Forcing with random variables in bounded arithmetic, and proof complexity

Jan Krajíček
Charles University in Prague
Faculty of Mathematics and Physics

preliminary version, March 2010

(There will be further revisions and some more material may be included, especially in Part VIII.)

Contents

Preface

Proof complexity is concerned with the mathematical analysis of the informal concept of a feasible proof when the qualification "feasible" is interpreted in a complexity-theoretic sense. The most important measure of complexity of a proof is its length when it is thought of as a string over a finite alphabet. The basic question proof complexity studies is to estimate (from below as well as from above) the minimal possible length of a proof of a formula. Measuring the complexity of a proof by its length may seem crude at first but it is analogous to measuring the complexity of an algorithm by its time complexity.

In the context of propositional logic the main question is whether there exists a proof system in which every propositional tautology has a short proof, a proof bounded in length by a polynomial in the length of the formula. With a suitably general definition of what it is a "proof system", the question is equivalent to the problem whether the computational complexity class NP is closed under complementation.

In the setting of first-order logic one considers theories whose principal axiom scheme is the scheme of induction but accepted only for predicates on binary strings that have limited computational complexity. These are the so called bounded arithmetic theories. A typical question is this: Can we prove more universally valid properties of strings if we assume induction for NP predicates than if we have only induction for polynomial-time predicates?

These two strands of proof complexity are, in fact, very much bounded together and in a precise technical sense one can think of proof systems as non-uniform versions of theories. The two problems mentioned, and their variants, are also quite linked with fundamental problems of computational complexity theory such as the P versus NP problem, the existence of one-way functions, or the possibility of a universal derandomization.

Answering in the negative the propositional problem, that is, showing

that NP is not closed under complementation, implies of course that P and NP are different. Answering in the negative the first-order question would not necessarily separate P and NP but it would show (among other) that the conjecture that P differs from NP is consistent with Cook's theory PV (standing for Polynomially Verifiable). PV has names for all polynomial time algorithms and axioms codifying how these algorithms are built from each other. This would be quite significant as most of contemporary computational complexity theory can be formalized in PV or in its mild extensions.

Interestingly, it is known that the consistency of the statement that P differs from NP with PV follows from a super-polynomial lower bound on the length of Extended Frege propositional proofs of *any* sequence of tautologies. In fact, the consistency[1] of the conjecture P $\neq$ NP with an arbitrary theory T (axiomatized by schemes) follows from any super-polynomial lower bound for a specific propositional proof system P depending on T. This is perhaps one of the reasons why proving lengths-of-proofs lower bounds appears quite difficult even for seemingly simple proof systems.

Obviously, when studying a theory it is quite useful to have a rich class of models. In particular, a separation of two theories may be proved by exhibiting a suitable model. However, it is also known that proving lengths-of-proofs lower bounds for propositional proof systems is equivalent to constructing extensions of particular bounded arithmetic models. For this reason we are interested in methods for constructing models of various bounded arithmetic theories.

In these lecture notes we describe a new class of models of bounded arithmetic. The models are Boolean-valued and are based on random variables. We suggest that the body of results obtained in these notes shows that this provides a coherent and rich framework for studying bounded arithmetic and propositional proof complexity. In particular, we propose this as a framework in which it is possible to think about unconditional independence results for bounded arithmetic theories and about lengths-of-proofs lower bounds for strong proof systems.

[1] The issue of possible formal unsolvability of the P versus NP problem attracted some popular interest recently but unfortunately also authors ignorant of the topic or of logic in general, producing expositions that are - to quote a famous man - not even wrong.

Acknowledgements

I thank to S. A. Cook (Toronto), E. Jeřábek (Prague), P. Pudlák (Prague), N. Thapen (Prague), L. van den Dries (Urbana), S. Todorcevic (Paris/Toronto) and I. Tzameret (Prague) for discussions of related topics and for comments on drafts of parts of the book. I am especially indebted to P. Nguyen (Prague/Montreal) for extensive comments on the draft of the first seven parts, and to S. A. Cook (Toronto) for enduring alone my presentation of attempts at the construction of a model required in Chapter 22 in May 2006 during the program *Logic and Algorithms* at the Isaac Newton Institute in Cambridge.

I lectured about parts of this work at seminars in Oxford in May and November '04, in the IAS in Princeton in February'05, in the Isaac Newton Institute in Cambridge in April'06, and at the meeting Computability in Europe in Swansea in July'06. I also gave a semester long exposition of parts of this work at the Charles University in Prague in Fall'04. I thank participants of all these events for a number of comments.

Support acknowledgments:

I started to work on this book while I was a member of the Institute for Advanced Study in Princeton in Spring'04, supported by the NSF grant DMS-0111298. Parts of this work were done while visiting the Mathematical Institute in Oxford supported by the EPSRC grant GR/S16997/01, the Institute for Advanced Study in Princeton supported by the NSF grant DMS-0111298, and the Isaac Newton Institute in Cambridge (program Logic and Algorithms) supported by an EPSRC grant N09176. Also partially supported over the years by various Czech grant agencies: grants A1019401, IAA100190902 and AV0Z10190503 (AS CR), by MSM0021620839, and by grant LC505 (Eduard Čech Center). A grant from the John F. Templeton Foundation provided a partial support during 2008 - 2010.

For a large part of the work on this project, until Spring 2009, I have been a member of the Institute of Mathematics of the Academy of Sciences of the Czech Republic in Prague.

Introduction

Propositional proof complexity studies the lengths of propositional proofs or equivalently the time complexity of non-deterministic algorithms accepting some coNP-complete set. The main problem is the NP versus coNP problem, a question whether the computational complexity class NP is closed under complementation. Central objects studied are propositional proof systems (non-deterministic algorithms accepting the set of propositional tautologies). Time lower bounds then correspond to lengths-of-proofs lower bounds.

Bounded arithmetic is a generic name for a collection of first-order and second-order theories of arithmetic linked to propositional proof systems (and to a variety of other computational complexity topics). The qualification *bounded* refers to the fact that the induction axiom is typically restricted to a subclass of bounded formulas.

The links between propositional proof systems and bounded arithmetic theories have many facets but informally one can view them as two sides of the same thing: The former is a non-uniform version of the latter. In particular, it is known that proving lengths-of-proofs lower bounds for propositional proof systems is very much related to proving independence results for bounded arithmetic theories. In fact, proving such lower bounds is *equivalent* to constructing non-elementary extensions of particular models of bounded arithmetic theories. This offers a very clean and coherent framework for thinking about lengths-of-proofs lower bounds, one that has been quite successful in the past (let us mention just Ajtai's [2] lower bound for the lengths of proofs of the pigeonhole principle in constant-depth Frege systems, cf. Chapter 21).

In this book we introduce a new method for constructing bounded arithmetic models, and hence for proving independence results and lengths-of-proofs lower bounds. Brief description could be *forcing with random variables* but it has also features of non-standard analysis and of definable ultraprod-

ucts. The novelty lies neither in using forcing in bounded arithmetic or proof complexity (see Remarks on the literature below) nor in forcing with random variables (that is well-established in set theory, cf.[99, 48]), but rather in finding a way how to do this meaningfully in arithmetic, and further in using families of random variables that are sampled by algorithms restricted in a particular way (different from one application to another) rather than using the family of all random variables with a given sample space and range.

The models are built from random variables defined on a sample space Ω which is a non-standard finite set (often parameterized by a subset of $\{0,1\}^n$ with a non-standard n), and sampled by functions of some restricted computational complexity. This is considered inside an $\aleph_1$-saturated non-standard model of true arithmetic. One could equivalently work with sequences of bigger and bigger sample spaces and random variables defined on them, and consider their limit behavior using a suitable ultrafilter on $\mathbf{N}$, simulating indirectly the ultraproduct[2] construction. However, the use of a non-standard model from the beginning simplifies things considerably. This is analogous to the situation in non-standard analysis: While proofs using infinitesimals (and other features of non-standard analysis) can be translated into the $\epsilon - \delta$ formalization, the intuition or clarity of the original argument may be lost in the translation.

Random variables induce probabilistic distributions and probabilities of events. In particular, two random variables may be neither equal nor unequal; rather they may be equal with some probability. However, there is a fundamental problem: Probabilities cannot be used as truth-values if classical logic is to be preserved. The (almost) right choice for the truth-value is the subset of the sample space consisting of those samples for which the two random variables are equal. At the heart of our construction is the realization that we can employ a bit of non-standard analysis (namely Loeb's measure [79]) at this point: If one identifies two such truth-values (subsets of the sample space) if their symmetric difference has an infinitesimal measure one gets a *complete* Boolean algebra - this is the single most important feature of our method. Evaluation of first-order formulas in complete Boolean algebras is very natural and faithful (as it is well-known from Boole [10] for propositional logic and from Rasiowa-Sikorski [89] for predicate logic).

The models we get are not classical but are Boolean-valued[3]. But that is

[2]This construction is explained in a way accessible to readers without a basic logic education in Chapter A.

[3]One can get classical models by applying a bit of logic: First apply the Löwenheim-

perfectly sufficient for the purpose of independence results (and lengths-of-proofs lower bounds): In order to demonstrate that a sentence is not provable from a set of axioms it is enough to show that its truth value in some model is smaller than the truth value of any finite conjunction of the axioms.

Although some of the models appear interesting in their own right we interpret the construction primarily as

> *A method that reduces an independence result or a lengths-of-proofs lower bound to a combinatorial/complexity-theoretic statement about random variables.*

The combinatorial/complexity-theoretic statement we refer to here expresses that the truth-value of a particular sentence (in a particular model) is some particular value, typically $1_{\mathcal{B}}$ or $0_{\mathcal{B}}$. The validity of such a statement is a property of the particular family of random variables forming the model. For the families we consider it can be often formulated as a statement that an algorithm of a certain type can (or cannot) perform some computational task successfully for a high fraction of inputs.

Organization of the book. The book is divided into eight parts and an appendix. Part I (*Basics*) describes the general framework of the construction and develops a few basic properties of the method. This includes witnessing of quantifiers in the structures and linking the validity in the structures with the probability in the standard model. Part II (*Second order structures*) extends the set-up to two sorted structures, with one sort for numbers and the other one for bounded sets.

In Part III (AC^0 *world*) we construct two structures. The first one is a structure based on random variables computed by shallow decision trees. This is a quite rudimentary example and its basic properties are mirrored in several later models. The second structure is based on deep decision trees and it is a model of theory V^0_1. In Part IV ($AC^0(2)$ *world*) we construct an algebraic structure based on random variables defined by algebraic decision trees. This structure is a model of the theory $Q_2V^0_1$, extending V^0_1 by a bounded quantifier allowing to count the parity of a bounded set. The key

Skolem theorem to the whole model-theoretic situation (i.e. not only the model but also the Boolean algebra and the truth valuation) to replace it by a countable one, and then apply the Rasiowa-Sikorski theorem [90] to collapse the Boolean algebra to the two-element Boolean algebra while preserving joins and meets used for defining the truth values, and hence collapsing the model to a classical one.

step in analyzing of both the deep tree model and the algebraic model is bounded quantifier elimination. The combinatorial heart of these elimination procedures are provided by the Hastad's switching lemma and by the Razborov-Smolensky's approximation method respectively. In both Parts III and IV we use the models to derive anew a few known undefinability results, witnessing theorems and independence results for the theories. The purpose of including this material (as well as examples in Part VII) is to demonstrate that the method is a viable alternative to the usual proof-theoretic approach based on some form of a normalization of proofs.

Part V (*Towards proof complexity*) describes a general approach using the models for lengths-of-proofs lower bounds. This follows to a large extent Ajtai's method in [2], but with some important twists. In Part VI (*Proof complexity of F_d and $F_d(\oplus)$*) we use this approach to give a new proof of an exponential lower bound for PHP proofs in constant depth Frege systems. Then we discuss a long standing open problem to prove the same lower bound also for constant depth Frege systems with the parity gate. We do not manage to construct a model that would prove the elusive lower bound but we review some possibly relevant material about algebraic proof systems and, more importantly, we discuss in detail the issues any construction of the wanted structure has to tackle (in particular, the necessity of partially defined random variables). The models considered in this part are quite analogous to models in Parts III and IV.

The structures in Part III - VI are second-order. In Part VII (*Polynomial time and higher worlds*) we return to the first order formalization and we construct several models for theories like PV, S^1_2 and T^1_2 and derive in this way some of the most important known witnessing theorems and conditional independence results in bounded arithmetic. In this part we also note a link between pseudorandom sets and a Löwenheim-Skolem phenomenon. Further, we define a model of PV naturally interpreting structural complexity results about random oracle.

In Part VIII (*Proof complexity of EF and beyond*) we first overview aims of proof complexity of strong proof systems and recall, in particular, background facts relevant to EF and to bounded arithmetic theories related to EF. We expose in some detail the emerging theory of proof complexity generators directed towards constructing examples of hard tautologies. We also discuss several conjectures regarding these generators: on the hardness of proving circuit lower bounds, Razborov's conjecture about the Nisan-Wigderson generator and Extended Frege system, a conjecture about using random sparse

Nisan-Wigderson generators as gadgets in gadget generators, and related Rudich's demi-bit conjecture. Then we construct a few models relevant to some of these conjectures.

The main text is supplemented by an appendix Chapter A in which we present in a self-contained and quite elementary way the construction of an ultrapower extension of the standard model of natural numbers. We also try to convey, using several examples, some mental picture about the model so that even a reader not familiar with non-standard methods can develop some intuition and follow the arguments in the main text.

Remarks on the literature. A form of forcing has been applied in bounded arithmetic earlier. Paris-Wilkie [85] and later Ajtai [2, 3, 4] and Riis [97] used simple variant of Robinson's model-theoretic forcing (although combined with an involved combinatorial reduction in [2, 3, 4]). Wilkie[4] described a construction of Boolean valued models of the theory S^1_2 and reproved using it a relation - known previously from Cook [27] and Buss [11] - between S^1_2 and the Extended Frege proof system EF. His construction has been further extended by Krajíček [54, 56, 59, 55] to a wider context. With a slight simplification one can describe the Boolean algebras involved in these constructions as Lindenbaum algebras but not based on provable equivalence of formulas (or circuits) as they are defined classically but rather on *feasibly provable* (i.e. with proofs of polynomially bounded length) equivalence. This works well in the sense that any valid lower bound can be proved, in principle, by such a forcing. But on the other hand the algebras are defined using the notion of a feasible proof about which we are supposed to say something by the construction in the first place, and so it is in a sense a vicious circle. Takeuti and Yasumoto [102, 103] changed the feasibly provable equivalence to simply "true equivalence" - breaking this vicious circle - but it apparently did not help much as we know very little about the power of Boolean circuits of feasible size. Most importantly, the algebras used in all these constructions are not complete but are closed only under some definable unions. That makes it very hard to use them.

Background. This is an investigation in bounded arithmetic and in proof complexity and we expect that, ideally, the reader is familiar with established basic definitions, facts, methods and aims of the field. The relevant background in bounded arithmetic and proof complexity can be found in

[4]Unpublished lecture at the *Int. Congress on Logic, Methodology and Philosophy of Science* in Moscow, 1987.

Krajíček [55] but some reader may find useful shorter explanations of some basic points in [52, 56, 60, 61] or an excellent survey by Pudlák [88]. Nevertheless we always briefly review the relevant theories and propositional proof systems before they are studied, and thus a reader with at least a minimal logic background should be able to study the book. In addition Chapter 27 gives some very general proof complexity background. Recently Cook and Nguyen [29] offered an excellent exposition of basic theories and their relations to proof systems. A reader looking for a background in model theory, and non-standard models in particular, may consult Chang and Keisler [21], Marker [82] (ultrapowers are there in Exercises 2.5.19 - 2.5.22 and 4.5.37) or Kaye [51].

Despite the natural character and simplicity of Boolean-valued models they were discovered only in late 1960s by P. Vopěnka, and by D. Scott and R. Solovay as their versions of Cohen's forcing; paper by Scott [99] is a beautiful exposition of the basic ideas aimed at non-logicians, best to this date (it also contains detailed bibliographical/historical comments[5]). Paper [99] as well as virtually all later expositions (e.g. in [104, 48]) consider only Boolean-valued models of set theory. Mansfield [81] attempts a general theory but concentrates on model-theoretic properties of the class of such models as opposed to properties of particular models and gives a version that yields only elementary extensions and hence is unsuitable for independence results.

[5]Takeuti reports in [102] that Gödel recognized in Boolean-valued models a model-theoretic version of a reinterpretation of logical operations he developed earlier but never used for independence results as it was too complicated.

Part I

Basics

Chapter 1

The definition of the models

1.1 The ambient model of arithmetic

Let L_{all} be the language containing symbols for every relation and function on the natural numbers $\mathbf{N}$; each symbol from L_{all} has a canonical interpretation in $\mathbf{N}$. Let $\mathcal{M}$ be an $\aleph_1$-saturated model[1] of the true arithmetic in the language L_{all}. Such a model exists by general model-theoretic constructions, cf. [42]. Definable sets mean definable with parameters, unless specified otherwise.

The $\aleph_1$-saturation implies the following:

(1) If a_k, $k \in \mathbf{N}$, is a countable family of elements of $\mathcal{M}$ then there exists a non-standard $t \in \mathcal{M}$ and a sequence $(b_i)_{i<t} \in \mathcal{M}$ such that $b_k = a_k$ for all $k \in \mathbf{N}$.

We shall often denote this sequence of length t simply $(a_i)_{i<t}$.

For example, if all elements $\{a_k\}_{k\in\mathbf{N}}$ obey some definable property P then - by induction in $\mathcal{M}$ (aka overspill, see Chapter A) - also some b_s with a non-standard index $s \leq t$ will obey P. Such element b_s will serve well as "a limit" (interpreted here informally) of the sequence $\{a_k\}_{k\in\mathbf{N}}$.

Another property implied by the $\aleph_1$-saturation (and equal to it if we used a countable language) is the following:

(2) If A_k, $k \in \mathbf{N}$, is a countable family of definable subsets of $\mathcal{M}$ such that $\bigcap_{i<k} A_i \neq \emptyset$ for all $k \geq 1$, then $\bigcap_k A_k \neq \emptyset$.

[1] In appendix Chapter A we give an elementary and self-contained construction (the so called ultrapower) of such a model.

However, the intersection $\bigcap_k A_k$ does not need to be definable in $\mathcal{M}$.

These two statements are essentially the only consequences of the $\aleph_1$-saturation we will use.

The ambient model $\mathcal{M}$ will suffice for our purposes everywhere in this book. However, in general we could take for $\mathcal{M}$ an $\aleph_1$-saturated elementary extension of $\mathbf{N}$ in a many sorted language having names not only for all elements of $\mathbf{N}$ and relations and functions on $\mathbf{N}$ as L_{all} has, but also names for all families of sets, families of families of sets, etc., for the whole so called *superstructure* (this is commonly done in non-standard analysis and this terminology is used there). In such a rich model the properties above would hold also for sequences of sets, families, etc..

