_0-formula $\psi'(x_1, \ldots, x_k, y_1, y_2, S)$, where S is a $(k+2)$-ary relation variable. Replace each occurrence $\mathrm{BIT}(z_i, z_j)$ in ψ by $z_i \in z_j$, each positive occurrence $P(z_i)$ in ψ by $z_i \in y_1$, each negative occurrence $\neg P(z_i)$ in ψ by $z_i \in y_2$, and each occurrence $R(z_{i_1}, \ldots, z_{i_k})$ by $S(z_{i_1}, \ldots, z_{i_k}, y_1, y_2)$. Finally, replace subformulas of the form $(\exists z)(\theta)$ by $(\exists z \in y_1)(\theta) \vee (\exists z \in y_2)(\theta)$, and subformulas of the form $(\forall z)(\theta)$ by $(\forall z \in y_1)(\theta) \wedge (\forall z \in y_2)(\theta)$. In the following, for every $w \in \{0, 1\}^*$, let $\mathbf{M}(w) = (\{0, \ldots, |w| - 1\}, P, <, \mathrm{BIT})$ be the unique word with BIT such that $\chi(P) = w$. Let $\mathrm{n}(w)$ be the number represented in binary by w.

Claim 1 For every $n \geq 0$ and $w \in \{0, 1\}^n$, the following are equivalent:

(i) $\mathbf{M}(w) \models \mathrm{LFP}_{\overline{x},R}\psi(\overline{x}, R)[\overline{0}]$

(ii) $e(\mathbf{BIT}_{2^n}) \models \mathrm{LFP}_{\overline{x},y_1,y_2,R'}\psi'(\overline{x}, y_1, y_2, R')[\overline{\emptyset}, e(\mathrm{n}(w)), e(\mathrm{n}(\tilde{w}))]$

Proof of Claim 1: Let $w \in \{0, 1\}^n$. For every $\overline{p} = (p_1, \ldots, p_k) \in \{0, \ldots, n-1\}^k$, let

$$e(\overline{p}) = (e(p_1), \ldots, e(p_k), e(\mathrm{n}(w)), e(\mathrm{n}(\tilde{w}))).$$

Let $M = \mathbf{M}(w)$, and let $N = \mathbf{BIT}_{2^n}$. We prove by induction on t, that $\overline{p} \in I^t_\psi(M)$ if and only if $e(\overline{p}) \in I^t_{\psi'}(N)$, for every $\overline{p} \in \{0, \ldots, n-1\}^k$. Since $e(0) = \emptyset$, the claim will follow. For a fixed t, write $I = I^t_{\psi}(M)$ and $I' = I^t_{\psi'}(N)$. The base case of the induction $t = 0$ is obvious. Suppose then that the claim holds for t; we prove that it also holds for $t + 1$. We need to prove the following stronger statement: for every subformula of $\psi(\overline{x}, R)$, say $\psi_s(\overline{z}, R)$, it holds that $M \models \psi_s(\overline{z}, R)[\overline{p}, I]$ if and only if $N \models \psi'_s(\overline{z}, y_1, y_2, R')[e(\overline{p}), I']$, for every $\overline{p} \in \{0, \ldots, n-1\}^r$. Here $\psi'_s(\overline{z}, y_1, y_2)$ is the result of applying the syntactic transformation over ψ_s. The proof is by induction on the structure of ψ_s. Consider first the atomic subformulas of the form $P(z_i)$, $\mathrm{BIT}(z_i, z_j)$, and $R(z_{i_1}, \ldots, z_{i_k})$. In the first case, $M \models P(z_i)[p_i]$ if and only if the p_i-th symbol w from the right is one, or equivalently, if and only if $N \models z_i \in y_1[e(p_i), e(\mathrm{n}(w))]$. The second case is also clear, namely, $M \models \mathrm{BIT}(z_i, z_j)[p_i, p_j]$ if and only if $N \models z_i \in z_j[e(p_i), e(p_j)]$. For the third case we use the induction hypothesis on t; namely, $M \models R(z_{i_1}, \ldots, z_{i_k})[\overline{p}, I]$ if and only if $(p_{i_1}, \ldots, p_{i_k}) \in I$, if and only if $(e(p_{i_1}), \ldots, e(p_{i_k}), e(\mathrm{n}(w)), e(\mathrm{n}(\tilde{w}))) \in I'$ by induction hypothesis, if and only if

$$N \models R'(z_{i_1}, \ldots, z_{i_k}, y_1, y_2)[e(p_{i_1}), \ldots, e(p_{i_k}), e(\mathrm{n}(w)), e(\mathrm{n}(\tilde{w})), I'].$$

When $\psi_s(\overline{z}, R)$ is of the form $\neg\theta_s(\overline{z}, R)$ or $\theta_s(\overline{z}, R) \wedge \chi_s(\overline{z}, R)$, we use the induction hypothesis on the structure of the formula. Consider finally the case where $\psi_s(\overline{z}, R)$ is of the form $(\forall z_i)\theta_s(\overline{z}, R)$. We have that $M \models (\forall z_i)\theta_s(\overline{z}, R)[\overline{p}, I]$ if and only if $M \models \theta_s(\overline{z}, R)[\overline{p}^i_p, I]$, for every $p \in \{0, \ldots, n-1\}$. Here $\overline{p}^i_p = (p_1, \ldots, p_{i-1}, p, p_{i+1}, \ldots, p_r)$. Now, by induction hypothesis on the structure of the formula, this is equivalent to $N \models \theta'_s(\overline{z}, y_1, y_2, R)[e(\overline{p}^i_p), I']$ for every $p \in \{0, \ldots, n-1\}$. Observe that e is a bijection between $\{0, \ldots, n-1\}$ and $e(\mathrm{n}(w)) \cup e(\mathrm{n}(\tilde{w}))$ from the definition of $\tilde{w}$. Thus, this is equivalent to $N \models \theta'_s(\overline{z}, y_1, y_2, R)[e(\overline{p})^i_a, I']$

for every $a \in e(\mathrm{n}(w)) \cup e(\mathrm{n}(\tilde{w}))$. Finally, this is equivalent to

$$N \models (\forall z_i \in y_1)(\psi'_s(\overline{z}, y_1, y_2)) \wedge (\forall z_i \in y_2)(\psi'_s(\overline{z}, y_1, y_2))[e(\overline{p}), I'].$$

The claim follows.

Since ψ' is a Δ_0-formula, the hypothesis of the theorem implies that the least fixed-point of ψ' is definable by a first-order formula on $\mathcal{BFR}$. It follows that the query

$$Q(\mathbf{BIT}_{2^n}) = \{(\mathrm{n}(w), \mathrm{n}(\tilde{w})) : w \in A,\ |w| = n\}$$

is first-order definable on $\mathcal{B}$. The result of Barrington, Immerman, and Straubing implies then that the encoding of Q, interpreted as a query on words with BIT, is computable in $\mathbf{LH}$. It follows that $L' = \{1^{2^n}\#0^{2^n}\#w\#\tilde{w} : w \in L,\ |w| = n\}$ is in $\mathbf{LH}$. A de-padding argument completes the proof that $\mathbf{P} \subseteq \mathbf{LINH}$. Since $\mathbf{LINH} \subseteq$ DSPACE(n^2) [BDG90], the space-hierarchy theorem implies that $\mathbf{P} \neq \mathbf{PSPACE}$. □

It should be noted that from a result of Dawar and Hella [DH95] it follows that if LFP $\subseteq$ FO on $\mathcal{BFR}$, then $\mathbf{P} \neq \mathbf{PSPACE}$. Theorem 1 shows that the separation of $\mathbf{P}$ from $\mathbf{PSPACE}$ can be derived from the weaker hypothesis that $\mathrm{LFP}(\Delta_0) \subseteq$ FO on $\mathcal{BFR}$.

3.3 Restricted $\mathrm{LFP}(\Delta_0)$ Collapses to FO

As mentioned earlier, Dawar, Doets, Lindell, and Weinstein [DDLW98] showed that there is a first-order formula of the vocabulary $\{\in\}$ that defines a linear order on the class $\mathcal{NFR}$ of near finite ranks, that is, structures of the form $(M, \in)$ such that $V_n \subseteq M \subseteq V_{n+1}$.

For this, they considered the following Δ_0-formula $\psi(x, y, S)$

$$(\exists y' \in y)(\forall x' \in x)(y' \notin x \wedge (x' \notin y \rightarrow S(x', y')))$$

and showed that its least fixed-point is definable by a first-order formula on $\mathcal{NFR}$. Observe that the occurrence of the relation symbol S in ψ involves only the bound variables x' and y'. We now abstract from this observation and introduce the following concept.

Definition 10 *A Δ_0-formula $\varphi(x_1, \ldots, x_k, R)$ is restricted if every occurrence of the relation symbol R involves only bound variables of φ.*

The main result of this section is that the least fixed-point of every positive restricted Δ_0-formula is first-order definable on $\mathcal{NFR}$ and, hence, on $\mathcal{BFR}$ as well. In fact, we show that it is definable by a first-order formula of low syntactic complexity. The class of Σ-formulas is the smallest collection of formulas containing the Δ_0-formulas and closed under conjunction, disjunction, bounded quantifications $(\exists y \in x)$, $(\forall y \in x)$, and existential quantification $\exists y$. The collection of Π formulas is defined dually by allowing closure under universal quantification $\forall y$. We say that a query Q on a class $\mathcal{C}$ is Δ-definable if it is definable on $\mathcal{C}$ by a Σ-formula and by a Π formula.

Theorem 2 *Let $\varphi(x_1, \ldots, x_k, R)$ be a restricted Δ_0-formula that is positive in the k-ary relation symbol R. The least fixed-point of φ is first-order definable on $\mathcal{NFR}$. In fact, it is Δ-definable on $\mathcal{NFR}$.*

The proof of the above theorem is inspired by the argument in [DDLW98] showing that the least fixed-point of the Δ_0-formula $\psi(x, y, S)$ is first-order definable on $\mathcal{NFR}$. We need, however, to establish certain absoluteness properties of arbitrary Δ_0-formulas, as well as certain structural properties of arbitrary restricted Δ_0-formulas that will be used heavily in the sequel. We begin with a basic definition from set theory (see [Bar75, page 34]).

Let $\mathbf{M}$ and $\mathbf{N}$ be two structures over the vocabulary $L \cup \{\in\}$. We say that $\mathbf{N}$ is an *end extension* of $\mathbf{M}$, and write $\mathbf{M} \subseteq_{\mathrm{end}} \mathbf{N}$, if $\mathbf{M}$ is a substructure of $\mathbf{N}$ and for every $a \in M$ it is the case that $\{b \in N : b \in^{\mathbf{N}} a\} = \{b \in M : b \in^{\mathbf{M}} a\}$.

Lemma 1 (Absoluteness of Δ_0-formulas [Bar75, page 35]) *If $\varphi(x_1, \ldots, x_k)$ is a Δ_0-formula and $\mathbf{M} \subseteq_{\mathrm{end}} \mathbf{N}$, then for every $(a_1, \ldots, a_k) \in M^k$ we have that $\mathbf{M} \models \varphi[a_1, \ldots, a_k]$ if and only if $\mathbf{N} \models \varphi[a_1, \ldots, a_k]$.*

Proof: The proof is by induction on the structure of φ. If φ is atomic, the claim follows from the fact that $\mathbf{M}$ is a substructure of $\mathbf{N}$. If $\varphi = \neg\psi$ or $\varphi = \psi \wedge \theta$, we use the induction hypothesis in the obvious way. Suppose next that $\varphi = (\exists y \in x_i)(\psi)$. For the forward direction of the equivalence we only use the fact that $\mathbf{M}$ is a substructure of $\mathbf{N}$, and not the fact that $\mathbf{N}$ is actually an end-extension of $\mathbf{M}$. Here is the argument: If $\mathbf{M} \models \varphi[a_1, \ldots, a_k]$, then there exists some $a \in^{\mathbf{M}} a_i$ such that $\mathbf{M} \models \psi[a_1, \ldots, a_k, a]$. Clearly, $a \in M$. Now, by induction hypothesis, $\mathbf{N} \models \psi[a_1, \ldots, a_k, a]$. Since $\mathbf{M}$ is a substructure of $\mathbf{N}$, we have $a \in N$ and $a \in^{\mathbf{N}} a_i$. Therefore $\mathbf{N} \models (\exists y \in x_i)(\psi)[a_1, \ldots, a_k]$. For the reverse direction of the implication we use the hypothesis that $\mathbf{M} \subseteq_{\mathrm{end}} \mathbf{N}$ as follows: If $\mathbf{N} \models (\exists y \in x_i)(\psi)[a_1, \ldots, a_k]$, then there exists some $a \in^{\mathbf{N}} a_i$ such that $\mathbf{N} \models \psi[a_1, \ldots, a_k, a]$. Since $\mathbf{M} \subseteq_{\mathrm{end}} \mathbf{N}$ and $a_i \in M$, we have $a \in M$ and $a \in^{\mathbf{M}} a_i$. Therefore, by induction hypothesis $\mathbf{M} \models \psi[a_1, \ldots, a_k, a]$, which means that $\mathbf{M} \models (\exists y \in x_i)(\psi)[a_1, \ldots, a_k]$ as required. □

If $\varphi(x_1,\ldots,x_s,R)$ is a Δ_0-formula, then the set of *R-free indices* of φ, denoted by $U(\varphi)$, is the set of indices of variables that are free in φ and appear in at least one occurrence of R in φ. For example, if $\varphi(x_1,x_2,x_3,R)$ is the formula $(\forall x_4 \in x_3)R(x_1,x_2,x_4)$, then $U(\varphi)=\{1,2\}$. Note that a Δ_0-formula φ is restricted if and only if $U(\varphi)=\emptyset$. Similarly, we define the set of *free indices* of φ, denoted by $F(\varphi)$, as the set of indices of variables that are free in φ. Thus $U(\varphi)\subseteq F(\varphi)$. From now on, we identify structures $(M,\in)$ over $\{\in\}$ with their universe M. In the following two lemmas, we establish certain important properties of Δ_0 and of restricted Δ_0-formulas on near finite ranks. Recall from basic set theory that the transitive closure of a set is defined inductively as follows: $\mathrm{TC}(a)=a\cup\bigcup\{\mathrm{TC}(b):b\in a\}$. Observe that if $a\in b$, then $\mathrm{TC}(a)\subseteq\mathrm{TC}(b)$.

Lemma 2 *Let $\varphi(x_1,\ldots,x_s,R)$ be a Δ_0-formula of $\{\in,R\}$, where R is a k-ary relation symbol, and let M be a near finite rank such that $V_n\subseteq M\subseteq V_{n+1}$. For every $m\le n$, every relation $A\subseteq M^k$, and every tuple $\overline{a}=(a_1,\ldots,a_s)\in V^s_{m+1}\cap M^s$, we have that $M\models\varphi[\overline{a},A]$ if and only if $M\models\varphi[\overline{a},A\cap(\bigcup\{\mathrm{TC}(a_i):i\in F(\varphi)\}\cup\{a_i:i\in U(\varphi)\})^k]$. In particular, if φ is a restricted Δ_0-formula positive in R, then $M\models\varphi[\overline{a},A]$ if and only if $M\models\varphi[\overline{a},A\cap V^k_m]$.*

Proof: We proceed by induction on the construction of φ. The base cases are trivial, and so is the case in which φ is of the form $\neg\psi$. Suppose that φ is of the form $\psi_1\wedge\psi_2$. Let $B=\bigcup\{\mathrm{TC}(a_i):i\in F(\varphi)\}$ and $B_j=\bigcup\{\mathrm{TC}(a_i):i\in F(\psi_i)\}$ for $j=1,2$. Then, $M\models\varphi[\overline{a},A\cap(B\cup\{a_i:i\in U(\varphi)\})^k]$ if and only if $M\models\psi_j[\overline{a},A\cap(B\cup\{a_i:i\in U(\varphi)\})^k]$ for $j=1,2$. By induction hypothesis, this is equivalent to

$$M\models\psi_j[\overline{a},A\cap(B\cup\{a_i:i\in U(\varphi)\})^k\cap(B_j\cup\{a_i:i\in U(\psi_i)\})^k]$$

for $j = 1, 2$. Since $F(\psi_j) \subseteq F(\varphi)$ and $U(\psi_j) \subseteq U(\varphi)$, this is equivalent to

$$M \models \psi_j[\overline{a}, A \cap (B_i \cup \{a_i : i \in U(\psi_i)\})^k]$$

for $j = 1, 2$. By induction hypothesis again, this is equivalent to $M \models \psi_j[\overline{a}, A]$ for $j = 1, 2$, and therefore to $M \models \varphi[\overline{a}, A]$. Suppose next that φ is of the form $(\exists x_i \in x_j)\psi$. In this case, $M \models \varphi[\overline{a}, A]$ if and only if there is some $a \in M$ such that $a \in a_j$ and $M \models \psi[\overline{b}, A]$, where $\overline{b} = (a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_s)$. Therefore, by induction hypothesis, $M \models \varphi[\overline{a}, A]$ if and only if there is some $a \in M$ such that $a \in a_j$ and

$$M \models \psi[\overline{b}, A \cap (\bigcup\{\mathrm{TC}(b_l) : l \in F(\psi)\} \cup \{b_l : l \in U(\psi)\})^k].$$

Since for every $a \in a_j$ we have that $\mathrm{TC}(a) \subseteq \mathrm{TC}(a_j)$ and $a \in \mathrm{TC}(a_j)$, it is the case that

$$\bigcup\{\mathrm{TC}(b_l) : l \in F(\psi)\} \cup \{b_l : l \in U(\psi)\} = \bigcup\{\mathrm{TC}(a_l) : l \in F(\varphi)\} \cup \{a_l : l \in U(\varphi)\}.$$

Consequently, $M \models \varphi[\overline{a}, A]$ if and only if there is some $a \in M$ such that $a \in a_j$ and

$$M \models \psi[\overline{b}, A \cap (\bigcup\{\mathrm{TC}(a_i) : i \in F(\varphi)\} \cup \{a_i : i \in U(\varphi)\})^k],$$

which means that $M \models \varphi[\overline{a}, A \cap (\bigcup\{\mathrm{TC}(a_i) : i \in F(\varphi)\} \cup \{a_i : i \in U(\varphi)\})^k]$, as required.
□

The next lemma yields an absoluteness property of Δ_0-inductions on near finite ranks and also reveals that every positive restricted Δ_0-formula has a unique fixed-point on near finite ranks.

Lemma 3 *Let $\varphi(x_1, \ldots, x_k, R)$ be a Δ_0-formula of $\{\in, R\}$ that is positive in the k-ary relation symbol R, and let M be a near finite rank such that $V_n \subseteq M \subset V_{n+1}$. For every $m \leq n$*

and every $t \geq 0$, we have that $I^t_\varphi(M) \cap V^k_m = I^t_\varphi(V_m)$ and $J^t_\varphi(M) \cap V^k_m = J^t_\varphi(V_m)$. If, in addition, φ is a restricted Δ_0-formula, then $I_\varphi(M) = J_\varphi(M)$.

Proof: The first statement is proved by induction on t as follows. Assume that for every $s < t$ we have that $I^s_\varphi(M) \cap V^k_m = I^s_\varphi(V_m)$. Let $\overline{a} \in V^k_m$. By definition, $\overline{a} \in I^t_\varphi(M)$ if and only if $M \models \varphi(\overline{x}, R)[\overline{a}, \bigcup_{s<t} I^s_\varphi(M)]$. By absoluteness, $M \models \varphi(\overline{x}, R)[\overline{a}, \bigcup_{s<t} I^s_\varphi(M)]$ if and only if $V_m \models \varphi(\overline{x}, R)[\overline{a}, (\bigcup_{s<t} I^s_\varphi(M)) \cap V^k_m]$, which in turn is equivalent to $V_m \models \varphi(\overline{x}, R)[\overline{a}, \bigcup_{s<t}(I^s_\varphi(M) \cap V^k_m)]$ by distributing $\cup$ over $\cap$. By induction hypothesis, we have $I^s_\varphi(M) \cap V^k_m = I^s_\varphi(V_m)$ so that $V_m \models \varphi(\overline{x}, R)[\overline{a}, \bigcup_{s<t}(I^s_\varphi(M) \cap V^k_m)]$ if and only if $V_m \models \varphi(\overline{x}, R)[\overline{a}, \bigcup_{s<t} I^s_\varphi(V_m)]$. By definition, this means $\overline{a} \in I^t_\varphi(V_m)$ as required. To prove $J^t_\varphi(M) \cap V^k_m = J^t_\varphi(V_m)$ reason similarly. We turn to the second statement. Suppose that φ is a restricted formula. We show that every post fixed-point of φ is included in the least fixed-point; this will imply that the greatest fixed-point and the least fixed-point coincide. Let $A \subseteq M^k$ be a post fixed-point of φ on M; that is, $A \subseteq \varphi^M(A)$. Let $\overline{a} \in A$, so that $M \models \varphi(\overline{x}, R)[\overline{a}, A]$. We prove that $\overline{a} \in I_\varphi(M)$ by induction on $j = \max\{\mathrm{rank}(a_i) : i = 1, \ldots, k\}$. Obviously, $\overline{a} \in V^k_j$, so that by Lemma 2 we have that $M \models \varphi(\overline{x}, R)[\overline{a}, A \cap V^k_{j-1}]$. By induction hypothesis on j we have that $A \cap V^k_{j-1} \subseteq I_\varphi(M)$, and so by monotonicity, $M \models \varphi(\overline{x}, R)[\overline{a}, I_\varphi(M)]$. Since $I_\varphi(M)$ is a fixed-point, we have $\overline{a} \in I_\varphi(M)$ as required. □

We now have all the necessary tools to show that restricted LFP(Δ_0) collapses to FO on the class $\mathcal{NFR}$ of near finite ranks.

Proof of Theorem 2: The key idea is that there is a first-order formula $\theta(x_1, \ldots, x_k)$ that *approximates* the greatest fixed-point $J_\varphi(M)$ on near finite ranks M; the formula θ is based on the characterization of $J_\varphi(M)$ as the union of all post fixed-points of φ. This approximation can be improved to yield an exact first-order definition of $J_\varphi(M)$ by first replacing every occurrence of R in φ by θ, and then iterating φ a constant number of times. The result will then follow from Lemma 3, which asserts that the greatest fixed-point of the restricted Δ_0-formula φ coincides with its least fixed-point. This type of argument was used in [DDLW98] to show that the least fixed-point of the Δ_0-formula $\psi(x, y, S)$ in the beginning of Section 3.3 is first-order definable on $\mathcal{NFR}$. Here, we have to deploy the machinery of Lemmas 1, 2 and 3 to show that the argument can be extended and applied to every restricted Δ_0-formula.

We will construct a Σ-formula $\psi(x_1, \ldots, x_k)$ that defines the greatest fixed-point $J_\varphi(M)$ on every $M \in \mathcal{NFR}$ such that $V_n \subseteq M \subseteq V_{n+1}$ for some $n > 2k - 1$ (for near finite ranks M below V_{2k} we can obtain a Δ_0 definition of $J_\varphi(M)$ by iterating φ a sufficient number of times). The construction of ψ is carried out in three steps. For the first step, define a Δ_0-formula $\mathrm{post}(y)$ expressing that y is the encoding of a post fixed-point of φ on M. We use the standard encoding of a pair $\langle x, y\rangle$ by the set $\{\{x\}, \{x, y\}\}$, and of a tuple $\langle x_1, \ldots, x_k\rangle$ by $\langle\langle x_1, \ldots, x_{k-1}\rangle, x_k\rangle$. Then $\mathrm{post}(y)$ is the formula

$$\mathrm{rel}_k(y) \wedge (\forall\langle z_1, \ldots, z_k\rangle \in y)(\widehat{\varphi}(z_1, \ldots, z_k, R/y)),$$

where $\mathrm{rel}_k(y)$ is a Δ_0-formula expressing that y is a set of encodings of k-tuples, and $\widehat{\varphi}$ is obtained from φ by replacing atomic formulas $R(z_{i_1}, \ldots, z_{i_k})$ by $\langle z_{i_1}, \ldots, z_{i_k}\rangle \in y$. Let $A \subseteq M^k$ be such that the set $\langle A\rangle = \{\langle a_1, \ldots, a_k\rangle : (a_1, \ldots, a_k) \in A\}$ is in M; it is not hard to show that $M \models \mathrm{post}[\langle A\rangle]$ if and only if A is a post fixed-point of $\varphi^M(A)$ on M. For the second step, let $\theta(x_1, \ldots, x_k)$ be the Σ-formula

$$(\exists y)(\mathrm{post}(y) \wedge \langle x_1, \ldots, x_k\rangle \in y)$$

expressing that $(x_1, \ldots, x_k)$ belongs to some post fixed-point of φ.

Claim 2 $J_\varphi(M) \cap V^k_{n-2k+1} \subseteq \theta^M \subseteq J_\varphi(M)$.

Proof of Claim 2: Since φ is Δ_0, Lemma 3 implies that $J_\varphi(M) \cap V^k_{n-2k+1} = J_\varphi(V_{n-2k+1})$. Thus, for the first inclusion it suffices to show that if $\overline{a} = (a_1, \ldots, a_k) \in J_\varphi(V_{n-2k+1})$, then $M \models \theta[\overline{a}]$. We need a witness r for the quantifier $(\exists y)$ in θ. Put $A = J_\varphi(V_{n-2k+1})$ and $r = \langle A\rangle$. Since $A \subseteq V^k_{n-2k+1}$, we have that $r \in V_n \subseteq M$. Moreover, since $\overline{a} \in A$, we have that $\langle\overline{a}\rangle \in r$; hence, $M \models (\langle\overline{x}\rangle \in y)[\overline{a}, r]$. We can now show that $M \models \mathrm{post}(y)[r]$ using Lemma 1 and the fact that $V_{n-2k+1} \subseteq_{\mathrm{end}} M$. For the second inclusion, let $\overline{a} = (a_1, \ldots, a_k) \in M^k$ be such that $M \models \theta(\overline{x})[\overline{a}]$; that is, $M \models \langle\overline{x}\rangle \in y[\overline{a}, r]$ for some $r \in M$ such that $M \models \mathrm{post}(y)[r]$. Since $M \models \mathrm{post}(y)[r]$, we know that r encodes a post fixed-point of φ on M, say A. Moreover $\overline{a} \in A$, and so $\overline{a} \in J_\varphi(M)$ because $J_\varphi(M)$ is the union of all post fixed-points of φ on M.

Thus, θ yields an approximation of the greatest fixed-point $J_\varphi(M)$ of φ. For the third step, we iterate φ a number of times (in fact, $2k$ times) to obtain progressively better approximations. Let $\theta_0(x_1, \ldots, x_k) = \theta(x_1, \ldots, x_k)$ and $\theta_i(x_1, \ldots, x_k) = \varphi(x_1, \ldots, x_k, R/\theta_{i-1})$, for $i \in \{1, \ldots, 2k\}$. Note that each θ_i is a Σ-formula.

Claim 3 $J_\varphi(M) \cap V^k_{n-2k+1+i} \subseteq \theta^M_i \subseteq J_\varphi(M)$, for every $i = 0, \ldots, 2k$.