1.2 The Boolean algebras

Let $\Omega \in \mathcal{M}$ be an arbitrary infinite set. It will be called a **sample space**. As it is an element of $\mathcal{M}$, it is $\mathcal{M}$-finite. Let $N = |\Omega|$ be the size of Ω in the sense of $\mathcal{M}$. It is necessarily non-standard.

Let $\mathcal{A} := \{A \in \mathcal{M} \mid A \subseteq \Omega\}$. This is a Boolean algebra but not a σ-algebra as the class of definable sets is not closed under all countable unions (for example, while it contains all singletons it does not contain the countable set of those elements of Ω having only standardly many predecessors in Ω). The **counting measure** (i.e. the uniform probability) on $\mathcal{A}$ is defined by:

$$A \in \mathcal{A} \to \ |A|/N \ .$$

Its values are the $\mathcal{M}$-rationals. A positive $\mathcal{M}$-rational is called infinitesimal if it is smaller that all fractions $\frac{1}{k}$, $k \in \mathbf{N}$.

Define an ideal $\mathcal{I} \subseteq \mathcal{A}$ by:

$$A \in \mathcal{I} \ \text{ iff } \ |A|/N \ \text{ is infinitesimal .}$$

$\mathcal{I}$ is not definable in $\mathcal{M}$ (otherwise the set of natural numbers $\mathbf{N}$ would be definable, violating the overspill in $\mathcal{M}$). Using $\mathcal{I}$ define Boolean algebra $\mathcal{B} := \mathcal{A}/\mathcal{I}$.

The induced measure on $\mathcal{B}$ (the so called Loeb's measure) will be denoted μ. Hence $\mu(b)$ for $b \in \mathcal{B}$ is the standard part of $|B|/N$ (i.e. the unique standard real infinitesimally close to it) for any $B \in \mathcal{A}$ such that $B/\mathcal{I} = b$.

It is a measure in the ordinary sense: The values of μ lie in the reals $\mathbf{R}$. It is σ-additive and a strict measure: $\mu(b) > 0$ if $b \neq 0_{\mathcal{B}}$.

The following key lemma is a combination of two well-known facts, one from non-standard analysis and one from measure theory.

Lemma 1.2.1 *$\mathcal{B}$ is a complete Boolean algebra.*

Proof :

First, as in the construction of Loeb's measure [79], we use the $\aleph_1$-saturation to show that

Claim 1: *$\mathcal{B}$ is a σ-algebra and the measure μ is σ-additive.*

To establish the claim let $\{b_k\}_{k\in\mathbf{N}}$ be a countable subset of $\mathcal{B}$. Assume $b_k = B_k/\mathcal{I}$ for B_k's from $\mathcal{A}$. We may assume, without a loss of generality, that $B_0 \subseteq B_1 \subseteq \ldots$. It is enough to find $C \in \mathcal{A}$ such that $B_k \subseteq C$ for all $k \in \mathbf{N}$ and $\mu(C) = \lim_{k\to\infty} \mu(B_k)$. It holds that for any $k \geq 1$ there is $n_k \geq 1$ such that for all $m > \ell \geq n_k$

$$\frac{|B_\ell|}{N} \leq \frac{|B_m|}{N} \leq \frac{|B_\ell|}{N} + \frac{1}{k} \ .$$

We may assume, by taking subsequence $\{B_{n_k}\}_{k\in\mathbf{N}}$, that $n_k = k$.

Take a non-standard extension $\{B_i\}_{i<t}$ of $\{B_k\}_{k\in\mathbf{N}}$ guaranteed to exists by the $\aleph_1$-saturation (each B_k is an element of $\mathcal{M}$). Consider the following property P parameterized by s:

$$B_s \in \mathcal{A} \ \wedge \ \forall i \leq s; B_i \subseteq B_s \wedge \frac{|B_i|}{N} \leq \frac{|B_s|}{N} \leq \frac{|B_i|}{N} + \frac{1}{i} \ .$$

Property P is obeyed by all standard s and hence, by induction in $\mathcal{M}$, also by some non-standard $s_0 < t$.

It is easy to verify that $C := B_{s_0}$ has the required properties.

Claim 2: *$\mathcal{B}$ satisfies the ccc condition: Any antichain is at most countable.*

This holds because the measure is strictly positive, i.e. $\mu(b) > 0$ for $b \neq O_{\mathcal{B}}$: Any antichain can contain only finitely many (non-zero) elements with measure in each interval $(1/(n+1), 1/n]$, $n \geq 1$.

As a consequence we get

Claim 3: *Any family of elements of $\mathcal{B}$ has the same set of upper bounds as one of its countable subfamilies.*

To see this note that a family has the same set of upper bounds as the ideal it generates which in turn has the same set of upper bounds as any maximal antichain it contains - such antichains are countable by ccc (Claim 2), and each element of the antichain is majorized by a union of a finite number of elements of the family).

We can conclude the proof that $\mathcal{B}$ is complete: The union of any family can be defined as the union of a countable subfamily (Claim 3), and any countable family has a union by the σ-additivity (Claim 1).

q.e.d.

We conclude this section by pointing out two facts that are generally *not true* in $\mathcal{B}$. The first observation is that the quotient operation $\ldots/\mathcal{I}$ that defines $\mathcal{B}$ from $\mathcal{A}$ does not commute with infinite unions: It is not true that it would generally hold that $(\bigcup_i A_i)/\mathcal{I} = \bigvee_i (A_i/\mathcal{I})$. In fact, the left-hand side may not be defined (i.e. may not be in $\mathcal{A}$), and even when it is the equality may not hold (e.g. if A_i' run over all singleton subsets of Ω then the left-hand side is $\Omega/\mathcal{I} = 1_{\mathcal{B}}$ while the right-hand is $\bigvee_i 0_{\mathcal{B}} = 0_{\mathcal{B}}$).

The second failure is well-know: Infinite distributive law

$$\bigwedge_{i\in I}\bigvee_{j\in J} b_{ij} = \bigvee_{f}\bigwedge_{i\in I} b_{i,f(i)} \ ,$$

where f range over all functions from I to J, may not hold (the left-hand side always majorizes the right-hand side though). However, it holds when I is finite.

Furthermore, for measure algebras (and hence for $\mathcal{B}$, see Chapter 2) *weak* distributive law

$$\bigwedge_{i\in I}\bigvee_{j\in J} b_{ij} = \bigvee_{g}\bigwedge_{i\in I}\bigvee_{j\in g(i)} b_{i,j} \ ,$$

holds, where g is a map assigning to elements of I finite subsets of J.

See [104, 48] for details.

1.3 The models $K(F)$

Let Ω, $\mathcal{A}$, $\mathcal{I}$ and $\mathcal{B}$ be as in the previous section.

Let $L \subseteq L_{all}$ and let F be a non-empty set of functions such that:

1. $F \subseteq \mathcal{M}$.

2. The domain of any function $\alpha \in F$ is Ω (hence $\alpha : \Omega \to \mathcal{M}$).

3. F is closed under L-functions and contains all L-constants.

The L-functions are interpreted by composition:

$$f(\alpha_1, \ldots, \alpha_k)(\omega) := f(\alpha_1(\omega), \ldots, \alpha_k(\omega)) \ ,$$

for an L-function f of arity $k \geq 1$.

Any such F will be called an **L-closed family** of random variables. Note, in particular, that F itself need not to be $\mathcal{M}$-definable; in fact, it almost all cases it will not be in our applications.

Definition 1.3.1 *Assume that F is an L-closed family of random variables. $K(F)$ will denote a Boolean-valued L-structure defined as follows.*

The universe of $K(F)$ is F. The Boolean valuation of L-sentences with parameters from F has values in $\mathcal{B}$ and is given by the following inductive conditions:

- $[\![\alpha = \beta]\!] := \{\omega \in \Omega \mid \alpha(\omega) = \beta(\omega)\}/\mathcal{I}$.
- $[\![R(\alpha_1, \ldots, \alpha_k)]\!] := \{\omega \in \Omega \mid R(\alpha_1(\omega), \ldots, \alpha_k(\omega))\}/\mathcal{I}$, *any L-relation R.*
- $[\![\ldots]\!]$ *commutes with* $\neg, \vee, \wedge$.
- $[\![\exists x A(x)]\!] := \bigvee_{\alpha \in F} [\![A(\alpha)]\!]$.
- $[\![\forall x A(x)]\!] := \bigwedge_{\alpha \in F} [\![A(\alpha)]\!]$.

Note that the values of $[\![\exists x A(x)]\!]$ and $[\![\forall x A(x)]\!]$ are well-defined as $\mathcal{B}$ is complete by Lemma 1.2.1. The notation $K(F)$ is sound as Ω is determined by F and $\mathcal{A}$, $\mathcal{I}$ and $\mathcal{B}$ are determined by Ω.

1.4 Valid sentences

We say that a sentence A is **valid** in $K(F)$ iff its Boolean value is $1_{\mathcal{B}}$. It is straightforward to verify that all axioms of first order predicate calculus in any particular formalization, including axioms of equality, are valid in any $K(F)$, and that the value $1_{\mathcal{B}}$ is preserved by inference rules. Further, an implication $B \to A$ is valid iff the truth value of the succedent majorizes the truth value of the antecedent:

$$[\![B]\!] \leq [\![A]\!] .$$

Hence we have the following.

Lemma 1.4.1 *Let T be a set of L-sentences and A an L-sentence (parameters from $K(F)$ are allowed). If A is provable from T in predicate calculus with equality then*

$$\bigwedge_{B \in T_0} [\![B]\!] \leq [\![A]\!]$$

for some finite $T_0 \subseteq T$.

In particular, if A is logically valid then $[\![A]\!] = 1_{\mathcal{B}}$ (as $\bigwedge \emptyset = 1_{\mathcal{B}}$), i.e. A is valid in $K(F)$.

Although we are primarily interested in models of bounded arithmetic theories and functions in the families F will typically have a small computational complexity, note that $K(F)$ can be a model of a very strong theory. For example, taking for F all function (in $\mathcal{M}$) with domain Ω yields a model where $Th(\mathbf{N})$ is valid.

For Γ a prefix class of formulas and $L \subseteq L_{all}$ let $Th_\Gamma(L)$ denote the set of true Γ-sentences valid in the canonical L-structure on $\mathbf{N}$. The following lemma is obvious.

Lemma 1.4.2 *Let F be an L-closed family. Then $Th_\forall(L)$ is $K(F)$-valid.*

If F contains (constant functions) $\mathbf{N}$ then $Th_{\exists\forall}(L)$ is $K(F)$-valid.

This seems to block a priori constructions of non-Π^b_1-elementary extensions which are needed for lengths-of-proofs lower bounds. However, there is a simple way out: A statement that can be expressed by a Π^b_1-formula or even an atomic formula in one language does not need to be so expressible

in another language. For example, if we have a function symbol for an algorithm checking that a truth assignment satisfies a set of clauses then such a statement is expressible by an atomic formula. But without such a symbol we may need a Π^b_2-formula to express it (every clauses contains a satisfied literal).

Lemma 1.4.2 will be also used in Section 4 in a crucial way to foster a link between the truth in $\mathbf{N}$ and the validity in $K(F)$.

1.5 Possible generalizations

We could have started the construction of $\mathcal{B}$ from any probability space $(\Omega, \mathcal{A}, \nu)$ that is definable in $\mathcal{M}$. In particular, both sets Ω and $\mathcal{A}$ have to be definable in $\mathcal{M}$ (the definability of the whole $\mathcal{A}$ - the definability of each element of $\mathcal{A}$ is not enough - is needed for Loeb's construction) and ν must be a definable finitely additive measure on $\mathcal{A}$ (with values in $\mathcal{M}$-rationals). One also needs to add a condition on α's from F: Each $\alpha^{(-1)}(a)$ for $a \in \mathcal{M}$ has to be measurable, i.e. in $\mathcal{A}$.

An example of this more general space is a probability space on an $\mathcal{M}$-finite Ω with a non-uniform distribution. This will be used, for example, in Chapter 25.

An example of a space with an $\mathcal{M}$-infinite Ω is $\Omega := \{w \in \mathcal{M} \mid |w| \geq n\}$, $\mathcal{A}$ consisting of all $\mathcal{M}$-finite Boolean combinations of r.e. subsets of Ω (defined using a code for the Boolean combination and a universal Σ^0_1-formula) and measure ν giving to a string w the weight $2^{n-1-2|w|}$.

However, we have no application for these more general constructions here and we restrict to the counting measure on $\mathcal{M}$-finite Ω.

Two random variables α, β on $\{0,1\}^n$ are "almost equal" (i.e. the formula $\alpha = \beta$ is $K(F)$-valid) if they differ for an infinitesimal fraction of samples $\omega \in \{0,1\}^n$. In complexity theory however, a stronger condition would be used: The random variables should differ in a negligible fraction, i.e. less than n^{-k}, any $k \in \mathbf{N}$. Whenever necessary (not in this book though) this can be remedied analogously to complexity theory: Build the model from the same random variables but computed on a tuple of independent samples.

More precisely, let F be an L-closed family on Ω, and let $t \geq 1$, $t \in \mathcal{M}$, be arbitrary. For $\alpha \in F$ define $\alpha^{(t)} : \Omega^t \to M^t$ by:

$$\alpha^{(t)}(\omega_1, \ldots, \omega_t) := (\alpha(\omega_1), \ldots, \alpha(\omega_t)) \ .$$

Denote by $F^{(t)}$ the collection of all $\alpha^{(t)}$; it is also an L-closed family.

Interpret an L-relation R on elements $a^i = (a^i_1, \ldots, a^i_t) \in \mathcal{M}^t$, $i \leq k$, as

$$R(a^1, \ldots, a^k) := \bigwedge_{j \leq t} R(a^1_j, \ldots, a^k_j) \ ,$$

i.e. we assign the Boolean-value by the equation:

$$[\![R((\alpha^1)^{(t)}, \ldots, (\alpha^k)^{(t)})]\!] := \{(\omega_1, \ldots, \omega_t) \in \Omega^t \mid \bigwedge_{j \leq t} R(\alpha^1(\omega_j), \ldots, \alpha^k(\omega_j))\}/\mathcal{I} \ .$$

Let $\mathcal{A}^t$ be the t-product of $\mathcal{A}$, a probability space on Ω^t, and let $\mathcal{B}^{(t)}$ be the complete Boolean algebra resulting from the construction in Section 1.2. (It is not the t-folded product of $\mathcal{B}$; in fact, such $\mathcal{B}^t$ is not even defined when t is non-standard.) The $F^{(t)}$-formulas are evaluated in $\mathcal{B}^{(t)}$.

Then for $\alpha^{(t)} = \beta^{(t)}$ to be valid in $K(F^{(t)})$ it is not sufficient that $p := Prob_{\omega \in \Omega}[\alpha(\omega) \neq \beta(\omega)]$ is infinitesimal: It must be that $p \cdot t$ is infinitesimal.

Chapter 2

Measure on $\mathcal{B}$

Boolean algebra $\mathcal{B}$ carries a strict probabilistic measure μ. We shall interpret it in this chapter as a metric and also link it with probability.

2.1 A metric on $\mathcal{B}$

The map

$$A \longrightarrow \mu([\![A]\!])$$

assigns real numbers to L-sentences, and satisfies

- $0 \leq \mu([\![A]\!]) \leq 1$,
- $\mu([\![\neg A]\!]) = 1 - \mu([\![A]\!])$
- $\mu([\![A \vee B]\!]) + \mu([\![A \wedge B]\!]) = \mu([\![A]\!]) + \mu([\![B]\!])$

and with suitable F (see Chapter 3) also

- $\mu([\![\exists x A(x)]\!]) = \sup_{\alpha \in F} \mu([\![A(\alpha)]\!])$ and
- $\mu([\![\forall x A(x)]\!]) = \inf_{\alpha \in F} \mu([\![A(\alpha)]\!]))$.

Further it satisfies the all-important inequality

- $\mu([\![A]\!]) \leq \mu([\![B]\!])$ if A logically implies B.

However, the number $\mu([\![A]\!])$ cannot be interpreted as the probability of A. The passage from the Boolean-value to probability is more subtle (Section 2.2). The right interpretation of this number seems to be in terms of a metric on $\mathcal{B}$. Define for $a, b \in \mathcal{B}$:

$$d(a,b) := \mu(a \triangle b)$$

where $a \triangle b$ is the symmetric difference $(a \wedge \neg b) \vee (\neg a \wedge b)$, denoted in Boolean complexity also $a \oplus b$ (the sum).

The following lemma is verified by looking at the Venn diagram.

Lemma 2.1.1 *Function d is a metric on $\mathcal{B}$. Further, the following equality and two inequalities hold for any $a, a', b, b' \in \mathcal{B}$:*

1. $d(a,b) = d(\neg a, \neg b)$.
2. $d(a \wedge b, a' \wedge b') \leq d(a,a') + d(b,b')$.
3. $d(a \vee b, a' \vee b') \leq d(a,a') + d(b,b')$.

2.2 From Boolean value to probability

In this section we investigate a relation between the probability of a sentence in $\mathcal{M}$ and the μ-measure of its Boolean value in $K(F)$.

The following lemma follows from Definition 1.3.1 and from the fact that the quotient operation $A \in \mathcal{A} \to A/\mathcal{I} \in \mathcal{B}$ commutes with $\neg, \vee, \wedge$ (binary).

Lemma 2.2.1 *Let F be an L-closed family, $A(x)$ be an open L-formula with the only free variable x, and let $\alpha \in F$.*

Then for all standard $\epsilon > 0$:

$$\Pr_{\omega \in \Omega}[A(\alpha(\omega))] \;\geq\; \mu([\![A(\alpha)]\!]) - \epsilon \;.$$

(The probability expression is defined in $\mathcal{M}$.)

In order to be able to apply later Lemma 3.3.3 we need some extension of Lemma 2.2.1 to universal formulas. However, in general it is not true that $\Pr_{\omega \in \Omega}[\forall z A(\alpha(\omega), z)]$ can be bounded from below by a term in $\mu([\![\forall z A(\alpha, z)]\!])$. For example, it can be that $\forall x \exists! z \neg A(x,z)$ but that no $\gamma \in F$ finds such z

for more than a negligible fraction of x's. Then $[\![\forall z A(\mathrm{id}_\Omega, z)]\!] = 1_{\mathcal{B}}$ while $\Pr_{\omega\in\Omega}[\forall z A(\omega, z)] = 0$.

The way out is to choose suitable families F. Call any function $\xi \in L_{all}$ satisfying:

$$\mathbf{N} \models \forall x\ [\exists z \neg A(x,z) \rightarrow \neg A(x, \xi(x))]$$

a counter-example function for $\forall z A(x,z)$.

Lemma 2.2.2 *Assume that $A(x,z)$ is an open L-formula, ξ is a counter-example function for $\forall z A(x,z)$, and F is an L-closed family closed under ξ.*

Then for any $\alpha \in F$ and any standard $\epsilon > 0$:

$$\Pr_{\omega\in\Omega}[\forall z A(\alpha(\omega), z)] \ \geq\ \mu([\![\forall z A(\alpha, z)]\!]) - \epsilon\ .$$

Proof :

Let $[\![\forall z A(\alpha, z)]\!] = b$. For all $\gamma \in F$ we have $[\![A(\alpha,\gamma)]\!] \geq b$, so

$$\Pr_{\omega\in\Omega}[A(\alpha(\omega), \gamma(\omega))] \ \geq\ \mu(b) - \epsilon\ ,$$

any $\epsilon > 0$ by Lemma 2.2.1.

For a particular $\gamma(\omega) := \xi(\alpha(\omega))$ we also have

$$(\forall z A(\alpha(\omega), z)) \equiv A(\alpha(\omega), \gamma(\omega))\ ,$$

so:

$$\Pr_{\omega\in\Omega}[\forall z A(\alpha(\omega), z)] \ =\ \Pr_{\omega\in\Omega}[A(\alpha(\omega), \gamma(\omega))]$$

and the lemma follows.

q.e.d.

Chapter 3

Witnessing quantifiers

Consider an L-sentence A of the form

$$\exists x_1 \forall y_1 \ldots \exists x_k \forall y_k B(x_1, y_1, \ldots, x_k, y_k)$$

with B open. Let K be an L-structure. Two players, $\exists$-player and $\forall$-player, pick sequentially witnesses $\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k \in F$ to the quantifiers. The $\exists$-player wants to make the truth value $[\![B(\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k)]\!]$ as large as possible by selecting $\alpha_1, \ldots, \alpha_k$ suitably, while the $\forall$-player's objective is the opposite, to make it as small as possible by picking $\beta_1, \ldots, \beta_k$ well.

If K were a classical structure (i.e. $\mathcal{B}$ were the 2-element algebra) the situation would be simple: If A is true then the $\exists$-player has a strategy to achieve value $1_{\mathcal{B}}$, and if A is false the $\forall$-player has a strategy to reach value $0_{\mathcal{B}}$. However, if K is a Boolean-valued structure there may not be optimal moves $\alpha_1, \beta_1, \ldots$ that would ensure that

$$[\![B(\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k)]\!] = [\![\exists x_1 \forall y_1 \ldots \exists x_k \forall y_k B(x_1, y_1, \ldots, x_k, y_k)]\!]$$

and, in general, the players need to collaborate to get as close (in terms of the metric on $\mathcal{B}$) to

$$[\![\exists x_1 \forall y_1 \ldots \exists x_k \forall y_k B(x_1, y_1, \ldots, x_k, y_k)]\!]$$

as possible.

In this Chapter we will study under which conditions posed on F we can guarantee the existence of suitable moves (called often "witnesses") $\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k$. Let us mention two reasons why we are interested in it.

Consider a sentence A of the form $\forall x \exists y B(x,y)$, B open. Assume that $\mathrm{id}_\Omega \in F$, where $\Omega = \{0,1\}^n$. If $[\![A]\!] = b$ then $[\![\exists y B(\mathrm{id}_\Omega, y)]\!] \geq b$. Assume that we can witness y by some $\alpha \in F$ such that:

$$[\![\exists y B(\mathrm{id}_\Omega, y)]\!] = [\![B(\mathrm{id}_\Omega, \alpha)]\!]$$

By Lemma 2.2.1 then:

$$Prob_{\omega\in\Omega}[B(\omega, \alpha(\omega)] \geq \mu(b) - \epsilon \ ,$$

for all standard $\epsilon > 0$. Hence we have reduced the question of approximating $[\![A]\!]$ (in $\mathcal{B}$ as a metric space) to a question about the witnessing power of F-functions in the ambient model $\mathcal{M}$, and consequently also in $\mathbf{N}$.

For the second example assume that we can find witnesses $\alpha_1, \ldots, \beta_k$ to quantifiers in $A(\gamma) := \exists x_1 \forall y_1 \ldots \exists x_k \forall y_k B(x_1, y_1, \ldots, x_k, y_k, \gamma)$ such that

$$[\![B(\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k, \gamma)]\!] = [\![\exists x_1 \forall y_1 \ldots \exists x_k \forall y_k B(x_1, y_1, \ldots, x_k, y_k, \gamma)]\!]$$

(B is again open and has a parameter $\gamma \in F$). The left-hand side value can be computed explicitly:

$$\{\omega \mid B(\alpha_1(\omega), \beta_1(\omega), \ldots, \alpha_k(\omega), \beta_k(\omega), \gamma(\omega))\}/\mathcal{I} \ .$$

So in this case we are able to represent the truth value of an arbitrarily complex formula by a set defined by an open formula. Such computation of the truth values of a complex formula by the truth value of a simpler one is, in principle, possible because the value $[\![A(\gamma)]\!]$ has, in general, nothing to do with

$$\{\omega \mid \exists x_1 \forall y_1 \ldots \exists x_k \forall y_k B(x_1, y_1, \ldots, x_k, y_k, \gamma(\omega))\}/\mathcal{I} \ .$$

When we shall in future quantify in statements over $\epsilon > 0$ or $\delta > 0$ it is tacitly over *standard* ϵ or δ. Infinitesimal positive quantity will be always written as $1/t$ with t non-standard.

3.1 Propositional approximation of truth values

Let A be a sentence (possibly with parameters) of the form $\exists x B(x)$; B needs not to be open here. The next statement was used already in the proof of

Lemma 1.2.1 and we state it here for completeness. It is a consequence of the ccc property of $\mathcal{B}$ (the second part is a consequence of the σ-additivity of μ).

Lemma 3.1.1 *There are countably many $\alpha_k \in F$, $k \in \mathbf{N}$, such that*

$$[\![\exists x B(x)]\!] = \bigvee_{k\in\mathbf{N}} [\![B(\alpha_k)]\!] \ .$$

In particular, for any $\epsilon > 0$ there are finitely many $\alpha_1, \ldots, \alpha_\ell \in F$ such that

$$d([\![\exists x B(x)]\!], \bigvee_{k\leq\ell} [\![B(\alpha_k)]\!]) < \epsilon \ .$$

Analogous statement holds also for approximating the truth value of a formula starting with $\forall$ by $\bigwedge$ of its instances.

Iterating Lemma 3.1.1 yields the following statement.

Lemma 3.1.2 *Let A be an L-sentence of the form*

$$\exists x_1 \forall y_1 \ldots \exists x_k \forall y_k B(x_1, y_1, \ldots, x_k, y_k)$$

with B open (parameters are allowed). Let $\epsilon > 0$ be arbitrary.

Then there are $\ell \in \mathbf{N}$ and elements of F:

$$\alpha_1^{i_1}, \beta_1^{i_1,j_1}, \alpha_2^{i_1,j_1,i_2}, \ldots, \beta_k^{i_1,\ldots,j_k}$$

with $i_1, \ldots, j_k < \ell$, such that the distance between $[\![A]\!]$ and

$$\bigvee_{i_1}\bigwedge_{j_1}\ldots\bigvee_{i_k}\bigwedge_{j_k} [\![B(\alpha_1^{i_1}, \beta_1^{i_1,j_1}, \ldots, \beta_k^{i_1,\ldots,j_k})]\!]$$

is less than ϵ.

In particular, the distance between $[\![A]\!]$ and

$$\{\omega \in \Omega \mid \bigvee_{i_1}\bigwedge_{j_1}\ldots\bigvee_{i_k}\bigwedge_{j_k} B(\alpha_1^{i_1}(\omega), \beta_1^{i_1,j_1}(\omega), \ldots, \beta_k^{i_1,\ldots,j_k}(\omega))\}/\mathcal{I}$$

is less than ϵ.

Proof :

Fix small enough $\delta > 0$ (how small we be clear at the end of the proof), a standard rational. Denote:

$$C(x_1) := \forall y_1 \exists x_2 \forall y_2 \ldots B(x_1, y_1, \ldots) \ .$$

By Lemma 3.1.1 there are $\alpha_1^0, \ldots, \alpha_1^{u-1} \in F$, $u \in \mathbf{N}$, such that

$$d([\![A]\!], \bigvee_{i<u} [\![C(\alpha_1^i)]\!]) \ \leq \ \delta \ .$$

Next denote for $i < u$:

$$D_i(y_1) := \exists x_2 \forall y_2 \ldots B(\alpha_1^i, y_1, x_2, y_2, \ldots) \ .$$

By Lemma 3.1.1 again there are $\beta_1^{i,j} \in F$, $j < v$ some standard v, such that:

$$d([\![C(\alpha_1^i)]\!], \bigwedge_{j<v} [\![D_i(\beta_1^{i,j})]\!]) \ \leq \ \delta/u$$

for all $i < u$. By Lemma 2.1.1 then:

$$d([\![A]\!], \bigvee_{i<u} \bigwedge_{j<v} [\![\exists x_2 \forall y_2 \ldots B(\alpha_1^i, \beta_1^{i,j}, x_2, y_2, \ldots)]\!]) \ \leq \ 2\delta \ .$$

Proceeding analogously for $\exists x_2, \forall y_2, \ldots$ we find in F elements

$$\alpha_1^{i_1}, \beta_1^{i_1,j_1}, \alpha_2^{i_1,j_1,i_2}, \ldots, \beta_k^{i_1,\ldots,j_k}$$

with all $i_1, \ldots, j_k$ ranging over some standard finite domain such that the distance of $[\![A]\!]$ from

$$\bigvee_{i_1} \bigwedge_{j_1} \ldots [\![B(\alpha_1^{i_1}, \beta_1^{i_1,j_1}, \ldots, \beta_k^{i_1,\ldots,j_k})]\!]$$

is at most $2k\delta$. However, this truth value equals, modulo the ideal $\mathcal{I}$, to the set of all $\omega \in \Omega$ satisfying (as evaluated in $\mathcal{M}$) the Boolean formula formed from formulas $B(\alpha_1^{i_1}(\omega), \beta_1^{i_1,j_1}(\omega), \ldots, \beta_k^{i_1,\ldots,j_k}(\omega))$:

$$\bigvee_{i_1} \bigwedge_{j_1} \ldots B(\alpha_1^{i_1}(\omega), \beta_1^{i_1,j_1}(\omega), \ldots, \beta_k^{i_1,\ldots,j_k}(\omega)) \ .$$

By picking $\delta > 0$ less than $\epsilon/2k$ we get the lemma.

q.e.d.

3.2 Witnessing in definable families

Our next lemma gives a sample statement and a proof whose variants will be used repeatedly. The construction shows that the $\aleph_1$-saturation can be used in a place where in set theory we could have used a straightforward induction on $k \in \mathbf{N}$.[1]

Lemma 3.2.1 *Assume that F is definable in $\mathcal{M}$. Let $b_k := B_k/\mathcal{I}$ for some $B_k \in \mathcal{A}$ and $\alpha_k \in F$, for all $k \in \mathbf{N}$. Assume that $b_k \wedge b_\ell = 0_\mathcal{B}$ for all $k \neq \ell \in \mathbf{N}$ (i.e. b_ks form an antichain).*

Further assume that for all $k \in \mathbf{N}$ there is $\beta \in F$ such that for all $\ell = 1, \ldots, k$:

$$(1) \qquad \beta(\omega) = \alpha_\ell(\omega) \ \text{ for } \ \omega \in B_\ell \setminus \bigcup_{i<\ell} B_i \ .$$

Then there exists $\beta \in F$ such that for all $k \in \mathbf{N}$:

$$(2) \qquad [\![\beta = \alpha_k]\!] \ \geq b_k \ .$$

Proof :

Let $S = \{(B_i, \alpha_i)\}_{i<t}$ be a non-standard extension of $\{(B_k, \alpha_k)\}_{k\in\mathbf{N}}$ provided by the $\aleph_1$-saturation. Using S, and the fact that F is definable, we can define for all $k \in \mathbf{N}$ the set C_k of all $\beta \in F$ satisfying condition (1).

Clearly $C_0 \supseteq C_1 \supseteq \ldots$ and each $C_k \neq \emptyset$ (by the hypothesis of the lemma). Hence, again by the $\aleph_1$-saturation, $\bigcap_{k\in\mathbf{N}} C_k \neq \emptyset$, and any β from the intersection satisfies condition (2).

q.e.d.

We derive a simple corollary about witnessing of quantifiers for particular families F. First a definition.

Definition 3.2.2 *Family F is* **closed under definitions** *if for any $\alpha_0, \alpha_1 \in F$ and any $B \in \mathcal{A}$ there is $\beta \in F$ such that:*

$$\beta(\omega) = \begin{cases} \alpha_0(\omega) & \text{if } \omega \in B \\ \alpha_1(\omega) & \text{otherwise.} \end{cases}$$

[1] Paradoxically there is "no arithmetic in arithmetic", meaning that in arithmetic we do not have the set of natural numbers: PA is equivalent (in the sense of mutual interpretability) to ZFC with the Axiom of Infinity replaced by its negation. This seems to block a transplantation of Cohen's forcing to arithmetic.

Clearly the hypothesis (1) of Lemma 3.2.1 is satisfied for families closed under definitions by cases.

Lemma 3.2.3 *Assume that F is definable in $\mathcal{M}$ and that it is closed under definitions by cases.*

Then for each sentence $\exists x C(x)$ there is an $\alpha \in F$ such that

$$[\![\exists x C(x)]\!] = [\![C(\alpha)]\!]$$

(C needs not to be open).

In fact, for each sentence A of the form

$$\exists x_1 \forall y_1 \ldots \exists x_k \forall y_k B(x_1, y_1, \ldots, x_k, y_k)$$

(B arbitrary, possibly with parameters), there are $\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k \in F$ such that:

$$\begin{gathered}
[\![A]\!] \;= \\
[\![\forall y_1 \exists x_2 \forall y_2 \ldots \exists x_k \forall y_k B(\alpha_1, y_1, \ldots, x_k, y_k)]\!] \;= \\
[\![\exists x_2 \forall y_2 \ldots \exists x_k \forall y_k B(\alpha_1, \beta_1, \ldots, x_k, y_k)]\!] \;= \\
\ldots \\
[\![\forall y_k B(\alpha_1, \beta_1, \ldots, \alpha_k, y_k)]\!] \;= \\
[\![B(\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k)]\!] \; .
\end{gathered}$$

Proof :

The second part of the lemma is proved by induction on the number of quantifiers in A using the first part.

By Lemma 3.1.1 there are α_k, $k \in \mathbf{N}$, such that $[\![\exists x C(x)]\!] = \bigvee_{k \in \mathbf{N}} [\![C(\alpha_k)]\!]$. Define $b_0 := [\![C(\alpha_0)]\!]$ and for $k > 0$ put:

$$b_k \;:=\; [\![C(\alpha_k)]\!] \setminus \bigvee_{\ell < k} b_\ell \; .$$

By induction one can verify that for all k:

$$\bigvee_{i \leq k} [\![C(\alpha_i)]\!] = \bigvee_{i \leq k} b_i$$

and, in particular,

$$\bigvee_{\alpha \in F} [\![C(\alpha)]\!] = \bigvee_k b_k \; .$$

Clearly any two different b_k's are incompatible. By Lemma 3.2.1 there is $\beta \in F$ such that

$$[\![\beta = \alpha_{\dot{k}}]\!] \geq b_k \ , \quad \text{for all } k \in \mathbf{N} \ .$$

But then clearly

$$[\![C(\beta)]\!] \leq [\![\exists x C(x)]\!] = \bigvee_{\alpha \in F} [\![C(\alpha)]\!] = \bigvee_{k \in \mathbf{N}} b_k \leq [\![C(\beta)]\!] \ .$$

The last inequality uses $b_k \leq [\![C(\alpha_k)]\!] \wedge [\![\beta = \alpha_k]\!]$ and hence $b_k \leq [\![C(\beta)]\!]$, for all $k \in \mathbf{N}$.

q.e.d.

Note that having witnesses $\alpha_1, \ldots, \beta_k$ such that

$$[\![A]\!] = [\![B(\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k)]\!]$$

does not, in general, imply that the witnesses also obey the remaining equations stated in the lemma.

The problem with the witnessing statement 3.2.3 is that families F we encounter never meet the two requirements:

1. F is closed under definition by cases.

2. F is definable in $\mathcal{M}$.

An example of F to keep in mind is a family F_{PV} consisting of polynomial time functions (i.e. random variables on $\{0,1\}^n$ sampled by p-time algorithms). F_{PV} satisfies neither of the two conditions. The first condition fails for F_{PV} as a set $B \in \mathcal{A}$ from Definition 3.2.2 can be defined by a formula with quantifiers and we have no a priori polynomial upper bound on an algorithm deciding the membership in B. Family F_{PV} is also not definable in $\mathcal{M}$ as otherwise one could define $\mathbf{N}$ (as the set of k such that F_{PV} contains a function of growth faster than n^k).

We shall see in the next two sections that it is possible to relax the conditions so that Lemma 3.2.3 still holds but interesting families will satisfy the new conditions.

3.3 Definition by cases by open formulas

Definition 3.3.1 *We shall say that F is* **closed under definitions by cases by open L-formulas** *iff whenever $\alpha, \beta \in F$ and $B(x)$ is an open L-formula with free variable x then there is $\gamma \in F$ such that:*

$$\gamma(\omega) = \begin{cases} \alpha(\omega) & \text{if } B(\alpha(\omega)) \text{ holds} \\ \beta(\omega) & \text{otherwise.} \end{cases}$$

Note that the example family F_{PV} of polynomial time functions is closed under definitions by cases by open L-formulas, as long as all relations and functions in L are computable by polynomial time algorithms.

Lemma 3.3.2 *Let F be an L-closed family closed under definition by cases by open $L(F)$-formulas. Let $B(x)$ be an open L-formula (possibly with parameters from F).*

Then for every $\epsilon > 0$ there is $\alpha \in F$ such that

$$d([\![\exists x B(x)]\!], [\![B(\alpha)]\!]) < \epsilon \ .$$

In particular, if $\exists x B(x)$ is $K(F)$-valid then for every $\epsilon > 0$ there is $\alpha \in F$ such that

$$\mu([\![B(\alpha)]\!]) > 1 - \epsilon \ .$$

Proof :

For any two $\beta_1, \beta_2 \in F$ there is (by the hypothesis that F is closed under definitions by cases by open L-formulas) $\beta_3 \in F$ such that $\beta_3(\omega) = \beta_1(\omega)$ if $B(\beta_1(\omega))$ holds, and $\beta_3(\omega) = \beta_2(\omega)$ otherwise. Then it follows rather easily from the definition of β_3 that:

$$[\![B(\beta_3)]\!] = [\![B(\beta_1)]\!] \vee [\![B(\beta_2)]\!] \ .$$

Combining this with Lemma 3.1.1 we get a countable sequence $\beta_0, \beta_1, \ldots \in F$ such that

$$[\![B(\beta_0)]\!] \leq [\![B(\beta_1)]\!] \leq \ldots$$

and such that $\lim_{k\to\infty} \mu([\![B(\beta_k)]\!]) = \mu([\![\exists x B(x)]\!])$. The lemma follows.

q.e.d.