Proof of Claim 3: Lemma 3 implies that $J_\varphi(M) \cap V^k_{n-2k+1+i} = J_\varphi(V_{n-2k+1+i})$. We proceed by induction on i. Claim 2 takes care of the case $i = 0$. Assume that Claim 3 holds for $i - 1$. For the first inclusion, if $\overline{a} = (a_1, \ldots, a_k) \in J_\varphi(M) \cap V^k_{n-2k+1+i}$, then $M \models \varphi[\overline{a}, J_\varphi(M)]$. Since $\overline{a} \in V^k_{n-2k+1+i} \cap M^k$ and φ is a restricted Δ_0-formula, Lemma 2 implies that $M \models \varphi[\overline{a}, J_\varphi(M) \cap V^k_{n-2k+i}]$. Note that this is the first place in this proof where we use the assumption that the Δ_0-formula φ is restricted. By induction hypothesis, $J_\varphi(M) \cap V^k_{n-2k+i} \subseteq \theta^M_{i-1}$; therefore, $M \models \varphi[\overline{a}, \theta^M_{i-1}]$, since φ is monotone. It follows that $M \models \varphi(\overline{x}, R/\theta_{i-1})[\overline{a}]$ and $M \models \theta_i[\overline{a}]$. For the second inclusion, let $\overline{a} = (a_1, \ldots, a_k) \in M^k$ be such that $M \models \theta_i(\overline{x})[\overline{a}]$, so that $M \models \varphi(\overline{x}, R/\theta_{i-1})[\overline{a}]$ by the definition of θ_i. Therefore, $M \models \varphi(\overline{x}, R)[\overline{a}, \theta^M_{i-1}]$, and since $\theta^M_{i-1} \subseteq J_\varphi(M)$ by induction hypothesis, $M \models \varphi(\overline{x}, R)[\overline{a}, J_\varphi(M)]$ by monotonicity. We conclude that $\overline{a} \in J_\varphi(M)$ because $J_\varphi(M)$ is a fixed-point.

Finally, let ψ be the Σ-formula $\theta_{2k}(x_1, \ldots, x_k)$. Claim 3 implies that ψ defines the greatest fixed-point $J_\varphi(M)$ of φ on every near finite rank M such that $V_n \subseteq M \subseteq V_{n+1}$ for some $n > 2k - 1$. Let $\tilde{\varphi}(\overline{x}, R)$ be the dual formula $\neg\varphi(\overline{x}, \neg R)$ of φ. Note that $\tilde{\varphi}$ is also a restricted Δ_0-formula. Moreover, $J_{\tilde{\varphi}}(M) = M - I_\varphi(M)$ as we saw in Chapter 2. Therefore, $I_\varphi(M)$ is Π definable by taking the negation of the Σ-formula that defines $J_{\tilde{\varphi}}(M)$. The Δ-definability of $I_\varphi(M)$ follows immediately, since $I_\varphi(M) = J_\varphi(M)$ by Lemma 3. This completes the proof of Theorem 2. □

Example 8 In spite of the stringent syntactic requirements imposed, several natural queries can be expressed using restricted LFP(Δ_0)-formulas. Thus, Theorem 2 provides a versatile tool for showing that such queries are first-order definable on $\mathcal{NFR}$. Here, we illustrate this technique by considering the query *(finite) ordinal addition* Q_{add}, where if M is a near finite rank, then

$$Q_{\mathrm{add}}(M) = \{(\alpha, \beta, \gamma) \in M^3 : \alpha, \beta, \gamma \text{ are ordinals and } \gamma = \alpha + \beta\}.$$

The standard recursive specification of ordinal addition does not lead to a restricted Δ_0-formula. Consider, however, the following alternate recursive specification: if α, β, γ are ordinals, then $\gamma = \alpha + \beta$ if and only if

$$(\alpha = 0 \wedge \gamma = \beta) \vee (\beta = 0 \wedge \gamma = \alpha) \vee (\exists \alpha' \in \alpha)(\exists \beta' \in \beta)(\exists \gamma' \in \gamma)(\exists \delta' \in \gamma')$$

$$(\alpha = \alpha' + 1 \wedge \beta = \beta' + 1 \wedge \gamma = \gamma' + 1 \wedge \gamma' = \delta' + 1 \wedge \delta' = \alpha' + \beta').$$

This recursive specification can easily be transformed into the least fixed-point of a positive restricted Δ_0-formula that defines Q_{add}; to this end, we use the fact that being an ordinal is Δ_0-definable, and that $\alpha + 1 = \alpha \cup \{\alpha\}$.

Example 9 We exhibit a second application of Theorem 2 that will be of use in a later section. We wish to show that the set of finite ranks that belong to a near finite rank M is uniformly Δ-definable. More precisely, for every $M \in \mathcal{NFR}$, let

$$Q_{\mathrm{rk}}(M) = \{a \in M : a = V_m \text{ for some } m \geq 0\}.$$

We show that the query Q_{rk} is definable as the least fixed-point of a positive restricted Δ_0-formula, and therefore, it is Δ-definable on $\mathcal{NFR}$. Let $\varphi(x, R)$ be the formula $x = \emptyset \vee (\exists y \in x)(R(y) \wedge \psi(x, y))$ where $\psi(x, y)$ is the Δ_0-formula

$$(\forall z \in x)(\forall z' \in z)(z \in y) \wedge (\exists z \in x)(z = \emptyset) \wedge$$

$$\wedge (\forall z \in x)(\forall z' \in y)(\exists u \in x)(u = z \cup \{z'\}).$$

Here, the formula $x = \emptyset$ is an abbreviation for the Δ_0-formula $(\forall z \in x)(z \neq z)$, and $u = z \cup \{z'\}$ is an abbreviation for the Δ_0-formula $(\forall u' \in u)(u' \in z \vee u' = z') \wedge (\forall u' \in z)(u' \in u) \wedge z' \in u$. Observe that $\varphi(x, R)$ is a restricted Δ_0-formula that is positive in R. Let $M \in \mathcal{NFR}$ be such that $V_n \subseteq M \subset V_{n+1}$.

Claim 4 For every $m < n$ and $a \in M$ it holds that $M \models \psi[a, V_m]$ if and only if $a = V_{m+1}$.

Proof: Fix $m < n$ and $a \in M$. The implication from right to left is easy. For the other implication, the first conjunct of $\psi(x, y)$ assures that if $M \models \psi[a, V_m]$, then every $a' \in a$ is such that $a' \subseteq V_m$, so $a \subseteq \mathcal{P}(V_m) = V_{m+1}$. We show next that if $M \models \psi[a, V_m]$, then $V_{m+1} \subseteq a$ too, so $a = V_{m+1}$. This will prove the claim. Suppose thus that $M \models \psi[a, V_m]$ and let $b = \{b_1, \ldots, b_r\} \in V_{m+1}$. We show by induction on r that $b \in a$. If $r = 0$, then $b = \emptyset$ so $b \in a$ according to the second conjunct of $\psi(x, y)$. Suppose next that $r > 0$. Clearly, $b' = \{b_1, \ldots, b_{r-1}\}$ is a member of V_{m+1}. Therefore, by induction hypothesis, $b' \in a$. It follows from the third conjunct of $\psi(x, y)$ that $b' \cup \{b_r\} \in a$ since $b_r \in V_m$. It follows that $b \in a$ as was to be proved. □

The proof that $I_\varphi(M) = Q_{\mathrm{rk}}(M)$ is completed by proving $I^t_\varphi(M) = \{V_m : m < t\}$ by induction on t.

3.4 Unary $\mathrm{LFP}(\Delta_0)$ and Binary $\mathrm{LFP}(\Delta_0)$ Collapse to FO

For every positive integer k, let *k-ary* $\mathrm{LFP}(\Delta_0)$ denote the fragment of $\mathrm{LFP}(\Delta_0)$ that allows the formation of least fixed-points of positive Δ_0-formulas $\varphi(x_1, \ldots, x_k, R)$ such that R is a relation symbol of arity k. The following simple observation reveals that the smallest of these fragments collapses to FO on $\mathcal{NFR}$.

Proposition 4 *If $\varphi(x, R)$ is a unary Δ_0-formula that is positive in R, then there is a unary restricted Δ_0-formula $\varphi^*(x, R)$ that is positive in R and such that $I_\varphi(\mathbf{M}) = I_{\varphi^*}(\mathbf{M})$ on every structure $\mathbf{M}$. Consequently, unary* LFP(Δ_0) *collapses to* FO *on $\mathcal{NFR}$.*

Proof: Let $\varphi^*(x, R)$ be the restricted Δ_0-formula obtained from $\varphi(x, R)$ by replacing every occurrence of $R(x)$ by $x \neq x$, while preserving all occurrences $R(z)$ with z a bound variable. We prove, by induction on the ordinals, that $I^\gamma_\varphi(\mathbf{M}) = I^\gamma_{\varphi^*}(\mathbf{M})$ for every ordinal γ. We first prove $I^\gamma_\varphi(\mathbf{M}) \subseteq I^\gamma_{\varphi^*}(\mathbf{M})$. If $a \in I^\gamma_\varphi(\mathbf{M})$, then $\mathbf{M} \models \varphi[a, \bigcup_{\delta<\gamma} I^\delta_\varphi(\mathbf{M})]$ by definition. By induction hypothesis on γ we have that $\mathbf{M} \models \varphi[a, \bigcup_{\delta<\gamma} I^\delta_{\varphi^*}(\mathbf{M})]$. Consider two cases: either $a \in \bigcup_{\delta<\gamma} I^\delta_{\varphi^*}(\mathbf{M})$ or $a \notin \bigcup_{\delta<\gamma} I^\delta_{\varphi^*}(\mathbf{M})$. In the first case we have that $a \in I^\gamma_{\varphi^*}(\mathbf{M})$ and we are done. In the second case, an easy induction on the structure of φ shows that $\mathbf{M} \models \varphi^*[a, \bigcup_{\delta<\gamma} I^\delta_{\varphi^*}(\mathbf{M})]$, from which $a \in I^\gamma_{\varphi^*}(\mathbf{M})$ follows by definition. The proof of the inclusion $I^\gamma_{\varphi^*}(\mathbf{M}) \subseteq I^\gamma_\varphi(\mathbf{M})$ is identical: $a \in I^\gamma_{\varphi^*}(\mathbf{M})$ implies $\mathbf{M} \models \varphi^*[a, \bigcup_{\delta<\gamma} I^\delta_{\varphi^*}(\mathbf{M})]$, which implies $\mathbf{M} \models \varphi^*[a, \bigcup_{\delta<\gamma} I^\delta_\varphi(\mathbf{M})]$ by induction hypothesis, which implies $a \in I^\gamma_\varphi(\mathbf{M})$ by a simple case analysis as before. Theorem 2 implies then that unary LFP(Δ_0) collapses to FO on $\mathcal{NFR}$. □

By Theorem 1, if LFP(Δ_0) collapses to FO on $\mathcal{BFR}$, then $\mathbf{P} \neq \mathbf{PSPACE}$. The proof of this theorem makes a crucial use of the hypothesis that k-ary LFP(Δ_0) collapses to FO for *every* $k \geq 1$. In view of Proposition 4, one may investigate whether k-ary LFP(Δ_0) collapses to FO on $\mathcal{BFR}$ for *particular* values of k bigger than 1. The main result of this section is that binary LFP(Δ_0) also collapses to FO on $\mathcal{BFR}$.

Theorem 3 *Let $\varphi(x_1, x_2, R)$ be a Δ_0-formula that is positive in the binary relation symbol R. The least fixed-point of φ and the greatest fixed-point of φ are first-order definable on $\mathcal{BFR}$. In fact, the least fixed-point is Π-definable, whereas the greatest fixed-point is Σ-definable on $\mathcal{BFR}$.*

Before we start the proof of Theorem 3, we need a short digression. In Example 9, we saw that the query Q_{rk} is Δ-definable on $\mathcal{NFR}$. We use this fact to show that for every $i \geq 0$, there exists a Σ-formula $\beta_i(x)$ such that for every $n \geq i$ and $V_n \subseteq M \subset V_{n+1}$ it holds that $\beta_i^M = V_{n-i}$. For $i > 1$, we let $\beta_i(x)$ be the formula

$$(\exists y)(Q_{\mathrm{rk}}(y) \wedge (\exists z_1 \in y) \cdots (\exists z_{i-1} \in y)(x \in z_1 \wedge \bigwedge_{j=1}^{i-2} z_j \in z_{j+1})).$$

For $i = 1$, let $\beta_i(x)$ be the formula $(\exists y)(Q_{\mathrm{rk}}(y) \wedge x \in y)$, and for $i = 0$ let $\beta_i(x)$ be the formula $(\exists y)(Q_{\mathrm{rk}}(y) \wedge (\forall z \in x)(z \in y))$. In all cases, the verification of the correctness is straightforward.

Proof of Theorem 3: As in the proof of Theorem 2, the first key idea is that the greatest fixed-point $J_\varphi(M)$ of φ on M can be approximated by a first-order formula. In fact, we can start with the Σ-formula $\theta(x_1, x_2)$ featured in that proof, because Claim 2 uses only the hypothesis that φ is a Δ_0-formula (and not the additional hypothesis in Theorem 2 that φ is restricted). Thus $J_\varphi(M) \cap V_{n-3} \subseteq \theta^M \subseteq J_\varphi(M)$ for every $M \in \mathcal{BFR}$ such that $V_n \subseteq M \subset V_{n+1}$. As a matter of fact, $(\beta_3(x_1) \wedge \beta_3(x_2) \wedge \theta(x_1, x_2))^M = J_\varphi(V_{n-3})$ according to the discussion preceding this proof. Our goal is to improve on this approximation of $J_\varphi(M)$ by defining Σ-formulas θ_i such that $J_\varphi(V_{n-3+i}) = \theta_i^{V_n}$, for $i = 1, 2, 3$ and $J_\varphi(M) = \theta_4^M$. Since $\varphi(x_1, x_2, R)$ may not be a restricted formula, a difficulty arises from the potential presence in φ of subformulas of the form $R(x_i, x_j)$, $R(x_i, z)$, and $R(z, x_i)$, where $i, j \in \{1, 2\}$ and z is a bound variable of φ. Actually, in building the formulas θ_i, the most serious difficulty is caused by the subformulas $R(x_i, z)$ and $R(z, x_i)$. Note that, since φ is Δ_0 and z is a bound variable of φ, every element of M witnessing z must be a member of $\mathrm{TC}(x_1) \cup \mathrm{TC}(x_2)$. Therefore, for every choice of x_i, the set of elements of M witnessing $R(x_i, z)$ (or $R(z, x_i)$) can be partitioned into three sets: a subset of x_1, a subset of x_2, and a subset of $(\mathrm{TC}(x_1) \cup \mathrm{TC}(x_2)) - (x_1 \cup x_2)$. In turn, this makes it possible to use first-order existential quantifiers over M to quantify these sets because every $M \in \mathcal{BFR}$ is closed under taking subsets and rank reduction.

We will build the desired Σ-formula θ_i from θ_{i-1}, for $i = 1, 2, 3, 4$, where we take θ_0 to be $\beta_3(x_1) \wedge \beta_3(x_2) \wedge \theta(x_1, x_2)$. Let S be the set $\{11, 12, 21, 22\}$. For each $T \subseteq S$ containing 12, we will define a formula $\delta_T(x_1, x_2)$. Then each $\theta_i(x_1, x_2)$ will be the formula

$$\beta_{3-i}(x_1) \wedge \beta_{3-i}(x_2) \wedge \Big(\bigvee_{T \in \mathcal{P}} \delta_T(x_1, x_2)\Big),$$

where $\mathcal{P}$ is the set of subsets of S containing 12. The first two subformulas of $\theta_i(x_1, x_2)$ are introduced to ensure that x_1 and x_2 belong to V_{n-3+i}. Here, $\beta_{-1}(x)$ is simply $x = x$. Let O be the set $\{10, 01, 20, 02\}$ and let $y^1_{j_1 j_2}, y^2_{j_1 j_2}, y^3_{j_1 j_2}$ be three new variables for each $j_1 j_2 \in O$. From now on, we use the abbreviations $\overline{x} = (x_1, x_2)$, $\overline{y} = (y^k_{j_1 j_2} : k = 1, 2, 3, j_1 j_2 \in O)$, and $\overline{z} = (z_1, z_2)$. Let $\delta_T(\overline{x})$ be the formula $(\exists \overline{y})(\delta'_T(\overline{x}, \overline{y}) \wedge \delta''_T(\overline{x}, \overline{y}))$ where

$$\begin{aligned} \delta'_T &\equiv \textstyle\bigwedge_{j_1 j_2 \in O} \rho_{j_1 j_2}(\overline{x}, \overline{y}), \\ \delta''_T &\equiv \textstyle\bigwedge_{i_1 i_2 \in T} \varphi(x_{i_1}, x_{i_2}, R(\overline{z})/\sigma_T(\overline{z}, \overline{x}, \overline{y})), \end{aligned}$$

and $\rho_{j_1 j_2}(\overline{x}, \overline{y})$ is the formula

$$\bigwedge_{k=1}^{3} (\forall x_0 \in y^k_{j_1 j_2})\varphi(x_{j_1}, x_{j_2}, R(\overline{z})/\sigma_T(\overline{z}, \overline{x}, \overline{y})),$$

while $\sigma_T(\overline{z}, \overline{x}, \overline{y})$ is the formula $\theta_{i-1}(\overline{z}) \vee \sigma'_T(\overline{z}, \overline{x}, \overline{y}) \vee \sigma''_T(\overline{z}, \overline{x}, \overline{y})$ where

$$\begin{aligned} \sigma'_T &\equiv \textstyle\bigvee_{k=1}^{3} \bigvee_{j_1 j_2 \in O} (\exists x_0 \in y^k_{j_1 j_2})(z_1 = x_{j_1} \wedge z_2 = x_{j_2}), \\ \sigma''_T &\equiv \textstyle\bigvee_{i_1 i_2 \in T} (z_1 = x_{i_1} \wedge z_2 = x_{i_2}), \end{aligned}$$

with x_0 a fresh variable. Note that $\rho_{j_1 j_2}$ says that $\bigcup_{k=1}^{3} y^k_{j_1 j_2}$ is the set of witnesses for bound variables that relate to x_1 or to x_2. Since θ_0 is a Σ-formula, it is easy to see that each θ_i is also a Σ-formula.

Claim 5 $\theta_i^M = J_\varphi(M) \cap V^2_{n-3+i}$ for $i = 0, \ldots, 4$.

Proof of Claim 5: The claim holds for $i = 0$, since θ_0 is $\beta_3(x_1) \wedge \beta_3(x_2) \wedge \theta(x_1, x_2)$. Fix $i \in \{1, 2, 3, 4\}$ and assume that the claim holds for $i - 1$; we show that it holds for i. We show first that $\theta_i^M \subseteq J_\varphi(M) \cap V^2_{n-3+i}$. Let $\overline{a} = (a_1, a_2) \in M^2$ be such that $M \models \theta_i[\overline{a}]$. From the definition of θ_i, it follows that $\overline{a} \in V^2_{n-3+i}$, and $M \models \delta_T[\overline{a}]$ for some $T \subseteq S$ with $12 \in T$. For every $k = 1, 2, 3$, and $j_1 j_2 \in O$, let $b^k_{j_1 j_2} \in M$ be a witness for the quantifier $\exists y^k_{j_1 j_2}$ in δ_T, and let $\overline{b}$ be the corresponding witness for $\overline{y}$. Since $12 \in T$, we have that $M \models \varphi[a_1, a_2, \sigma_T^M(\overline{a}, \overline{b})]$. It suffices to show that $\sigma_T^M(\overline{a}, \overline{b})$ is a post fixed-point of φ on M; then, the monotonicity of φ will imply that $\overline{a}$ is in $J_\varphi(M)$. We need to verify that $\sigma_T^M(\overline{a}, \overline{b}) \subseteq \varphi^M(\sigma_T^M(\overline{a}, \overline{b}))$. Let $\overline{c} \in M^2$ be such that $M \models \sigma_T[\overline{c}, \overline{a}, \overline{b}]$. Then $\overline{c}$ must satisfy one of the three disjuncts of σ_T. Assume first that $M \models \theta_{i-1}[\overline{c}]$. By absoluteness and the induction hypothesis that $\theta^M_{i-1} = J_\varphi(M) \cap V^2_{n-3+i-1} = J_\varphi(V_{n-3+i-1})$, we easily have that

θ^M_{i-1} is a post fixed-point of φ on M. Thus $M \models \varphi[\overline{c}, \theta^M_{i-1}]$. But $\theta^M_{i-1} \subseteq \sigma^M_T(\overline{a}, \overline{b})$ and hence, by monotonicity, $M \models \varphi[\overline{c}, \sigma^M_T(\overline{a}, \overline{b})]$, as required. Assume now that $\overline{c}$ satisfies the second disjunct. Then, there exist $k \in \{1, 2, 3\}$, $j_1 j_2 \in O$ and $a_0 \in b^k_{j_1 j_2}$ such that $\overline{c} = (a_{j_1}, a_{j_2})$. From the choice of $b^k_{j_1 j_2}$ and the definition of $\rho_{j_1 j_2}$, we know that $M \models \varphi[a_{j_1}, a_{j_2}, \sigma^M_T(\overline{a}, \overline{b})]$; therefore, $M \models \varphi[\overline{c}, \sigma^M_T(\overline{a}, \overline{b})]$, as required again. The case of the third disjunct is handled in a similar manner.

Consider next the inclusion $J_\varphi(M) \cap V^2_{n-3+i} \subseteq \theta^M_i$. If $\overline{a} \in J_\varphi(M) \cap V^2_{n-3+i}$, then $M \models \varphi[\overline{a}, J_\varphi(M) \cap V^2_{n-3+i}]$; note that if $i < 4$, then $J_\varphi(M) \cap V^2_{n-3+i} = J_\varphi(V_{n-3+i})$, and if $i = 4$, then $J_\varphi(M) \cap V^2_{n-3+i} = J_\varphi(M)$. Let T be the biggest subset of S containing 12 and such that for every $i_1 i_2 \in T$ it is the case that $M \models \varphi[a_{i_1}, a_{i_2}, J_\varphi(M) \cap V^2_{n-3+i}]$. Let A be the set

$$J_\varphi(M) \cap V^2_{n-3+i} \cap (\mathrm{TC}(a_1) \cup \mathrm{TC}(a_2) \cup \{a_1, a_2\})^2.$$

By Lemma 2 and monotonicity of φ, for every $i_1 i_2 \in T$ we have that $M \models \varphi[a_{i_1}, a_{i_2}, A]$. We construct witnesses $b^k_{j_1 j_2}$ for $y^k_{j_1 j_2}$ in δ_T such that $A \subseteq \sigma^M_T(\overline{a}, \overline{b})$, which will prove the claim. Fix $j_1 j_2 \in O$. We first define a set Q as follows: if $j_1 = 0$, then $Q = \{a \in \mathrm{TC}(a_1) \cup \mathrm{TC}(a_2) : (a, a_{j_2}) \in A\}$; if $j_2 = 0$, then $Q = \{a \in \mathrm{TC}(a_1) \cup \mathrm{TC}(a_2) : (a_{j_1}, a) \in A\}$. Then $b^1_{j_1 j_2} = Q \cap a_1$, $b^2_{j_1 j_2} = Q \cap a_2$, and $b^3_{j_1 j_2} = Q \cap ((\mathrm{TC}(a_1) \cup \mathrm{TC}(a_2)) - (a_1 \cup a_2))$. We need to check three properties: (i) $b^k_{j_1 j_2} \in M$, (ii) $A \subseteq \sigma^M_T(\overline{a}, \overline{b})$, and (iii) $b^k_{j_1 j_2}$ is a witness for $y^k_{j_1 j_2}$ in δ_T. For (i), observe that $b^1_{j_1 j_2} \subseteq a_1$, $b^2_{j_1 j_2} \subseteq a_2$, and $b^3_{j_1 j_2} \subseteq V_{n-3+i-2}$. Since each $M \in \mathcal{BFR}$ is closed under taking subsets and $\mathcal{P}(V_{n-3+i-2}) = V_{n-3+i-1} \subseteq V_n \subseteq M$, (i) follows. We prove (ii) next. Suppose that $\overline{c} \in A$. Then, one of the following cases holds, for some $i, j \in \{1, 2\}$:

(i) $\overline{c} \in J_\varphi(M) \cap V^2_{n-3+i-1}$,

(ii) $\overline{c} \in J_\varphi(M) \cap a_i \times \{a_j\}$,

(iii) $\overline{c} \in J_\varphi(M) \cap \{a_i\} \times a_j$,

(iv) $\overline{c} \in J_\varphi(M) \cap ((\mathrm{TC}(a_1) \cup \mathrm{TC}(a_2)) - (a_1 \cup a_2)) \times \{a_j\}$,

(v) $\overline{c} \in J_\varphi(M) \cap \{a_j\} \times ((\mathrm{TC}(a_1) \cup \mathrm{TC}(a_2)) - (a_1 \cup a_2))$,

(vi) $\overline{c} \in J_\varphi(M) \cap \{(a_i, a_j)\}$.

In the first case we have that $\overline{c} \in J_\varphi(V_{n-3+i-1})$, and therefore, by induction hypothesis on i we have that $\overline{c} \in \theta^M_{i-1}$. Thus, $\overline{c} \in \sigma^M_T(\overline{a}, \overline{b})$. The next two cases are handled by the second disjunct of σ_T and the choices of $b^1_{j_1 j_2}$ and $b^2_{j_1 j_2}$. The next two cases are handled by the second disjunct of σ_T and the choice of $b^3_{j_1 j_2}$. The last case is handled by the third disjunct of σ_T and the choice of T. It remains to see that $b^k_{j_1 j_2}$ is a witness for $y^k_{j_1 j_2}$ in δ_T. From the choice of T and the fact that $A \subseteq \sigma^M_T(\overline{a}, \overline{b})$ we have that $M \models \delta''_T[\overline{a}, \overline{b}]$. All we have to show is that $M \models \delta'_T[\overline{a}, \overline{b}]$. Observe that for every $a_0 \in b^k_{j_1 j_2}$ we have that $(a_{j_1}, a_{j_2}) \in A \subseteq J_\varphi(M)$. Therefore, $M \models \varphi[a_{j_1}, a_{j_2}, J_\varphi(M)]$. According to Lemma 2 and monotonicity of φ, we have that

$$M \models \varphi[a_{j_1}, a_{j_2}, J_\varphi(M) \cap (\mathrm{TC}(a_{j_1}) \cup \mathrm{TC}(a_{j_2}) \cup \{a_{j_1}, a_{j_2}\})^2].$$

But

$$J_\varphi(M) \cap (\mathrm{TC}(a_{j_1}) \cup \mathrm{TC}(a_{j_2}) \cup \{a_{j_1}, a_{j_2}\})^2 \subseteq A \subseteq \sigma^M_T(\overline{a}, \overline{b})$$

because $a_0 \in \mathrm{TC}(a_1) \cup \mathrm{TC}(a_2)$ and so $\mathrm{TC}(a_0) \subseteq \mathrm{TC}(a_1) \cup \mathrm{TC}(a_2)$. Therefore, $M \models \varphi[a_{j_1}, a_{j_2}, \sigma^M_T(\overline{a}, \overline{b})]$ by monotonicity. Conclude that $M \models \delta''_T[\overline{a}, \overline{b}]$. □

The inquisitive reader may wonder whether the argument of Theorem 3 extends to higher arities. The answer is that it does *not*, for the reason that we cannot encode binary relations on V_{n-1} by elements of V_n, whereas it is possible to encode unary relations (sets) on V_{n-1} by elements of V_n. We note, however, that Theorem 3 extends to binary Δ_0-formulas of an expanded vocabulary that, in addition to $\in$, contains other relation symbols, as long as they are interpreted by Δ-definable queries on $\mathcal{BFR}$.