Now we extend Lemma 3.3.2 to more complex formulas.

Lemma 3.3.3 *Let F be an L-closed family closed under definition by cases by open $L(F)$-formulas (L-formulas with parameters from F).*

Let $\exists x C(x)$ be an L-sentence, C an arbitrary formula with parameters from F. Then for any $\epsilon > 0$ there is an $\alpha \in F$ such that

$$d([\![\exists x C(x)]\!], [\![C(\alpha)]\!]) < \epsilon \ .$$

In particular, if $\exists x C(x)$ is $K(F)$-valid then

$$\mu([\![C(\alpha)]\!]) > 1 - \epsilon \ .$$

In fact, for each sentence A of the form

$$\exists x_1 \forall y_1 \ldots \exists x_k \forall y_k B(x_1, y_1, \ldots, x_k, y_k)$$

(B open, possibly with parameters), there are $\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k \in F$ such that:

$$d([\![A]\!], [\![\forall y_1 \exists x_2 \forall y_2 \ldots \exists x_k \forall y_k B(\alpha_1, y_1, \ldots, x_k, y_k)]\!]) < \epsilon$$

$$d([\![\forall y_1 \exists x_2 \ldots \exists x_k \forall y_k B(\alpha_1, y_1, x_2, \ldots, x_k, y_k)]\!], [\![\exists x_2 \ldots \forall y_k B(\alpha_1, \beta_1, x_2, \ldots, y_k)]\!]) < \epsilon$$

$$\ldots$$

$$d([\![\forall y_k B(\alpha_1, \beta_1, \ldots, \alpha_k, y_k)]\!], [\![B(\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k)]\!]) < \epsilon \ .$$

Proof :

The lemma appears similar to Lemma 3.2.3 where the second part followed from the first part. However, here it is the other way around; we need to prove the second part while the first one is a special case of it.

We shall describe the construction for $k = 1$, i.e. for A of the form

$$\exists x \forall y B(x, y) \ ,$$

B open. This will display the key idea; the general case $k \geq 1$ is proved analogously by induction on the quantifier complexity of the formula.

The following claim follows from Lemma 3.3.2: Apply it to $\exists y \neg B(\alpha, y)$ (here we use the hypothesis that F is closed under definitions by cases by open $L(F)$-formulas).

Claim 1: *For any $\alpha \in F$ and $\delta > 0$ there is $\beta \in F$ such that*

$$d([\![\forall y B(\alpha, y)]\!], [\![B(\alpha, \beta)]\!]) < \delta \ .$$

The next claim is a bit subtler.

Claim 2: *For any $\alpha_1, \alpha_2 \in F$ and $\xi > 0$ there is $\alpha_3 \in F$ such that*

$$d([\![\forall y B(\alpha_3, y)]\!], [\![\forall y B(\alpha_1, y)]\!] \vee [\![\forall y B(\alpha_2, y)]\!]) < \xi \ .$$

To prove the claim we first apply Claim 1 to α_1, α_2 and $\delta > 0$ (to be specified later) to get $\beta_1, \beta_2 \in F$ such that

$$(1) \qquad d([\![B(\alpha_i, \beta_i)]\!], [\![\forall y B(\alpha_i, y)]\!] < \delta \ \text{ , for } i = 1, 2.$$

Define pair α_3, β_3:

$$\alpha_3(\omega), \beta_3(\omega) = \begin{cases} \alpha_1(\omega), \beta_1(\omega) & \text{if } B(\alpha_1(\omega), \beta_1(\omega)) \text{ holds} \\ \alpha_2(\omega), \beta_2(\omega) & \text{otherwise.} \end{cases}$$

Clearly $[\![B(\alpha_3, \beta_3)]\!] = [\![B(\alpha_1, \beta_1)]\!] \vee [\![B(\alpha_2, \beta_2)]\!]$. Observe that:

$$(2) \qquad d([\![\forall y B(\alpha_3, y)]\!], [\![B(\alpha_3, \beta_3)]\!]) < 3\delta \ .$$

This is because otherwise we could find (via Claim 1) $\gamma \in F$ such that

$$d([\![\forall y B(\alpha_3, y)]\!], [\![B(\alpha_3, \gamma)]\!]) < \delta$$

and use it to modify β_i, $i = 1$ or $i = 2$, to decrease the measure of $[\![B(\alpha_i, \beta_i)]\!]$ by more than δ, which is impossible by the choice of β_i's.

Putting (1) and (2) together, and using Lemma 2.1.1 we get:

$$(3) \qquad d([\![\forall y B(\alpha_3, y)]\!], [\![\forall y B(\alpha_1, y)]\!] \vee [\![\forall y B(\alpha_2, y)]\!]) < 5\delta \ ,$$

and taking $\delta := \xi/5$ yields the claim.

To conclude the proof of the lemma let $\alpha_1, \alpha_2, \ldots$ be a countable family such that

$$\bigvee_k [\![\forall y B(\alpha_k, y)]\!] = [\![\exists x \forall y B A(x, y)]\!] \ .$$

Put $\hat{\alpha}_1 := \alpha_1$ and apply Claim 2 to pairs $\hat{\alpha}_k, \alpha_{k+1}$ and parameters $\xi_k := \epsilon/2^{k+1}$ to get $\hat{\alpha}_{k+1}$ for $k = 1, 2, \ldots$. Then

$$d([\![\forall y B(\hat{\alpha}_k, y)]\!], \bigvee_{j \leq k} [\![\forall y B(\alpha_j, y)]\!]) < \epsilon/2(\frac{2^k - 1}{2^k}) \ .$$

Hence for k so large that

$$d(\bigvee_{j \leq k} [\![\forall y B(\alpha_j, y)]\!], [\![\exists x \forall y B(x, y)]\!]) < \epsilon/2$$

$\hat{\alpha}_k$ is the wanted element.

q.e.d.

3.4 Compact families

The following concept will substitute in Section 3.5 the definability of F used in Lemma 3.2.3.

Definition 3.4.1 *Let $F \subseteq \mathcal{M}$ be a family. F is* **compact** *iff there exists an L_{all}-formula $H(x,y)$ such that for*

$$F_a := \{b \in \mathcal{M} \mid \mathcal{M} \models H(a,b)\}$$

the following two properties hold:

1. $\bigcap_{k\in\mathbf{N}} F_k = F$.
2. $F_k \supseteq F_{k+1}$, *for all* $k \in \mathbf{N}$.

The second property is only technical: If $H(x,y)$ satisfies the first one then $H'(x,y) := \forall z \leq x H(z,y)$ will satisfy both (and, in fact, the second one will be satisfied for all $a \in \mathcal{M}$).

Our example family F_{PV} of polynomial time computable functions is not compact. However, in many applications we can replace it by a bigger family F that is compact: The family of functions computable by circuits of subexponential size (here we take $\Omega = \{0,1\}^n$ and the qualification "subexponential" means less than 2^{n^ξ}, some infinitesimal ξ, i.e. size $2^{n^{o(1)}}$). Definable sets F_k witnessing that F is compact consists of functions computable by circuits of size less than $2^{n^{1/k}}$. Note that we switched in this example to non-uniform algorithms in order to gain the definability of F_k's. Of course, we could have produced a tighter compact envelope F' of F_{PV} by taking for F'_k, for example, the set of functions computable by circuits of size less than $n^{\log^{(k)}(n)}$ ($\log^{(k)}$ is k-times iterated logarithm). However, the subexponential family yields stronger independence results and lower bounds.

The following lemma justifies the name **compact**: If we think of nonstandard extensions of sequences as their "limits" then it says that any sequence of elements of a compact family F has a limit in F.

Lemma 3.4.2 *Let F be a compact family. Assume C_k, $k \in \mathbf{N}$, are definable sets such that*

$$F \cap \bigcap_{\ell<k} C_\ell \neq \emptyset \ ,$$

for all $k \geq 1$. Then:

$$F \cap \bigcap_{k \in \mathbf{N}} C_k \neq \emptyset \ .$$

Further, if $\{\alpha_k\}_{k \in \mathbf{N}}$ is a countable sequence of elements of F and $(\alpha_i)_{i<t}$ its arbitrary non-standard extension then there is non-standard $s < t$ such that

$$\alpha_i \in F, \quad \textit{for all } i \leq s \ .$$

Proof :

Let $H(x, y)$ be the formula certifying that F is compact. Put:

$$F_a := \{b \in \mathcal{M} \mid H(a, b)\} \ .$$

Define sets

$$D_k := C_k \cap F_k \ .$$

These are definable (as F_k's are) and satisfy

$$\bigcap_{\ell<k} D_\ell \neq \emptyset$$

as

$$\bigcap_{\ell<k} D_\ell \supseteq \bigcap_{\ell<k} C_\ell \cap F \neq \emptyset \ .$$

Here we use $F_k \supseteq F$.

By the $\aleph_1$-saturation we have:

$$\bigcap_{k \in \mathbf{N}} D_k \neq \emptyset \ .$$

But any $\alpha \in \bigcap_{k \in \mathbf{N}} D_k$ is, in particular, an element of $\bigcap_{k \in \mathbf{N}} F_k = F$.

For the second part and any non-standard extension $(\alpha_i)_{i<t}$ of $\{\alpha_k\}_{k \in \mathbf{N}}$ consider the definable property $P(i)$:

$$\alpha_i \in F_i \wedge F_0 \supseteq F_1 \supseteq \ldots \supseteq F_i \ .$$

$P(k)$ holds for all $k \in \mathbf{N}$ and so by induction in $\mathcal{M}$ it hold sup to some non-standard $s < t$. By $P(i)$ it holds that $\alpha_i \in F_i \subseteq \bigcap_{k \in \mathbf{N}} F_k = F$, for all $i \leq s$.

q.e.d.

3.5 Propositional computation of truth values

We shall strengthen Lemmas 3.1.2 and 3.3.3 in this section for the case of compact families. The strengthening lies in the elimination of the error term ϵ.

Theorem 3.5.1 *Let F be an L-closed compact family. Let $s \in \mathcal{M}$ be an arbitrary non-standard number.*

Let A be an L-sentence, possibly with parameters from F, and assume that A has the form:

$$\exists x_1 \forall y_1 \ldots \exists x_k \forall y_k B(x_1, y_1, \ldots, x_k, y_k)$$

with B open.

Then there are elements of F

$$\alpha_1^{i_1}, \beta_1^{i_1,j_1}, \alpha_2^{i_1,j_1,i_2}, \ldots, \beta_k^{i_1,\ldots,j_k}$$

with $i_1, \ldots, j_k < s$ such that $[\![A]\!]$ equals modulo the ideal $\mathcal{I}$ to the set - defined in $\mathcal{M}$ - of all $\omega \in \Omega$ satisfying:

$$\bigvee_{i_1} \bigwedge_{j_1} \ldots \bigvee_{i_k} \bigwedge_{j_k} B(\alpha_1^{i_1}(\omega), \beta_1^{i_1,j_1}(\omega), \ldots, \beta_k^{i_1,\ldots,j_k}(\omega)) \ .$$

Proof :

Fix arbitrary $\epsilon > 0$, a standard rational. By Lemma 3.1.2 there are $r \in \mathbf{N}$ and elements of F:

$$\alpha_1^{i_1}, \beta_1^{i_1,j_1}, \alpha_2^{i_1,j_1,i_2}, \ldots, \beta_k^{i_1,\ldots,j_k}$$

with all $i_1, \ldots, j_k < r$ such that the distance of $[\![A]\!]$ from

$$\bigvee_{i_1} \bigwedge_{j_1} \ldots [\![B(\alpha_1^{i_1}, \beta_1^{i_1,j_1}, \ldots, \beta_k^{i_1,\ldots,j_k})]\!]$$

is at most ϵ. However, this $[\![\ldots]\!]$-value equals, modulo the ideal $\mathcal{I}$, to the set of all $\omega \in \Omega$ satisfying (as evaluated in $\mathcal{M}$) the Boolean formula formed from formulas $B(\alpha_1^{i_1}(\omega), \beta_1^{i_1,j_1}(\omega), \ldots, \beta_k^{i_1,\ldots,j_k}(\omega))$:

$$\bigvee_{i_1} \bigwedge_{j_1} \ldots B(\alpha_1^{i_1}(\omega), \beta_1^{i_1,j_1}(\omega), \ldots, \beta_k^{i_1,\ldots,j_k}(\omega)) \ .$$

Applying this for $\epsilon := 1/2\ell$ yields the following claim.

Claim: *For any $\ell \in \mathbf{N}$ there are $r_\ell \in \mathbf{N}$ and $\alpha_1^{i_1}, \beta_1^{i_1,j_1}, \ldots, \beta_k^{i_1,\ldots,j_k} \in F$ with $i_1, \ldots, j_k < r_\ell$ such that the distance of $[\![A]\!]$ and of $V_\ell/\mathcal{I}$, where V_ℓ is the set of all $\omega \in \Omega$ satisfying the formula U_ℓ:*

$$\bigvee_{i_1<r_\ell} \bigwedge_{j_1<r_\ell} \ldots B(\alpha_1^{i_1}(\omega), \beta_1^{i_1,j_1}(\omega), \ldots, \beta_k^{i_1,\ldots,j_k}(\omega))$$

is at most $1/2\ell$.

Let $(U_w)_{w<t}$ be a non-standard extension of $(U_\ell)_{\ell\in\mathbf{N}}$ provided by the $\aleph_1$-saturation. Let $V \in \mathcal{A}$ be a set such that $[\![A]\!] = V/\mathcal{I}$.

By the hypothesis that F is compact, $F = \bigcap_{k\in\mathbf{N}} F_k$ for some definable family $(F_a)_a$ (defined by an L_{all} formula H - cf. Definition 3.4.1), and $F_k \supseteq F_{k+1}$ for all $k \in \mathbf{N}$.

Consider the following three properties parameterized by w:

1. U_w is a formula of the form

   $$\bigvee_{i_1<r} \bigwedge_{j_1<r} \ldots B(\alpha_1^{i_1}(\omega), \beta_1^{i_1,j_1}(\omega), \ldots, \beta_k^{i_1,\ldots,j_k}(\omega))$$

   for some $r < s$, and with $\alpha_1^{i_1}, \beta_1^{i_1,j_1}, \ldots, \beta_k^{i_1,\ldots,j_k}$ from F_w.

2. The symmetric difference $V \triangle V_w$ has the counting measure at most $1/w$.

3. $F_1 \supseteq F_2 \supseteq \ldots \supseteq F_w$.

For any standard w all three properties are met: The arity r_w is standard and so $r_w < s$, and αs and βs occurring in U_w are even from $F \subseteq F_w$. The second property is also satisfied: The counting measure of the symmetric difference $V \triangle V_w$ differs from the distance of $[\![A]\!]$ and $V_w/\mathcal{I}$ by at most an infinitesimal amount (in particular, by less than $1/2w$) and the distance itself is at most $1/2w$. The third property holds by the definition of compact families.

The properties are all definable in $\mathcal{M}$ (using formula H and V and $(U_w)_{w<t}$ as parameters) and hence by overspill must hold up to some non-standard $p < t$.

The αs and βs occurring in U_p are from F_p which is contained in all F_k for $k \in \mathbf{N}$, hence in F. The counting measure of $V \triangle V_p$ is $< 1/p$. This is infinitesimal, so

$$[\![A]\!] = V/\mathcal{I} = V_p/\mathcal{I} \ .$$

q.e.d.

Consider the simplest case when $A(\beta)$ has the form $\exists x B(x,\beta)$, B open and β the only parameter from F. Then (for compact F):

$$[\![A(\beta)]\!] \;=\; \{\omega\in\Omega \mid \bigvee_{i<s} B(\alpha^i(\omega),\beta(\omega))\}/\mathcal{I}$$

for some αs from F. This does not mean that A equals in $\mathcal{M}$ to $\bigvee_{i<s} B(\alpha^i,\beta)$, as $[\![A(\beta)]\!]$ generally does not equal (modulo $\mathcal{I}$) to the set of ωs satisfying $A(\beta(\omega))$. Also, the right-hand side is not equal to $\bigvee_{i<s}[\![B(\alpha^i,\beta)]\!]$ as taking the quotient by $\mathcal{I}$ does not commute with infinite disjunctions. Hence we should be careful in interpreting the theorem.

It is now clear that we can analogously strengthen Lemma 3.3.3.

Theorem 3.5.2 *Let F be an L-closed family that is*

1. *closed under definition by cases by open L-formulas, and*
2. *compact.*

Let A be an L-sentence of the form

$$\exists x_1\forall y_1\ldots\exists x_k\forall y_k B(x_1,y_1,\ldots,x_k,y_k)$$

with B open (and with parameters).

Then there are $\alpha_1,\beta_1,\ldots,\alpha_k,\beta_k\in F$ such that for all $i=1,\ldots,k$:

$$[\![\forall y_i\exists x_{i+1}\forall y_{i+1}\ldots\exists x_k\forall y_k B(\alpha_1,\beta_1,\ldots,\alpha_i,y_i,x_{i+1},y_{i+1},\ldots,x_k,y_k)]\!] = [\![A]\!]$$

and

$$[\![\exists x_{i+1}\forall y_{i+1}\ldots\exists x_k\forall y_k B(\alpha_1,\beta_1,\ldots,\alpha_i,\beta_i,x_{i+1},y_{i+1},\ldots,x_k,y_k)]\!] = [\![A]\!]\ .$$

For compact families that we shall consider in Part III, based on various types of decision trees, Theorem 3.5.2 actually follows from Theorem 3.5.1 (formulas $B(\alpha_1^{i_1}(\omega),\beta_1^{i_1,j_1}(\omega),\ldots,\beta_k^{i_1,\ldots,j_k}(\omega))$ are then computed by decision trees and hence the whole formula too).

Chapter 4

The truth in N and the validity in $K(F)$

In this section we apply a particular Skolemization to a formula in order to link its truth-values in $\mathbf{N}$ and $K(F)$.