Example 10 The aforementioned extension of Theorem 3 can be used to show that the *Even Cardinality* query

$$Q_{\text{even}}(M) = \{a \in M : \text{the cardinality of } a \text{ is even}\}$$

is first-order definable on $\mathcal{BFR}$. For this, one can write a Δ_0-formula $\varphi(x, y, R)$ of the vocabulary $\{\in, <, R\}$, whose least fixed-point defines the binary query "y is an even element of x with respect to the linear order $<$" on $\mathcal{BFR}$. Here, $<$ is interpreted by the Δ-definable linear order on $\mathcal{BFR}$ described in the beginning of Section 3.3.

3.5 On the Logical Complexity of the Formulas

We turn next to a discussion on the logical complexity of the first-order formulas defining the fixed-points. We saw already that these formulas can be chosen to be Σ or Π. It is well known that on V_ω, every Σ-formula is equivalent to a Σ_1-formula, that is, a Σ-formula all whose unbounded quantifiers precede the bounded ones. This is called the Σ-reflection principle [Bar75]. Dually, every Π-formula is equivalent to a Π_1-formula. Unfortunately, this does not hold in general on finite substructures of V_ω and therefore we cannot conclude directly that the Σ-formula ψ of Theorem 2 is equivalent to a Σ_1-formula. Quite interestingly, it turns out that in Theorem 2, the formula ψ is actually equivalent to a Σ_1-formula on $\mathcal{NFR}$. The main idea is that, the only unbounded existential quantifier in ψ has an absolute witness. We discuss this fact more carefully.

Recall that ψ was obtained from the syntactic iteration of $\varphi(x_1, \ldots, x_k, R)$ over the formula

$$\theta(x_1, \ldots, x_k) = (\exists y)(\mathrm{post}(y) \wedge \langle x_1, \ldots, x_k \rangle \in y).$$

The formula $\theta(x_1, \ldots, x_k)$ defines a superset of the greatest fixed-point of φ on V_{n-2k+1}. The syntactic iterations of φ over θ complete the definition of the greatest fixed-point of φ on the remaining $2k - 1$ levels (a constant number of them). Graphically, the formula ψ takes the form

$$\psi(x_1, \ldots, x_n) = \varphi(x_1, \ldots, x_n, R/\varphi(R/\varphi(\cdots\varphi(R/\theta)\cdots))).$$

Since φ is positive in R and Δ_0, and since θ is Σ_1, the resulting formula is Σ with a unique unbounded existential quantifier $(\exists y)$. Recall that the intended meaning of that quantifier is the existence of a post fixed-point of φ on V_{n-2k+1} that contains the tuple $(x_1, \ldots, x_k)$. Such a post fixed-point can be encoded within the structure. But we know that the greatest-fixed-point is the maximum of all post fixed-points with respect to inclusion; therefore, if a witness exists for y, the encoding of $J_\varphi(V_{n-2k+1})$ would also be a witness for it. The absoluteness of this witness allows us to pull out the existential quantifier to the front of ψ to obtain a Σ_1-formula. We formalize this argument in the following claim. Let $\chi(x_1, \ldots, x_k, y)$ be the formula $\mathrm{post}(y) \wedge \langle x_1, \ldots, x_k \rangle \in y$, so that $\theta(x_1, \ldots, x_k) = (\exists y)\chi(x_1, \ldots, x_k, y)$.

Lemma 4 $M \models \varphi(x_1, \ldots, x_n, R/(\exists y)\chi) \leftrightarrow (\exists y)\varphi(x_1, \ldots, x_n, R/\chi)$.

Proof: Since φ is positive in R, the variable y is existentially quantified in the formula $\varphi(\overline{x}, R/(\exists y)\chi)$. Therefore, as y does not appear in φ and we can always push down an existential quantifier, $M \models (\exists y)\varphi(\overline{x}, R/\chi) \rightarrow \varphi(\overline{x}, R/(\exists y)\chi)$. The other direction is proved by induction on the structure of φ. Assume that all negations occur in front of the atoms.

If φ is atomic or negated atomic of the form $x_i \in x_j$, $x_i \notin x_j$, $x_i = x_j$ or $x_i \neq x_j$ then this is clear. If φ is atomic of the form $R(x_{i_1}, \ldots, x_{i_k})$ then, if $M \models \varphi(\overline{x}, R/(\exists y)\chi)[\overline{a}]$ we have that $M \models (\exists y)\chi(\overline{x}, y)[\overline{a}]$, and therefore $M \models (\exists y)\varphi(\overline{x}, R/\chi)[\overline{a}]$. Moreover, since φ is positive in R, it cannot be atomic of the form $\neg R(x_{i_1}, \ldots, x_{i_k})$. The case $\varphi = \psi \wedge \theta$ will be treated together with the case $\varphi = (\forall x \in x_i)\psi$. If $\varphi = (\exists x \in x_i)\psi$ then $M \models \varphi(\overline{x}, R/(\exists y)\chi)[\overline{a}]$ implies that $M \models \psi(\overline{x}, x, R/(\exists y)\chi)[\overline{a}, a]$ for some $a \in M$ such that $a \in a_i$. By induction hypothesis, $M \models (\exists y)\psi(\overline{x}, x, R/\chi)[\overline{a}, a]$. As we can always swap existential quantifiers we have that $M \models (\exists y)(\exists x \in x_i)\psi(\overline{x}, x, R/\chi)[\overline{a}]$ and therefore $M \models (\exists y)\varphi(\overline{x}, R/\chi)[\overline{a}]$.

Consider finally the case $\varphi = (\forall x \in x_i)\psi$; the case $\varphi = \psi \wedge \theta$ is similar. Suppose that $M \models \varphi(\overline{x}, R/(\exists y)\chi)[\overline{a}]$ and fix any $a \in M$ such that $a \in a_i$. Then $M \models \psi(\overline{x}, x, R/(\exists y)\chi)[\overline{a}, a]$ and by induction hypothesis $M \models (\exists y)\psi(\overline{x}, x, R/\chi)[\overline{a}, a]$. Let $b \in M$ be a witness for y in the above satisfaction. Recall that $\chi(\overline{x}, y) = \mathrm{post}(y) \wedge \langle \overline{x} \rangle \in y$ and in particular b needs to be the encoding of a relation on V_{n-2k+1}. Therefore, $M \models \chi(\overline{x}, y)[\overline{a}, b]$ if and only if b is the encoding of a post fixed-point of φ on V_{n-2k+1} and $\langle \overline{a} \rangle \in b$. But the greatest fixed-point is the union of all post fixed-points and therefore $\langle J_\varphi(V_{n-2k+1})\rangle \in M$ is a witness for y as good as b. Let $c = \langle J_\varphi(V_{n-2k+1})\rangle$. Now, $M \models \psi(\overline{x}, x, R/\chi(\overline{x}, y))[\overline{a}, a, c]$. But a was arbitrarily chosen in a_i and therefore $M \models (\forall x \in x_i)\psi(\overline{x}, x, R/\chi(\overline{x}, y))[\overline{a}, c]$; that is, $M \models (\exists y)\varphi(\overline{x}, R/\chi)[\overline{a}]$ as needed. As said, the case $\varphi = \psi \wedge \theta$ is very similar; the absolute witness for y is again c. □

From the claim we know that the only unbounded existential quantifier in ψ can be pulled up from the front of the subformula χ to the front of the subformula θ_1. But y does not occur in φ and therefore, it can actually be pulled out to the front of $\theta_{2k} = \psi$ itself. The resulting Σ_1-formula is

$$(\exists y)\varphi(\overline{x}, R/\varphi(R/\varphi(\cdots\varphi(R/\chi)\cdots)))$$

Overall, we built a proof for the following:

Theorem 4 *Let $\varphi(x_1, \ldots, x_k, R)$ be a restricted Δ_0-formula that is positive in the k-ary relation symbol R. The least fixed-point of φ is Δ_1-definable on $\mathcal{NFR}$.*

As mentioned before, the fact that the linear order is FO-definable on BIT structures follows from the fact that its inductive definition is restricted Δ_0. Therefore, the preceding theorem not only shows that the canonical linear order is FO-definable on $\mathcal{NFR}$, but also establishes its Δ_1-definability.

We conclude this section with the observation that we have not used the hypothesis that φ is restricted in the proof of Lemma 4. This indicates that a similar argument could be carried out to prove that the least fixed-points of unary and binary Δ_0-formulas are Δ_1-definable. As a matter of fact, the unary case is immediate from its proof and does not need any new argument. However, the binary case looks much more complicated due to the existential quantifiers in the formulas δ_T.

Chapter 4

Descriptive Complexity of the Fixed-Points of Bounded Formulas

4.1 Key Lemma

In this chapter we will focus on the complexity-theoretic aspects of the least fixed-points of positive Δ_0-formulas. First we will analyze their closure functions, that is, the number of steps that the formula needs to be iterated until it reaches the least fixed-point. This is a measure of their complexity. Then we will use these results to obtain an exact characterization of the complexity classes underlying the fragments of LFP. Before we start with these tasks, we need to recall a key lemma from Chapter 3 and re-formulate it in a slightly different form. We will be interested in arbitrary finite structures with built-in membership relation. Recall that $\mathcal{BFR}$ is the class of structures of the form $(P_n, \in)$, where $P_n = \{e(0), \ldots, e(n-1)\}$ and e is the Ackermann isomorphism.

Definition 11 *Let L be a relational vocabulary. We say that a finite structure $\mathbf{M}$ over $L \cup \{\in\}$ has built-in membership relation if $(M, \in^{\mathbf{M}}) \in \mathcal{BFR}$, where M is the universe of $\mathbf{M}$.*

Recall that the transitive closure of a set a, denoted by $\mathrm{TC}(a)$, is defined inductively as follows: $\mathrm{TC}(a) = \bigcup\{\mathrm{TC}(b) : b \in a\}$. The reflexive transitive closure of a, denoted by $\mathrm{RTC}(a)$, is $\{a\} \cup \mathrm{TC}(a)$. The Lemma says that the satisfiability of a Δ_0-formula only depends on the reflexive transitive closure of its arguments.

Lemma 5 *Let L be a relational vocabulary, let $\mathbf{M}$ be a structure over $L \cup \{\in\}$ with built-in membership relation, and let $\varphi(x_1, \dots, x_s, X)$ be a Δ_0-formula of $L \cup \{\in, X\}$, where X is a k-ary relation variable. For every $A \subseteq M^k$, and every tuple $\overline{a} = (a_1, \dots, a_s) \in M^s$, we have that $\mathbf{M} \models \varphi[\overline{a}, A]$ if and only if $\mathbf{M} \models \varphi[\overline{a}, A \cap (\bigcup\{\mathrm{RTC}(a_i) : i \in \{1, \dots, s\}\})^k]$. Moreover, if φ is restricted Δ_0 and $\overline{a} \in V^s_{m+1}$ for some $m \geq 0$, then $\mathbf{M} \models \varphi[\overline{a}, A]$ if and only if $\mathbf{M} \models \varphi[\overline{a}, A \cap V^k_m]$.*

Proof: The proof of this is entirely identical to that of Lemma 2 in Chapter 3. The proof by induction requires the definition of the set of R-free indices $U(\varphi)$, the set of free indices $F(\varphi)$, and a slightly stronger induction hypothesis. Namely, that $\mathbf{M} \models \varphi[\overline{a}, A]$ if and only if

$$\mathbf{M} \models \varphi[\overline{a}, A \cap (\bigcup\{\mathrm{TC}(a_i) : i \in F(\varphi)\} \cup \{a_i : i \in U(\varphi)\})^k].$$

Then, since $\bigcup\{\mathrm{TC}(a_i) : i \in F(\varphi)\} \cup \{a_i : i \in U(\varphi)\} \subseteq \{\mathrm{RTC}(a_i) : i \in \{1, \dots, s\}\}$, the statement of the Lemma follows. □

We will also need a slightly more general form of Lemma 3 to work with arbitrary relational vocabularies without constants.

Lemma 6 *Let L be a relational vocabulary, let $\mathbf{M}$ be a structure over $L \cup \{\in\}$ with built-in membership relation, and let $\varphi(x_1, \ldots, x_k, X)$ be a Δ_0-formula of $L \cup \{\in, X\}$ that is positive in the k-ary relation variable X. For every $m \geq 0$, let $\mathbf{M}_m$ be the substructure of $\mathbf{M}$ with universe $M \cap V_m$. Then, $I^t_\varphi(\mathbf{M}) \cap V^k_m = I^t_\varphi(\mathbf{M}_m)$ for every $t \geq 0$.*

Proof: The lemma is proved by induction on t as follows. Assume that for every $s < t$ we have that $I^s_\varphi(\mathbf{M}) \cap V^k_m = I^s_\varphi(\mathbf{M}_m)$. Let $\overline{a} \in V^k_m$. By definition, $\overline{a} \in I^t_\varphi(\mathbf{M})$ if and only if $\mathbf{M} \models \varphi(\overline{x}, R)[\overline{a}, \bigcup_{s<t} I^s_\varphi(\mathbf{M})]$. Note that $\mathbf{M}$ is an end-extension of $\mathbf{M}_m$. By absoluteness, $\mathbf{M} \models \varphi(\overline{x}, R)[\overline{a}, \bigcup_{s<t} I^s_\varphi(\mathbf{M})]$ if and only if $\mathbf{M}_m \models \varphi(\overline{x}, R)[\overline{a}, (\bigcup_{s<t} I^s_\varphi(\mathbf{M})) \cap V^k_m]$, which in turn is equivalent to $\mathbf{M}_m \models \varphi(\overline{x}, R)[\overline{a}, \bigcup_{s<t}(I^s_\varphi(\mathbf{M}) \cap V^k_m)]$ by distributing $\cup$ over $\cap$. By induction hypothesis, we have $I^s_\varphi(\mathbf{M}) \cap V^k_m = I^s_\varphi(\mathbf{M}_m)$ so that $\mathbf{M}_m \models \varphi(\overline{x}, R)[\overline{a}, \bigcup_{s<t}(I^s_\varphi(\mathbf{M}) \cap V^k_m)]$ if and only if $\mathbf{M}_m \models \varphi(\overline{x}, R)[\overline{a}, \bigcup_{s<t} I^s_\varphi(\mathbf{M}_m)]$. By definition, this means $\overline{a} \in I^t_\varphi(\mathbf{M}_m)$ as required. □

4.2 Closure Functions of Positive Δ_0-Formulas

Let $\varphi(x_1, \ldots, x_k, R)$ be an arbitrary positive first-order formula. From the preliminaries, recall that if $\mathbf{M}$ is a finite structure, then $\mathrm{cl}_\varphi(\mathbf{M})$ is the smallest integer m such that $I^m_\varphi(\mathbf{M}) = \bigcup_{m'<m} I^{m'}_\varphi(\mathbf{M})$. In general, the rate of growth of $\mathrm{cl}_\varphi(\mathbf{M})$ can be as high as a polynomial in the cardinality of the universe of $\mathbf{M}$. Here, we analyze the rate of growth of the closure function of positive Δ_0-formulas. We establish that if φ is a restricted Δ_0-formula, then $\mathrm{cl}_\varphi(\mathbf{M})$ is bounded by the iterated logarithm of $|M|$, and if φ is a general Δ_0-formula, then $\mathrm{cl}_\varphi(\mathbf{M})$ is bounded by a polylogarithm of $|M|$.

Theorem 5 *Let L be a relational vocabulary, and let $\varphi(x_1, \ldots, x_k, X)$ be a restricted Δ_0-formula of $L \cup \{\in, X\}$ that is positive in the k-ary relation variable X. Then,*

$$\mathrm{cl}_\varphi(\mathbf{M}) \leq 2 + \log^*(|M|)$$

for every finite structure $\mathbf{M}$ over $L \cup \{\in\}$ with built-in membership relation.

Proof: Let $p \geq 0$ be such that $V_p \subset M \subseteq V_{p+1}$. For every $n \geq 1$, let $\mathbf{M}_n$ be the substructure of $\mathbf{M}$ with universe $M \cap V_n$. We show that $\mathrm{cl}_\varphi(\mathbf{M}_n) \leq \mathrm{cl}_\varphi(\mathbf{M}_{n-1}) + 1$ holds for all $n \geq 1$. Since $\mathrm{cl}_\varphi(\mathbf{M}_0) \leq 1$, it will follow that

$$\mathrm{cl}_\varphi(\mathbf{M}) \leq p + 1 \leq 2 + \log^*(|V_p|) \leq 2 + \log^*(|M|).$$

Let m be the closure ordinal of φ on $\mathbf{M}_{n-1}$. It is enough to show that $I_\varphi^{m+2}(\mathbf{M}_n) \subseteq I_\varphi^{m+1}(\mathbf{M}_n)$. So let us assume that $\overline{a} \in I_\varphi^{m+2}(\mathbf{M}_n)$, so that $\mathbf{M}_n \models \varphi[\overline{a}, I_\varphi^{m+1}(\mathbf{M}_n)]$. Lemma 5 implies that $\mathbf{M}_n \models \varphi[\overline{a}, I_\varphi^{m+1}(\mathbf{M}_n) \cap V_{n-1}^k]$, since φ is restricted and $\overline{a} \in V_n$. By Lemma 6 we have $I_\varphi^{m+1}(\mathbf{M}_n) \cap V_{n-1}^k = I_\varphi^{m+1}(\mathbf{M}_{n-1})$, and the choice of m implies $I_\varphi^{m+1}(\mathbf{M}_{n-1}) = I_\varphi^{m}(\mathbf{M}_{n-1})$. It follows that $\mathbf{M}_n \models \varphi[\overline{a}, I_\varphi^{m}(\mathbf{M}_{n-1})]$. Since $I_\varphi^{m}(\mathbf{M}_{n-1}) \subseteq I_\varphi^{m}(\mathbf{M}_n)$ by Lemma 6 again, monotonicity gives $\mathbf{M}_n \models \varphi[\overline{a}, I_\varphi^{m}(\mathbf{M}_n)]$. Therefore, $\overline{a} \in I_\varphi^{m+1}(\mathbf{M}_n)$ as required. □

Our second result concerns the closure functions of positive Δ_0-formulas that are not necessarily restricted. Since the reflexive transitive closure of a finite set is relatively small, Lemma 5 allows us to put polylogarithmic bounds on the closure functions of Δ_0-formulas.

Theorem 6 *Let L be a relational vocabulary, and let $\varphi(x_1, \ldots, x_k, X)$ be a Δ_0-formula of $L \cup \{\in, X\}$ that is positive in the k-ary relation variable X. Then,*

$$\mathrm{cl}_\varphi(\mathbf{M}) \leq (\log(|M| - 1) + k)^k$$

for every finite structure $\mathbf{M}$ over $L \cup \{\in\}$ with built-in membership relation.

Proof: During this proof, we will identify a natural number n and its encoding as a finite set $e(n)$. Thus, $M = \{0, \ldots, |M|-1\}$. Put $t = (\log(|M|-1)+k)^k$, and assume for contradiction that $\mathrm{cl}_\varphi(\mathbf{M}) > t$. Let $\overline{a}_0 = (a_{0,1}, \ldots, a_{0,k}) \in I_\varphi(\mathbf{M})$ be such that $|\overline{a}_0| > t$, where $|\overline{a}|$ denotes the minimal m such that $\overline{a} \in I^m_\varphi(\mathbf{M})$ if $\overline{a} \in I_\varphi(\mathbf{M})$, and ∞ if $\overline{a} \notin I_\varphi(\mathbf{M})$. In the following, let I^m be an abbreviation for $I^m_\varphi(\mathbf{M})$. We build a sequence $\overline{a}_0, \overline{a}_1, \ldots, \overline{a}_t$ such that $|\overline{a}_i| = |\overline{a}_0| - i$, and $\overline{a}_i \in S^k$ for every $i = 0, \ldots, t$, where $S = \{0, \ldots, \log(|M| - 1) - 1\} \cup \{a_{0,1}, \ldots, a_{0,k}\}$. This will prove the theorem since the cardinality of S^k is at most t.

For every $\overline{a} = (a_1, \ldots, a_k) \in M^k$, let $S(\overline{a})$ denote the set $\{0, \ldots, \log(|M| - 1)\} \cup \{a_1, \ldots, a_k\}$. Observe that $\bigcup\{\mathrm{RTC}(a_i) : i \in F(\varphi)\} \subseteq S(\overline{a})$ since every element in $\mathrm{TC}(a_i)$ is a *bit* position of an element in $\{0, \ldots, |M| - 1\}$. Assuming $\overline{a}_i = (a_{i,1}, \ldots, a_{i,k})$ is already defined, we define $\overline{a}_{i+1} = (a_{i+1,1}, \ldots, a_{i+1,k})$. Let $m = |\overline{a}_i|$. Then, $\mathbf{M} \models \varphi[\overline{a}_i, I^{m-1}]$. Lemma 5 and monotonicity imply that $\mathbf{M} \models \varphi[\overline{a}_i, I^{m-1} \cap S(\overline{a}_i)^k]$. Observe that since $\overline{a}_i \in S(\overline{a}_0)^k$ by assumption, we have that $S(\overline{a}_i) \subseteq S(\overline{a}_0)$. Now let us consider two cases: (i) $I^{m-1} \cap S(\overline{a}_0)^k \subseteq I^{m-2}$, or (ii) $I^{m-1} \cap S(\overline{a}_0)^k \not\subseteq I^{m-2}$. In case (i) we have that $\mathbf{M} \models \varphi[\overline{a}_i, I^{m-2}]$ by monotonicity. Hence, $\overline{a}_i \in I^{m-1}$ which contradicts the minimality of $m = |\overline{a}_i|$. In case (ii), there must exist some $\overline{a}_{i+1} \in I^{m-1} \cap S(\overline{a}_0)^k$ that does not belong to I^{m-2}. Observe that $|\overline{a}_{i+1}| = |\overline{a}_i| - 1$, and $\overline{a}_{i+1} \in S(\overline{a}_0)^k$ as required. This completes the proof of the theorem. □

We can show that the upper bounds provided by Theorem 5 are tight. First, it can be shown that the restricted Δ_0-formula whose least fixed-point defines the standard linear order requires $\log^*(|V_n|)$ steps to close in V_n. Indeed, the induction *grows by ranks*, which means that V_0 is linearly ordered first, then V_1 is linearly ordered, then V_2, and so on. This takes $n = 1 + \log^*(|V_n|)$ steps on V_n.

Second, we construct below a 4-ary positive Δ_0-formula that requires $\log(|V_n|)$ steps to close on V_n; it should be an easy exercise to extend the construction to all polylogarithmic bounds. Our strategy will be to build a *system of simultaneous inductions*, and then translate the system into a single Δ_0-formula that implements it.

Recall that every first-order formula $\varphi(x_1, \ldots, x_r, X)$ that is positive in the r-ary relation variable X defines a monotone operator from M^r to M^r on every structure $\mathbf{M}$. For the same reason, a pair of formulas $(\varphi_1(x_1, \ldots, x_r, X, Y), \varphi_2(x_1, \ldots, x_s, X, Y))$ that are both positive in the r-ary relation symbol X and the s-ary relation symbol Y defines a monotone operator from $M^r \times M^s$ to $M^r \times M^s$. Since the operator is monotone, it reaches a least fixed-point $I_{\varphi_1,\varphi_2}(\mathbf{M}) = (X^\infty(\mathbf{M}), Y^\infty(\mathbf{M}))$ that is called the simultaneous least fixed-point of φ_1 and φ_2 on $\mathbf{M}$.

Consider the following system of positive Δ_0-formulas:

$$\begin{aligned}
\varphi_1(x, y, R, S) \quad &\equiv \quad (\exists y' \in y)(\forall x' \in x)(y' \notin y \wedge (x' \notin x \rightarrow R(x', y')))\\
\varphi_2(x, y, R, S) \quad &\equiv \quad x = \emptyset \vee (\exists u \in y)(R(u, x) \wedge\\
&\qquad (\forall v \in y)(R(v, u) \vee R(x, v)) \wedge S(u, y)).
\end{aligned}$$

Here, $x = \emptyset$ is an abbreviation for $(\forall u \in x)(x \neq x)$. We first show that the simultaneous least fixed-point $I_{\varphi_1,\varphi_2}(V_n) = (R^\infty, S^\infty)$ requires $\log(|V_n|)$ steps to be reached. Observe that φ_1 is simply defining the linear order $\prec$ on V_n, and is independent of S; therefore, in $\log^*(|V_n|)$ steps, the relation R will be the linear order. Consider next the set $A = \{(a, V_{n-1}) : a \in V_{n-1}\} \subseteq V_n^2$. We claim that $A \subseteq S^\infty$ and that each pair in A gets into S^∞ in a different stage. Since $|A| = |V_{n-1}| = \log(|V_n|)$, the claim will follow. Suppose that $a_0 \prec a_1 \prec \cdots \prec a_r$ is the complete enumeration of V_{n-1}. From the construction of φ_2 we have that for each i such that $0 \leq i < r$, and each stage t, the pair (a_{i+1}, V_{n-1}) does not get into S^t if (a_i, V_{n-1}) is not already in S^{t-1}. Moreover, each such pair eventually gets into S^∞ because $(a_0, V_{n-1}) = (\emptyset, V_{n-1})$ gets in in the first stage. This completes the argument that I_{φ_1,φ_2} requires $\log(|V_n|)$ steps to be reached.

We now build the 4-ary Δ_0-induction that implements the above system. The standard way of doing this requires at least two constants or, in case constants are not available, existential quantification [Mos74]. Fortunately, we are lucky and in this very particular case we can keep within Δ_0-formulas without constants. One of the reasons is that S does not occur in φ_1 as we will see. Let T be a new 4-ary relation symbol, and consider the following Δ_0-formula:

$$\begin{aligned}
&\varphi(s,t,x,y,T) \equiv \\
&\qquad (s = t \wedge \varphi_1(x,y,R/\{(u,v) : T(s,s,u,v)\}, S/s = s)) \vee \\
&\qquad (s \neq t \wedge \varphi_2(x,y,R/\{(u,v) : T(s,s,u,v)\}, S/\{(u,v) : T(s,t,u,v)\})).
\end{aligned}$$

It is easily seen that the least fixed-point of φ implements the simultaneous least fixed-point of φ_1 and φ_2 in the following sense: the projection $(\exists s)(\exists t)(s = t \wedge T^\infty(s,t,x,y))$ coincides with R^∞, and the projection $(\exists s)(\exists t)(s \neq t \wedge T^\infty(s,t,x,y))$ coincides with S^∞. Moreover, the same argument as before shows that the closure ordinal is at least $\log(|V_n|)$ on V_n.

The polylogarithmic bound is also tight in the following sense: there exists a positive Π-formula that requires $|M|$ steps to close on M. Such an example is the following induction defined in terms of the linear order:

$$\varphi(x,S) \quad \equiv \quad x = \emptyset \vee (\forall u)(u \prec x \rightarrow S(u)).$$

Since the linear order $\prec$ is Δ-definable on $\mathcal{NFR}$, the formula $\varphi(x,S)$ can be written as a Π-formula. It should be clear that $I_\varphi(M) = M$ but each element gets in in a different stage. Therefore, the closure function of φ grows linearly with $|M|$. The question whether some Σ-formula requires a linear number of steps to close is open.

4.3 Complexity of Evaluating $\mathrm{LFP}(\Delta_0)$

Let $L = \{R_1, \ldots, R_s\}$ be a relational vocabulary, and let $\mathbf{M} = (M, R_1^{\mathbf{M}}, \ldots, R_s^{\mathbf{M}})$ be a finite structure over L. We will always identify the universe of $\mathbf{M}$, denoted M, with the initial segment of the natural numbers of cardinality $|M|$; thus, $M = \{0, \ldots, |M|-1\}$.