Consider an L-formula of the form $\exists y C(\overline{x}, y)$, with $\overline{x}$ a k-tuple of free variables, C open. Let $f(\overline{x})$ be a new k-ary function symbol, and let

$$Sk[\exists y C(\overline{x}, y); f]$$

be the universal closure of the formula

$$C(\overline{x}, y) \rightarrow C(\overline{x}, f(\overline{x})) \ .$$

Clearly $Sk[\exists y C(\overline{x}, y); f]$ implies (in predicate calculus) the equivalence

$$\exists y C(\overline{x}, y) \equiv C(\overline{x}, f(\overline{x})) \ .$$

For an L-formula of the form $\forall y C(\overline{x}, y)$, again C open with parameters from F, we define

$$Sk[\forall y C(\overline{x}, y); f]$$

to be

$$Sk[\exists y \neg C(\overline{x}, y); f] \ .$$

This axiom then implies

$$\forall y C(\overline{x}, y) \equiv C(\overline{x}, f(\overline{x})) \ .$$

This reduction can be iterated. Let A be an L-sentence of the form

$$Q_1 x_1 \dots Q_k x_k B(x_1, \dots, x_k) \ ,$$

B open and possibly with parameters. Introduce $(k-1)$-ary function symbol f_k and axiom $Sk[Q_k x_k B(x_1, \dots, x_k); f_k]$. Then A is equivalent, modulo this axiom, to

$$Q_1 x_1 \dots Q_{k-1} x_{k-1} B(x_1, \dots, x_{k-1}, f_k(x_1, \dots, x_{k-1})) \ .$$

Analogously introduce $(i-1)$-ary function symbols f_i and corresponding axioms, for $i = k-1, k-2, \dots 1$ to successively reduce the quantifier complexity of the formula, the last formula

$$B(f_1, f_2(f_1), \dots, f_k(f_1, f_2(f_1), \dots, f_{k-1}(f_1, f_2(f_1), \dots)))$$

(any occurrence of f_i is substituted for an occurrence of x_i - we do not show it explicitly) being a quantifier-free sentence (f_1 is a constant).

Definition 4.0.3 *For a sentence A of the form as above denote by*

$$Sk[A; f_1, \dots, f_k]$$

the conjunction of the k universal axioms introduced in the reduction process, and by A_{Sk} the quantifier-free sentence obtained at the end.

Any k-tuple of functions $f_1, \dots, f_k$ (of the appropriate arity) defined on $\mathbf{N}$ *and satisfying $Sk[A; f_1, \dots, f_k]$ will be called a* **Skolem tuple** *for A.*

The next lemma follows from the construction.

Lemma 4.0.4 *For a sentence A as above the axiom $Sk[A; f_1, \dots, f_k]$ implies in predicate logic alone the equivalence*

$$A \equiv A_{Sk} \ .$$

We aim at the following statement.

Lemma 4.0.5 *Assume that A is an L-sentence of the form as above that has no parameters from F. Assume also that L contains (symbols from L_{all} for) a Skolem tuple for A, and constants for all elements of* $\mathbf{N}$.

Then for any L-closed family F it holds that:

$$\mathbf{N} \models A \quad \textit{iff} \quad [\![A]\!] = 1_{\mathcal{B}} \ .$$

Proof :

Let $f_1, \ldots, f_k$ be a Skolem tuple for A that is in L. By Lemma 1.4.2 the axiom $Sk[A; f_1, \ldots, f_k]$ is $K(F)$-valid. By Lemma 4.0.4 then A is equivalent to A_{Sk} in both $\mathbf{N}$ and $K(F)$.

But by Lemma 1.4.2 again $\mathbf{N} \models A_{Sk}$ holds if and only if $[\![A_{Sk}]\!] = 1_{\mathcal{B}}$.

q.e.d.

We shall use this lemma to our advantage in both directions, to pull facts from $\mathbf{N}$ to $K(F)$ and vice versa.

Part II

Second order structures

Chapter 5

Structures $K(F, G)$

We have developed some basic facts about models $K(F)$ in the preceding part, all in the first-order setting. It is often more transparent however, to define various bounded arithmetic theories in a second-order language. This concerns, in particular, theories related to weak complexity classes like AC^0 and $AC^0(2)$ we will be concerned with in future parts of the book.

"Second-order language" is a misnomer in this context: No second order logic is involved. The languages of these second-order theories are many-sorted first-order languages having besides the number-sort also a sort for sets (maybe also for relations or functions, when convenient). The intended range of sets are *bounded* sets, i.e. sets admitting an upper bound on their elements. One thinks of sets as representing binary strings and numbers as representing positions of bits or lengths of strings. In connection with computational complexity theory it is important not to require that the number sort is closed under fast growing functions.

Of course, one could use only numbers and consider sets as being coded by numbers. However, a typical subset of $[n]$ (i.e. a string of length close to n) is coded by a number proportional to 2^n. As we do not want the exponential function to be defined on all of number sort, we would have to consider only partially defined functions. That is not difficult to do but the second-order formalism appears more convenient.

In fact, we could have allowed also unbounded sets. Although we are interested only in provability of bounded formulas, it is well-know that allowing unbounded sets does not change the strength of the theories considered as far as bounded formulas are concerned (cf. Buss [11], e.g. theory $V^0_1(BD)$ and other).

Paris and Wilkie [85] were first to consider a second-order version of a bounded arithmetic theory; they studied theory $I\Delta_0(R)$ extending Parikh's $I\Delta_0$ from [84]. Buss [11] then defined a variety of not so trivial second-order theories and many later authors recognized the advantages of the second-order formalism. Cook and Nguyen [29] offer an elegant and comprehensive treatment of second order theories related to various weak computational complexity classes, as well as relevant bibliographic information.

5.1 Language L^2 and the hierarchy of bounded formulas

Let $L \subseteq L_{all}$ be any first-order language. We shall take eventually a fairly rich language L_n (see Section 5.2) but usually the following basic language L_{PA} is adopted:

- Constants 0 and 1.
- Functions $x + y$, $x \cdot y$.
- Relation $x < y$.

Any language $L \subseteq L_{all}$ determines **language $\mathbf{L^2}$**: it is obtained from L by adding another sort of variables $X, Y, \ldots$ whose intended interpretation are bounded sets. Although the logic is always first-order we shall often call the elements of the set sort simply "second-order".

There is the equality symbol for sets (we consider equality as a logical symbol automatically included) which we will denote also $=$. In addition there is

- Relation $x \in X$ between numbers and sets (the membership relation).

Cook and Nguyen [29] also include function symbol $|X|$ for a function assigning a number to a set. Its intended meaning is the least upper bound on elements of X. Buss [11] considered instead variables of the form X^t, the superscript being a first-order term bounding the elements of X. We will adopt neither formalism and instead will simply consider interpretation of the set variables by bounded sets; in our structures the axiom

$$\forall X \exists x \forall y (y \in X \rightarrow y < x)$$

will be always valid. We shall use the abbreviation $\mathbf{X} < \mathbf{x}$ for the formula $\forall y(y \in X \to y < x)$; hence the axiom is simply

$$\forall X \exists x X < x \ .$$

As in this book we are concerned with model-theoretic constructions and not with proof-theoretic arguments, such details of the syntax are not important.

We now define a hierarchy of bounded L^2-formulas. **Bounded formulas** are those having all quantifiers (first-order or second-order) bounded, i.e. of the form: $\exists x < t$, $\forall x < t$ or $\exists X < t$, $\forall X < t$ where t is a first-order term.

The class of all bounded formulas without the set-quantifiers is denoted $\Sigma_0^{1,b}$. It has important subclass Σ_∞^b: the class of **first-order bounded L^2-formulas**, i.e. bounded L-formulas. In particular, Σ_∞^b-formulas do not have even free second-order variables.

Define simultaneously two hierarchies of classes of bounded L^2-formulas, $\Sigma_i^{1,b}$ and $\Pi_i^{1,b}$ for $i \geq 1$. These are the smallest classes of bounded formulas such that it holds:

- $\Sigma_{i-1}^{1,b} \subseteq \Sigma_i^{1,b} \cap \Pi_i^{1,b}$.
- Both $\Sigma_i^{1,b}$ and $\Pi_i^{1,b}$ are closed under $\vee$ and $\wedge$.
- Both $\Sigma_i^{1,b}$ and $\Pi_i^{1,b}$ are closed under bounded number-quantifiers.
- If $A \in \Sigma_i^{1,b}$ then $\neg A \in \Pi_i^{1,b}$, and vice versa.

Two particular classes of bounded L^2-formulas are **strict** $\Sigma_1^{1,b}$-formulas and **strict** $\Pi_1^{1,b}$-formulas, denoted $s\Sigma_1^{1,b}$ and $s\Pi_1^{1,b}$. These are of the form

$$QX_1 < t_1, \ldots, X_k < t_k \ \varphi$$

with Q being $\exists$ or $\forall$ respectively, and φ a $\Sigma_0^{1,b}$-formula. Cook and Nguyen [29] define only the hierarchy if strict bounded formulas, denoted Σ_i^B and Π_i^B there.

5.2 Cut $\mathcal{M}_n$, languages L_n and L_n^2

This section introduces a few objects used for the rest of the book. We first

- **fix a non-standard number** $n \in \mathcal{M}$.

The parameter n is fixed throughout the book. No particular properties of n are assumed in the following definitions but in future sections we may pose on n some additional requirement (e.g. that it is even).

Define a cut $\mathcal{M}_n \subseteq_e \mathcal{M}$ by:

$$\mathcal{M}_n := \bigcup_{t>\mathbf{N}} \{b \in \mathcal{M} \mid b \leq 2^{n^{1/t}}\} .$$

(This cut is called a **large canonical model** in [61, 62].) Hence elements of $\mathcal{M}_n$ are numbers that are subexponential in terms of n.

Let $L_n \subseteq L_{all}$ be the language consisting of:

- All relation symbols from L_{all}, and
- those function symbols $f \in L_{all}$ such that: $f''\mathcal{M}_n \subseteq \mathcal{M}_n$.

For example, if f is (a function symbol for) a polynomially bounded function:

$$|f(x)| \leq |x|^{O(1)}$$

then $f''\mathcal{M}_n \subseteq \mathcal{M}_n$ and $f \in L_n$.

In $\mathcal{M}$ there are no differences between numbers and $\mathcal{M}$-finite (i.e. with all elements bounded above by an element of $\mathcal{M}$) definable sets as any such set is coded by a number (and vice versa). However, in $\mathcal{M}_n$ this is no longer true: the code of a subset of $[n]$ may well be outside of $\mathcal{M}_n$.

Hence the extension of language L_n to the second-order language L_n^2 defined in Section 5.1, will be essential.

5.3 Definition of the structures

All theories under consideration have some pairing function on numbers, e.g. the usual: $\langle 0, 0\rangle$ is encoded by 0 and for $i + j > 0$

$$\langle i, j\rangle := \frac{(i+j)(i+j-1)}{2} + i .$$

Bounded functions and relations (i.e. with bounded domains and ranges) can be thus encoded by bounded sets. However, it is technically advantageous to have bounded unary functions directly as the second order objects in the structures; sets are simply functions with values 0 and 1.

Hence we make the following definition.

Definition 5.3.1 *Assume $L \subseteq L_n$. A model $K(F,G)$ for a theory in L^2 consists of an L-closed family F of functions on a sample space Ω, as in Definition 1.3.1, and a family G of some functions $\Theta \in \mathcal{M}$ assigning to $\omega \in \Omega$ a function $\Theta_\omega \in \mathcal{M}$ (it is thus $\mathcal{M}$-finite) that maps a subset $dom(\Theta_\omega)$ of $\mathcal{M}_n$ into $\mathcal{M}_n$.*

The definition extends Definition 1.3.1 by defining how $\Theta \in G$ operates on F: For any $\alpha \in F$

$$\Theta(\alpha)(\omega) = \begin{cases} \Theta_\omega(\alpha(\omega)) & \text{if } \alpha(\omega) \in dom(\Theta_\omega) \\ 0 & \text{otherwise.} \end{cases}$$

It is required that for all $\Theta \in G$ and all $\alpha \in F$

$$\Theta(\alpha) \in F$$

too. The value of equality is defined identically as for first-order objects:

$$[\![\Theta = \Xi]\!] \;:=\; \{\omega \in \Omega \mid \theta_\omega = \Xi_\omega\}/\mathcal{I} \;.$$

Further we add two inductive clauses for second order quantifiers:

3. $[\![\exists X A(X)]\!] := \bigvee_{\Theta \in G} [\![A(\Theta)]\!]$,
4. $[\![\forall X A(X)]\!] := \bigwedge_{\Theta \in G} [\![A(\Theta)]\!]$.

Let us illustrate the definition by two examples. Let $\Theta, \Gamma \in G$ be (unary) functions and $\alpha, \beta \in F$. The value $[\![\Theta(\alpha) = \Gamma(\beta)]\!]$ is computed as follows: By the definition $\Theta(\alpha) = \alpha'$ and $\Gamma(\beta) = \beta'$ for some $\alpha', \beta' \in F$, hence $[\![\Theta(\alpha) = \Gamma(\beta)]\!] = [\![\alpha' = \beta']\!]$.

As a second example let us see how we represent in G a binary relation: A unary function $\Theta \in G$ and $\langle x, y\rangle$ (the pairing function on numbers, an L_n-function) define relation $\hat{\Theta}(x,y)$ by:

$$[\![\hat{\Theta}(\alpha,\beta)]\!] = [\![\Theta(\langle\alpha,\beta\rangle) = 1]\!] \;.$$

We may want to redefine Θ (by changing all its values different from 1 to 0) to get also the dual property:

$$[\![\neg\hat{\Theta}(\alpha,\beta)]\!] = [\![\Theta(\langle\alpha,\beta\rangle) = 0]\!]$$

but this might violate the property that Θ must map F into F.

In the following lemma one just needs to verify the second order equality axioms, which is straightforward.

Lemma 5.3.2 *All L^2-sentences valid in many sorted first-order logic with equality are valid in the two-sorted model $K(F,G)$. In particular, instances of the equality axioms*

$$\alpha = \beta \rightarrow \Theta(\alpha) = \Theta(\beta)$$

and

$$\Theta = \Xi \rightarrow \Theta(\alpha) = \Xi(\alpha)$$

are valid in $K(F,G)$.

5.4 Equality of functions, extensionality and possible generalizations

We shall make now a brief digression and we will discuss the issue of the definition of the equality between functions and the axiom of extensionality.

Let us restrict ourselves to sets (i.e. 0/1 - valued functions) in this discussion. We write $x \in X$ in place of $X(x) = 1$. Definition 5.3.1 implies the following clause:

- $[\![\alpha \in \Theta]\!] = \{\omega \in \Omega \mid \alpha(\omega) \in \Theta_\omega\}/\mathcal{I}$,

and

- $[\![\Theta = \Xi]\!] = \{\omega \in \Omega \mid \Theta_\omega = \Xi_\omega\}/\mathcal{I}$.

With this definition we automatically got the validity of the relevant equality axioms in Lemma 5.3.2

$$\alpha = \beta \rightarrow \alpha \in \Theta \equiv \beta \in \Theta$$

and

$$\Theta = \Xi \rightarrow \alpha \in \Theta \equiv \alpha \in \Xi \ .$$

But note that the extensionality

$$\forall X, Y \exists x (X = Y \vee x \in X \not\equiv x \in Y)$$

needs not to be valid in a structure $K(F,G)$ (although it will be in the models we shall define later). For example, let F consists of constants bigger than 1 while G contains two random variables, one giving constantly set $\{0\}$ and the other one giving constantly set $\{1\}$.

However, it is possible to "arrange" extensionality by redefining the second order equality $[\![X = Y]\!]$ as

$$[\![\Theta = \Xi]\!] := \bigwedge_{\alpha} [\![\alpha \in \theta \equiv \alpha \in \Xi]\!]$$

(this definition would automatically yield the second equality axioms).

In fact, one can proceed even more generally (we make the observation here although we will not take it up in this book). By taking the interpretation of sets to be also random variables in $\mathcal{M}$ we got the first equality axioms for free. However, these equality axioms are the *only* conditions we need to pose on sets in the model in order for Lemma 5.3.2 to hold. In particular, we could take for G a collection of some maps

$$\mathcal{U} \ : \ F \to F$$

(we use letter $\mathcal{U}$ instead of a Greek letter to avoid a confusion) such that the following inequality holds for all $\alpha, \beta \in F$:

$$[\![\alpha = \beta]\!] \ \leq \ [\![\mathcal{U}(\alpha) = \mathcal{U}(\beta)]\!] \ .$$

This guarantees the first equality axioms and then by defining

$$[\![\mathcal{U} = \mathcal{V}]\!] := \bigwedge_{\alpha \in F} [\![\mathcal{U}(\alpha) = \mathcal{V}(\alpha)]\!]$$

gets us the second equality axioms and the extensionality for free.

An advantage of our treatment of the second order objects is that all results proved for $K(F)$ apply equally well to $K(F, G)$; the splitting of random variables into two families F and G is just for convenience but otherwise they are treated equally as in Definition 1.3.1. A little more general approach would be to have one family and a new unary predicate $Number(x)$ that defines the sort of numbers (and $\neg Number(x)$ defines the second order sort). Our definition with separate families F and G would correspond in this more general set-up to requiring that $[\![Number(\alpha)]\!]$ is either $1_{\mathcal{B}}$ or $0_{\mathcal{B}}$ for all α.

5.5 Absoluteness of $\forall\Sigma^b_\infty$-sentences of language L_n

We now observe a simple but useful preservation property of structures $K(F, G)$.

Lemma 5.5.1 *Let A be $\forall\Sigma^b_\infty$ L_n-sentence (no parameters from F or G). Assume $K(F,G)$ is a second-order structure such that F is L_n-closed.*

Then it holds that:

$$\mathbf{N} \models A \quad \textit{iff} \quad [\![A]\!] = 1_{\mathcal{B}} \ .$$

Proof :

In accordance with Lemma 4.0.5 it is sufficient to show that all $\forall\Sigma^b_\infty$ L_n-sentences have a Skolem tuple in L_n. This is established by induction on the number of bounded quantifiers in A, using the following claim.

Claim: *Let $A(\overline{x}, y)$ be an open L_n-formula and $t(\overline{x})$ and L_n-term. Then there is a function $f(\overline{x}) \in L_n$ such that both*

- $\forall \overline{x}\ f(\overline{x}) \leq t(\overline{x})$, *and*
- $Sko[\exists y \leq t(\overline{x}, A(\overline{x}, y); f]$

are true in $\mathbf{N}$ and valid in $K(F,G)$.

The claim follows from the fact that L_n contains any function majorized by an L_n-term, and from Lemma 1.4.2.

q.e.d.

Part III

AC^0 world

Chapter 6

Theories $I\Delta_0$, $I\Delta_0(R)$ and V_1^0

In this part we shall consider models of second-order theories like $I\Delta_0(R)$ or V_1^0, related to constant depth Frege systems and to AC^0-circuits. The common approach to the analysis of these models is the Skolemization (Chapter 9) which utilizes the switching lemma from Boolean complexity (Chapter 11).