Finite structures are encoded as words over the alphabet $\{0,1,\#\}$ according to the following convention. For every relation symbol $R_i \in L$ of arity r, we let $\chi(R_i^{\mathbf{M}})$ be the characteristic sequence of $R_i^{\mathbf{M}}$. That is, $\chi(R_i^{\mathbf{M}}) = a_0 a_1 \ldots a_{n^r-1}$, where $a_m \in \{0,1\}$, and $a_m = 1$ if and only if $(m_{r-1}, \ldots, m_0) \in R_i^{\mathbf{M}}$ where $(m_{r-1}, \ldots, m_0)$ is the n-ary representation of m. That is, $m = \sum_{i=0}^{r-1} m_i n^i$. Then, the encoding of $\mathbf{M}$ is just

$$\langle \mathbf{M} \rangle = 1^n \# \chi(R_1^{\mathbf{M}}) \# \ldots \# \chi(R_s^{\mathbf{M}}).$$

We extend the encoding to include individuals as follows. For every $a_1, \ldots, a_k \in M$, let $\langle \mathbf{M}, a_1, \ldots, a_k \rangle = \langle \mathbf{M} \rangle \# \mathrm{b}_n(a_1) \# \ldots \# \mathrm{b}_n(a_k)$.

Definition 12 *Let L be a relational vocabulary, let D be a class of finite structures over L, and let Q be a k-ary query on D. We say that Q is computable in a complexity class $\mathcal{C}$ on D if there exists a language $A \in \mathcal{C}$ such that for every $\mathbf{M} \in D$ and $a_1, \ldots, a_k \in M$, we have that $(a_1, \ldots, a_k) \in Q(\mathbf{M})$ if and only if $\langle \mathbf{M}, a_1, \ldots, a_k \rangle \in A$.*

The polylogarithmic bounds on the closure functions of positive Δ_0-formulas provided by Theorem 6, together with a result of Immerman [Imm89], imply that every query that is definable as the fixed-point of a Δ_0-formula is computable in $\mathbf{NC}$, the parallel complexity class. However, we can keep the machine sequential as noted in the following

Lemma 7 *Let L be a relational vocabulary, let C be a class of finite structures over L with built-in membership relation, and let $\varphi(x_1, \ldots, x_k, X)$ be a Δ_0-formula that is positive in the k-ary relation variable X. Then, the query on C defined by the formula $(\mathrm{LFP}_{\overline{x},X}\varphi)$ is computable in $\mathbf{DPOLYLOGTIME}$ on C.*

Proof: The idea is that the standard fixed-point computation will only take a polylogarithmic number of iterations by Theorem 6, and each iteration is computable in polylogarithmic-time because φ is a Δ_0-formula. More precisely, on input $\langle \mathbf{M}, a_1, \ldots, a_k \rangle$, the polylogarithmic-time Turing machine will proceed as follows. The machine first determines the cardinality of M, say n. To this end, it determines the length m of the input in $O(\log m)$ steps using its random access to the input (see the preliminaries for this trick), and then it executes a straightforward computation to extract n from m (here we use the fact that our encodings are carefully chosen so that their length is determined by the cardinality of $\mathbf{M}$, the arities of the relation symbols in L, and k). Let $B = \{0, \ldots, \log(n-1) - 1\} \cup \{a_1, \ldots, a_k\}$. The machine will keep, in a separate tape, an encoding of a k-ary relation on B; this will require $O((\log n)^k)$ bits of information. Then, it starts a loop that is to be repeated $(\log(n-1) + k)^k$ times. In each iteration, the machine cycles through all k-tuples $(b_1, \ldots, b_k)$ in B^k, and evaluates $\mathbf{M} \models \varphi[b_1, \ldots, b_k, R]$ where R is the k-ary relation encoded in the separate tape. Atomic formulas from L are resolved by random access to the input, and atomic formulas of the form $X(u_1, \ldots, u_k)$ are resolved by accessing the position of tuple $(u_1, \ldots, u_k)$ in the encoding of R. Observe that each relevant tuple $(u_1, \ldots, u_k)$ will be available since $\mathbf{M} \models \varphi[b_1, \ldots, b_k, R]$ if and only if

$$\mathbf{M} \models \varphi[b_1, \ldots, b_k, R \cap (\bigcup\{\mathrm{RTC}(b_i) : i \in F(\varphi)\})^k],$$

and $\bigcup\{\mathrm{RTC}(b_i) : i \in F(\varphi)\} \subseteq B$ for every $(b_1, \ldots, b_k) \in B^k$, since every element of $\mathrm{TC}(a_i)$ is a *bit* position of an element in $\{0, \ldots, n-1\}$. For the same reason, each quantifier is bounded by some b_i, and therefore, the variable it bounds ranges over at most $\log(n-1)$ elements of the universe. Hence, the computation can be done in time $O((\log n)^r)$ where r depends on the number of quantifiers of φ. When the evaluation of $\mathbf{M} \models \varphi[b_1, \ldots, b_k, R]$ is complete, the machine updates accordingly the position corresponding to tuple $(b_1, \ldots, b_k)$ in the encoding of R. Finally, the machine will only have to check whether the tuple $(a_1, \ldots, a_k)$ belongs to R at the end of the loop. □

4.4 Circuit Uniformity and the Complexity Class AC^0

It is traditional in the area of computational complexity to use the Turing machine as the underlying computational model. This is partly due to a natural inheritance from the field of recursion theory, and partly due to the very convenient robustness properties that the model has. No less interesting is the Boolean circuit model, which might seem closer to what actual machines of our time really do. The truth, however, is that these two models are tightly related through the theory of *uniformity of circuits*. Since we will make strong use of it in this chapter, let us devote this whole section to it.

A Boolean circuit is a directed acyclic graph with exactly one node of out-degree zerp, whose nodes are called gates. The gates of in-degree zero are called the inputs, and the gate of out-degree zero is called the output. Each input gate is labeled by a Boolean variable x_i, and each non-input gate is labeled by a Boolean function $g : \{0,1\}^k \rightarrow \{0,1\}$ whose arity k coincides with the in-degree of the node. The circuit computes a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ in the obvious way. Here n is the number of inputs. Typically, each gate of the circuit has in-degree at most two, and the Boolean functions labeling the gates are taken from a complete basis such as $\{AND, OR, NOT\}$. Also, we may assume that all NOT gates appear before all AND or OR gates by using the De Morgan rule. We note that this transformation leads to a circuit whose number of gates is at most twice the original.

It is intuitively obvious that the larger the circuit, the more complex the Boolean function that it computes can be. This leads to a classification of Boolean functions by the size of its minimal circuit computing it, where size is measured by the number of gates of the circuit. Moreover, certain structural properties of the circuit, such as its depth measured by the length of the longest path, may increase or decrease its computational power.

Definition 13 *Let C be a Boolean circuit with n inputs. The size of C, denoted by $s(C)$, is the number of gates of C. The depth of C, denoted by $d(C)$, is the length of the longest directed path of C. The alternation-depth of C, denoted by $a(C)$, is the maximum number of alternations between AND and OR gates in any directed path of C.*

Boolean circuits are thought to compute Boolean functions. However, we can also think of families of Boolean circuits as recognizers of languages over the alphabet $\{0,1\}$ through the so-called characteristic function:

Definition 14 *Let $L \subseteq \{0,1\}^*$ be a language, and let $C_1, C_2, \ldots$ be a family of circuits such that C_n has exactly n inputs and one output. We say that $C_1, C_2, \ldots$ decides L if and only if for every $n \geq 1$ and every $(a_1, \ldots, a_n) \in \{0,1\}^n$, the circuit C_n outputs 1 on input $(a_1, \ldots, a_n)$ if and only if $a_1 \ldots a_n \in L$.*

These definitions lead us to the definition of natural complexity classes by imposing natural restrictions on size, depth, and alternation-depth of circuits.

Definition 15 *For every $i \geq 0$, let $\mathbf{AC}^i$ be the class of languages that are decidable by circuit families $C_1, C_2, \ldots$ such that $s(C_n) \leq n^{O(1)}$ and $a(C_n) \leq O((\log n)^i)$. Similarly, for every $i \geq 0$, let $\mathbf{NC}^i$ be the class of languages that are decidable by circuit families $C_1, C_2, \ldots$ such that $s(C_n) \leq n^{O(1)}$ and $d(C_n) \leq O((\log n)^i)$.*

It is known that $\mathbf{NC}^i \subseteq \mathbf{AC}^i \subseteq \mathbf{NC}^{i+1}$ and so $\mathbf{AC} = \bigcup_{i\geq 0}\mathbf{AC}^i = \bigcup_{i\geq 0}\mathbf{NC}^i = \mathbf{NC}$. The complexity classes that we just presented are strongly non-uniform in the sense that the underlying computational device changes at each input length. This leads to the somewhat unpleasant fact that each such class contains non-computable languages. This is in sharp contrast with the Turing machine model in which a fixed device is designed to work for all input lengths, and of course, any recognized language is (semi-)computable. The connection between the two models is made through the concept of *uniformity*.

Intuitively, a circuit family $C_1, C_2, \ldots$ is called uniform if there exists a fixed Turing machine that constructs C_n when given n as input. In order to be able to talk about uniformity of sublinear complexity, we will need the more refined but technical notion due to Barrington, Immerman, and Straubing [BIS90]. In their definition, which we make precise in the sequel, circuits are recognized rather than constructed. Let $C_1, C_2, \ldots$ be a circuit family and let $s_n = s(C_n)$ be the size of C_n. The gates in C_n may be numbered in $\{0, \ldots, s_n - 1\}$, and may be given a type $t \in \{0, 1, 2, 3\}$ according to whether they are AND gates, OR gates, positive inputs, or negative inputs respectively. The direct connection language of $C_1, C_2, \ldots$ is the set of words of the form $1^n\#\mathrm{b}_{s_n}(a)\#\mathrm{b}_{s_n}(b)\#\mathrm{b}_2(t)$, where gate b is an input to gate a in C_n, and the type of gate b is $t \in \{0, 1, 2, 3\}$. Recall that $\mathrm{b}_m(n)$ is the unique binary representation of n of length m, padded with leading zeros if necessary.

Definition 16 *Let $\mathcal{C}$ be a complexity class, let $C_1, C_2, \ldots$ be a circuit family, and let D be its direct connection language. We say that C is $\mathcal{C}$-uniform if there exists a language $L \in \mathcal{C}$ such that for every $n \geq 1$, we have $1^n\#a\#b\#t \in L$ if and only if $1^n\#a\#b\#t \in D$.*

We want to stress the technical point that the condition in Definition 16 is not the same as the condition that D belongs to $\mathcal{C}$. Indeed, the language L in Definition 16 may contain other words that are not of the form $1^n\#a\#b\#t$. However, this highly technical point does not affect the intuitions at all. Technically speaking, we need the condition stated in the definition because sublinear-time deterministic machines are not able to check that its input word has the required form $1^n\#a\#b\#t$.

Now, every circuit-based complexity class, such as those defined in Definition 15, have uniform counterparts by letting the circuit families be uniform. When $\mathcal{C}$ is a natural complexity class, the issue of having non-computable languages disappears. Moreover, the combination of natural restrictions on the circuits and natural uniformity conditions lead to illuminating capturing results. Thus, to cite some examples, it is quite pleasant that the class of languages accepted by $\mathbf{DLOGTIME}$-uniform families of polynomial-size circuits coincides with the complexity class $\mathbf{P}$, and that $\mathbf{DLOGTIME}$-uniform $\mathbf{NC}$ coincides with the parallel complexity class defined by the PRAM model with polynomially many processors and polylogarithmic-time (see [Vol99]). Many other connections of this type can be established.

One such important connection was made by Barrington, Immerman and Straubing [BIS90] who noticed that $\mathbf{DLOGTIME}$-uniform $\mathbf{AC}^0$ happens to coincide exactly with the Logarithmic-Time Hierarchy $\mathbf{LH}$. A key step in their proof is the result mentioned in Section 3.2 of Chapter 3 that first-order logic captures $\mathbf{LH}$ on finite structures with BIT. Equivalently now, FO captures $\mathbf{DLOGTIME}$-uniform $\mathbf{AC}^0$ on finite structures with BIT. In fact, the proof of the Barrington-Immerman-Straubing Theorem says a lot more: A query Q on a class of finite structures D with built-in membership relation is computable in $\mathcal{C}$-uniform $\mathbf{AC}^0$

if and only if there exists a query R on $\mathcal{BFR}$ that is computable in the complexity class $\mathcal{C}$ and such that Q is first-order definable on D using R as a built-in predicate.

The results of this chapter will elaborate further on the connections between uniform circuits, complexity classes, and logical languages, with the goal of identifying the descriptive complexity of $\mathrm{LFP}(\Delta_0)$.

4.5 Capturing Results without Input Predicates

A closer examination of Lemma 7 in the case of $\mathcal{BFR}$ reveals that $\mathrm{LFP}(\Delta_0)$-definable queries are also computable in non-uniform $\mathbf{AC}^0$; the reason is that they depend only on $O(\log n)$ bits of the input, and every such query is in non-uniform $\mathbf{AC}^0$. Indeed, every Boolean function on $O(\log n)$ bits can be computed by a circuit of alternation-depth two and size $n^{O(1)}$ that implements its truth-table. The interesting question at this point is the following: which uniform version of $\mathbf{AC}^0$ is captured by $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ on $\mathcal{BFR}$? Here, $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ denotes the first-order closure of $\mathrm{LFP}(\Delta_0)$. That is, in addition to $\mathrm{LFP}(\Delta_0)$ formulas, we also allow atomic formulas, conjunctions, negations and first-order quantification. Our next theorem is the answer to this question.

Technically speaking, Boolean circuits can only recognize languages over the alphabet $\{0, 1\}$. However, if we are interested in languages over larger alphabets Σ, we can fix an appropriate homomorphism $h : \Sigma^* \to \{0, 1\}^*$ and convey to say that circuits accept languages $L \subseteq \Sigma^*$ if and only if they accept $h(L)$. For our particular purposes, we will use $\Sigma = \{0, 1, \#\}$, and we can fix h to be the unique homomorphism such that $h(0) = 00$,

$h(1) = 11$ and $h(\#) = 01$.

Theorem 7 *Let Q be a query on $\mathcal{BFR}$. The following are equivalent:*

(i) Q is computable in $\mathbf{LH}^{\mathbf{P}}$ on $\mathcal{BFR}$,

(ii) Q is computable in $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$ on $\mathcal{BFR}$,

(iii) Q is definable in $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ on $\mathcal{BFR}$.

Proof: We close a cycle of implications. We first show that (i) implies (ii). Assume that Q is computable in $\mathbf{LH}^A$ for some $A \in \mathbf{P}$. We may assume $A \subseteq \{0,1\}^*$. For every n, let $F_n(x_1, \ldots, x_n)$ be the following DNF-formula

$$\bigvee_{\overline{a} \in A \cap \{0,1\}^n} \left(\bigwedge_{a_i = 1} x_i \wedge \bigwedge_{a_i = 0} \neg x_i \right).$$

Observe that $F_n(a_1, \ldots, a_n)$ is true if and only if the word $a_1 \ldots a_n$ belongs to A. The sequence $(F_1, F_2, \ldots)$, interpreted as a sequence of circuits of alternation-depth two, is of size exponential in n, but $\mathbf{P}$-uniform: the words of its direct connection language are of the form $1^n \# a \# b \# t$ with $a, b \in \{0,1\}^{O(n)}$, and deciding membership can be done in polynomial-time since $A \in \mathbf{P}$. We now build a $\mathbf{DPOLYLOGTIME}$-uniform family of $\mathbf{AC}^0$ circuits to compute Q. Let M be an oracle alternating Turing machine witnessing that Q is computable in $\mathbf{LH}^A$, and assume that M queries its oracle at most once in each computation path. This is a standard trick in alternating machines; it consists of existentially guessing the answers, write them down on a separate tape together with the nondeterministic branch taken at each step, and at the end of the computation, universally branch to check the correctness of every guess by deterministically re-simulating the computation path until the challenged query is asked. Let $c \log n$ be a bound on the running-time of M on inputs of length n. Observe that the length of each oracle query is bounded by $c \log n$ too. As in [BIS90], we may see the computation trees of M as a $\mathbf{DLOGTIME}$-uniform family of $\mathbf{AC}^0$ circuits, except for the oracle queries, which may be resolved by $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$ circuits; namely, we let queries of length $m \leq c \log n$ be resolved by the circuit F_m above (exponential-size in $m \leq c \log n$ is polynomial-size in n, and polynomial-time uniformity for length $m \leq c \log n$ is polylogarithmic-time uniformity for length n). It follows that Q is computable in $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$.

We see that (ii) implies (iii). Let Q be computable in $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$. By the Barrington-Immerman-Straubing Theorem (see Section 4.4), we know that Q is then first-order definable with an additional built-in query $R = (R_1, R_2, \ldots)$ on $\mathcal{BFR}$ that is computable in polylogarithmic-time. We show how to replace every occurrence of this built-in query by a formula of $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$. For every n and $\overline{a} = (a_1, \ldots, a_k) \in \{0, \ldots, n-1\}^k$, let $\mathbf{M}_{\overline{a}} = (\{0, \ldots, \log(n-1) - 1\}, \in, P_1^n, \ldots, P_k^n)$, where $P_i^n = \{m : m \in a_i\}$. Since R is a query computable in polylogarithmic-time, the language

$$\{\langle \mathbf{BIT}_n \rangle \# \mathrm{b}_n(a_1) \# \ldots \# \mathrm{b}_n(a_k) : (a_1, \ldots, a_k) \in R_n,\ n \geq 1\}$$

is decidable in polylogarithmic-time on inputs of the appropriate form. A simple unpadding argument shows then that the language $\{\langle \mathbf{M}_{\overline{a}} \rangle : \overline{a} \in \bigcup_{n \geq 1} R_n\}$ is in $\mathbf{P}$ (the length of $\langle \mathbf{M}_{\overline{a}} \rangle$ is logarithmic in the length of $\langle \{0, \ldots, n-1\}, a_1, \ldots, a_k \rangle$). Hence, by the Immerman-Vardi Theorem, the boolean query $Q = \{\mathbf{M}_{\overline{a}} : \overline{a} \in \bigcup_{n \geq 1} R_n\}$ is definable in least fixed-point logic on the class of all structures of the form $\mathbf{M}_{\overline{a}}$. We may even assume that Q is definable by a sentence of the form $(\mathrm{LFP}_{\overline{x},X}\varphi)(\overline{0})$ in which φ is a first-order formula, and 0 is a constant for zero. Let $\varphi'(y,z,p_1,\ldots,p_k,\overline{x},X)$ be the first-order formula of the vocabulary $\{\in\}$ that results from the following substitution in φ: replace each occurrence of an atomic formula of the form $P_i(u)$ by $u \in p_i$; replace each atomic formula of the form $X(\overline{u})$ by $X(y,z,p_1,\ldots,p_k,\overline{u})$; and replace each subformula of the form $(\exists u)(\psi)$ by $(\exists u \in y)(\psi') \vee (\exists u \in z)(\psi')$, where ψ' is the result of applying recursively the substitutions. Clearly, φ' is a Δ_0 formula. Moreover, it is not hard to see that for every $a_1,\ldots,a_k \in \{0,\ldots,n-1\}$, we have that $\mathbf{M}_{\overline{a}} \models (\mathrm{LFP}_{\overline{x},X}\varphi)(\overline{0})$ if and only if

$$(\{0,\ldots,\log(n-1)-1\},\in) \models (\mathrm{LFP}_{y,z,\overline{p},\overline{x},X}\varphi')(r,s,a_1,\ldots,a_k,\overline{0}),$$

where s is the largest power of two in the universe, and $r = s-1$. Observe that the binary representations of s and r are dual words; that is, $j \in s$ if and only if $j \notin r$ for every $j \leq \log(\log(n-1)-1)$. Since r and s are first-order definable with $\in$, we have shown that R is uniformly definable on $\mathcal{BFR}$ by a sentence of $\mathrm{FO}+\mathrm{LFP}(\Delta_0)$.

It remains to see that (iii) implies (i). Let $\varphi(x_1,\ldots,x_s)$ be a formula witnessing that Q is definable in $\mathrm{FO}+\mathrm{LFP}(\Delta_0)$. Without loss of generality, we may assume the following normal form for φ:

$$(Q_1y_1)\cdots(Q_ry_r)(\bigwedge_{i=1}^{s}(\psi_{i,1}\vee\ldots\vee\psi_{i,t}\vee\neg\psi_{i,t+1}\vee\neg\psi_{i,u})),$$

where each Q_i is $\exists$ or $\forall$, and each $\psi_{i,j}$ is either an atomic formula, or a formula of the form $(\mathrm{LFP}_{\overline{z},Z}\theta)(z_1,\ldots,z_s)$, with θ a Δ_0 formula. For every i,j, let $Q_{i,j}$ be the query on $\mathcal{BFR}$ defined by $\psi_{i,j}$, and let A be the following language over the alphabet $\{0,1,\#\}$:

$$\{n\#\mathrm{b}_n(b_1)\#\ldots\#\mathrm{b}_n(b_s)\#i\#j : (b_1,\ldots,b_s)\in Q_{i,j}(\mathbf{BIT}_n)\}.$$

This language will be our oracle set. That it belongs to $\mathbf{P}$ will be shown later. An alternating Turing machine with oracle A may simulate φ as indicated next. On input $\langle \mathbf{BIT}_n, a_1, \ldots, a_k \rangle$ where $a_1, \ldots, a_k \in \{0, \ldots, n-1\}$, the machine behaves as follows. First, it computes n. To this end, it existentially guesses the position of the leftmost $\#$ in the input, and universally branches to check that every smaller position contains a symbol other than $\#$. Then, following the alternation pattern of the quantifier prefix of φ, the machine existentially or universally guesses r words $w_1, \ldots, w_r$ of length $\log(n-1)$ each. The i-th word w_i is meant to be the binary representation of an element $b_i \in \{0, \ldots, n-1\}$ that is to interpret the first-order variable y_i. The machine proceeds then to evaluate each atomic formula $\psi_{i,j}$ as follows. Assume $\psi_{i,j} = \psi_{i,j}(z_1, \ldots, z_s)$, where each variable z_k is either an x_l or a y_l. The machine will write an oracle query of the form $n\#d_1\#\ldots\#d_s\#i\#j$, where $d_k = \mathrm{b}_n(b_l)$ if $z_k = y_l$, and $d_k = \mathrm{b}_n(a_l)$ if $z_k = x_l$. Observe that the length of this query is $O(\log n)$, and is easy to recover from the input (existentially guess each $\mathrm{b}_n(a_l)$ and universally branch to check that all guesses match the input). Clearly, the answer to this query is yes if and only if $\mathbf{BIT}_n \models \psi_{i,j}[d_1, \ldots, d_s]$ by the definition of the oracle set A.

All it remains to show is that the language A belongs to $\mathbf{P}$. This is fairly easy. If $\psi_{i,j}$ is an atomic formula, there is almost nothing to see: equalities are checked at once, and atomic formulas of the form $z_i \in z_j$ are also straightforward to check. If $\psi_{i,j}$ is a formula of the form $(\mathrm{LFP}_{\overline{z},Z}\theta)(z_1, \ldots, z_s)$ with θ being a Δ_0 formula, then the query it defines is computable in $\mathbf{DPOLYLOGTIME}$ on $\mathcal{BFR}$ by Lemma 7. Therefore, since the length of $n\#d_1\#\ldots\#d_s\#i\#j$ is logarithmic in the length of $\langle \mathbf{BIT}_n, d_1, \ldots, d_s \rangle$, a simple unpadding argument puts A in $\mathbf{P}$. □

As a corollary, we characterize the question on whether all polynomial-time decidable languages are rudimentary. The relationship between $\mathbf{P}$ and $\mathbf{RUD}$ remains unknown. It is known however that $\mathbf{NL} \subseteq \mathbf{RUD}$ [Nep70, Nep73], where $\mathbf{NL}$ is the class of languages accepted in nondeterministic logarithmic-space.

Corollary 1 *The following are equivalent:*

(i) $\mathrm{FO} + \mathrm{LFP}(\Delta_0) \subseteq \mathrm{FO}$ *on* $\mathcal{BFR}$*,*

(ii) $\mathbf{P} \subseteq \mathbf{RUD}$*,*

(iii) $\mathbf{P} \subseteq \mathbf{LINH}$.

Proof: Since $\mathbf{RUD} = \mathbf{LINH} = \mathrm{ATIME}(O(n), O(1))$, it is enough to show that (i) and (iii) are equivalent. The implication from (i) to (iii) follows from Theorem 3.2 in Chapter 3. For the other implication, assume that $\mathbf{P} \subseteq \mathbf{LINH}$, and let Q be a query on $\mathcal{BFR}$ that is definable by a $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ formula. By Theorem 7 we have Q is computable in $\mathbf{LH}^{\mathbf{P}}$, and so in $\mathbf{LH}^{\mathbf{LINH}}$ by hypothesis. Let M be an oracle alternating Turing machine witnessing that Q is computable in $\mathbf{LH}^{A}$ for some $A \in \mathbf{LINH}$, and let N be an alternating Turing machine witnessing that $A \in \mathbf{LINH}$. Since an oracle Turing machine running in logarithmic-time can only ask logarithmically long queries, oracle queries of M may be answered by N in logarithmic-time with respect to the input to M. The number of alternations being constant, it follows that Q is computable in $\mathbf{LH}$. Hence, Q is first-order definable on $\mathcal{BFR}$. □

4.6 Capturing Results with Input Predicates

The natural question at this point is what happens when input predicates, in addition to the membership (BIT) relation, are available. That is, we fix a relational vocabulary L, and we wonder what is captured by $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ on classes of finite structures over L with built-in membership relation. Unfortunately, we are able to provide only an exact answer in the case that L is a unary vocabulary. We show that $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ still captures $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$ on such classes. The proof of this fact is not as simple as before because the straightforward evaluation of the $\mathrm{LFP}(\Delta_0)$ subformulas requires polylogarithmic-time and cannot be done in $\mathbf{AC}^0$ directly. Indeed, it is not known whether $\mathbf{DPOLYLOGTIME} \subseteq \mathbf{AC}^0$. The solution to overcome this difficulty is to make use of the fact that Δ_0-formulas depend only on the reflexive transitive closure of their arguments. This makes the proof slightly more involved, but we obtain a clean result:

Theorem 8 *Let L be a unary vocabulary, let C be a class of finite structures over L with built-in membership relation, and let Q be a query on C. Then, the following are equivalent:*

(i) Q is computable in $\mathbf{LH}^{\mathbf{P}}$ on C,

(ii) Q is computable in $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$ on C,

(iii) Q is definable in $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ on C.

Proof: The proofs that (i) implies (ii), and that (ii) implies (iii), go through as in Theorem 7 essentially without change. The proof that (iii) implies (i) uses an argument similar to the one in the proof of Lemma 7. Recall from Lemma 5 that if φ is a Δ_0 formula, then

$\mathbf{M} \models \varphi[a_1, \ldots, a_s, A]$ if and only if $\mathbf{M} \models \varphi[a_1, \ldots, a_s, A \cap (\bigcup\{\mathrm{RTC}(a_i) : i \in F(\varphi)\})^k]$. Iterated application of this lemma with each of the relation symbols of L shows then that $\mathbf{M} \models \varphi[a_1, \ldots, a_s, A]$ if and only if

$$\mathbf{M} \cap B \models \varphi[a_1, \ldots, a_s, A \cap B^k],$$

where $B = \bigcup\{\mathrm{RTC}(a_i) : i \in F(\varphi)\}$, and $\mathbf{M} \cap B$ is the substructure of $\mathbf{M}$ generated by B. In turn, we remark that $B \subseteq \{0, \ldots, \log(|M| - 1) - 1\} \cup \{a_1, \ldots, a_s\}$ since each element of $\mathrm{TC}(a_i)$ is a *bit* position of an element in $\{0, \ldots, |M| - 1\}$. Moreover, a straightforward argument reveals that $\mathbf{M} \cap B'$ is an end-extension of $\mathbf{M} \cap B$, where $B' = \{0, \ldots, \log(|M| - 1) - 1\} \cup \{a_1, \ldots, a_s\}$. Hence, $\mathbf{M} \models \varphi[a_1, \ldots, a_s, A]$ if and only if $\mathbf{M} \cap B \models \varphi[a_1, \ldots, a_s, A \cap B^k]$, and by absoluteness, if and only if $\mathbf{M} \cap B' \models \varphi[a_1, \ldots, a_s, A \cap B'^k]$. With these observations in hand, we claim that:

Claim 1 *If Q is definable in* $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ *on C, then Q is definable by a formula of* $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ *in which no relation symbol from L appears within the scope of a fixed-point operator.*

Proof: The main idea is that since every $\mathrm{LFP}(\Delta_0)$ formula will depend only on $O(\log n)$ bits of the input predicates by the remarks above (here is the crucial point where we use the fact that the vocabulary is unary), we can existentially quantify these bits outside the $\mathrm{LFP}(\Delta_0)$-formula, and pass them to it as input variables. Formally, the argument is as follows. Assume for simplicity that L consists of a unique relation symbol R; the general case is as easy. Let φ be a formula defining Q on C. Replace each occurrence in φ of a subformula of the form

$(\mathrm{LFP}_{\overline{x},X}\theta)(x_1,\ldots,x_k)$ with θ a Δ_0 formula, by the formula

$$(\exists v)((r \in v \leftrightarrow R(r)) \wedge (\forall z \in s)(z \in v \leftrightarrow R(z)) \wedge$$

$$\bigvee_{w\in\{0,1\}^k} (\bigwedge_{w_i=1} R(x_i) \wedge \bigwedge_{w_i=0} \neg R(x_i) \wedge (\mathrm{LFP}_{v,\overline{x},X'}\theta^w)(v,\overline{x}))),$$

where θ^w is the result of replacing each atomic formula of the form $R(u)$, with u a bound variable, by $u \in v$, each atomic formula of the form $R(x_i)$ by $x_i = x_i$, if $w_i = 1$, each atomic formula of the form $R(x_j)$ by $x_j \neq x_j$, if $w_j = 0$, and each atomic formula of the form $X(\overline{u})$ by $X'(v,\overline{u})$. Here, r and s are existentially quantified variables set to the largest power of two of the universe, and $r - 1$ respectively (observe that the binary representations of r and s are dual words). Observe that if v is a witness for the first-order variable of this formula, then its binary representation is encoding the first $\log(n-1)$ bits of R. By the remarks preceding the claim, it is straightforward to check using standard absoluteness arguments that the modified formula is defining Q on C, as required.

The rest of the proof that (iii) implies (i) is now almost identical to the proof of Theorem 7. Namely, access to the input predicates is only required when simulating the first-order part of the formula, and the simulation of the $\mathrm{LFP}(\Delta_0)$-parts of the formula may be asked to an oracle set in $\mathbf{P}$. □

Observe that the argument of Theorem 8 does not go through for vocabularies of higher arities. In the case of a binary relation, for example, the reason is that there are $O((\log|M|)^2)$ significant bits in the substructure $\mathbf{M} \cap \{0,\ldots,\log(|M|-1)\}$. Although we do not provide with an exact characterization of $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ for vocabularies of higher arities, we are able to compare the expressive power of $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ with a familiar complexity class. Recall from the introduction that $\mathbf{RUD}_{n^{1/r}} = \mathrm{ATIME}(O(n^{1/r}), O(1))$ (see Corollary 5 in [AG91]).

Theorem 9 *Let L be a relational vocabulary of maximum arity r, and let C be the class of all finite structures over L with built-in membership relation. If $\mathbf{P} \subseteq \mathbf{RUD}_{n^{1/r}}$, then* $\mathrm{FO} + \mathrm{LFP}(\Delta_0) \subseteq \mathrm{FO}$ *on C.*

Proof: Assume $\mathbf{P} \subseteq \mathbf{RUD}_{n^{1/r}}$, and let Q be a query on C definable in $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$. It is enough to show that Q is computable in $\mathbf{LH}$ on C. Even easier, it is enough to show that each $\mathrm{FO}+\mathrm{LFP}(\Delta_0)$-formula can be evaluated in $\mathbf{LH}$ on the appropriate inputs. Let $\varphi(x_1, \ldots, x_k)$ be such a formula. Lemma 7 says that deciding whether $\mathbf{M} \models \varphi[a_1, \ldots, a_k]$ can be done in polylogarithmic-time in $|M|$. Moreover, the same absoluteness argument as in the proof of Theorem 8 reveals that $\mathbf{M} \models \varphi[a_1, \ldots, a_k]$ if and only if $\mathbf{M} \cap B' \models \varphi[a_1, \ldots, a_k]$, where $B' = \{0, \ldots, \log(|M| - 1) - 1\} \cup \{a_1, \ldots, a_k\}$. Since only $O((\log |M|)^r)$ bits are relevant in $\mathbf{M} \cap B'$, the same computation can can be carried over an unpadded input that only contains these bits. The computation time is now polynomial in the length of the (unpadded) input, and therefore, by hypothesis, the same language is decidable in $\mathrm{ATIME}(O(n^{1/r}), O(1)) = \mathbf{RUD}_{n^{1/r}}$ on the appropriate inputs. Since the length of these inputs is $O((\log |M|)^r)$, the alternating computation can be carried over the original inputs in time

$$O(((\log |M|)^r)^{1/r}) = O(\log |M|),$$

and still a constant number of alternations. That is, on the original inputs, the evaluation of φ can be done in $\mathbf{LH}$ as required. □

4.7 Consequences

As mentioned in the introduction, Theorem 8 sets the link to an important problem related to the uniformity of circuits for integer division. Beame, Cook, and Hoover [BCH86] showed that the problem of dividing two integer numbers of n bits can be computed by $\mathbf{P}$-uniform logarithmic-depth circuits ($\mathbf{P}$-uniform $\mathbf{NC}^1$). The result was improved by Reif [Rei87] (see also [IL95]) who showed that the problem could be computed by $\mathbf{P}$-uniform bounded-depth circuits with majority gates ($\mathbf{P}$-uniform $\mathbf{TC}^0$). However, it was not known until Hesse's solution [Hes01] whether the uniformity condition can be relaxed to $\mathbf{DLOGTIME}$-uniformity, as it is the case for the $\mathbf{TC}^0$ circuits for addition, subtraction, and multiplication (see Barrington, Immerman and Straubing [BIS90]).

It is known that majority gates of polylogarithmically-many bits may be simulated by $\mathbf{DLOGTIME}$-uniform $\mathbf{AC}^0$ circuits. We review that construction (see [Weg87] for a similar one). A circuit $\mathrm{TH}_k(x_1, \ldots, x_m)$ computing whether at least k of the input bits $x_1, \ldots, x_m$ are one is recursively built as follows:

$$\mathrm{TH}_k(x_1, \ldots, x_m) = \bigvee_{\substack{i_1+\cdots+i_s \geq k \\ i_j \leq m/s}} \bigwedge_{j=1}^{s} \mathrm{TH}_{i_j}(x_{(j-1)m/s+1}, \ldots, x_{jm/s}),$$

where $m = (\log n)^{O(1)}$, and s is suitably chosen so that the size of the circuit is polynomial in n, and the depth is a constant independent of n. The choice $s = (\log n)^{\epsilon}$ works for sufficiently small ϵ. It is not hard to see that these circuits are $\mathbf{DLOGTIME}$-uniform: a clever numbering of gates will tell all the required information to the $\mathbf{DLOGTIME}$ algorithm that computes the direct connection language. The well-known power of $\mathbf{AC}^0$ circuits to do arithmetic on numbers with polylogarithmically-many significant bits follows from Reif's result, and the known algorithms for addition, subtraction and multiplication. However, while addition, subtraction and multiplication of polylogarithmically-long numbers admit $\mathbf{DLOGTIME}$-uniform $\mathbf{AC}^0$ such circuits, the known algorithms for division before Hesse's fell short since they give only $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$ circuits. We note that Theorem 8 implies that division of numbers with polylogarithmically-many significant bits is definable in $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ on the class of finite words with built-in membership relation. We do not know, however, of a direct proof of this fact that yields any new good insight.

Chapter 5

Datalog on Random CNF Formulas

5.1 About Datalog and Infinitary Logic

Let us start by defining Datalog, which, as we will see, corresponds to the existential positive fragment of least fixed-point logic. We also revisit the inclusion of Datalog into the existential positive fragment of infinitary logic with finitely many variables.

Let $L = \{R_1, \ldots, R_r\}$ be a relational vocabulary and let $X_1, \ldots, X_s$ be second-order variables. Let r_i be the arity of R_i, and let s_i be the arity of X_i. A Datalog program over $L \cup \{X_1, \ldots, X_s\}$ is a set of rules of the form

$$t_0 \quad :- \quad t_1, \ldots, t_p,$$

where t_0 is an atomic formula of the form $X_i(x_1, \ldots, x_{s_i})$, and each t_j with $j > 0$ is an atomic formula of the form $R_i(x_1, \ldots, x_{r_i})$ or $X_i(x_1, \ldots, x_{s_i})$. Here the x_i are first-order variables that are not necessarily different.

Note that the heads of the rules (left part) are always predicates formed from second-order variables. These are called the *intensional database* predicates (IDBs). The predicates in the vocabulary L always appear in the bodies of the rules (right part), and are called the *extensional database* predicates (EDBs). One of the IDBs is called the *goal* of the program, and it might be of arity zero (a propositional letter). A Datalog program is a recursive specification of the IDBs from the EDBs with the least fixed-points semantics. Informally, if $\mathbf{M}$ is a structure over the vocabulary L of the EDBs, the IDBs are given the following semantics: initialize the IDBs to be empty, and repeatedly apply the rules of the program whose bodies (right part) are satisfied by $\mathbf{M}$ and the current value of the IDBs, until no new facts can be added to the IDBs. A precise definition of the semantics of a Datalog program requires the concept of systems of simultaneous inductions that we introduce next.

Let

$$\varphi_1(x_1, \ldots, x_{r_1}, X_1, \ldots, X_s), \ldots, \varphi_s(x_1, \ldots, x_{r_s}, X_1, \ldots, X_s)$$

be a sequence of first-order formulas of $L \cup \{X_1, \ldots, X_s\}$ each of which is positive in each X_i. On every structure $\mathbf{M}$ over L, the sequence of formulas $(\varphi_1, \ldots, \varphi_s)$ defines s monotone operators $F_i : M^{s_1} \times \cdots \times M^{s_s} \to M^{s_i}$ by setting

$$F_i(A_1, \ldots, A_s) = \{(a_1, \ldots, a_{s_i}) \in M^{s_i} : \mathbf{M} \models \varphi_i[a_1, \ldots, a_{s_i}, A_1, \ldots, A_s]\}.$$

The stages are defined by setting $F^0 = (\emptyset, \ldots, \emptyset)$ and $F^{m+1} = (F_1(F^m), \ldots, F_s(F^m))$. If M is finite, the system reaches a least fixed-point $F^\infty(\mathbf{M}) = (F_1^\infty(\mathbf{M}), \ldots, F_s^\infty(\mathbf{M}))$ in a finite number of steps.

Each Datalog program specifies a system of simultaneous inductions. Indeed, each IDB predicate X_i has an associated formula $\varphi_i(x_1, \ldots, x_{s_i}, X_1, \ldots, X_s)$ that is the disjunction of the existential closure of the bodies of the rules that have X_i in its head. The proper construction of φ_i requires some renaming of variables in order for each head to have the form $X_i(x_1, \ldots, x_{s_i})$. Then, the interpretation of a program with goal predicate X_g on a structure $\mathbf{M}$ over L is simply $\varphi_g^\infty(\mathbf{M})$, where $(\varphi_1^\infty(\mathbf{M}), \ldots, \varphi_s^\infty(\mathbf{M}))$ is the least fixed-point of the simultaneous system $(\varphi_1, \ldots, \varphi_s)$. The interpretation of a Datalog program P on $\mathbf{M}$ is denoted by $P^{\mathbf{M}}$. This defines the semantics of Datalog. Let us see an example.

Example 11 Consider the following Datalog program over the vocabulary of directed graphs $\{E\}$, where X and Y are binary and unary second-order variables respectively:

$$\begin{aligned} X(x,y) &\ :-\ \ E(x,y) \\ X(x,y) &\ :-\ \ X(x,z), E(z,y) \\ Y(x) &\ :-\ \ X(x,x). \end{aligned}$$

The Datalog program recursively defines X as the set of all pairs of nodes (x, y) such that y is reachable from x in the graph. Also, it defines Y in terms of X as those nodes that are reachable from themselves. Thus, if Y is the goal predicate of the program, its interpretation on a graph is the set of nodes that belong to a (non-necessarily simple) cycle. The corresponding system is

$$\begin{aligned} \varphi_X(x,y,X,Y) &= E(x,y) \vee (\exists z)(X(x,z) \wedge E(z,y)) \\ \varphi_Y(x,X,Y) &= X(x,x). \end{aligned}$$

We note that in this particular case, the system of simultaneous inductions is quite trivial since, in fact, X and Y variables do not mix. Nonetheless, we will have genuine simultaneous inductions in the general case. Example 13 below is a genuine one.

A careful writing of the system of simultaneous inductions corresponding to a Datalog program reveals that the defining formulas are existential positive. That is, each component of the system is a first-order formula that is formed from the atomic formulas and equalities by means of conjunctions, disjunctions and existential quantification only.

We are interested on the number of first-order variables of a Datalog program as a measure of its complexity. Thus, following Kolaitis and Vardi [KV00a], let k-Datalog be the class of all Datalog programs with at most k variables and IDB predicates of arity at most k. In the following Theorem we show that each k-Datalog program is equivalent to one component of a system of simultaneous inductions defined by an existential positive formula with at most k variables. This was already noted by Kolaitis and Vardi [KV00a] but we provide a full proof for completeness.

Theorem 10 *Let $L = \{R_1, \dots, R_r\}$ be a relational vocabulary, let $X_1, \dots, X_s$ be second-order variables, and let s_i be the arity of X_i. Let P be a k-Datalog program over $L \cup \{X_1, \dots, X_s\}$ with goal predicate X_1 such that $k \geq s_i$ for every $i \in \{1, \dots, s\}$. Then, there exists a system of existential positive formulas*

$$\varphi_1(x_1, \dots, x_{s_1}, X_1, \dots, X_s), \dots, \varphi_s(x_1, \dots, x_{s_s}, X_1, \dots, X_s)$$

with all variables among $\{x_1, \dots, x_k\}$ such that for every structure $\mathbf{M}$ over L if holds that $P^{\mathbf{M}} = \varphi_1^{\infty}(\mathbf{M})$.

Proof: Let $V = \{x_1, \ldots, x_k\}$ be a set of variables that contains all variables that occur in the program. Before we build our system, we need to do a renaming of the variables of the rules in such a way that all heads with the same second-order variable are identical. This is done as follows. Suppose that R is a rule t_0 :– $t_1, \ldots, t_p$. The head of the rule is of the form $X_i(x_{\pi(1)}, \ldots, x_{\pi(s_i)})$ for some function $\pi : \{1, \ldots, s_i\} \to \{1, \ldots, k\}$. Now, let $\rho : \{1, \ldots, k\} \to \{1, \ldots, k\}$ be any permutation such that $\rho(\pi(j)) = \min\{j' : \pi(j') = \pi(j)\}$, where j and j' range over $\{1, \ldots, s_i\}$. Such a permutation exists because the minimal j' such that $\pi(j') = p$ is clearly unique for any $p \in \{1, \ldots, k\}$, and because $k \geq s_i$. This is the only place in the proof where we need that k is at least as big as the arity of any IDB predicate. We let

$$\varphi_R = (\exists x_{i_1}) \cdots (\exists x_{i_q})(t'_1 \wedge \cdots \wedge t'_p),$$

where t'_i is the result of renaming each variable x_j by $x_{\rho(j)}$ in t_i. Here, $i_1, \ldots, i_q$ is the set of all indices of variables in $\{1, \ldots, k\} - \{\rho(\pi(1)), \ldots, \rho(\pi(s_i))\}$. In other words, φ_R is the result of renaming each variable of the rule according to the permutation ρ, and existentially quantifying all variables that are not within the new names $x_{\rho(\pi(1))}, \ldots, x_{\rho(\pi(s_i))}$ of the variables $x_{\pi(1)}, \ldots, x_{\pi(s_i)}$ in the head. Finally, let $C = \{(j, j') : \pi(j) = \pi(j')\}$ be the set of clashes of π, and let

$$\psi_R(x_1, \ldots, x_{s_i}) = \varphi_R \wedge \bigwedge_{(j,j') \in C} x_j = x_{j'}.$$

By now, we have succeeded to rename the variables of the rule R in such a way that the *head*, if we were to write the rule again, has the form $X_i(x_1, \ldots, x_{s_i})$. Now, if $R_1, \ldots, R_t$ are all the rules of the program having X_i in its head, we let

$$\varphi_i(x_1,\ldots,x_{s_i},X_1,\ldots,X_s) = \psi_{R_1} \vee \cdots \vee \psi_{R_t}.$$

From its very construction, the first component of the system $(\varphi_1^\infty(\mathbf{M}),\ldots,\varphi_s^\infty(\mathbf{M}))$ is defining the semantics of the program P. □

Following [KV00a] we will denote the existential positive fragment of first-order logic with k variables by $\exists\mathrm{FO}^k$. Similarly, we let $\exists L^k_{\infty\omega}$ be the existential positive fragment of infinitary logic with k variables. That is, $\exists L^k_{\infty\omega}$ is the smallest class of formulas with at most k variables that is obtained by closing the atomic formulas under arbitrary conjunctions, arbitrary disjunctions, and existential quantification only. It was shown in [KV00a] that every component of the least fixed-point of a system of $\exists\mathrm{FO}^k$ formulas is expressible in $\exists L^k_{\infty\omega}$. In view of Theorem 10 we obtain:

Corollary 2 *Let k be a positive integer. Then k-Datalog $\subseteq \exists L^k_{\infty\omega}$, where k-Datalog is the class of all Datalog programs with at most k variables and IDB predicates of arity at most k.*

5.2 Formulas as Structures and Satisfiability as Homomorphism

Let $\{v_1,\ldots,v_n\}$ be a set of propositional variables. A literal is a variable v_i or the negation of a variable $\neg v_i$. A clause is a multiset of literals. A k-CNF formula is a multiset of clauses each of which has at most k literals. It is customary to write clauses $\{l_1,\ldots,l_k\}$ as disjunctions $l_1 \vee \cdots \vee l_k$, and CNF formulas as conjunctions of clauses $C_1 \wedge \cdots \wedge C_m$. A k-CNF formula F is satisfiable if there exists a truth assignment $f : \{v_1,\ldots,v_n\} \to \{0,1\}$ that satisfies every clause C of F. That is, if $C = \{\neg v_{i_1}, \ldots, \neg v_{i_s}, v_{j_1}, \ldots, v_{j_{k-s}}\}$ is a clause of C, then either $f(v_{i_p}) = 0$ for some $p \in \{1, \ldots, s\}$, or $f(v_{j_p}) = 1$ for some $p \in \{1, \ldots, k-s\}$.

There is a natural encoding of k-CNF formulas into finite relational structures. Let $L_k = \{R_0, \ldots, R_k\}$ be the vocabulary of $k+1$ relations of arity k. We encode a k-CNF formula F with variables $v_1, \ldots, v_n$ as a finite structure $\mathbf{M} = (M, R_0^{\mathbf{M}}, \ldots, R_k^{\mathbf{M}})$ over L_k. The universe M is the set of variables $\{v_1, \ldots, v_n\}$, and for each $s \in \{0, \ldots, k\}$ the relation $R_s^{\mathbf{M}}$ encodes the set of clauses of F with exactly s negated variables. More precisely, $R_s^{\mathbf{M}}$ consists of all k-tuples of the form

$$(v_{i_1}, \ldots, v_{i_s}, v_{j_1}, \ldots, v_{j_{k-s}}) \in \{v_1, \ldots, v_n\}^k$$

such that $\{\neg v_{i_1}, \ldots, \neg v_{i_s}, v_{j_1}, \ldots, v_{j_{k-s}}\}$ is a clause of F. The encoding of F by such a finite structure will be denoted by $M(F)$.

Example 12 Consider the 3-CNF formula

$$F = (v_1 \vee \neg v_2 \vee v_3) \wedge (\neg v_2 \vee \neg v_3 \vee v_4) \wedge (v_1 \vee \neg v_3 \vee \neg v_4)$$

over the variables $v_1, \ldots, v_4$. Recall that, strictly speaking, the clauses of F are sets of literals. Thus, $C_1 = \{\neg v_2, v_1, v_3\}$, $C_2 = \{\neg v_2, \neg v_3, v_4\}$ and $C_3 = \{\neg v_3, \neg v_4, v_1\}$. The structure encoding F is $M(F) = (M, R_0, R_1, R_2, R_3)$ where $M = \{v_1, v_2, v_3, v_4\}$, $R_0 = \emptyset$, $R_1 = \{(v_2, v_1, v_3), (v_2, v_3, v_1)\}$, $R_2 = \{(v_2, v_3, v_4), (v_3, v_2, v_4), (v_3, v_4, v_1), (v_4, v_3, v_1)\}$ and $R_3 = \emptyset$.

The encoding by means of finite structures allows us to use logic to define properties of k-CNF formulas. For example, the simple property that no clause is tautological is expressed by the following universal negative first-order formula:

$$(\forall x_1) \cdots (\forall x_k) \left(\bigwedge_{s=0}^{k} \left(R_s(x_1, \ldots, x_k) \rightarrow \bigwedge_{i \leq s < j} x_i \neq x_j \right) \right).$$

More sophisticated properties can be expressed in more powerful logics as our next example reveals.

Example 13 The purpose of this example is to show that the property "F is an unsatisfiable 2-CNF formula" is definable by a 3-Datalog program. The main idea amounts to the well-known fact that clauses of two literals may be seen as arcs of a directed graph, and that the unsatisfiability of a 2-CNF formula is characterized by a reachability property of that graph. Let F be a 2-CNF formula on the variables $v_1, \ldots, v_n$. Let $G = (V, E)$ be the following directed graph. The set of nodes V is the set of all possible literals $v_1, \ldots, v_n, \neg v_1, \ldots, \neg v_n$. The set of arcs is the set of all pairs of literals (l_1, l_2) such that $\{\neg l_1, l_2\}$ is a clause of F. Then, it is not hard to check that F is unsatisfiable if and only if there exists a variable v_i such that $\neg v_i$ is reachable from v_i and v_i is reachable from $\neg v_i$. We express this as a 3-Datalog program over the vocabulary L_2 of 2-CNF formulas with four binary IDB predicates PP, PN, NP, NN, and a propositional IDB goal P. The meaning of $PP(v_i, v_j)$ will be that v_j is reachable from v_i, the meaning of $PN(v_i, v_j)$ will be that $\neg v_j$ is reachable from v_i, the meaning of $NP(v_i, v_j)$ will be that v_j is reachable from $\neg v_i$, and the meaning of $NN(v_i, v_j)$ will be that $\neg v_j$ is reachable from $\neg v_i$. Thus, the names PP, PN, NP and NN stand for Positive-Positive, Positive-Negative, Negative-Positive and Negative-Negative respectively. The recursive definitions of each of these predicates by means of a Datalog program follows:

$$\begin{aligned}
PP(x,y) &\; :- \; R_1(x,y)\\
PN(x,y) &\; :- \; R_2(x,y)\\
NP(x,y) &\; :- \; R_0(x,y)\\
NN(x,y) &\; :- \; R_1(y,x)\\
PP(x,y) &\; :- \; PP(x,z), PP(z,y)\\
PP(x,y) &\; :- \; PN(x,z), NP(z,y)\\
PN(x,y) &\; :- \; PP(x,z), PN(z,y)\\
PN(x,y) &\; :- \; PN(x,z), NN(z,y)\\
NP(x,y) &\; :- \; NP(x,z), PP(z,y)\\
NP(x,y) &\; :- \; NN(x,z), NP(z,y)\\
NN(x,y) &\; :- \; NP(x,z), PN(z,y)\\
NN(x,y) &\; :- \; NN(x,z), NN(z,y)\\
P &\; :- \; PN(x,x), NP(x,x).
\end{aligned}$$

The recursive definition of $PP(v_i, v_j)$, for instance, reflects the fact that there are exactly two ways of reaching v_j from v_i by a path of length at least two: either we first reach a positive literal v_k, and from that we reach v_j, or we first reach a negative literal $\neg v_k$, and from that we reach v_j. The interpretation of the rest of recursive definitions is similar.

Before we conclude this section, let us introduce an alternative way of interpreting satisfiability of k-CNF formulas. Recall that a homomorphisms between two relational structures $\mathbf{M}$ and $\mathbf{N}$ over the same vocabulary L is a mapping $f : M \to N$ such that if $(a_1, \ldots, a_r) \in R^{\mathbf{M}}$ for some R in L, then $(f(a_1), \ldots, f(a_r)) \in R^{\mathbf{N}}$. Now, let F be a k-CNF formula. Recall that a truth assignment is a mapping from the set of variables of F to $\{0, 1\}$. Moreover, a truth assignment satisfies a clause $v_{j_1} \vee \cdots \vee v_{j_k}$ exactly when f does not map the tuple $(v_{j_1}, \ldots, v_{j_k})$ to the tuple $(0, \ldots, 0)$, and similarly for the other types of clauses that have one, two, or more negations. Thus, we observe, as Feder and Vardi [FV98] do, that the satisfying truth assignments of F are exactly the homomorphisms between $M(F)$, the structure encoding F, and the following structure:

$$\begin{aligned}
T &= (\{0,1\}, R_0^T, R_1^T, \ldots, R_k^T) \\
R_0^T &= \{0,1\}^k - \{(0,0,\ldots,0)\} \\
R_1^T &= \{0,1\}^k - \{(1,0,\ldots,0)\} \\
\ldots \\
R_k^T &= \{0,1\}^k - \{(1,1,\ldots,1)\}).
\end{aligned}$$

The structure T will be called the *template* structure in the future. This interpretation of satisfiability in terms of homomorphisms will be exploited later.