First bounded arithmetic theory $I\Delta_0$ has been defined by Parikh [84]. This is a first-order theory, a subsystem of Peano arithmetic. Its language is the L_{PA} from Section 5.1 (although sometimes the successor function is taken in place of constant 1) and the axioms are the universal closures of the following formulas::

1. $0 + 1 = 1$
2. $x + 1 \neq 0$
3. $x + 1 = y + 1 \rightarrow x = y$
4. $x \neq 0 \rightarrow \exists y(y + 1 = x)$
5. $x + 0 = x$
6. $x + (y + 1) = (x + y) + 1$
7. $x \cdot 0 = 0$
8. $x \cdot (y + 1) = (x \cdot y) + x$

which comprise the so called *Robinson's Q*, together with the following set (a smaller one would suffice but there is no need to be particularly economic) of axioms about ordering (we use $x \leq y$ to abbreviate $x < y \vee x = y$):

1. $(x \leq y \wedge y \leq x) \rightarrow x = y$
2. $x \leq x + y$
3. $x \leq x + 0$
4. $x \leq y \vee y \leq x$
5. $x \leq y \equiv x < y + 1$

and the all-important scheme IND of bounded induction

$$A(0, \overline{y}) \wedge \forall x(A(x, \overline{y}) \rightarrow A(x + 1, \overline{y})) \rightarrow \forall x A(x, \overline{y})$$

where A is a bounded L_{PA} formula ($\overline{y}$ are implicitly universally quantified in front of the axiom). The set of axioms except the induction axioms will be called PA^-. We refer the reader to [55] for more information about $I\Delta_0$.

Theory $I\Delta_0(R)$, where $R(x, y)$ is a binary relation symbol, has been considered by Paris and Wilkie [85]. The particular choice of a binary relation symbol for R rather than unary or ternary or even a function symbol is to a large extent irrelevant. A binary relation symbol allows for an easy formalization of various interesting combinatorial principles about graphs, e.g. the pigeonhole principle.

This theory extends $I\Delta_0$ by first allowing R in the language and then accepting as an axiom any instance of the induction scheme for any bounded formula in the extended language.

One can think of $I\Delta_0(R)$ as a sort-of second order theory: there is one variable for an unknown binary relation which cannot be quantified.

Theory V^0_1 defined by Buss [11] (our version is more like $V^0_1(BD)$ of [11]) or its version V^0 in [29] extends $I\Delta_0$ to a proper second-order theory, i.e. a theory in language L^2_{PA}.

Axioms of V^0_1 are PA^- axioms above together with the induction axioms:

$$B(0, \overline{y}) \wedge \forall x(B(x, \overline{y}) \rightarrow B(x + 1, \overline{y})) \rightarrow \forall x B(x, \overline{y})$$

where B is a $\Sigma^{1,b}_0$-formula, and with the scheme of bounded comprehension

$$\exists X \forall y < x \ (y \in X \equiv A(y, \overline{z}))$$

accepted also for all $\Sigma^{1,b}_0$-formulas A.

Theory V^0_1 is very close to $I\Delta_0(R)$ in the following sense. If we take any model N of $I\Delta_0(R)$ and add to it as second-order objects all bounded subsets of N definable in N by a Σ^b_∞-formula in the language of $I\Delta_0(R)$ (with parameters from N) then the resulting $L^2_{PA(R)}$-structure is a model of V^0_1.

The reader may look into [55] for more information about these theories.

In future chapters we will be concerned with the task to construct "interesting" structures $K(F, G)$ in which V^0_1 or stronger theories are valid. Of course, one can always achieve that by Skolemizing the theory first (for V^0_1 this can be done by introducing AC^0-computable maps from strings to strings, cf.[29]) and applying Chapter 4. Using the witnessing theorems from Chapter 3 for the analysis of the resulting model is then quite analogous to using Herbrand theorem on the Skolemized theory, and the model seems consequently quite dull. In other words, the qualification "interesting" is essential.

Chapter 7

Shallow Boolean decision tree model

In this chapter we construct a rudimentary example of a second-order structure $K(F_{rud}, G_{rud})$, and we shall establish its basic properties. The particular models used in later chapters will all be related to this rudimentary example. In particular, it will be clear that the basic properties of $K(F_{rud}, G_{rud})$ established here hold also (and for exactly the same reasons) in these models. This will allow us to concentrate on establishing specific properties of each of these particular models.

7.1 Family F_{rud}

This whole section is one long definition; we shall not split it into a sequence of smaller definitions with their own numbering. Recall the definition of the cut $\mathcal{M}_n$ from 5.2, and use a standard notation $[n] := \{1, \ldots, n\}$.

The sample space is

$$\Omega := \{\omega \in \mathcal{M} \mid \omega \subseteq [n]\} \ .$$

To define the family F_{rud} we have to define first an auxiliary concept of a labeled tree. A decision tree T is defined as usual: It is a finite binary tree whose inner nodes (i.e. non-leafs) are labeled by queries of the form $i \in_? \omega$, for $i \in [n]$. The two outgoing edges from such a node are labeled yes and no, respectively. The depth of a tree, denoted $dp(T)$, is the length of the longest path (i.e. the number of edges on it) from the root to a leaf. A depth 0 tree

thus consists of just one node, a leaf. A labeling of T is a map ℓ assigning to leafs of T values in $\mathcal{M}_n$.

Let (T, ℓ) be a labeled tree. Any $\omega \in \Omega$ determines a unique path through T ending in a leaf that we shall denote $T(\omega)$. In this way (T, ℓ) defines a function

$$\omega \in \Omega \to \ell(T(\omega)) \in \mathcal{M}_n \ .$$

Family F_{rud} consists of all functions

$$\alpha \ : \ \Omega \to \mathcal{M}_n$$

from $\mathcal{M}$ such that there is labeled tree $(T, \ell) \in \mathcal{M}$ such that

- $\alpha(\omega) = \ell(T(\omega))$, for all $\omega \in \Omega$, and
- the depth $dp(T)$ of T satisfies $dp(T) < n^{1/t}$, for some $t > \mathbf{N}$.

In particular, depth 0 trees labeled by elements of $\mathcal{M}_n$ represent in F_{rud} constants from $\mathcal{M}_n$.

7.2 Family G_{rud}

The definition of G_{rud} (Definition 7.2.3) requires some preliminary notions.

Definition 7.2.1 *Let $m \in \mathcal{M}_n$ and let $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1}) \in \mathcal{M}$ be an m-tuple of elements of F_{rud}.*

For any $\alpha \in F_{rud}$ define $\hat{\beta}(\alpha) : \Omega \to \mathcal{M}_n$ by

$$\hat{\beta}(\alpha)(\omega) = \begin{cases} \beta_{\alpha(\omega)}(\omega) & \text{if } \alpha(\omega) < m \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 7.2.2 *Let $m \in \mathcal{M}_n$ and let $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1}) \in \mathcal{M}$ be an m-tuple of elements of F_{rud}.*

Then function $\hat{\beta}$ from Definition 7.2.1 maps F_{rud} into F_{rud}.

Proof :

As $\hat{\beta} \in \mathcal{M}$, the induction in $\mathcal{M}$ implies that there is a maximal depth of a tree computing one of β_i, $i < m$. This depth is therefore bounded above by n raised to an infinitesimal. The depth of the tree computing any $\alpha \in F$ is bounded by a term of the same form, and hence so is the depth of the tree computing $\hat{\beta}(\alpha)$.

q.e.d.

Definition 7.2.3 *Family G_{rud} consists of all random variables Θ computed by some $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1})$, where $m \in \mathcal{M}_n$ and $\hat{\beta} \in \mathcal{M}$ is a m-tuple of elements of F_{rud}.*

In other words, for Θ computed by $\hat{\beta}$ the function Θ_ω has the graph $\{(i, \beta_i(\omega)) \mid i < m\}$.

7.3 Properties of F_{rud} and G_{rud}

Let $L_n(F_{rud}, G_{rud})$ be the language L_n augmented by names for elements of F_{rud} and G_{rud}.

Lemma 7.3.1 *F_{rud} is an L_n-closed family. It is closed under definitions by open $L_n(F_{rud}, G_{rud})$-formulas, and it is compact. Also G_{rud} is compact.*

Proof :

Let $f(x, y)$ be a binary function symbol from L_n (functions of higher arity are treated analogously). Let $\alpha, \beta \in F_{rud}$ be two random variables computed by trees (T_α, ℓ_α) and (T_β, ℓ_β) respectively. The element $f(\alpha, \beta)$ is computed by (T, ℓ) where T is obtained from T_α by replacing each leaf by a copy of T_β, and where the label $\ell(T(\omega))$ is defined to be

$$\ell(T(\omega)) := f(\ell_\alpha(T_\alpha(\omega)), \ell_\beta(T_\beta(\omega))) \ .$$

This is a sound definition as both $T_\alpha(\omega)$ and $T_\beta(\omega)$ are determined by $T(\omega)$, and $dp(T)$ clearly obeys the bounds imposed by the definition of F_{rud} if $dp(T_\alpha)$ and $dp(T_\beta)$ do. This verifies the L_n-closeness of F_{rud}.

The truth values of atomic sentences

$$R(\alpha_1, \ldots, \alpha_k) \ \text{ or } \ f(\alpha_1, \ldots, \alpha_k) = g(\alpha_1, \ldots, \alpha_k) \ ,$$

for $R, f, g \in L_n$, can be clearly computed by decision trees composed from the labeled decision trees computing the individual α_is. The depth of the tree is the sum of the depths of the individual trees and hence obeys the required bounds.

The truth value of an atomic sentence of the form $\Theta(\alpha) = \gamma$ is computed by combining the tree T extending T_α by adding trees computing Θ given by

$\hat{\beta} = (\beta_0, \ldots, \beta_{m-1})$ (at a leaf of T_α labeled by i compute β_i and label the new leafs accordingly), with the tree computing γ. Hence it is again computed by a tree from F_{rud}. It follows that F_{rud} is closed under definitions by cases by open $L_n(F_{rud}, G_{rud})$-formulas.

Finally, to verify the compactness of F_{rud}, define sets F_a to consists of those functions

$$\alpha \ : \ \Omega \to \{b \in \mathcal{M} \mid b < 2^{n^{1/a}}\}$$

computed by a tree of the depth less than $n^{1/a}$. So $F = \bigcap_{k \in \mathbf{N}} F_k$. Similarly, sets G_a consisting of functions computed by $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1})$, with $m < 2^{n^{1/a}}$ and all $\beta_i \in F_a$, certify that G_{rud} is compact too.

q.e.d.

Chapter 8

Open comprehension and open induction

Arranging that some modicum of induction is valid in a model is a key requirement. First, the theories in question have usually induction as their principal axiom. Second, some induction is also needed in order to deduce lower bounds for propositional proof system (cf. Part V). Generally, induction for a wider class of formulas yields lower bounds for a stronger proof system.

In this chapter we start investigating how to arrange induction in a model.

8.1 The $\langle\langle \ldots \rangle\rangle$ notation

In analyzing practically all models we will compute the truth values of various open sentences by considering the subset of the sample space where the instance of the sentence is true. This has been actually already done in Chapter 3 but now a need for some abbreviated notation will become especially pressing. We introduce it now.

Let $K(F,G)$ be an arbitrary L^2-structure, any $L \subseteq L_{all}$.

Definition 8.1.1 *For an L^2-sentence $B(\alpha_1, \ldots, \alpha_k, \Theta_1, \ldots, \Theta_\ell)$ with all parameters from F and G shown define:*

$$\langle\langle B(\alpha_1, \ldots, \alpha_k, \Theta_1, \ldots, \Theta_\ell) \rangle\rangle \ :=$$

$$\{\omega \in \Omega \mid B(\alpha_1(\omega), \ldots, \alpha_k(\omega), (\Theta_1)_\omega, \ldots, (\Theta_\ell)_\omega)\} \ .$$

The next lemma follows from the definition of the Boolean evaluation $[\![\ldots]\!]$ and the fact that taking quotient modulo the ideal $\mathcal{I}$ commutes with finite Boolean combinations.

Lemma 8.1.2 *Let $K(F,G)$ be an arbitrary L^2-structure and assume B is an open L^2-sentence with parameters from F or G. Then:*

$$[\![B]\!] \;=\; \langle\langle B\rangle\rangle/\mathcal{I}\ .$$

Note that for a general formula there is no a priori relation between these two Boolean values (one in $\mathcal{A}$ and the other in $\mathcal{B}$). For example, let A be a sentence of the form $\exists y R(\alpha, y)$, where $\alpha \in F$ and $R(x,y)$ is a relation from L such that $\forall x \exists y R(x,y)$ holds true in $\mathbf{N}$ but no $\beta \in F$ finds a witness for $\exists y R(\alpha(\omega), y)$ with a non-infinitesimal probability. Then $[\![A]\!] = 0_{\mathcal{B}}$ while $\langle\langle A\rangle\rangle = \Omega = 1_{\mathcal{A}}$.

8.2 Open comprehension in $K(F_{rud}, G_{rud})$

We note in this section that comprehension for open formulas is a corollary of two simple technical lemmas.

Lemma 8.2.1 *Let $A(y)$ be an open $L^2_n(F_{rud}, G_{rud})$-formula with the only free variable x.*

Then there is $t > \mathbf{N}$ such that for all $i \in \mathcal{M}_n$ the membership in $\langle\langle A(i)\rangle\rangle$ is computed by a tree with the leafs labeled by 1 *(for* yes*) or* 0 *(for* no*) of the depth bounded above by $n^{1/t}$. In particular, the membership is computed by an element of F_{rud}.*

Proof :

Instances $t(i)$ of any term $t(y)$ can be computed by trees satisfying the restrictions; this is verified by induction on the complexity of the term. Equations among terms and their Boolean combinations can be decided when all terms involved are computed; this composition of finitely many trees again obeys the restrictions.

q.e.d.

Taking for m an element of $\mathcal{M}_n$ that is bigger than all labels of α yields the following statement.

Lemma 8.2.2 *Any $\alpha \in F_{rud}$ is bounded above by some $m \in \mathcal{M}$ in the sense that $[\![\alpha < m]\!] = 1_{\mathcal{B}}$. In fact, one can find m such that even $\langle\langle \alpha < m \rangle\rangle = \Omega$.*

By the lemma it suffices to consider comprehension of the form

$$\exists X \forall y < m \ (y \in X \equiv A(y))$$

with A allowing parameters. Taking as a witness for X the element of G_{rud} computed $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1})$, where trees β_i are the trees provided by Lemma 8.2.1 computing the membership in $\langle\langle A(i) \rangle\rangle$, yields the open comprehension.

Corollary 8.2.3 *The axiom of comprehension is valid in $K(F_{rud}, G_{rud})$ for all open formulas.*

8.3 Open induction in $K(F_{rud}, G_{rud})$

Assume we have an open formula $A(x)$ with parameters from F_{rud} and G_{rud} that we do not show explicitly. We want to show that the induction axiom:

$$\neg A(0) \ \vee \ A(m) \ \vee \ \exists x < m \ (A(x) \wedge \neg A(x+1))$$

is valid in $K(F_{rud}, G_{rud})$. Here $m \in \mathcal{M}_n$; by Lemma 8.2.2 it is without a loss of generality to consider induction up to an element of $\mathcal{M}_n$ rather than up to an element of F_{rud}.

We may also assume without a loss of generality that $[\![A(0)]\!] = 1_{\mathcal{B}}$ and $[\![A(m)]\!] = 0_{\mathcal{B}}$ and, in fact, that

$$\langle\langle A(0) \rangle\rangle = \Omega \ \text{ and } \ \langle\langle A(m) \rangle\rangle = \emptyset \ .$$

This can be achieved by replacing $A(x)$ by another open formula

$$A'(x) \ := \ x = 0 \vee (x < m \wedge A(x))$$

if necessary, induction for which is equivalent to the induction for the original formula (over a few axioms about linear ordering).

Lemma 8.3.1 *Assume that there is $\alpha \in F_{rud}$ such that for all $\omega \in \Omega$ if $\alpha(\omega) = i$ then $i < m$ and*

$$\omega \in \langle\langle A(i) \rangle\rangle \setminus \langle\langle A(i+1) \rangle\rangle \ .$$

Then

$$[\![\alpha < m \wedge A(\alpha) \wedge \neg A(\alpha+1)]\!] = 1_{\mathcal{B}} \ ,$$

i.e. α *witnesses the induction axiom for* $A(x)$.

Proof :

Clearly $[\![\alpha < m]\!] = 1_{\mathcal{B}}$. By the hypothesis

$$\langle\langle A(\alpha)\rangle\rangle \ = \ \bigcup_{i<m} (\{\omega \in \Omega \mid \alpha(\omega) = i\} \cap \langle\langle A(i)\rangle\rangle)$$

$$= \ \bigcup_{i<m} \{\omega \in \Omega \mid \alpha(\omega) = i\} \ = \ \Omega \ .$$

So $[\![A(\alpha)]\!] = 1_{\mathcal{B}}$, and similarly one shows that $[\![\neg A(\alpha+1)]\!] = 1_{\mathcal{B}}$.

q.e.d.

It is easy to define by binary search a function $\alpha : \Omega \to M$, $\alpha \in \mathcal{M}$, that satisfies the hypothesis of the preceding lemma. Namely, given $\omega \in \Omega$ construct pairs (i_t, j_t) such that $(i_0, j_0) = (0, m)$ and each (i_{t+1}, j_{t+1}) is either the second or the first half of (i_t, j_t), depending on whether or not the midpoint $k := \lfloor \frac{i_t + j_t}{2} \rfloor$ satisfies $\omega \in \langle\langle A(k)\rangle\rangle$.

The issue remains whether this α is an element of F_{rud}. By Lemma 8.2.1 there is a bound $d = n^{1/t}$, $t > \mathbf{N}$, on the depth of trees computing the membership in all $\langle\langle A(k)\rangle\rangle$. It is then easy to see that α is computed by a tree of depth $\leq d \cdot \log(m)$. As $m \in \mathcal{M}_n$, $\log(m)$ is also bounded by n raised to an infinitesimal and so is $d \cdot \log(m)$. Hence $\alpha \in F_{rud}$ and we have the following lemma.

Lemma 8.3.2 *Open induction is valid in* $K(F_{rud}, G_{rud})$.

8.4 Short open induction

The ordinary induction axioms, as in Section 8.3, talk about validity of induction on intervals $[0, m]$, where $m \in \mathcal{M}_n$ is arbitrary. If one considers induction for a class of formulas closed under Boolean combinations and bounded quantification (as is, for example, the class $\Sigma_0^{1,b}$) then, in fact, the general induction follows already from a "short induction", an induction on

intervals $[0, \log(m)]$. This is showed by Solovay's method of shortening cuts (cf.[37, 55]) that we shall describe for completeness below.

Assume that it holds

$$A(0) \wedge \forall x(A(x) \rightarrow A(x+1))$$

but $\neg A(m)$.

Define $B(x) := \forall y \leq xA(y)$. It follows (again using only a few ordering axioms) that B also satisfies

$$B(0) \wedge \forall x(B(x) \rightarrow B(x+1)) \wedge \neg B(m)$$

but in addition it is closed downwards:

$$(B(x) \wedge y \leq x) \rightarrow B(y) \ ,$$

i.e. B is a cut.