5.3 Random Model of CNF Formulas

Let n, m and k be positive natural numbers. Let $F(n, m, k)$ be the set of all k-CNF formulas on the variables $\{v_1, \ldots, v_n\}$ with exactly m clauses each of which has exactly k literals on distinct variables. We let $\mathcal{F}(n, m, k)$ be the uniform distribution on the sample space $F(n, m, k)$. Observe that a particular formula $F = C_1 \wedge \ldots \wedge C_m$ of the sample space has probability

$$\left[2^k \binom{n}{k}\right]^{-m}.$$

Alternatively, $\mathcal{F}(n, m, k)$ can be described as the result of independently repeating the following experiment m times: choose exactly k variables from $\{v_1, \ldots, v_k\}$, and negate each variable independently with probability $1/2$. We will use this interpretation whenever it is convenient. The ratio m/n is denoted by Δ, and is called the clause density. Usually, Δ is fixed to a constant and therefore $m = \Delta n$ is determined by n. We are interested in studying the asymptotic properties of a randomly chosen formula $F \sim \mathcal{F}(n, \Delta n, k)$ as n approaches infinity.

It is well known that when the clause density Δ exceeds a certain constant that only depends on k, a randomly chosen formula is almost surely unsatisfiable. Let us remind the simple argument. Let $f : \{0, 1\}^n \to \{0, 1\}$ be a truth assignment. Observe that probability that a randomly chosen clause is falsified by f is $1/2^k$ since all chosen variables need to be assigned the wrong polarity. Therefore, the probability that the clause is satisfied by f is $(1 - 1/2^k)$. Now, by independence of the clauses, the probability that a randomly chosen formula F is satisfied by f is

$$\left(1 - \frac{1}{2^k}\right)^m \leq e^{-m/2^k}.$$

Here we used the fact that $1 + x \leq e^x$ for every real number x. If we let X be the random variable that counts the number of satisfying assignments of a randomly chosen formula F, we have that $\mathrm{E}[X] = 2^n e^{-m/2^k}$ by linearity of expectation. By Markov's inequality

$$\Pr[X > 0] \leq \mathrm{E}[X] = 2^n e^{-m/2^k} = 2^n e^{-\Delta n/2^k}.$$

Finally, if $\Delta > 2^k \ln(2)$, the formula is unsatisfiable with probability approaching 1 as n approaches ∞. When $k = 3$, this bound gives the well-known number 5.19.

5.4 Main Result and Discussion of Proof Techniques

Let $3CNF_n$ be the class of all structures of the form $M(F)$ where F is a 3-CNF formula on at most n variables. Let $3SAT_n \subseteq 3CNF_n$ be the class of all structures of the form $M(F)$ where F is a satisfiable 3-CNF formula on at most n variables. Similarly, let $3UNSAT_n \subseteq 3CNF_n$ be the class of all structures of the form $M(F)$ where F is an unsatisfiable 3-CNF formula on at most n variables. The main result of this chapter is the following:

Theorem 11 *For all constants $\delta > 0$, $\Delta > 0$, and for all sufficiently large n, if $C_n \subseteq 3CNF_n$ is such that*

(i) $C_n \subseteq 3UNSAT_n$ and

(ii) $\Pr[M(F) \in C_n] \geq \delta$ when F is drawn from $\mathcal{F}(n, \Delta n, 3)$,

then C_n is not definable in $\exists L^k_{\infty\omega}$ on $3CNF_n$ for $k \leq n/(\ln(n))^2$.

In words, what this result says is that every property of 3-CNF formulas that implies unsatisfiability and is expressible in $\exists L^k_{\infty\omega}$ with $k \leq n/(\ln(n))^2$ must have asymptotic probability 0. Let $3CNF = \bigcup_{n \geq 1} 3CNF_n$ and $3UNSAT = \bigcup_{n \geq 1} 3UNSAT_n$. Since k-Datalog is included in $\exists L^k_{\infty\omega}$ by Corollary 2, the following follows:

Corollary 3 *Every Datalog Boolean query on $3CNF$ that implies $3UNSAT$ must have asymptotic probability 0 in $\mathcal{F}(n, \Delta n, 3)$.*

Theorem 11 is an inexpressibility result since it asserts that certain Boolean queries are not expressible in $\exists L^k_{\infty\omega}$ on $3CNF$. The purpose of the rest of this section is to discuss the techniques that are available to prove such a result.

Consider the following game played by two players, Spoiler and Duplicator, on a board formed by two structures $\mathbf{A}$ and $\mathbf{B}$ over the same relational vocabulary. Each player has k pebbles numbered $\{1, \ldots, k\}$. At each round of the game, Spoiler can make one of two different moves: either he places a free pebble over an element of $\mathbf{A}$, or he removes a pebble from a pebbled element of $\mathbf{A}$. To each move of Spoiler, Duplicator must respond by placing her corresponding pebble over an element of $\mathbf{B}$, or removing her corresponding pebble from $\mathbf{B}$ respectively. By corresponding pebble we mean the pebble that has the same number as that chosen by Spoiler. If a round is reached in which the correspondence $a_i \mapsto b_i$ that assigns pebbled elements of $\mathbf{A}$ to the corresponding pebbled elements of $\mathbf{B}$ is not a partial homomorphism between $\mathbf{A}$ and $\mathbf{B}$, then we say that Spoiler wins. Otherwise, Duplicator wins. Recall that a partial homomorphism from $\mathbf{A}$ to $\mathbf{B}$ is a function $f : A' \rightarrow B'$, where $A' \subseteq A$ and $B' \subseteq B$, such that if $a_1, \ldots, a_r \in A'$ and $(a_1, \ldots, a_r) \in R^{\mathbf{A}}$, then $(f(a_1), \ldots, f(a_r)) \in R^{\mathbf{B}}$. The following definition makes the game formal:

Definition 17 *Let L be a relational vocabulary, and let $\mathbf{A}$ and $\mathbf{B}$ be two structures over L. A* *winning strategy for Duplicator in the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$ is a non-empty set $\mathcal{H}$ of partial homomorphisms from $\mathbf{A}$ to $\mathbf{B}$ such that:*

(i) If $f \in \mathcal{H}$, then $|\mathrm{Dom}(f)| \leq k$.

(ii) If $f \in \mathcal{H}$ and $g \subseteq f$, then $g \in \mathcal{H}$.

(iii) If $f \in \mathcal{H}$, $|\mathrm{Dom}(h)| < k$ and $a \in A$, then there is some $b \in B$ and $g \in \mathcal{H}$ such that $f \subseteq g$ and $b \in \mathrm{Dom}(g)$.

The main result about existential k-pebble games is that they characterize expressibility in the logic $\exists L^k_{\infty\omega}$ in the following sense:

Theorem 12 [KV00a] *Let L be a relational vocabulary, let $\mathcal{C}$ be a class of finite structures over L, and let Q be a Boolean query on $\mathcal{C}$. Then, the following are equivalent:*

(i) Q is definable in $\exists L^k_{\infty\omega}$ on $\mathcal{C}$,

(ii) For every two structures $\mathbf{A}, \mathbf{B} \in \mathcal{C}$, if $\mathbf{A} \models Q$ and Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$, then $\mathbf{B} \models Q$.

Equipped with the tool of the existential pebble games, the proof of Theorem 11 will proceed as follows. Suppose that C_n satisfies conditions (i) and (ii) in the hypothesis of the theorem, and that C_n is definable on $3CNF_n$ by an $\exists L^k_{\infty\omega}$-sentence φ with $k \leq n/(\ln(n))^2$. The core of the proof will be a probabilistic argument that, using (ii), will show that there exists some 3-CNF formula F such that $M(F) \in C_n$ and Duplicator wins the existential k-pebble game on $M(F)$ and T, where T is the template structure defined in Section 5.2. Then we apply Theorem 12 and infer that $T \models \varphi$, and so $T \in C_n$ because $T \in 3CNF_n$ and φ

defines C_n on $3CNF_n$. By (i), T must be the encoding of an unsatisfiable 3-CNF formula. However, this is absurd because the identity mapping is a trivial homomorphism from T to T. Indeed, T is the encoding of the unique 3-CNF formula on the variables $\{v_1, v_2\}$ that contains every clause except those falsified by the truth assignment $v_1 \mapsto 0$ and $v_2 \mapsto 1$. Consequently, T is the encoding of a satisfiable 3-CNF formula.

5.5 Extension Axioms and the Matching Game

In order to execute the proof idea that we outlined in Section 5.4 we will need to develop some combinatorial machinery. We do it in this section.

A directed graph $G = (M, E)$ is called *bipartite* if there is a partition of the set of nodes M into two sets U and V such that $E \subseteq U \times V$. A *partial matching* π *of* G is a subset of the edges not sharing an endpoint. If $(u, v) \in \pi$, we write $\pi(u) = v$. Note that this notation is not ambiguous. We say that π *is defined on* u, and we define the *domain* $\mathrm{Dom}(\pi)$ *of* π as the set of nodes of U on which π is defined. We also define the *range* $\mathrm{Ran}(\pi)$ *of* π as $\{\pi(u) : u \in \mathrm{Dom}(\pi)\}$. If $A \subseteq U$, we say that A *is matchable into* V *in* G if there exists a partial matching π of G such that $\mathrm{Dom}(\pi) = A$. If $u \in U$, we let $N_G(u) = \{v \in V : (u, v) \in E\}$, and if $v \in V$, we let $N_G(v) = \{u \in U : (u, v) \in E\}$. The left-degree (resp. right-degree) of G is the maximum size of $N_G(u)$ as u ranges over U (resp. V). If $A \subseteq U$, then $N_G(A) = \bigcup_{u \in A} N_G(u)$. For $\epsilon > 0$, we say that G *is an* (s, ϵ)*-bipartite expander* if for every $A \subseteq U$ of size at most s, the size of $N_G(A)$ is at least $(1+\epsilon)|A|$. Observe that in an (s, ϵ)-bipartite expander, every subset of U of size at most s is matchable into V.

This is true by Hall's Theorem:

Theorem 13 (Hall's Theorem) *If $G = (U \cup V, E)$ is a bipartite graph on U and V, and $A \subseteq U$, then A is matchable into V in G if and only if $|N_G(B)| \geq |B|$ for every $B \subseteq A$.*

Observe also that an immediate consequence of Hall's Theorem is that if $A \subseteq U$ is minimally non-matchable into V, then $|N_G(A)| < |A|$. We will use this observation later.

Let $G = (U \cup V, E)$ be a bipartite graph on U and V. The *matching game on G with k fingers* was introduced by Ben-Sasson and Galesi. The game is played by two players: Prover and Disprover. Each player has k fingers numbered $\{1, \ldots, k\}$. In each round of the game, Prover may place a finger over an uncovered node in U or remove a finger from a covered node in U. If Prover places a finger over node $u \in U$, Disprover must place her corresponding finger over an uncovered node in $N_G(u) \subseteq V$. If Prover removes a finger from a node in U, Disprover must remove her corresponding finger from V. The game is over when Disprover is not able to answer to a move of the Prover. In that case, we say that Prover wins the game. If Disprover can make the game go on forever, we say that Disprover wins the game. Notice that at every non-final round, the fingers placed on G determine a partial matching of G. The goal of Disprover is to *maintain* a partial matching forever.

Ben-Sasson and Galesi found a sufficient condition on the graph G that guarantees a win for Disprover. They proved that if G is an (s, ϵ)-bipartite expander, then Disprover wins the matching game with $r \leq \epsilon s/(2 + \epsilon)$ fingers. Although our main result would already follow from this, we prefer to give an alternative and self-contained proof of a similar result with the aim of isolating the extension axioms on which the strategy of Disprover relies. The result that we obtain is that if G is an (s, ϵ)-bipartite expander of left-degree at most l, then Disprover wins the matching game with $r \leq \epsilon s/(l+\epsilon)$ fingers. Although our result is weaker than the one of Ben-Sasson and Galesi, our proof is somewhat simpler (Claim 4.6 and 4.7 in [BSG01] are avoided). Moreover, we will consider bipartite graphs whose left-degree is 3, and therefore the weakness of our result is really minor.

Let us isolate an important property of bipartite graphs.

Definition 18 *Let $r \leq s$, let $G = (U \cup V, E)$ be a bipartite graph on U and V, and let $A \subseteq U$ and $B \subseteq V$ be such that $|A| = |B| \leq r$. We say that (G, A, B) satisfies the (r, s)-matching property if the following holds: For every $C \subseteq U - A$ of size $s - |B|$, there exists a partial matching from C to $V - B$ in G.*

The following is the main result of this section.

Theorem 14 *If G is an (s, ϵ)-bipartite expander of left-degree at most l, and $r \leq \epsilon s/(l+\epsilon)$, then Disprover wins the matching game on G with r fingers.*

Proof: The strategy of Disprover is to guarantee that $(G, \mathrm{Dom}(\pi_t), \mathrm{Ran}(\pi_t))$ satisfies the (r, s)-matching property, where π_t is the partial matching of the game at the end of round t. Observe that $(G, \emptyset, \emptyset)$ satisfies the (r, s)-matching property because in an (s, ϵ)-bipartite expander, every subset of U of size s is matchable into V. The following two lemmas will be used to show that Disprover can reply to the moves that Prover makes according to her strategy.

Lemma 8 (Extension Axiom) *If (G, A, B) satisfies the (r, s)-matching property, $u \in U - A$, and $|A| = |B| < r$, then there exists $v \in N_G(u) \cap (V - B)$ such that $(G, A \cup \{u\}, B \cup \{v\})$ satisfies the (r, s)-matching property.*

Proof: Let $\{v_1, \ldots, v_e\} = N_G(u) \cap (V - B)$. Clearly $e \leq l$ because the left-degree of G is at most l. Also $e \geq 1$ because $u \in U - A$ and every subset of $U - A$ of size $s - |B| > s - r \geq 0$ is matchable into $V - B$. Let $A' = A \cup \{u\}$, and for every $i \in \{1, \ldots, e\}$ let $B^i = B \cup \{v_i\}$. Note that $|A'| = |B^i| \leq r$ since $|A| = |B| < r$. Suppose for contradiction that (G, A', B^i) does not satisfy the (r, s)-matching property for any $i \in \{1, \ldots, e\}$. Let $C^i \subseteq U - A'$ be of size $s - |B^i|$ and non-matchable into $V - B^i$. Let $D^i \subseteq C^i$ be minimally non-matchable into $V - B^i$. Necessarily, $|N_G(D^i) \cap (V - B^i)| < |D^i|$ by Hall's Theorem. The rest of neighbors of D^i can only be in B^i, so

$$|D^i| + |B^i| > |N_G(D^i)| \geq (1 + \epsilon)|D^i| \tag{5.1}$$

for every $i \in \{1, \ldots, e\}$. Recall that $|D^i| \leq |C^i| = s - |B^i| \leq s$ and the expansion property of G for the second inequality. We derive a lower bound on $|D^i|$ for some $i \in \{1, \ldots, e\}$. We first claim that $\bigcup_{i=1}^{e} D^i \cup \{u\}$ is not matchable into $V - B$. Indeed, suppose that $\pi \subseteq E$ is such a matching, and let E^i be the image $\pi(D^i)$ of D^i. Then $v_i \in E^i$ since D^i is minimally non-matchable into $V - B^i$. Therefore, u cannot be matched to any of its neighbors in $V - B$; a contradiction. Now, since $\bigcup_{i=1}^{e} D^i \cup \{u\} \subseteq U - A$ and (G, A, B) satisfies the (r, s)-matching property, we conclude that

$$\left| \bigcup_{i=1}^{e} D^i \cup \{u\} \right| > s - |B|.$$

Therefore, using $|B^i| = |B| + 1$, there must exist some $i \in \{1, \ldots, e\}$ such that

$$|D^i| > \frac{s - |B^i|}{e} \geq \frac{s - |B^i|}{l}.$$

Plugging this bound into (5.1) we get

$$|B^i| > \frac{\epsilon s}{l + \epsilon} \geq r,$$

a contradiction. □

Lemma 9 (Retraction Axiom) *If (G, A, B) satisfies the (r, s)-matching property and $(u, v) \in E \cap (A \times B)$, then $(G, A - \{u\}, B - \{v\})$ satisfies the (r, s)-matching property.*

Proof: Let $A' = A - \{u\}$ and $B' = B - \{v\}$. Clearly $|A'| = |B'| \leq r$. Let $C \subseteq U - A'$ be of size $s - |B'|$. Suppose first that $u \in C$. Then $C - \{u\} \subseteq U - A$ and has size $s - |B'| - 1 = s - |B|$. Therefore, $C - \{u\}$ is matchable into $V - B$ since (G, A, B) satisfies the (r, s)-matching property. It follows that C is matchable into $V - B'$ by adding the edge (u, v) to that matching. Suppose next that $u \notin C$. For every $w \in C$, the set $C - \{w\} \subseteq U - A$ is matchable into $V - B \subseteq V - B'$ since (G, A, B) satisfies the (r, s)-matching property. It follows that if C is not matchable into $V - B'$, then it is minimally non-matchable, and so $|N_G(C) \cap (V - B')| < |C|$ by Hall's Theorem. The rest of neighbors of C can only be in B' and therefore

$$|C| + |B'| > |N_G(C)| \geq (1 + \epsilon)|C|.$$

Recall that $|C| = s - |B'| \leq s$ and the expansion property of G for the second inequality. We conclude that

$$|B'| > \frac{\epsilon s}{1 + \epsilon} \geq \frac{\epsilon s}{l + \epsilon} \geq r,$$

a contradiction. □

We complete the proof. Suppose that $(G, \mathrm{Dom}(\pi_t), \mathrm{Ran}(\pi_t))$ satisfies the (r, s)-matching property. Consider the move of Prover at time $t+1$. If Prover places a finger over an uncovered node $u \in U - \mathrm{Dom}(\pi_t)$, Disprover answers by some $v \in N_G(u) \cap (V - \mathrm{Ran}(\pi_t))$ such that $(G, \mathrm{Dom}(\pi_t) \cup \{u\}, \mathrm{Ran}(\pi_t) \cup \{v\})$ satisfies the (r, s)-matching property. Such a v exists by Lemma 8 because $|\mathrm{Dom}(\pi_t)| = |\mathrm{Ran}(\pi_t)| < r$. If Prover removes a finger from a node $u \in \mathrm{Dom}(\pi_t)$, Disprover removes the corresponding finger from $v = \pi_t(u)$, and $(G, \mathrm{Dom}(\pi_t) - \{u\}, \mathrm{Ran}(\pi_t) - \{v\})$ satisfies the (r, s)-matching property by Lemma 9. □

5.6 Proof of Main Result

Let $F = C_1 \wedge \ldots \wedge C_m$ be a 3-CNF formula on the variables $v_1, \ldots, v_n$. We define the *incidence graph $G(F)$ of F* as follows: $G(F)$ is a bipartite graph on the sets of nodes $\{C_1, \ldots, C_m\}$ and $\{v_1, \ldots, v_n\}$, and (C_i, v_j) is an edge in $G(F)$ if and only if $v_j \in Var(C_i)$. Here $Var(C_i)$ is used to denote the set of variables that occur in C_i. Notice that the left-degree of $G(F)$ is at most 3.

Let F be a 3-CNF formula, let V be a set of variables from F, and let $\mathcal{C} \subseteq F$ be the subset of the clauses of F that mention some variable from V. In symbols,

$$\mathcal{C} = \{C \in F : Var(C) \cap V \neq \emptyset\}.$$

Let π be a partial matching of $G(F)$ such that $\mathrm{Dom}(\pi) = \mathcal{C}$. We define the *partial truth assignment ρ corresponding to π and V* as follows: For every $x \notin V$, the partial truth assignment ρ is undefined on x. If $x \in V$ and $\pi(C) = x$ for some $C \in \mathcal{C}$, then $\rho(x) = 0$ if x occurs both positively and negatively in C, $\rho(x) = 1$ if x occurs positively in C only, and $\rho(x) = 0$ if x occurs negatively in C only. Finally, if $x \in V$ but $x \notin \mathrm{Ran}(\pi)$, then $\rho(x) = 0$. Note that ρ is well-defined because π is a partial matching. In the following, we say that a partial truth assignment ρ *falsifies a clause* C if ρ sets all literals of C to 0.

Lemma 10 *Let F be a 3-CNF formula and let ρ be the partial truth assignment corresponding to some partial matching π of $G(F)$ and some set of variables V. Then ρ does not falsify any clause from F.*

Proof: Let $\mathcal{C}$ be the set of clauses from F that mention some variable in V. Let C be an arbitrary clause from F. If $C \notin \mathcal{C}$, then every variable of C is left undefined by ρ and C cannot be falsified. If $C \in \mathcal{C}$, then $\pi(C)$ occurs in C and $\rho(\pi(C))$ is set to satisfy C. □

The next result connects the Matching Game and the Existential Pebble Game.

Theorem 15 *Let F be a 3-CNF formula such that $G(F)$ has right-degree at most d. If Disprover wins the matching game with r fingers on $G(F)$, then Duplicator wins the existential $\lfloor r/d \rfloor$-pebble game on $M(F)$ and T.*

Proof: Let $G = G(F)$ and $r' = \lfloor r/d \rfloor$. Thus, G is a bipartite graph between the clauses and the variables of F. Duplicator will simulate a play of the matching game on G on the side trying to satisfy the following condition: Clause C has a finger on in it in the matching game at time t if and only if some variable $x \in Var(C)$ has a pebble on it in the existential r'-pebble game at time t. In other words, if V_t is the set of pebbled variables in the existential r'-pebble game at the end of round t, and $\mathcal{C}_t$ is the set of clauses of F that mention some variable in V_t, then $\mathcal{C}_t$ is the set of fingered clauses in the matching game at the end of round t. Since at most r' variables are pebbled simultaneously in the existential r'-pebble game, and since the right degree of G is at most d, at most $r'd \leq r$ fingers will be required. Duplicator will make use of the winning strategy of Disprover in the matching game on G with r fingers. In the following, let π_t be the partial matching in the matching game at the end of round t. Note that $\pi_0 = \emptyset$ and $\mathrm{Dom}(\pi_t) = \mathcal{C}_t$. We consider the two possible moves of Spoiler at time $t+1$ separately.

If Spoiler puts a pebble on variable x, Duplicator responds as follows. Let $\mathcal{D} = \{C_1, \ldots, C_l\}$ be the set of clauses of F on which variable x occurs. Duplicator considers each clause of $\mathcal{D}$ one at a time, starting by C_1. If C_1 does not have a finger on it in the matching game at time t, Duplicator simulates the winning strategy of Disprover as if Prover had placed a finger over C_1, and proceeds to C_2. Otherwise, she proceeds to C_2 directly. This procedure is repeated until C_l is considered and treated. Let π_{t+1} be the partial matching in the matching game at that point. Then Duplicator replies in the existential r'-pebble game as follows: If some C_i exists such that $\pi_{t+1}(C_i) = x$, Duplicator answers 0 if x occurs both positively and negatively in C_i, answers 1 if x occurs positively in C_i only, and answers 0 if x occurs negatively in C_i only. Otherwise, Duplicator answers 0. Note for the record that Duplicator is answering consistently with the partial truth assignment corresponding to π_{t+1} and V_{t+1}, the partial matching and the set of pebbled variables in the existential r'-pebble game at time $t+1$.

If Spoiler removes a pebble from variable x, the answer of Duplicator is clear: she removes the corresponding pebble from T. However, she will need to do some bookkeeping in the matching game: Let $\mathcal{D} = \{C_1, \ldots, C_l\}$ be the set of clauses of F on which variable

x occurs. Duplicator considers each clause of $\mathcal{D}$ one at a time, starting by C_1. If some other variable $y \in Var(C_1) - \{x\}$ has a pebble on it in the existential r'-pebble game at time t, Duplicator proceeds to C_2. If there is no $y \in Var(C_1) - \{x\}$ as above, Duplicator simulates the matching game as if Prover had removed his finger from C_1. Then Duplicator proceeds to C_2. This procedure is repeated until C_l is considered and treated. Let π_{t+1} be the partial matching in the matching game at that point.

We need to argue two points: (1) that Duplicator can follow this strategy, and (2) that if she does, she wins the existential r'-pebble game on $M(F)$ and T. For (1), recall that each variable occurs in at most d clauses, and therefore, since at most $r' = \lfloor r/d \rfloor$ variables are pebbled in the existential r'-pebble game, at most r fingers are used in the matching game. It follows that Duplicator can simulate the winning strategy of Disprover in the matching game. For (2), suppose that Spoiler has not won at time t; we show that he does not win at time $t+1$. When Spoiler removes a pebble, Duplicator cannot loose since she removes the corresponding pebble from T. When Spoiler puts a new pebble, Duplicator answers according to the partial truth assignment corresponding to the partial matching π_{t+1} and the set of pebbled variables V_{t+1}, and this cannot falsify a clause from F by Lemma 10. This completes the proof of the Theorem. □

Finally, we are ready for the proof of the main result. Let us recall the statement of the theorem:

Theorem 16 *For all constants $\delta > 0$, $\Delta > 0$, and for all sufficiently large n, if $C_n \subseteq 3CNF_n$ is such that* *(i)* $C_n \subseteq$ *3UNSAT$_n$ and*

(ii) $\Pr[M(F) \in C_n] \geq \delta$ *when F is drawn from $\mathcal{F}(n, \Delta n, 3)$,*

then C_n is not definable in $\exists L^k_{\infty\omega}$ on $3CNF_n$ for $k \leq n/(\ln(n))^2$.

Proof: Fix $\delta > 0$, $\Delta > 0$, and let n be large. Let $k \leq n/(\ln(n))^2$. Suppose for contradiction that C_n satisfies (i) and (ii), and φ is an $\exists L^k_{\infty\omega}$ sentence defining C_n on $3CNF_n$. We will show that there exists a 3-CNF formula F such that $M(F) \in C_n$ and $G(F)$ is an $(n/\ln(n), 1/4)$-bipartite expander of right-degree at most $d = \ln(n)/13$ and left-degree at most 3. This will be enough because then: (1) $M(F) \models \varphi$ and (2) Duplicator wins the existential k-pebble game on $M(F)$ and T by Theorem 14 and Theorem 15. This means that $T \models \varphi$ by Theorem 12, so that $T \in C_n \subseteq$ 3UNSAT$_n$; a contradiction since T is isomorphic to a satisfiable 3-CNF formula on at most n variables (in fact, on two variables).

Let us prove the existence of F as above by the probabilistic method. First we bound the probability that the right degree of $G(F)$ is bigger than d. Let $d_R(G(F))$ denote the right-degree of the bipartite graph $G(F)$.

Lemma 11 *Let F be drawn from $\mathcal{F}(n, \Delta n, 3)$. Then,*

$$\Pr\left[d_R(G(F)) \geq d\right] \leq n\left(\frac{3e\Delta}{d}\right)^d.$$

Proof: Note that a variable v_i occurs in a random clause with probability $\binom{n-1}{2}\binom{n}{3}^{-1} = 3/n$. For every $i \in \{1, \ldots, n\}$, let X_i be the random variable that counts the number of clauses of F in which variable v_i occurs. By the above, the expectation of X_i is $\Delta n \cdot 3/n = 3\Delta$. More important,

$$\Pr[X_i \geq d] \leq \binom{\Delta n}{d}\left(\frac{3}{n}\right)^d \leq \left(\frac{\Delta n e}{d}\right)^d \left(\frac{3}{n}\right)^d = \left(\frac{3e\Delta}{d}\right)^d.$$

Here we used the well-known inequality $\binom{n}{k} \leq (ne/k)^k$ that is derivable using the fact that $k! \geq (k/e)^k$. Now, let X be the random variable that counts the number of $i \in \{1, \ldots, n\}$ such that $X_i \geq d$. The expectation of X is at most $n(3e\Delta/d)^d$. Therefore, the probability that $X > 0$ is at most $n(3e\Delta/d)^d$ by Markov's inequality. □

Next we bound the probability that $G(F)$ is not an (s, ϵ)-bipartite expander.

Lemma 12 *Let $s > 0$ and $\epsilon > 0$ be such that $(1+\epsilon)s \leq n$ and let F be drawn from $\mathcal{F}(n, \Delta n, 3)$. Then,*

$$\Pr\left[G(F) \textit{ not } (s,\epsilon)\textit{-bipartite expander}\right] \leq \frac{C(1-C^s)}{1-C},$$

where $C = \Delta e^{2+\epsilon}(1+\epsilon)^{2-\epsilon}(s/n)^{1-\epsilon}$.

Proof: Write $G = G(F) = (U_1 \cup U_2, E)$. Let $\gamma = 1+\epsilon$. We bound the probability that for a fixed set $U \subseteq U_1$ of size $i \leq s$ and a fixed set $V \subseteq U_2$ of size γi it holds that $N_G(U) \subseteq V$. For a fixed $u \in U$, the probability that $N_G(u) \subseteq V$ is bounded by $\binom{\gamma i}{3}\binom{n}{3}^{-1}$. Since each clause is drawn independently of the rest, the probability that $N_G(U) \subseteq V$ is bounded by

$$\left[\frac{\binom{\gamma i}{3}}{\binom{n}{3}}\right]^i \leq \left(\frac{\gamma i}{n}\right)^{3i}.$$

Here we used the fact that $\binom{n}{k} / \binom{m}{k} \leq (n/m)^k$ if $n \leq m$. Therefore, the probability that G is not an (s,ϵ)-bipartite expander is bounded by

$$\sum_{i=1}^{s}\binom{\Delta n}{i}\binom{n}{\gamma i}\left(\frac{\gamma i}{n}\right)^{3i} \leq \sum_{i=1}^{s}\left(\frac{\Delta n e}{i}\right)^i\left(\frac{ne}{\gamma i}\right)^{\gamma i}\left(\frac{\gamma i}{n}\right)^{3i} =$$

$$= \sum_{i=1}^{s} \left[\Delta e^{2+\epsilon}(1+\epsilon)^{2-\epsilon} \left(\frac{i}{n}\right)^{1-\epsilon} \right]^{i}.$$

The largest term into brackets is achieved when $i = s$. Thus, we can bound the sum by

$$\sum_{i=1}^{s} C^{i} \leq \frac{C(1-C^{s})}{1-C}$$

as was to be proved. □

Finally, we bound the probability that the right-degree of $G(F)$ is bigger than d, or $G(F)$ is not an (s, ϵ)-expander, or $M(F)$ does not belong to C_n by

$$n\left(\frac{3e\Delta}{d}\right)^{d} + \frac{C(1-C^{s})}{1-C} + 1 - \delta. \tag{5.2}$$

From our choice of $d = \ln(n)/13$, $s = n/\ln(n)$, and $\epsilon = 1/4$, we immediately see that (5.2) is strictly smaller than 1 for sufficiently large n (recall that $\delta > 0$ and $\Delta > 0$ are constants). Thus, some F must exist for which $G(F)$ is an (s, ϵ)-expander of right-degree at most d, and $M(F)$ belongs to C_n. This completes the proof of the Theorem. □

Chapter 6

Consequences for Proof Complexity

6.1 Easier Games

The aim of this section is to show that certain combinatorial statements such as the Pigeonhole Principle, when encoded as contradictory sets of clauses in the spirit of Cook and Reckhow [CR79], may yield structures on which playing pebble games is easier than expected.

The Pigeonhole Principle states that there is no one-to-one mapping from a set of $n + 1$ pigeons into a set of n holes. This statement may be expressed in propositional logic as follows: For $i \in \{1, \ldots, n + 1\}$ and $j \in \{1, \ldots, n\}$, let $p_{i,j}$ be a propositional variable meaning that pigeon i is sitting in hole j. The clause $P_i = p_{i,1} \vee \ldots \vee p_{i,n}$ says that pigeon i sits in some hole. The clause $H_k^{i,j} = \neg p_{i,k} \vee \neg p_{j,k}$ for $i \neq j$ says that pigeons i and j cannot sit in the same hole k. The conjunction PHP_n^{n+1} of all the clauses P_i and $H_k^{i,j}$ is an unsatisfiable CNF since it expresses the negation of the Pigeonhole Principle. Note that PHP_n^{n+1} is not a 3-CNF formula since the clauses P_i involve n literals. However, it is easy to turn it into an equivalent 3-CNF as follows: For every $i \in \{1, \ldots, n+1\}$ and $j \in \{0, \ldots, n\}$, let $y_{i,j}$ be a new propositional variable. Then EP_i is the following 3-CNF formula: $\neg y_{i,0} \wedge \bigwedge_{j=1}^{n} (y_{i,j-1} \vee p_{i,j} \vee \neg y_{i,j}) \wedge y_{i,n}$. Finally, the 3-CNF formula $EPHP_n^{n+1}$ expressing the negation of the Pigeonhole Principle is the conjunction of all clauses EP_i and all clauses $H_k^{i,j}$. Here is our promised result about pebble games and the Pigeonhole Principle:

Theorem 17 *Duplicator wins the existential n-pebble game on $M(EPHP_n^{n+1})$ and T.*

Proof: Consider the following strategy for Duplicator. She will keep track of a partial matching π_t in the complete bipartite graph K_n^{n+1} that satisfies the following property: Pigeon $i \in \{1, \ldots, n+1\}$ belongs to $\mathrm{Dom}(\pi_t)$ if and only if some variable of pigeon i is pebbled at the end of round t. Of course, the variables of pigeon i are $p_{i,1}, \ldots, p_{i,n}$ and $y_{i,0}, \ldots, y_{i,n}$. Notice that if there are p pebbles on the board at the end of round t, then $|\mathrm{Dom}(\pi_t)| = |\mathrm{Ran}(\pi_t)| \leq p$. Initially $\pi_0 = \emptyset$. Let us consider the possible moves of Spoiler at time $t+1$ separately:

If Spoiler places a pebble over a variable $p_{i,j}$, Duplicator acts as follows: If $i \in \mathrm{Dom}(\pi_t)$, Duplicator answers 1 if $j = \pi_t(i)$ and 0 otherwise. In that case, $\pi_{t+1} = \pi_t$. If $i \notin \mathrm{Dom}(\pi_t)$, Duplicator finds some $k \in \{1, \ldots, n\} - \mathrm{Ran}(\pi_t)$ and sets $\pi_{t+1} = \pi_t \cup \{(i,k)\}$. Note that such a k exists since at most $n-1$ pebbles are placed on the board at the end of round t, and therefore $|\mathrm{Dom}(\pi_t)| = |\mathrm{Ran}(\pi_t)| \leq n-1$. Then she answers 1 if $j = k$ and 0 otherwise.

If Spoiler places a pebble over a variable $y_{i,j}$, Duplicator acts as follows: If $i \in \mathrm{Dom}(\pi_t)$, Duplicator answers 1 if $j \geq \pi_t(i)$ and 0 otherwise. In that case, $\pi_{t+1} = \pi_t$. If $i \notin \mathrm{Dom}(\pi_t)$, Duplicator finds some $k \in \{1, \ldots, n\} - \mathrm{Ran}(\pi_t)$ and sets $\pi_{t+1} = \pi_t \cup \{(i,k)\}$.

Such a k exists for the same reason as above. Then she answers 1 if $j \geq k$ and 0 otherwise.

If Spoiler removes a pebble from a variable, the answer of Duplicator is clear: she removes the corresponding pebble in T. However, she will need to do some bookkeeping in the partial matching π_t. Suppose that the variable from which Spoiler removed the pebble belongs to pigeon i. If no other variable of pigeon i is pebbled at time t, Duplicator sets $\pi_{t+1} = \pi_t - \{(i, \pi_t(i))\}$. Otherwise, Duplicator sets $\pi_{t+1} = \pi_t$.

It is not hard to see that Duplicator wins the existential n-pebble game when playing according to this strategy since a clause is never falsified. □

We can do the same with a form of the Pigeonhole Principle that is based on a bipartite graph. Let U and V be disjoint sets of sizes $n+1$ and n respectively. Let $G = (U \cup V, E)$ be a bipartite graph on U and V. Obviously, there is no matching from U into V. This is expressed by the conjunction of the following set of clauses on the variables $\{x_{u,v} : (u,v) \in E\}$. For every $u \in U$, let $P_u = \bigvee_{(u,v)\in E} x_{u,v}$, and for every $(u,w), (v,w) \in E$, $u \neq v$, let $H_w^{u,v} = \neg x_{u,w} \vee \neg x_{v,w}$. We write $G\text{-}PHP$ for the conjunction of these clauses. Observe that if $G = K_n^{n+1}$, then $G\text{-}PHP$ coincides with PHP_n^{n+1}. Observe also that if G has left-degree at most k, then $G\text{-}PHP$ is a k-CNF.

We assign a partial truth assignment ρ to every partial matching π in G. For every $u \in \mathrm{Dom}(\pi)$, let $\rho(x_{u,\pi(u)}) = 1$ and $\rho(x_{u,v}) = 0$ for every $v \in V - \{\pi(u)\}$ such that $(u,v) \in E$.

Lemma 13 *Let $G = (U \cup V, E)$ be a bipartite graph, let π be a partial matching in G, and let ρ be the partial truth assignment corresponding to π. Then, ρ does not falsify any clause* *from G-PHP.*

Proof: The clauses of the form $\neg x_{u_1,v} \vee \neg x_{u_2,v}$ are not falsified by ρ since at most one of (u_1, v) or (u_2, v) is in the partial matching π. Similarly, the clauses of the form $x_{u,v_1} \vee \ldots \vee x_{u,v_d}$ are not falsified by ρ since if $\rho(x_{u,v_j}) = 0$ for some $j \in \{1, \ldots, d\}$, then $\rho(x_{u,v_i}) = 1$ for some other $i \in \{1, \ldots, d\}$. □

Theorem 18 *If $G = (U \cup V, E)$ is an (s, ϵ)-expander of left-degree at most l, and $r \leq \epsilon s/(l + \epsilon)$, then Duplicator wins the existential r-pebble game on $M($G-$PHP)$ and T.*

Proof: We design a strategy for Duplicator to win the existential r-pebble game on the structures $M($G-$PHP)$ and T. Duplicator will simulate the play of a matching game on G on the side trying to satisfy the following condition: Node $u \in U$ has a finger on it in the matching game at time t if and only if there exists some $v \in N_G(u)$ such that $x_{u,v}$ has a pebble on it in the existential r-pebble game at time t. This condition is clearly satisfied at time $t = 0$. Duplicator will make use of the winning strategy of Disprover in the matching game on G with r fingers. In the following, let π_t be the partial matching in the matching game at time t, so that $\pi_0 = \emptyset$. We consider the two possible moves of Spoiler at time $t + 1$ separately.

If at time $t + 1$ Spoiler removes a pebble from variable $x_{u,v}$ in the existential r-pebble game, Duplicator acts as follows: Note that u must have a finger on it at time t in the matching game since $x_{u,v}$ has a pebble at time t in the existential r-pebble game. Thus, $\pi_t(u)$ is well-defined. If some $w \in V - \{\pi_t(u)\}$ exists such that the variable $x_{u,w}$ is pebbled in the existential r-pebble game, Duplicator does nothing in the matching game, so that $\pi_{t+1} = \pi_t$. Then she removes the pebble from T corresponding to $x_{u,v}$ in the existential r-pebble game.

On the other hand, if no $w \in V - \{\pi_t(u)\}$ exists as above, Duplicator removes the fingers from nodes u and $\pi_t(u)$ in the matching game as if Prover had removed his finger from node u and Disprover had responded accordingly. Thus, $\pi_{t+1} = \pi_t - \{(u, \pi_t(u))\}$ in that case. Then, she removes the pebble from T corresponding to $x_{u,v}$ in the existential r-pebble game.

If at time $t + 1$ Spoiler places a pebble over variable $x_{u,v}$ in the existential r-pebble game, Duplicator acts as follows: If u has a finger on it in the matching game already, Duplicator does nothing in that game, so that $\pi_{t+1} = \pi_t$. Then, in the existential r-pebble game, she answers 1 if $v = \pi_t(u)$, and 0 otherwise. If u did not have a finger in the matching game, Duplicator finds some $w \in V$ to match u according to the winning strategy of Disprover as if Prover had placed a finger over u in the matching game. Thus, $\pi_{t+1} = \pi_t \cup \{(u, w)\}$ in that case. Then, in the existential r-pebble game, she answers 1 if $w = v$ and 0 otherwise.

We need to argue two points: (1) that Duplicator can follow this strategy, and (2) that if she does, she wins the existential r-pebble game on $M(F)$ and T. For (1), observe that since at most r pebbles are involved in the existential r-pebble game, at most r nodes from U have a finger in the matching game. It follows that Duplicator can always find the required w when needed according to the winning strategy of Disprover in the matching game. For (2), suppose that Spoiler has not won at time t; we show that he does not win at time $t + 1$. If Spoiler removed a pebble from variable $x_{u,v}$, say, Duplicator cannot lose there since she removed the corresponding pebble in T. If Spoiler placed a pebble over variable $x_{u,v}$, say, Duplicator will always answer according to the truth values inferred by a partial matching on G, and these never falsify a clause from $G\text{-}PHP$ by Lemma 13. This completes the proof of the Theorem. □

6.2 On the Complexity of Resolution

Resolution is probably the most popular proof system for propositional logic. There is only one rule: from clauses $C \cup \{x\}$ and $D \cup \{\neg x\}$ derive $C \cup D$. A Resolution refutation of a CNF formula F is a sequence of clauses ending with the empty clause $\{\}$, each of which is either (1) a clause of F, (2) a clause appearing earlier in the sequence, or (3) follows from two previous clauses in the sequence by the Resolution rule. The length of a Resolution refutation is the length of the sequence of clauses. The width of a Resolution refutation is the maximum number of distinct literals in a clause of the refutation.

Theorem 19 *There exists a $2k$-Datalog sentence φ such that for every 3-CNF formula F, the following holds:*

(i) If F has a Resolution refutation of width k, then $M(F) \models \varphi$.

(ii) If $M(F) \models \varphi$, then F has a Resolution refutation of width $2k$.

Proof: Consider the following set of Datalog rules. For every w and i such that $0 \leq i \leq w \leq 2k$, let S_i^w be a new predicate symbol of arity w. Consider the following rule that we call (A):

$$S_i^3(x_1, x_2, x_3) \quad :- \quad R_i(x_1, x_2, x_3)$$

The meaning of (A) is that every clause of F is derivable. Add also the following rules that we call (B) and (C) respectively:

$$S_i^w(x_1, \ldots, x_p/x_q, \ldots, x_q/x_p, \ldots, x_i, y_1, \ldots, y_{w-i}) \quad :- \quad S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i})$$

$$S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{p'}/y_{q'}, \ldots, y_{q'}/y_{p'}, \ldots, y_{w-i}) \quad :- \quad S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i})$$

where $1 \leq p < q \leq i$ in (B), $1 \leq p' < q' \leq w - i$ in (C). The meaning of (B) is that if the clause $\{\neg x_1, \ldots, \neg x_i, y_1, \ldots, y_{w-i}\}$ is a derivable, then any result of permuting two x-literals in it is also derivable. Similarly for (C). The following four clauses are called (D), (E), (F) and (G) respectively:

$$\begin{aligned}
S_i^w(x, \ldots, x, x, x_1, \ldots, x_a, y_1, \ldots, y_{w-i}) \quad &:- \quad S_i^w(x, \ldots, x, x_1, x_1, \ldots, x_a, y_1, \ldots, y_{w-i}) \\
S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_b, y, y, \ldots, y) \quad &:- \quad S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_b, y_b, y, \ldots, y) \\
S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i-1}, y_{w-i-1}) \quad &:- \quad S_{i+1}^w(x_1, x_1, \ldots, x_i, y_1, \ldots, y_{w-i-1}) \\
S_{i+1}^w(x_1, x_1, \ldots, x_i, y_1, \ldots, y_{w-i-1}) \quad &:- \quad S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i-1}, y_{w-i-1})
\end{aligned}$$

where $0 < a < i - 1$ in (D), $0 < b < w - i - 1$ in (E), and $0 < i < w - 1$ in (F) and (G). The meaning of rule (D) is that if $\{\neg x, \ldots, \neg x, \neg x_1, \neg x_1, \ldots, \neg x_a, y_1, \ldots, y_{w-i}\}$ is derivable, so is the result of contracting one $\neg x_1$ and adding one $\neg x$. Similarly for (E). The meaning of (F) is that if $\{\neg x_1, \neg x_1, \ldots, \neg x_i, y_1, \ldots, y_{w-i-1}\}$ is derivable, so is the result of contracting one $\neg x_1$ and adding one y_{w-i-1}. Similarly for (G). We introduce two rules that, together with (B), . . . ,(G) can be used to eliminate repeated literals. The rules are called (H), (I):

$$\begin{aligned}
S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i}) \quad &:- \quad S_i^{w+p}(x_1, \ldots, x_i, y_1, \ldots, y_{w-i}, y_{w-i}, \ldots, y_{w-i}) \\
S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i}) \quad &:- \quad S_{i+p}^{w+p}(x_1, \ldots, x_1, x_1, \ldots, x_i, y_1, \ldots, y_{w-i}),
\end{aligned}$$

as long as $w + p \leq 2k$, $0 \leq i < w$ in (H) and $0 < i \leq w$ in (I). Next we introduce the rule that simulates the Resolution rule that we call (J):

$$\begin{aligned}
&S_{r_1+r_2-1}^{w_1+w_2-2}(x_2, \ldots, x_{r_1}, x'_1, \ldots, x'_{r_2}, y_1, \ldots, y_{w_1-r_1}, y'_1, \ldots, y'_{w_2-r_2-1}) :- \\
&\quad :- \quad S_{r_1}^{w_1}(z, x_2, \ldots, x_{r_1}, y_1, \ldots, y_{w_1-r_1}), S_{r_2}^{w_2}(x'_1, \ldots, x'_{r_2}, y'_1, \ldots, y'_{w_2-r_2-1}, z),
\end{aligned}$$

where $w_1 + w_2 - 2 \leq 2k$, $0 < r_1 \leq w_1$ and $0 \leq r_2 < w_2$. The meaning of (J) is that if the clauses $\{\neg z, \neg x_2, \dots, \neg x_{r_1}, y_1, \dots, y_{w_1 - r_1}\}$ and $\{\neg x'_1, \dots, \neg x'_{r_2}, y'_1, \dots, y'_{w_2 - r_2 - 1}, z\}$ have been derived, then the resolvent clause is also derivable by an application of the Resolution rule. Note that the resulting clause may have repeated literals. Note also that we do not insist that variable z appears only once in each clause in order to apply the Resolution rule. The soundness of the rule is not lost and we will make use of this fact later. Finally, add a new predicate symbol G and the rule (K):

$$G \quad :- \quad S^1_0(x), S^1_k(x).$$

The meaning of (K) is that if $\{x\}$ and $\{\neg x\}$ are derivable, then F is refutable.

Let φ be the conjunction of all these Datalog rules (ABCDEFGHIJK). Simple inspection reveals that each rule can be written with at most $2k$ variables. Thus, φ is a $2k$-Datalog program. It is a routine task (but somewhat tedious) to check that if F has a Resolution refutation of width k then $M(F) \models \varphi$. Similarly, if $M(F) \models \varphi$ then F has a Resolution refutation of width $2k$. The proof follows.

We first prove that if F has a Resolution refutation of width k, then $M(F) \models \varphi$. Indeed, suppose that $C_1, C_2, \dots, C_m$ is a Resolution refutation of F of width k. We may assume without loss of generality that $C_{m-2} = \{x\}$, $C_{m-1} = \{\neg x\}$ and $C_m = \{\}$. For $j \in \{1, \dots, m-1\}$, let us write $C_j = \{\neg x_{j,1}, \dots, \neg x_{j,r_j}, y_{j,1}, \dots, y_{j,s_j}\}$ where all displayed variables are different and $r_j + s_j \leq k$. Let $w_j = r_j + s_j$ be the width of C_j. We will show, by induction on $j \in \{1, \dots, m-1\}$, that

$$S^{w_j}_{r_j}(x_{j,1}, \dots, x_{j,r_j}, y_{j,1}, \dots, y_{j,s_j})$$

is derivable from $M(F)$ and the Datalog rules. Fix $j \geq 1$ and suppose that the claim holds for all smaller values of j. If C_j is a clause of F, the conclusion is immediate from rule (A), and rules (B) , . . . ,(I). Suppose then that C_j follows from C_h and C_l by the Resolution rule. Here $h < l < j$. By induction hypothesis, we know that

$$S^{w_h}_{r_h}(x_{h,1}, \ldots, x_{h,r_h}, y_{h,1}, \ldots, y_{h,s_h})$$

$$S^{w_l}_{r_l}(x_{l,1}, \ldots, x_{l,r_l}, y_{l,1}, \ldots, y_{l,s_l})$$

are derivable. Since C_h and C_l are resolved together to obtain C_j, some variable must appear negatively in C_h and positively in C_l, or conversely. Without loss of generality, let us assume that the former holds. Using the rules (B) and (C), we may also assume that $x_{h,1} = y_{l,s_l}$ is the cut variable z. Now let us apply the rule (J) to obtain

$$S^{w_h+w_l-2}_{r_h+r_l-1}(x_{h,2}, \ldots, x_{h,r_h}, x_{l,1}, \ldots, x_{l,r_l}, y_{h,1}, \ldots, y_{h,s_h}, y_{l,1}, \ldots, y_{l,s_l-1}).$$

Then apply the rules (B), . . . ,(I) to obtain

$$S^{w_h+w_l-2}_{r_j}(x_{j,1}, \ldots, x_{j,r_j}, y_{j,1}, \ldots, y_{j,s_j}, y_{j,s_j}, \ldots, y_{j,s_j}).$$

or

$$S^{w_h+w_l-2}_{w_h+w_l-2}(x_{j,1}, \ldots, x_{j,1}, x_{j,1}, \ldots, x_{j,r_j})$$

depending on whether $s_j > 0$ or $s_j = 0$ respectively. Finally, apply the rule (H) or (I) to obtain

$$S^{w_j}_{r_j}(x_{j,1}, \ldots, x_{j,r_j}, y_{j,1}, \ldots, y_{j,s_j}).$$

We have proved that $S^1_0(x)$ and $S^1_1(x)$ are derivable. Thus, G is derivable, so $M(F) \models \varphi$.

Let us prove the reverse direction, namely that if $M(F) \models \varphi$, then F has a Resolution refutation of width $2k$. In order to prove that, we will need to introduce a mild extension of Resolution as an intermediary step. The new proof system incorporates the so-called weakening rule: from clause C, derive $C \cup D$ where D is an arbitrary clause. In Lemma 14 below, we prove that weakenings may be eliminated without increasing the width, and therefore allowing them cannot hurt. Next we prove that if

$$S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i})$$

holds, then the clause

$$\{\neg x_1, \ldots, \neg x_i, y_1, \ldots, y_{w-i}\} \tag{6.1}$$

has a Resolution derivation with weakening from F in width $2k$. The proof is by induction on the stage s at which the tuple $(x_1, \ldots, x_i, y_1, \ldots, y_{w-i})$ entered in the least fixed-point of the Datalog rules. Let $s > 0$ and suppose that the claim holds for all smaller values of s. If the tuple entered following rule (A), the claim is obvious since then (6.1) is an initial clause of F. If the tuple entered following one of the rules (B), . . . ,(I), the claim is also clear since clauses are sets of literals. Finally, if the tuple entered following the rule (J), we proceed as follows: Let us rename $x_1, \ldots, x_i, y_1, \ldots, y_{w-i}$ in the form of the conclusion of rule (J); that is

$$(x'_2, \ldots, x'_{r_1}, x''_1, \ldots, x''_{r_2}, y'_1, \ldots, y'_{w_1-r_1}, y''_1, \ldots, y''_{w_2-r_2-1}), \tag{6.2}$$

where both

$$S_{r_1}^{w_1}(z, x'_2, \ldots, x'_{r_1}, y'_1, \ldots, y'_{w_1-r_1})$$

$$S_{r_2}^{w_2}(x''_1, \ldots, x''_{r_2}, y''_1, \ldots, y''_{w_2-r_2-1}, z)$$

hold and have entered in a previous stage. If z does not occur in (6.2), then (6.1) is simply the result of resolving the clauses

$$\{\neg z, \neg x'_2, \ldots, \neg x'_{r_1}, y'_1, \ldots, y'_{w_1 - r_1}\}$$

$$\{\neg x''_1, \ldots, \neg x''_{r_2}, y''_1, \ldots, y''_{w_2 - r_2 - 1}, z\}$$

on z. If on the contrary, z occurs in (6.2), then we may obtain (6.1) by weakening over one of these two clauses. Which clause depends on whether z occurs positively or negatively. This completes the proof of the Theorem. □

Before we go on we need to address the following lemma that we left without proof. The result it asserts is well-known, but we include a complete proof for completeness.

Lemma 14 *Let $C_1, \ldots, C_r$ be clauses. If $\{C_1, \ldots, C_r\}$ has a Resolution refutation with weakening of width w and length s, then it also has a Resolution refutation without weakening of width at most w and length at most s.*

Proof: We prove, by induction on m, that if $D_1, \ldots, D_m$ is a Resolution derivation with weakening, then there exist a Resolution derivation $D'_1, \ldots, D'_m$ without weakening such that $D'_i \subseteq D_i$ for every $i \in \{1, \ldots, m\}$. Let $m \geq 1$ and suppose that the claim holds for all smaller values. Let $D'_1, \ldots, D'_{m-1}$ be the derivation claimed by the induction hypothesis. We consider three cases according to the nature of D_m. If D_m is an initial clause, let $D'_m = D_m$ and we are done. If D_m is derived from D_k with $k < m$ by weakening, we let $D'_m = D'_k$ and we are also done because $D'_k \subseteq D_k \subseteq D_m$. If D_m is derived from D_k and D_j with $k < j < m$ by a cut with $x \in D_k$ and $\neg x \in D_j$, we consider three cases. If x appears in D'_k and $\neg x$ appears in

D'_j, we let D'_m be the result of cutting D'_k and D'_j on x. Then

$$D'_m = D'_k \cup D'_j - \{x, \neg x\} \subseteq D_k \cup D_j - \{x, \neg x\} \subseteq D_m.$$

If x appears in D'_k but $\neg x$ does not appear in D'_j, we let $D'_m = D'_j$. Then

$$D'_m = D'_j = D'_j - \{x, \neg x\} \subseteq D_j - \{x, \neg x\} \subseteq D_m.$$

Finally, if x does not appear in D'_k, we let $D'_m = D'_k$ and again

$$D'_m = D'_k = D'_k - \{x, \neg x\} \subseteq D_k - \{x, \neg x\} \subseteq D_m.$$

Since we covered all possible cases, this completes the proof of the lemma. □

An immediate corollary of the proof of Theorem 11 and Theorem 19 is that every Resolution refutation of a a random 3-CNF formula requires width $n/2(\ln n)^2$ with probability approaching 1 as n approaches ∞. Now, Ben-Sasson and Wigderson [BSW01] proved that if F is a 3-CNF on n variables that has a Resolution refutation of length S, then F has a Resolution refutation of width $O\left(\sqrt{n \log S}\right)$. Thus, we obtain a slightly weaker bound than that of Chvátal and Szeméredi.