Now put

$$C(x) \ := \ \forall y < m(B(y) \rightarrow B(x+y)) \ .$$

It is easy to verify that C is also a cut and $\neg C(m)$ but C is closed more:

$$(x < m \wedge C(x)) \ \rightarrow \ (C(2x) \wedge C(2x+1)) \ .$$

To see this assume $C(x)$. To entail $C(2x)$ we need to verify that

$$\forall y < m(B(y) \rightarrow B(2x+y)) \ .$$

For any $y < m$, $B(y) \wedge C(x)$ imply $B(x+y)$, which again with $C(x)$ implies (as necessarily $x + y < m$) $B(2x + y)$. The closure of C under $x \rightarrow 2x + 1$ follows as it is a cut.

Now define the last formula in this sequence of definitions:

$$D(x) \ := \ \exists y < m(C(y) \wedge |y| = x) \ ,$$

i.e. D consists of the lengths of elements in C. Then clearly $D(0)$ and $\neg D(\log(m))$ hold. D is also a cut (closed under the successor function) because of the closure properties of C.

Putting it all together: if induction for A fails on $[0, m]$ then induction for D fails on $[0, \log(m)]$.

Now we will observe that it is easy to witness this "short" induction for open formulas in $K(F_{rud}, G_{rud})$.

Lemma 8.4.1 *Assume $D(x)$ is an open formula with parameters from F_{rud} or G_{rud} and let $m \in \mathcal{M}_n$.*

Then there is $\alpha \in F_{rud}$ such that

$$[\![(D(0) \wedge \neg D(\log(m))) \ \rightarrow \ (\alpha < \log(m) \wedge D(\alpha) \wedge \neg D(\alpha+1))]\!] \ = \ 1_{\mathcal{B}} \ .$$

Proof :

For $i = 0, 1, \ldots, \log(m)$ let β_i be a tree computing the membership into $\langle\langle D(i) \rangle\rangle$ (use Lemma 8.2.1). Putting all these $\log(m)+1$ trees below one another will produce a tree of depth still bounded above by n raised to an infinitesimal. Using the tree we can define a random variable $\alpha \in F_{rud}$ such that for all $\omega \in \Omega$ $\alpha(\omega)$ equals to the minimal $i < \log(m)$ such that $\omega \in \langle\langle D(i) \rangle\rangle$.

It is easy to verify, using Lemma 8.1.2, that α satisfies the requirement.

q.e.d.

Formula D resulting from Solovay's construction is not open. Hence this lemma is useful only in the presence of an elimination of bounded quantifiers.

Chapter 9

Comprehension and induction via quantifier elimination: a general reduction

The last chapter showed that open comprehension and open induction is fairly easy to verify in the structure $K(F_{rud}, G_{rud})$, and that will be the same for most of the models we will consider. A way to arrange comprehension and induction for more complex formulas is quantifier elimination.

In this chapter we present a few well-known definitions adopted to structures $K(F, G)$ and to the way they will be used later, and we show that they are enough for the general reduction of comprehension and induction for $\Sigma_0^{1,b}$-formulas to their open cases.

9.1 Bounded quantifier elimination

The qualification "bounded" in the title of the section has a double meaning: we will eliminate bounded first-order quantifiers and on bounded domains only.

Definition 9.1.1 *Let $K(F, G)$ be an L_n^2-structure. Let $B(x_1, \ldots, x_k)$ be a bounded first-order $L_n^2(F, G)$-formula, i.e. a Σ_∞^b-formula, possibly with parameters from F or G.*

We say that B admits bounded quantifier elimination in $K(F, G)$ if for

all $m \in \mathcal{M}_n$ there is an open $L^2_n(F,G)$-formula $A(x_1,\ldots,x_k)$ such that

$$[\![\forall x_1,\ldots,x_k < m\ B(x_1,\ldots,x_k) \equiv A(x_1,\ldots,x_k)]\!] \ = \ 1_{\mathcal{B}}\ .$$

Note that formula A may have other parameters from F and G over those present in B.

9.2 Skolem functions in $K(F,G)$ and quantifier elimination

Theorem 3.5.2 already provides a form of quantifier elimination for the rudimentary model (using Lemma 7.3.1). Namely, for a formula $A(x)$ of the form $Q_1y_1\ldots Q_ky_kA(x,\overline{y})$ with A open and any $\alpha \in F_{rud}$, there are $\gamma_1,\ldots,\gamma_k \in F_{rud}$ such that

$$[\![B(\alpha) \equiv A(\alpha,\gamma_1,\ldots,\gamma_k)]\!] \ = \ 1_{\mathcal{B}}\ .$$

In fact, we could allow second order quantifiers in B too, by Lemma 7.3.1.

That fails, however, short of what is required in Definition 9.1.1. We need a form of quantifier elimination that describes explicitly the dependence of γ_i's on α in order to get the parametric version required in Definition 9.1.1.

Definition 9.2.1 *Let $B(\overline{x},y)$ be an open first-order $L^2_n(F,G)$-formula, $\overline{x} = (x_1,\ldots,x_k)$. Let $m \in \mathcal{M}_n$.*

Any $\Theta \in G$ satisfying for all $\alpha_i \in F$, $i \le k$, the equation:

$$[\![\bigwedge_{i\le k} \alpha_i < m \to [\exists y < mB(\overline{\alpha},y) \to (\Theta(\overline{\alpha}) < m \wedge B(\overline{\alpha},\Theta(\overline{\alpha})))]]\!] \ = \ 1_{\mathcal{B}}$$

is called a Skolem function for $\exists yB(\overline{x},y)$ on m.

The phrase "Skolem function on m" allows us to abbreviate the formula; otherwise it would have to be written in a longer form as:

$$\bigwedge_{i\le k} x_i < m \ \to \ \exists y < mB(\overline{x},y)\ .$$

Definition 9.2.2 *A structure $K(F,G)$ is Skolem closed iff for all open first-order $L^2_n(F,G)$-formulas $B(\overline{x},y)$ and for all $m \in \mathcal{M}_n$ there is a $\Theta \in G$ that is a Skolem function for $\exists yB(\overline{x},y)$ on m.*

The following theorem is a straightforward analogue to the standard derivation of quantifier elimination from the existence of Skolem functions, see also Chapter 4.

Theorem 9.2.3 *Assume that an L^2_n-structure $K(F,G)$ is Skolem closed. Then any Σ^b_∞ $L^2_n(F,G)$-formula admits a bounded quantifier elimination in $K(F,G)$.*

9.3 Comprehension and induction for $\Sigma^{1,b}_0$-formulas

Here we just summarize for later applications the general reduction of general comprehension and induction to the open case via Skolemization.

Theorem 9.3.1 *Let $K(F,G)$ be an L^2_n-structure. Assume that:*

1. *Open comprehension is valid in $K(F,G)$.*
2. *Open induction is valid in $K(F,G)$.*
3. *$K(F,G)$ admits bounded quantifier elimination for all Σ^b_∞ $L^2_n(F,G)$-formulas.*

Then comprehension and induction for all $\Sigma^{1,b}_0$ $L^2_n(F,G)$-formulas is valid in $K(F,G)$. In particular, theory V^0_1 is valid in $K(F,G)$.

The conclusion holds also if condition 3. is replaced by 3'.:

3'. *$K(F,G)$ is Skolem closed.*

Proof :

Assume we have a $\Sigma^{1,b}_0$ $L^2_n(F,G)$-formula for which we want to verify, say, induction axiom up to $m \in \mathcal{M}_n$. The free variables of the formula except the induction variable are universally quantified in front of the axiom. Hence we substitute for these parameters any elements of F or G respectively, and verify the axiom for all theses instances.

Note that after the substitution we get a bounded first-order $L^2_n(F,G)$-formula $B(x)$. By the bounded quantifier elimination there is an open $L^2_n(F,G)$-formula $A(x)$ such that

$$[\![\forall x < m+1\ B(x) \equiv A(x)]\!] \ = \ 1_{\mathcal{B}}\ .$$

By the validity of open induction we also have

$$[\![\neg A(0) \ \vee \ A(m) \ \vee \ \exists x < m(A(x) \wedge \neg A(x+1))]\!] \ = \ 1_{\mathcal{B}} \ .$$

These two sentences (together with a few valid axioms about ordering) in predicate logic alone imply the induction for $B(x)$ on $[0, m]$, and the provability in predicate logic preserves the truth-value $1_{\mathcal{B}}$.

Comprehension axioms are verified analogously. This proves the first part of the theorem. The second part follows from Theorem 9.2.3.

q.e.d.

Chapter 10

Skolem functions, switching lemma, and the tree model

In this chapter we modify the rudimentary model based on shallow decision trees to a model using deep trees (we leave the adjective deep out), in order to get Skolem functions into family G and thus provide bounded quantifier elimination for $\Sigma_0^{1,b}$-formulas (in the next chapter).

10.1 Switching lemma

We start by recalling the classical topic of partial random restrictions and the switching lemma. This has been invented by Furst, Saxe and Sipser [33] and Ajtai [1], and strengthened by Yao [107] and Hastad [39], and later generalized for the purpose of proof complexity in [78, 87] and elsewhere (see Chapter 21 or [88]).

The prototype form of the switching lemma is the one due to Hastad [39]. We state it in a form useful for our purposes.

Definition 10.1.1 *Let $X \subseteq [n]$ be arbitrary. For such X let D_X be the set of all partial truth assignments $\rho : X \to \{0,1\}$ with domain $dom(\rho) = X$. For $\rho \in D_X$ define*

$$E_\rho \ := \ \{\omega : [n] \to \{0,1\} \mid \rho \subseteq \omega\} \ .$$

Note that if we identify the set of $\omega : [n] \to \{0,1\}$ with $\{0,1\}^n$ (i.e. with the sample space of the rudimentary model), E_ρ defines a subset of the sample space.

Theorem 10.1.2 (Hastad's switching lemma [39])
Let $1 \leq k, \ell$ and $0 \leq m < n$ be arbitrary parameters. Suppose propositional formula $\varphi(x_1, \ldots, x_n)$ is in a k-DNF form, i.e. it is a disjunction of terms (conjunctions of literals) of size at most k.

Pick a subset $X \subseteq [n]$ uniformly at random from all subsets of $[n]$ of size m and then pick uniformly at random a partial restriction $\rho \in D_X$.

The probability that the restricted formula $\varphi \downarrow \rho$ cannot be computed by a decision tree (over variables x_i with $i \notin X$) of depth at most $k \cdot \ell$ is bounded above by:

$$\left(\frac{10k(n-m)}{n} \right)^{\ell} .$$

As pointed out in Section 9.2, Theorem 3.5.2 already provides a form of quantifier elimination. In particular, for any formula of the form $\exists y < mB(\alpha, y)$ there is $\beta \in F_{rud}$ such that

$$[\![\exists y < mB(\alpha, y)]\!] = [\![\beta < m \wedge B(\alpha, \beta)]\!] .$$

But the method of the proof of Theorem 3.5.2, utilizing the ccc of $\mathcal{B}$ and the compactness of F_{rud}, does not give information about the dependence of β on α and, in particular, does not yield a Skolem function $\Theta \in G_{rud}$ that would satisfy

$$[\![\exists y < mB(\alpha, y)]\!] = [\![\Theta(\alpha) < m \wedge B(\alpha, \Theta(\alpha))]\!]$$

for all $\alpha \in F_{rud}$.

We will now describe an idea how the switching lemma can be used to construct such a Skolem function. The actual implementation of the idea requires a modification of the sample space as well as of the families F_{rud} and G_{rud}. Thus we describe the idea only informally here and develop it formally in Chapter 11, after describing the new structure $K(F_{tree}, G_{tree})$ in Section 10.2.

Take an open $L^2_n(F_{rud}, G_{rud})$-formula $B(x, y)$ and some bound $m \in \mathcal{M}_n$. Let $\alpha, \beta \in F_{rud}$ and assume:

$$[\![\exists y < mB(\alpha, y)]\!] = [\![\beta < m \wedge B(\alpha, \beta)]\!] .$$

Now let us compute using the $\langle\!\langle \ldots \rangle\!\rangle$ notation.

$$\langle\!\langle \beta < m \wedge B(\alpha, \beta) \rangle\!\rangle = \bigcup_{j<m} \langle\!\langle \beta = j \wedge B(\alpha, \beta) \rangle\!\rangle =$$

$$\bigcup_{j<m} \langle\langle \beta = j \wedge B(\alpha, j) \rangle\rangle \subseteq \bigcup_{j<m} \langle\langle B(\alpha, j) \rangle\rangle \ .$$

Here $j < m$ range over elements of $\mathcal{M}$, not F_{rud}. But note that the last expression satisfies:

$$\langle\langle \exists y < m B(\alpha, y) \rangle\rangle = \bigcup_{j<m} \langle\langle B(\alpha, j) \rangle\rangle \ .$$

Hence a way to find β witnessing the quantifier is to find a decision tree that on input ω finds some $j < m$ such that $\omega \in \langle\langle B(\alpha, j) \rangle\rangle$, if it exists. But if we could decide for any $1 \leq u < v < m$ by a tree from F_{rud} of depth d independent of u, v any disjunction

$$\bigvee_{u \leq j \leq v} \omega \in \langle\langle B(\alpha, j) \rangle\rangle$$

we could use a binary search to create a depth $d \cdot \log(m)$ tree finding some j.

By Lemma 8.2.1 formula $\omega \in \langle\langle B(\alpha, j) \rangle\rangle$ is computed by a tree β_j from F_{rud} of depth $n^{1/t}$ independent of j. Thus the statement $\omega \in \langle\langle B(\alpha, j) \rangle\rangle$ is equivalent to a disjunction of terms of size $n^{1/t}$: the terms correspond to paths in the tree β_j that end in a leaf labeled 1 (i.e. yes). Consequently, each disjunction

$$\bigvee_{u \leq j \leq v} \omega \in \langle\langle B(\alpha, j) \rangle\rangle$$

is equivalent to a k-DNF, with $k := n^{1/t}$. It is this place where the switching lemma is applied and provides us with a decision tree of depth $n^{2/t}$ computing the validity of the disjunction.

To construct the witness from α by a Skolem function needs some additional tricks that we will postpone until Chapter 11. There are two conceptual problems caused by the switching lemma with regard to the rudimentary shallow tree model $K(F_{rud}, G_{rud})$. The first issue is that the switching lemma guarantees the existence of the wanted decision tree only after a partial restriction on a fairly large domain (we will need $m \geq n - n^{1-\epsilon}$). Consequently, if we wanted to interpret the construction on a subset of the sample space $\{0,1\}^n$ satisfying the restriction, this subset would have an infinitesimal counting measure in Ω and hence contribute nothing to the truth-values in question.

The second issue is that albeit the probability of the restriction in the switching lemma to fail is very small, it is non-zero and hence one restriction

will not work for all formulas at once. In other words, we could construct the Skolem function in this way only for some set of formulas, not for all.

It turns out that both issues can be successfully addressed by defining a different sample space in the next section.

10.2 Tree model $K(F_{tree}, G_{tree})$

This section is devoted to the definition of the new sample space Ω_{tree} and the new structure $K(F_{tree}, G_{tree})$. The definition of Ω_{tree} may look at first a bit unfriendly but this is compensated for later by the smooth way it works.

For the definition we use as an auxiliary parameter a function assigning to any $k \in \mathcal{M}$ an $\mathcal{M}$-rational δ_k by the formula:

$$\delta_k := \frac{1 + 2^{k+1}}{2(1 + 2^k)} .$$

The only two properties of this function we actually use are:

(i) $0 < \delta_k < 1$, for all $k \geq 0$.

(ii) $\delta_0 \cdot \delta_1 \cdot \ldots \cdot \delta_k > \frac{1}{2}$, for all $k \geq 0$.

Any function with these two properties would serve equally well.

Definition 10.2.1 *For $k \geq 0$ define inductively a* level k tree*, or briefly a k-tree, as follows:*

- *A 0-tree is any decision tree T such that there is $X \subseteq [n]$ and it holds:*
  - *$|X| = n - n^{\delta_0}$ (rounded to the nearest integer).*
  - *T on each path queries, in the natural order on $[n]$, the membership of all elements of X (in input ω).*

  Set X is called the support *of each leaf in T.*

- *A $(k+1)$-tree is any tree T for which there exists an k-tree S such that the following holds.*

  If $X_\ell \subseteq [n]$ are the supports of leafs ℓ of S then there are subsets $Y_\ell \subseteq [n] \setminus X_\ell$ such that

- $|Y_\ell| = (n - |X_\ell|) - (n - |X_\ell|)^{\delta_{k+1}}$.
- *At each leaf ℓ T extends S by querying on each path, in the natural order on $[n]$, the membership of all elements of Y_ℓ.*

The support of any new leaf ℓ' in T that is below an old leaf ℓ of S is $X_\ell \cup Y_\ell$.

Let $Tree_k$ denotes the set of all k-trees.

A partial ordering $\preceq$ on these trees is defined as follows: for a k-tree T and an ℓ-tree it holds $T \preceq S$ iff T the initial part of S.

Note that if T and S are both k-trees then $T \preceq S$ iff $T = S$, i.e. different trees of the same level are $\preceq$-incomparable.

Lemma 10.2.2 *For any $k \geq 0$ (in $\mathcal{M}$) the depth of a k-tree is less than $n - n^{1/2}$.*

Proof :

The depth d_k of a k-tree is $n - n^s$ where $s = \delta_0 \cdot \delta_1 \cdot \ldots \cdot \delta_k = \frac{1+2^{k+1}}{2^{k+2}} > \frac{1}{2}$.

q.e.d.

For the next definition we need a fixed non-standard number h as a parameter. We could select h in some canonical way, e.g. $h = n$. But taking a new symbol h for it seems to stress more that its role is auxiliary and its value is, as long as it is non-standard, irrelevant. We will not refer to h in the forthcoming notation.

Definition 10.2.3 *The sample space Ω_{tree} is*

$$\bigcup_{T \in Tree_h} \{T\} \times \{0,1\}^n \ .$$

Now comes the key definition of the family of random variables F_{tree}. Recall from the proof of Lemma 10.2.2 that the depth d_k of k-trees is

$$d_k \ := \ n \ - \ n^{\frac{1+2^{k+1}}{2^{k+2}}} \ < \ n \ - \ n^{\frac{1}{2}} \ .$$

Definition 10.2.4 *Family F_{tree} consists of all functions $\alpha \in \mathcal{M}$,*

$$\alpha : \Omega_{tree} \to \mathcal{M}_n$$

for which there exists $k \in \mathbf{N}$, $t > \mathbf{N}$ and a tuple $(S_Z)_{Z \in Tree_k} \in \mathcal{M}$ of trees such that

- *Each S_Z extends Z by adding to leafs of Z some subtrees.*
- *The depth of all S_Z in the tuple is bounded by $d_k + n^{1/t}$.*
- *The value of α on sample $(T, \omega) \in \Omega_{tree}$ is computed by S_Z at ω, for the unique $Z \preceq T$.*

The parameter k is called the level of α.