Corollary 4 *Let F be drawn from $\mathcal{F}(n, 6n, 3)$. Then, almost surely, F is unsatisfiable and every Resolution refutation of F requires size $2^{\Omega(n/(\ln n)^4)}$.*

From Theorem 18 we can also obtain Haken's lower bound:

Corollary 5 *Every resolution refutation of PHP_n^{n+1} requires length $2^{\Omega(n)}$.*

Proof: It is known that there exist $(\Omega(n), 1/4)$-bipartite expanders of left-degree 3 on sets of size $n + 1$ and n (an easy probabilistic argument as in the proof Theorem 11 does it). Therefore, for such a bipartite expander $G = (U \cup V, E)$, every Resolution refutation of the set of clauses G-PHP requires width $\Omega(n)$ by Theorem 18. Since the number of variables of G-PHP is $O(n)$, we conclude from the size-width trade-off of Ben-Sasson and Wigderson that every Resolution refutation of G-PHP requires size $2^{\Omega(n)}$. Although the lower bound for PHP_n^{n+1} follows from this at once, we review the proof for completeness.

Suppose for contradiction that PHP_n^{n+1} has a resolution refutation of length $2^{o(n)}$. Let $C_1, \ldots, C_r$ be such a refutation. Let ρ be the partial truth assignment to the variables $\{p_{i,j}\}$ that sets $\rho(p_{i,j}) = 0$ if $(i, j) \notin E$. Here we are assuming without loss of generality that $U = \{1, \ldots, n + 1\}$ and $V = \{1, \ldots, n\}$. For every $i \in \{1, \ldots, r\}$, let $C_i' = \rho(C_i)$ be the result of applying ρ to C_i. That is, C_i' is 1 if ρ sets some literal of C_i to 1, and the subset of non-falsified literals of C_i otherwise. We claim that the sequence of clauses $C_1', \ldots, C_r'$ is a Resolution refutation with weakening of G-$PHP \cup \{1\}$. If this is true, then G-PHP has a Resolution refutation of length $2^{o(n)}$ since the clause 1 cannot be used in any cut.

We will show, by induction on $m \in \{1, \ldots, r\}$, that $C_1', \ldots, C_m'$ is a valid Resolution derivation from G-$PHP \cup \{1\}$. Suppose that $m \geq 1$ and that the claim is valid for every other smaller value. If C_m is an initial clause of PHP_n^{n+1}, it is easily seen that either $C_m' = 1$, or C_m' is an initial clause of G-PHP. Suppose next that C_m is derived by the resolution rule from C_k and C_l with $k < l < m$. Suppose that the cut variable is $p_{i,j}$. The immediate case is when $C_k' \neq 1$ and $C_l' \neq 1$ because then C_m' is simply the result of applying the resolution rule on them. Also, if $C_k' = 1$ and $C_l' = 1$, then $C_m' = 1$ because some satisfied literal must be inherited by C_m. Finally, if $C'_k = 1$ and $C'_l \neq 1$, say, there are two cases two consider. If the satisfied literal in C_k is not about variable $p_{i,j}$, then this literal is inherited in C_m and so $C'_m = 1$. If the satisfied literal in C_k is about variable $p_{i,j}$ indeed, then it must be $\neg p_{i,j}$. In that case, C'_l does not contain the literal $p_{i,j}$ because $\rho(p_{i,j}) = 0$. We conclude that C'_m is a weakening of C'_l and we are done. □

Chapter 7

Conclusions

7.1 General Discussion

The results of this dissertation advance the state of knowledge about the expressive power of least fixed-point logics on finite structures from two different perspectives. In the first part of the thesis we studied the effect of powerful built-in predicates, such as the membership relation between hereditarily finite sets, and in the second part we proved non-expressibility results for Datalog without built-in predicates. The results in the first part delineate the boundary where the fragments of LFP collapse to FO on finite structures with built-in membership relation, and provide new insights on computational aspects of LFP. The results in the second part establish inherent limitations in the expressive power of Datalog on random 3-CNF formulas. In both cases, the results tightly connect with well-known problems in complexity theory, such as questions about Boolean circuit uniformity in the first case, and questions about random satisfiability and proof complexity in the other. Moreover, we believe that the results are also interesting as results of finite model theory in their own right, independently of the complexity-theoretic connections.

We conclude with some remarks on the results presented here and with some open problems that warrant further investigation.

7.2 About Finite Set Theory

The result stating that the least fixed-points of positive restricted Δ_0-formulas are Δ_1-definable on $\mathcal{BFR}$ is a valuable expressibility tool. We demonstrated this by means of non-trivial examples, such as Example 9 in which we showed that the query

$$Q_{\mathrm{rk}}(M) = \{a \in M : a = V_m \text{ for some } m \geq 0\}$$

is definable as the least fixed-point of a positive restricted Δ_0-formula. We note that, although possible in principle, the direct construction of a Σ_1 and a Π_1-formula that defines Q_{rk} on $\mathcal{BFR}$ seems a harder approach. Second, the fact that the binary fragment of LFP(Δ_0) collapses to FO on $\mathcal{BFR}$ is both unexpected and useful. It is unexpected since at first sight it seems that quantification over binary relations of bounded elements is needed, while we show that monadic quantification over such elements suffices. It is useful because, once again, it provides a versatile expressibility tool.

Moreover, the set-theoretic framework suggests new ideas and new problems to consider that may provide new insights. Recall that the class of Σ-formulas is the smallest class of formulas that contains the Δ_0-formulas and is closed under conjunction, disjunction, and existential quantification only. Recall also that the class of Σ_1-formulas is the class of Σ-formulas in which all unbounded quantifiers precede the bounded ones. We note that it is known that $\Sigma = \Sigma_1$ on V_ω. Does this collapse propagate to $\mathcal{BFR}$?

Open Problem 1 *Is $\Sigma = \Sigma_1$ on $\mathcal{BFR}$?*

The positive answer to this question would mean that the class of Σ_1-formulas forms a very robust class. Moreover, such a result would constitute a highly non-trivial computational speedup. Consider the following Σ-formula

$$(\forall x_1 \in x)(\forall x_2 \in x)(\forall x_3 \in x)(\exists y)(\psi(x_1, x_2, x_3, y, x))$$

with ψ being a Δ_0-formula. Note that the evaluation of such a formula on $\mathcal{BFR}$ by a non-deterministic Turing machine seems to require $O((\log n)^4)$ steps of non-determinism, since for each triple x_1, x_2, and x_3 of elements of x, y, and z respectively, the machine needs to guess $\log n$ bits for the unbounded existential quantifier $(\exists y)$. However, $O(\log n)$ steps of non-determinism suffice to evaluate any Σ_1-formula on $\mathcal{BFR}$. On the other hand, the negative answer to the question might give new information on the well-known problems $\mathbf{NLIN} \overset{?}{=} \mathbf{co\text{-}NLIN}$, and $\mathbf{NLIN} \overset{?}{=} \mathbf{LINH}$.

Let us consider a second open problem suggested by the set-theoretic framework. We know, and we have used that intensively, that every Δ_0-definable predicate is absolute. In fact, by standard absoluteness arguments, even Δ-definable predicates on $\mathcal{BFR}$ are absolute. One can easily show, though, that $\Delta \neq \Delta_0$ on $\mathcal{BFR}$. Indeed, using the well-known result of Furst, Saxe and Sipser [FSS84] that $\mathbf{AC}^0$ circuits cannot compute the parity of its input, it is not hard to see that the query

$$Q_{\text{even}}(M) = \{a \in M : \text{the cardinality of } a \text{ is even}\}$$

is not Δ_0-definable on $\mathcal{BFR}$. Since we showed in Example 10 that $Q_{\text{even}}(M)$ is definable as the least fixed-point of a binary positive Δ_0-formula, we know that it is Π-definable. Moreover, it is easy to see that the greatest fixed-point of that formula coincides with the least fixed-point, and so the query is indeed Δ-definable on $\mathcal{BFR}$. It seems to us that the use of the circuit lower bound for parity to prove that $\Delta \neq \Delta_0$ is an overkill, since Δ_0-formulas are limited to quantify over the transitive closure of the input. Is it possible to prove this lower bound directly, by means of Ehrenfeucht-Fraïssé games?

Open Problem 2 *Prove that $\Delta \neq \Delta_0$ on $\mathcal{BFR}$ using Ehrenfeucht-Fraïssé games.*

The solution may give some insight on the following important question: how do we design winning strategies for the Duplicator in Ehrenfeucht-Fraïssé games when both arithmetic built-in predicates $+$ and $\times$ are available? We remark that successful strategies have been designed when only $+$ is available [Lyn82].

7.3 About Uniformity of Circuits

A descriptive complexity characterization of a complexity class, that is, the proof that a certain logic captures it, usually provides new information and motivates further the class. Moreover, if the logic that captures the class is natural enough, it allows us to build up very useful analogies. One of the most appealing analogies is the trichotomy provided by first-order logic FO, monadic second-order logic MSO, and full second-order logic SO. On finite structures with built-in membership relation, these three logics capture the Logarithmic-Time Hierarchy $\mathbf{LH}$ [BIS90], the Linear-Time Hierarchy $\mathbf{LINH}$, and the Polynomial-Time

Hierarchy $\mathbf{PH}$ [Sto77], respectively. Note that the fact that SO captures $\mathbf{PH}$ can be extended to arbitrary finite structures, but provably not in the other two cases.

The results of Chapter 4 showed that $\mathrm{FO}+\mathrm{LFP}(\Delta_0)$ captures $\mathbf{LH}^{\mathbf{P}}$, or equivalently $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$, on $\mathcal{BFR}$. In this case, the logic $\mathrm{FO}+\mathrm{LFP}(\Delta_0)$ arose naturally from the set-theoretic interpretation of the built-in predicate and the fact that the Δ_0-formulas had been successfully used in that field. The identification of the underlying complexity class leads to another interesting trichotomy. The logics FO, $\mathrm{FO}+\mathrm{LFP}(\Delta_0)$ and $\mathrm{FO}+\mathrm{LFP}$ capture $\mathbf{DLOGTIME}$-uniform $\mathbf{AC}^0$, $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$, and $\mathbf{P}$-uniform $\mathbf{AC}^0$, respectively, on $\mathcal{BFR}$. However, we want to make the point that complexity classes should not only be motivated by their underlying computational model or their descriptive characterization, but also by the computational problems that they contain. This leads us to the following remarks about $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$.

For 15 years after the time that Beame, Cook and Hoover [BCH86] introduced the technique of discrete logarithms for integer division, the most efficient Boolean circuits for integer division had only been $\mathbf{P}$-uniform, which means that the descriptions of the circuits were computable in polynomial-time, but no less. This situation was quite annoying since the uniformity of circuits for natural problems tends to be quite low in the complexity hierarchy, actually, as low as $\mathbf{DLOGTIME}$-uniform. Hesse's recent result that integer division is in $\mathbf{DLOGTIME}$-uniform $\mathbf{TC}^0$ resolved this issue with an elegant proof after a number of previous attempts [Hes01].

Recall now the discussion in Chapter 4 about the $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$ circuits for integer division of numbers of polylogarithmic length. Before Hesse's result, it was not known how to improve the uniformity of such circuits from $\mathbf{DPOLYLOGTIME}$ to $\mathbf{DLOGTIME}$. This was in sharp contrast with the $\mathbf{DLOGTIME}$-uniform $\mathbf{AC}^0$ circuits for the rest of arithmetic operations on numbers of polylogarithmic length such as addition and multiplication. We note that the restriction to numbers of polylogarithmic length is a necessary one since, from the strongest lower bounds for $\mathbf{AC}^0$ circuits, it follows that neither multiplication nor division of longer numbers can be done in $\mathbf{AC}^0$ [Has86, FSS84]. The fact that the problem about integer division has been solved, and therefore that division of numbers of polylogarithmic length is also in $\mathbf{DLOGTIME}$-uniform $\mathbf{AC}^0$, leaves us without candidates to separate $\mathbf{DPOLYLOGTIME}$-uniform $\mathbf{AC}^0$ from $\mathbf{DLOGTIME}$-uniform $\mathbf{AC}^0$. After all, it is still possible that $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ collapses to FO, as do the unary and binary fragments. This state of affairs motivates the following problem:

Open Problem 3 *Does ternary* $\mathrm{LFP}(\Delta_0)$ *collapse to* FO *on* $\mathcal{BFR}$*?*

As a matter of fact, we are inclined to think that $\mathbf{P} \subseteq \mathbf{LINH}$, just as $\mathbf{NL} \subseteq \mathbf{LINH}$ [Nep70, Nep73]. Recall, though, that a proof would imply $\mathbf{P} \neq \mathbf{PSPACE}$.

7.4 About Inexpressibility Results and Proof Complexity

The fact that strong logics capture complexity classes implies that tight inexpressibility results for these logics are very hard to obtain. Indeed, a proof that unsatisfiability of 3-CNF formulas is not expressible in existential second-order logic would imply $\mathbf{NP} \neq \textbf{co-}\mathbf{NP}$, and so $\mathbf{P} \neq \mathbf{NP}$. In contrast, inexpressibility results of this type are possible for logics, such as Datalog, that do not capture any particular complexity class. What is the value of these results?

Let us focus on Datalog. While Datalog fails short to capture any complexity class defined in terms of the standard computational models, many important algorithmic techniques can be "implemented" as Datalog programs. In particular, this holds true for algorithmic techniques capable of solving such problems as unsatisfiability of 2-CNF formulas, graph reachability, or the circuit value problem. Note that the latter problem is $\mathbf{P}$-complete but Datalog is not closed under logspace reductions, and that is why it falls short to capture $\mathbf{P}$.

The fact that natural classes of algorithms can be implemented in Datalog makes inexpressibility results powerful. A clear example of this is provided by the results in Chapter 6, where we showed that a $2k$-Datalog program is able to implement the proof search algorithm for Resolution of width k. This algorithm was studied already in 1977 by Galil [Gal77], and revisited by Ben-Sasson and Wigderson [BSW01]. Moreover, the important results of Ben-Sasson and Wigderson that relate Resolution width with Resolution size, together with our inexpressibility result for Datalog, allowed us to re-derive the known lower bounds for Resolution size. In this sense then, inexpressibility results for logics that fall short to capture complexity classes can still be used to derive complexity-theoretic lower bounds.

The main issue that remains open from the results of Chapters 5 and 6 is that of establishing further connections between inexpressibility results and lower bounds for proof complexity. The first observation we want to make in this respect is related to a potential application of proof-complexity techniques to finite-model-theory techniques. The second observation is related to a synergy in the reverse direction.

Inexpressibility results are usually proved by designing a winning strategy for the

Duplicator, one of the players of the appropriate Ehrenfeucht-Fraïssé game for the logic under consideration. For example, inexpressibility results for $\exists L^k_{\infty\omega}$ are proved by exhibiting a winning strategy for the Duplicator in the existential k-pebble game. It turns out that the design of winning strategies for the Duplicator is often a highly non-trivial task. To make things worse, there is a clear lack of techniques to describe the winning strategies in a precise mathematical form, other than by enumerating essentially all possible cases. However, there is an important exception: when extension axioms are available, winning strategies for the Duplicator are easy to describe. The proof of the result of Chapter 5 made strong use of extension axioms, which, in turn, were derived from the expansion properties of certain graphs. This suggests that further extension-type axioms are yet to be discovered, and the experience in using expander properties in propositional proof complexity may turn out to be helpful.

The second observation that we want to make with respect to the interaction between finite model theory and proof complexity is the following: it seems intuitively clear that if Datalog relates to Resolution in the precise sense of Theorem 19 of Chapter 6, then stronger fixed-point logics should relate to stronger proof systems. This suggests two problems. The first one is in finite model theory, proper.

Open Problem 4 *Extend the inexpressibility results of Datalog on random 3-CNF formulas to stronger fixed-point logics, such as Least Fixed-Point Logic LFP, or even full infinitary logic with finitely many variables $L^\omega_{\infty\omega}$.*

Clearly, the same problem could be posed for other logics such as first-order logic, existential monadic second-order logic (monadic $\mathbf{NP}$), or logics with counting quantifiers.

The last open problem of this section is stated in slightly more general terms. It should be considered as a family of open problems, one for each logic and proof system, rather than as concrete open problem with a precise statement.

Open Problem 5 *Establish further connections between inexpressibility results for natural logics and lower bounds for propositional proof systems.*

We hope that the results of this dissertation, together with the solution to some of these open problems, may contribute to the synergy between the fields of finite model theory and propositional proof complexity in their quest to classify the complexity of algorithmic problems.

Bibliography

[ABE02] A. Atserias, M. L. Bonet, and J. L. Esteban. Lower bounds for the weak pigeonhole principle and random formulas beyond resolution. *Information and Computation*, 176(2):136–152, 2002.

[Ach01] D. Achlioptas. A survey of lower bounds for random 3-SAT via differential equations. *Theoretical Computer Science*, 265(1–2):159–185, 2001.

[AG91] E. Allender and V. Gore. Rudimentary reductions revisited. *Information Processing Letters*, 40:89–95, 1991.

[AK99] A. Atserias and Ph. G. Kolaitis. First-order logic vs. fixed-point logic in finite set theory. In *14th IEEE Symposium on Logic in Computer Science*, pages 275–284, 1999.

[Ats00] A. Atserias. The descriptive complexity of the fixed-points of bounded formulas. In P. Clote and H. Schwichtenberg, editors, *Computer Science Logic '2000, 14th Annual Conference of the EACSL*, volume 1862 of *Lecture Notes in Computer Science*, pages 172–186. Springer-Verlag, 2000.

[Ats02] A. Atserias. Unsatisfiable random formulas are hard to certify. In *17th IEEE Symposium on Logic in Computer Science*, pages 325–334, 2002.

[Bar75] J. Barwise. *Admissible Sets and Structures*. Springer-Verlag, 1975.

[BCH86] P. W. Beame, S. A. Cook, and H. J. Hoover. Log depth circuits for division and related problems. *SIAM Journal of Computing*, 15(4):994–1003, 1986.

[BDG90] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity II*. Springer-Verlag, 1990.

[BFU93] A. Z. Broder, A. M. Frieze, and E. Upfal. On the satisfiability and maximum satisfiability or random 3-CNF formulas. In *4th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 322–330, 1993.

[BIS90] D.M. Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41(3):274–306, 1990.

[BKPS98] P. Beame, R. Karp, T. Pitassi, and M. Saks. On the complexity of unsatisfiability proofs for random k-CNF formulas. In *30th Annual ACM Symposium on the Theory of Computing*, pages 561–571, 1998.

[BN78] R. V. Book and M. Nivat. Linear languages and the intersection closures of classes of languages. *SIAM Journal of Computing*, 7(2):167–177, 1978.

[BSG01] E. Ben-Sasson and N. Galesi. Space complexity of random formulas in resolution. In *16th IEEE Conference on Computational Complexity*, pages 42–51, 2001.

[BSI99] E. Ben-Sasson and R. Impagliazzo. Random CNF's are hard for the polynomial calculus. In *40th Annual IEEE Symposium on Foundations of Computer Science*, pages 415–421, 1999.

[BSW01] E. Ben-Sasson and A. Wigderson. Short proofs are narrow–resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.

[Bus87] S. R. Buss. The boolean function value problem is in ALOGTIME. In *28th Annual IEEE Symposium on Foundations of Computer Science*, pages 123–131, 1987.

[Bus98] S. R. Buss. First-order proof theory of arithmetic. In *Handbook of proof theory*, pages 79–147. Elsevier Science, 1998.

[CH82] A. Chandra and D. Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25:99–128, 1982.

[CKS81] A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the ACM*, 28:114–133, 1981.

[CR79] S. Cook and R. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44:36–50, 1979.

[CR92] V. Chvátal and B. Reed. Mick gets some (the odds are on his side). In *33rd Annual IEEE Symposium on Foundations of Computer Science*, pages 620–627, 1992.

[CS88] V. Chvátal and E. Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, 1988.

[DDLW98] A. Dawar, K. Doets, S. Lindell, and S. Weinstein. Elementary properties of finite ranks. *Mathematical Logic Quarterly*, 44:349–353, 1998.

[DH95] A. Dawar and L. Hella. The expressive power of finitely many generalized quantifiers. *Information and Computation*, 123:172–184, 1995.

[DLW96] A. Dawar, S. Lindell, and S. Weinstein. First order logic, fixed point logic and linear order. In H. K. Büning, editor, *Computer Science Logic '95*, volume 1092 of *Lecture Notes in Computer Science*, pages 161–177. Springer-Verlag, 1996.

[Fag74] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. Karp, editor, *Complexity of Computation*, pages 27–41. SIAM-AMS Proceedings 7, 1974.

[Fag76] R. Fagin. Probabilities on finite models. *Journal of Symbolic Logic*, 41:50–58, 1976.

[FSS84] M. Furst, J. Saxe, and M. Sipser. Parity, circuits and the polynomial time hierarchy. *Mathematical Systems Theory*, 17:13–27, 1984.

[FV98] T. Feder and M. Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal of Computing*, 28(1):57–104, 1998.

[Gal77] Z. Galil. On resolution with clauses of bounded size. *SIAM Journal of Computing*, 6:444–459, 1977.

[GIS94] Y. Gurevich, N. Immerman, and S. Shelah. McColm's conjecture. In *9th IEEE Symposium on Logic in Computer Science*, pages 10–19, 1994.

[Grä92] E. Grädel. Capturing complexity classes by fragments of second-order logic. *Theoretical Computer Science*, 101:35–57, 1992.

[Gur88] Y. Gurevich. Logic and the challenge of computer science. In E. Börger, editor, *Current Trends in Theoretical Computer Science*, pages 1–57. Computer Science Press, 1988.

[Hak85] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

[Has86] J. Hastad. *Computational limitations for small depth circuits*. PhD thesis, Massachussets Institute of Technology, 1986.

[Hes01] W. Hesse. Division is in uniform TC^0. In *28th International Colloquium on Automata, Languages and Programming*, volume 2076 of *Lecture Notes in Computer Science*, pages 104–114. Springer-Verlag, 2001.

[IL95] N. Immerman and S. Landau. The complexity of iterated multiplication. *Information and Computation*, 116(1):103–116, 1995.

[Imm86] N. Immerman. Relational queries computable in polynomial time. *Information and Computation*, 68:86–104, 1986.

[Imm89] N. Immerman. Expressibility and parallel complexity. *SIAM Journal of Computing*, 18:625–638, 1989.

[Imm99] N. Immerman. *Descriptive Complexity*. Springer-Verlag, 1999.

[Jon69] N. Jones. Context-free languages and rudimentary attributes. *Mathematical Systems Theory*, 3:102–109, 1969.

[Jon75] N. D. Jones. Space-bounded reducibility among combinatorial problems. *Journal of Computer and System Sciences*, 11:68–85, 1975. Corrigendum: *Journal of Computer and System Sciences* 15:241, 1977.

[KKL02] A. C. Kaporis, L. M. Kirousis, and E. G. Lalas. The probabilistic analysis of a greedy satisfiability algorithm. In *10th Annual European Symposium on Algorithms*, volume 2461 of *Lecture Notes in Computer Science*, pages 574–585. Springer, 2002.

[KKS$^+$01] A. C. Kaporis, L. M. Kirousis, Y. C. Stamatiou, M. Vamvakari, and M. Zito. Coupon collectors, q-binomial coefficients and the unsatisfiability threshold. In *7th Italian Conference on Theoretical Computer Science*, volume 2202 of *Lecture Notes in Computer Science*, pages 328–338. Springer, 2001.

[KL00] A. Koponen and K. Luosto. Definability of group theoretic notions. Manuscript, 2000.

[KM81] B. Krishnamurthy and R. N. Moll. Examples of hard tautologies in the propositional calculus. In *13th Annual ACM Symposium on the Theory of Computing*, pages 28–37, 1981.

[KV90] Ph. G. Kolaitis and M. Y. Vardi. 0-1 laws and decision problems for fragments of second-order logic. *Information and Computation*, 87:302–338, 1990.

[KV92] Ph. G. Kolaitis and M. Y. Vardi. Fixpoint logic vs. infinitary logic in finite-model theory. In *7th IEEE Symposium on Logic in Computer Science*, pages 46–57, 1992.

[KV95] Ph. G. Kolaitis and M. Y. Vardi. On the expressive power of Datalog: tools and a case study. *Journal of Computer and System Sciences*, 51:110–134, 1995.

[KV00a] Ph. G. Kolaitis and M. Y. Vardi. Conjunctive-query containment and constraint satisfaction. *Journal of Computer and System Sciences*, 61(2):302–332, 2000.

[KV00b] Ph. G. Kolaitis and M. Y. Vardi. A game-theoretic approach to constraint satisfaction. In *7th National Conference on Artificial Intelligence*, pages 175–181, 2000.

[Lyn80] J. F. Lynch. Almost sure theories. *Annals of Mathematical Logic*, 18:91–135, 1980.

[Lyn82] J. Lynch. On sets of relations definable by addition. *Journal of Symbolic Logic*, 47(3):659–668, 1982.

[Mos74] Y. N. Moschovakis. *Elementary Induction on Abstract Structures*. North-Holland, 1974.

[Nep70] V. A. Nepomnjascij. Rudimentary predicates and Turing calculations. *Soviet Math. Dokl.*, 11:1462–1465, 1970.

[Nep73] V. A. Nepomnjascij. Rudimentary modelling of indeterminate Turing calculations. *Cybernetics*, 9(2):212–218, 1973.

[Pap85] C. H. Papadimitriou. A note on the expressive power of Prolog. *Bullentin of the EATCS*, 26:21–23, 1985.

[Rei87] J. H. Reif. On threshold circuits and polynomial computation. In *2nd IEEE Structure in Complexity Theory*, pages 118–123, 1987.

[Rob65] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[Ruz81] W. L. Ruzzo. On uniform circuit complexity. *Journal of Computer and System Sciences*, 22:365–383, 1981.

[Saz97] V. Y. Sazonov. On bounded set theory. In *Logic and Scientific Methods*, pages 85–103. Kluwer Academic Publishers, 1997.

[Sip83] M. Sipser. Borel sets and circuit complexity. In *15th Annual ACM Symposium on the Theory of Computing*, pages 61–69, 1983.

[SML96] B. Selman, D. G. Mitchell, and H. J. Levesque. Generating hard satisfiability problems. *Artificial Intelligence*, 81:17–29, 1996.

[Smu61] R. Smullyan. Theory of formal systems. In *Annals of Mathematics Studies*, volume 47. Princeton University Press, 1961.

[SS88] S. Shelah and J. Spencer. Zero-one laws for sparse random graphs. *Journal of the American Mathematical Society*, 1(1):97–115, 1988.

[Sto77] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

[Ull89] J. D. Ullman. Database and knowledge-base systems. *Computer Science Press*, 1989.

[Var82] M. Vardi. Complexity of relational query languages. In *14th Annual ACM Symposium on the Theory of Computing*, pages 137–146, 1982.

[Vol99] H. Vollmer. *Introduction to Circuit Complexity*. Springer-Verlag, 1999.

[Weg87] I. Wegener. *The Complexity of Boolean Functions*. John Wiley & Sons, 1987.

[Wil79] A. J. Wilkie. Applications of complexity theory to Σ_0-definability problems in arithmetic. In *Model Theory of Algebra and Arithmetic*, volume 834 of *Lecture Notes in Mathematics*, pages 363–369. Springer-Verlag, 1979.

[Woo81] A. R. Woods. *Some problems in logic and number theory, and their connections*. PhD thesis, University of Manchester, Department of Mathematics, 1981.

[Wra78] C. Wrathall. Rudimentary predicates and relative computation. *SIAM Journal of Computing*, 7(2):194–209, 1978.