Note, in particular, that any constant from $\mathcal{M}_n$ is represented by any tuple of trees whose all leafs (of all trees) are labeled by the constant.

In this definition we quantify over standard k and non-standard t, and hence the family F_{tree} is not definable in $\mathcal{M}$. While the definability of the family of random variables is not required by the method, the definability of the sample space is. For this reason we used the auxiliary parameter h and the set $Tree_h$ in place of undefinable $\bigcup_{k \in \mathbf{N}} Tree_k$.

We need a simple technical observation.

Lemma 10.2.5 *Let $m \in \mathcal{M}_n$ and let $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1}) \in \mathcal{M}$ be an m-tuple of elements of F_{tree}. For any $\alpha \in F_{tree}$ define function $\hat{\beta}(\alpha) : \Omega_{tree} \to \mathcal{M}_n$ by*

$$\hat{\beta}(\alpha)((T, \omega)) = \begin{cases} \beta_{\alpha((T,\omega))}((T, \omega)) & \text{if } \alpha((T, \omega)) < m \\ 0 & \text{otherwise.} \end{cases}$$

Then this function is in F_{tree} too.

Proof :

The proof is analogous to the proof of Lemma 7.2.2. First note that because the tuple of β_i's is in $\mathcal{M}$ there must be a common upper bound $k \in \mathbf{N}$ on their level. Hence we may assume without a loss of generality that α and all β_i are of the same level (the level can be always artificially increased by extending the trees Z in all possible ways).

Then note that the common parts $Z \in Tree_k$ of the trees computing random variables α and β_i's on $\{T\} \times \{0,1\}^n$, $Z \preceq T$, can be shared. That is, if we use a tree S_Z to compute that the value of α on a sample is i and then we want to further compute the value of β_i at the sample, we need not to go through the Z again. We just use the part of tree computing β_i below Z (it has a depth n raised to an infinitesimal).

q.e.d.

The lemma allows us to define family G_{tree} in the same way as was G_{rud} defined from F_{rud}.

Definition 10.2.6 *Let $m \in \mathcal{M}_n$ and let $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1}) \in \mathcal{M}$ be an m-tuple of elements of F_{tree}.*

For any $\alpha \in F_{tree}$ define $\hat{\beta}(\alpha) : \Omega_{tree} \to \mathcal{M}_n$ by

$$\hat{\beta}(\alpha)((T,\omega)) = \begin{cases} \beta_{\alpha((T,\omega))}((T,\omega)) & \text{if } \alpha((T,\omega)) < m \\ 0 & \text{otherwise.} \end{cases}$$

Family G_{tree} consists of all random variables Θ computed by some $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1})$, where $m \in \mathcal{M}_n$ and $\hat{\beta} \in \mathcal{M}$ is a m-tuple of elements of F_{tree}.

We conclude this section with a technical lemma.

Lemma 10.2.7 (1) *Let $A(x)$ be an open $L^2_n(F_{tree}, G_{tree})$-formula with the only free variable x. Then there are $k \in \mathbf{N}$ and $t > \mathbf{N}$ such that for all $i \in \mathcal{M}_n$ the membership in $\langle\langle A(i) \rangle\rangle$ is computed by a 0/1-valued element of F_{tree} having level k and the depth bounded above by $d_k + n^{1/t}$.*

(2) *Let $s > \mathbf{N}$ and $w := n^{1/s}$. Assume $(\alpha_1, \ldots, \alpha_w) \in \mathcal{M}$ is an w-tuple of elements of F_{tree}. Then there is $\beta \in F_{tree}$ computing simultaneously all α_i in the tuple (i.e. it computes the w-tuple of the values).*

(3) *Families F_{tree} and G_{tree} are closed under definition by cases by open $L^2_n(F_{tree}, G_{tree})$-formulas.*

Proof :

Part (1) is proved, via Lemma 10.2.5, analogously as Lemma 8.2.1. For Part (2) we may assume, as in the proof of Lemma 10.2.5, that the level of all α_i is the same, some $k \in \mathbf{N}$.

Each of the w functions α_i is computed by a tuple of trees $(S^i_Z)_Z$, $Z \in Tree_k$, with the properties as in Definition 10.2.4. As the tuple is an element of $\mathcal{M}$ there is one $t > \mathbf{N}$ such the the depth of all trees, any Z and i, is bounded above by $d_k + n^{1/t}$.

For any one $Z \in Tree_k$ one can compute the values of all α_i on the subset of the sample space $\bigcup_{T:Z \preceq T}\{T\} \times \{0,1\}^n \subseteq \Omega_{tree}$ by a tree S_Z constructed as follows: first compute as in Z and then successively as in the subtrees of S^i_Z starting at level d_k, for $i = 1, \ldots, w$. The depth of this tree is bounded above by $d_k + w \cdot n^{1/t}$. The term $w \cdot n^{1/t}$ is bounded by n raised to an infinitesimal, hence S_Z obeys the condition required in Definition 10.2.4.

Part (3) is now obvious.

q.e.d.

Chapter 11

Quantifier elimination in $K(F_{tree}, G_{tree})$

Our first aim in this chapter is to show that structure $K(F_{tree}, G_{tree})$ is Skolem closed and hence admits bounded quantifier elimination. We then observe that the argument yielding the open comprehension and open induction in $K(F_{rud}, G_{rud})$ works here too and hence we will get comprehension and induction for $\Sigma_0^{1,b}$-formulas, and hence theory V_1^0, valid in $K(F_{tree}, G_{tree})$ by Theorem 9.3.1.

11.1 Skolem functions

Theorem 11.1.1 *Structure $K(F_{tree}, G_{tree})$ is Skolem closed.*

Proof :

We need to show that for any open first-order $L_n^2(F_{tree}, G_{tree})$-formula $B(\overline{x}, y)$ and for any $m \in \mathcal{M}_n$ there is a $\Theta \in G_{tree}$ that is a Skolem function for $\exists y B(\overline{x}, y)$ on m.

Assume without a loss of generality (we have a pairing function) that $\overline{x}$ is just one variable x. Assume we have an m-tuple $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1}) \in \mathcal{M}$ of elements of F_{tree} such that for all $i < m$:

$$[\![\beta_i < m \wedge B(i, \beta_i)]\!] = [\![\exists y < m B(i, y)]\!] \ .$$

Then we would like to take for the Skolem function the $\Theta \in G_{tree}$ computed by $\hat{\beta}$. Of course, $\Theta(i) = \beta_i$ and so Θ behaves as a Skolem function for

$x = 0, \ldots, m-1$ but that is not enough. For example, it may happen that there is another m-tuple $\hat{\gamma} = (\gamma_0, \ldots, \gamma_{m-1}) \in \mathcal{M}$ of elements of F_{tree} behaving as $\hat{\beta}$ for all $i < m$:

$$[\![\beta_i < m \wedge B(i, \beta_i)]\!] = [\![\gamma_i < m \wedge B(i, \gamma_i)]\!]$$

but

$$\langle\!\langle \gamma_i < m \wedge B(i, \gamma_i) \rangle\!\rangle \setminus \langle\!\langle \beta_i < m \wedge B(i, \beta_i) \rangle\!\rangle$$

are non-empty. The only information we have about such difference sets is that their individual counting measures in Ω_{tree} are infinitesimal. But it may happen that there is an $\alpha \in F_{tree}$ with $[\![\alpha < m]\!] = 1_{\mathcal{B}}$ defining a distribution on $i < m$ so that the errors of individual β_i's combine into a set with standard positive probability. That is, with a positive probability

$$\alpha(\omega) = i \ \wedge \ i \notin \langle\!\langle \beta_i < m \wedge B(i, \beta_i) \rangle\!\rangle \ .$$

By a careful choice of $\hat{\beta}$ we need to exclude this possibility.

For a $\hat{\beta}$ of the form as above define:

$$E_i \ := \ \langle\!\langle \exists y < m B(i, y) \rangle\!\rangle \setminus \langle\!\langle \beta_i < m \wedge B(i, \beta_i) \rangle\!\rangle \ .$$

Claim 1: *Assume that the counting measure (in Ω_{tree}) of $\bigcup_{i<m} E_i \in \mathcal{A}$ is infinitesimal. Then $\Theta \in G_{tree}$ computed by $\hat{\beta}$ is a Skolem function for $\exists y < m B(x, y)$ on m.*

Assume for the sake of contradiction that for some $\alpha \in F_{tree}$:

$$[\![\Theta(\alpha) < m \wedge B(\alpha, \Theta(\alpha))]\!] < [\![\exists y < m B(\alpha, y)]\!] \ .$$

Applying Part (3) of Lemma 10.2.7 and Lemma 3.3.2 we get $\gamma \in F$ such that

$$[\![\Theta(\alpha) < m \wedge B(\alpha, \Theta(\alpha))]\!] < [\![\gamma < m \wedge B(\alpha, \gamma)]\!] \ .$$

Therefore

$$U \ := \ \langle\!\langle \gamma < m \wedge B(\alpha, \gamma) \rangle\!\rangle \setminus \langle\!\langle \Theta(\alpha) < m \wedge B(\alpha, \Theta(\alpha)) \rangle\!\rangle$$

has a non-infinitesimal counting measure in Ω_{tree}. But clearly

$$U \subseteq \bigcup_{i<m} \{(T, \omega) \in \Omega_{tree} \mid \alpha((T, \omega)) = i\} \cap E_i \subseteq \bigcup_{i<m} E_i$$

which is a contradiction. This proves the claim.

So we want to find $\hat{\beta}$ such that the error sets E_i satisfy the hypothesis of Claim 1. For this we need to understand how is the membership in $\langle\langle \exists y < mB(i,y) \rangle\rangle$ computed. We have:

$$\langle\langle \exists y < mB(i,y) \rangle\rangle \;=\; \bigcup_{j<m} \langle\langle B(i,j) \rangle\rangle \;.$$

By Part (1) of Lemma 10.2.7 the membership in sets $\langle\langle B(i,j) \rangle\rangle$ are computed by 0/1-valued functions $\xi_{i,j}$ from F_{tree} such that all $\xi_{i,j}$ have the same standard level k and depth bounded above by $d_k + n^{1/t}$, some common non-standard t .

Claim 2: *For any $i < m$, $0 \leq u < v < m$ and $s >$ **N** there are elements $\kappa_{i,u,v} \in F_{tree}$ of level $k+1$ depth $d_{k+1} + n^{\frac{t+s}{ts}}$, such that:*

$$Prob_{(T,\omega)\in\Omega_{tree}}[\; ((T,\omega) \in \bigcup_{u\leq j\leq v} \langle\langle B(i,j) \rangle\rangle) \;\not\equiv\; (\kappa_{i,u,v}((T,\omega)) = 1) \;]$$

$$< \; (\frac{10n^{1/t}(n-d_{k+1})}{n-d_k}\;)^{n^{1/s}} \;<\; (\frac{10n^{1/t}n^{\frac{1+2^{k+2}}{2^{k+3}}}}{n^{\frac{1+2^{k+1}}{2^{k+2}}}}\;)^{n^{1/s}} \;<\; 2^{-n^{1/s}} \;.$$

We shall use Theorem 10.1.2 to prove this key claim. Fix parameters i, u, v, s satisfying the restrictions of the claim. As i, u, v are fixed we ease the notation and do not reflect these parameters in the names of the forthcoming objects (trees and formulas). Also put $a := n^{1/t}$and $b := n^{1/s}$.

Let $(S^j_Z)_Z$, for $u \leq j \leq v$ and $Z \in Tree_k$, be the trees computing $\xi_{i,j}$. Hence each S^j_Z extends Z by attaching some trees of depth at most a to its leafs. Let $\mu^j_{Z,\ell}$ be the depth at most a tree attached to a leaf ℓ of Z in S^j_Z, and let $\psi^j_{Z,\ell}$ be the disjunction of all terms corresponding to paths in $\mu^j_{Z,\ell}$ ending with label 1 (for yes). Note that each $\psi^j_{Z,\ell}$ is an a-DNF formula.

The element $\kappa_{i,u,v} \in F_{tree}$ will be computed by trees $(R_W)_{W\in Tree_{k+1}}$. Tree R_W will extend the $(k+1)$-tree W by attaching certain trees $\tau_{W,\ell'}$ to its leafs ℓ', each $\tau_{W,\ell'}$ of the depth at most $a \cdot b = n^{\frac{t+s}{ts}}$.

We are now going to argue, using Theorem 10.1.2, that there exists such trees $\tau_{W,\ell'}$ such that the element $\kappa_{i,u,v}$ defined by the resulting trees $(R_W)_W$ satisfies the requirement of the claim.

Fix $Z \in Tree_k$ and its leaf ℓ. Let X_ℓ be the support of ℓ in Z, so $|X_\ell| = d_k$. Consider propositional formula

$$\varphi_{Z,\ell} := \bigvee_{u \le j \le v} \psi^j_{Z,\ell} \ .$$

It is an a-DNF formula, as $\psi^j_{Z,\ell}$ are. It has $n - d_k$ variables corresponding to elements of $[n] \setminus X_\ell$. By its definition the formula has the following property. Let $\Gamma_{Z,\ell} \subseteq \Omega_{tree}$ be the set of all $(T,\omega) \in \Omega_{tree}$ such that $Z \preceq T$ and ω leads in Z to the leaf ℓ. Then for $(T,\omega) \in \Gamma_{Z,\ell}$:

$$((T,\omega) \in \bigcup_{u \le j \le v} \langle\langle B(i,j) \rangle\rangle) \ \text{ iff } \ (\omega \text{ satisfies } \varphi_{Z,\ell}) \ .$$

For leafs ℓ of Z and sets $Y_\ell \subseteq [n] \setminus X_\ell$ of size $(n - d_k)^{\delta_{k+1}}$, all possible truth assignments on Y_ℓ are in a bijective correspondence with leafs ℓ' of a $(k+1)$-tree W extending Z by querying elements of Y_ℓ.

By Theorem 10.1.2 if we take a random such set $Y_\ell \subseteq [n] \setminus X_\ell$ and a random truth assignment $\sigma_{\ell'}$ (corresponding to leaf ℓ' in W) on Y_ℓ then we can find a decision tree $\tau_{W,\ell'}$ that

- is equivalent to $\varphi_{Z,\ell}$ for all ω that lead to ℓ' in W, and
- has the depth at most $a \cdot b$

with the probability of failing to do so at most

$$(\frac{10a(n - d_{k+1})}{n - d_k} \ )^b$$

(the quantity required in the claim). The claim now follows by noting that:

1. Sets $\Gamma_{Z,\ell}$, $Z \in Tree_k$ and ℓ a leaf in Z, partition the sample space Ω_{tree}, and
2. sets $\Gamma_{W,\ell'}$, $W \in Tree_{k+1}$ such that $Z \preceq W$ and ℓ' a leaf in W below ℓ, partition sets $\Gamma_{Z,\ell}$.

By the estimate above suitable trees $\tau_{W,\ell'}$ exists for all but a fraction of at most $(\frac{10a(n-d_{k+1})}{n-d_k} \ )^b$ pairs (W,ℓ') considered in part 2., and hence by 1. this also estimates the fraction of all pairs (W,ℓ'), $W \in Tree_{k+1}$ and leaf ℓ' in W this happens. This proves the claim.

To define elements β_i finding a j such that $(T,\omega) \in \langle\langle B(i,j)\rangle\rangle$ if it exists (with only a small probability of failing) we simulating the binary search, computing functions $\kappa_{i,u,v}$. This requires to compute $\log(m)$ of such functions on a path in the search, and the whole binary search can be performed by an element β_i of F_{tree} (as in Lemma 10.2.7, Part.2). The depth of the trees computing β_i is bounded above by $d_{k+1} + \log(m) \cdot a \cdot b$ and that obeys the restrictions for a $(k+1)$-level element of F_{tree}.

By Claim 2 each $\kappa_{i,u,v}$ makes an error with probability less than $2^{-n^{1/s}}$. There is at most m such pairs $u \leq v$ used in a binary search and there are m different instances i. Hence the counting measure of the error set $\bigcup_{i<m} E_i$ for $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1})$ defined in this way is less than

$$2^{-n^{1/s}} \cdot m^2 \ .$$

By taking s non-standard but small enough this quantity can be made infinitesimal.

q.e.d.

The theorem yields, via Theorem 9.2.3, the bounded quantifier elimination.

Corollary 11.1.2 *Model $K(F_{tree}, G_{tree})$ admits bounded quantifier elimination.*

We shall note for a future reference one more corollary of the construction.

Corollary 11.1.3 *For any Σ^b_∞-formula $A(x_1, \ldots, x_k)$, possibly with parameters from F_{tree} or G_{tree}, and for any $m \in \mathcal{M}_n$ and arbitrarily small but non-standard $s > \mathbf{N}$ there is an open $L^2_n(F_{tree}, G_{tree})$-formula $B(x_1, \ldots, x_k)$ such that*

$$[\![\forall x_1, \ldots, x_k < m \ (A(x_1, \ldots, x_k) \ \equiv \ B(x_1, \ldots, x_k))]\!] \ = \ 1_{\mathcal{B}}$$

and such that for any $\alpha_1, \ldots, \alpha_k \in F_{tree}$ the counting measure of the symmetric difference

$$\langle\langle \bigwedge_i \alpha_i < m \to A(\alpha_1, \ldots, \alpha_k)\rangle\rangle \ \triangle \ \langle\langle \bigwedge_i \alpha_i < m \to B(\alpha_1, \ldots, \alpha_k)\rangle\rangle$$

is less than $2^{-n^{1/s}}$. In particular, it is infinitesimal.

11.2 Comprehension and induction for $\Sigma_0^{1,b}$-formulas

Lemma 11.2.1 *Open comprehension and open induction are valid in* $K(F_{tree}, G_{tree})$.

Proof :

The proof is identical to the proof of Corollary 8.2.3 and Lemma 8.3.2 (via Lemma 8.3.1), with Lemma 10.2.7 replacing Lemma 8.2.1, and noting (again using Lemma 10.2.7) that the witness α constructed as in the proof of Lemma 8.3.1 is in F_{tree}.

q.e.d.

Lemma 11.2.1 and Corollary 11.1.2 imply via Theorem 9.3.1 the following statement.

Theorem 11.2.2 *Comprehension and induction for* $L_n^2(F_{tree}, G_{tree})$ $\Sigma_0^{1,b}$ *formulas, is valid in* $K(F_{tree}, G_{tree})$. *In particular, theory* V_1^0 *is valid in the structure.*

Chapter 12

Witnessing, independence and definability in V_1^0

A proof-theoretic approach to witnessing theorems for V_1^0 is to Skolemize the theory first via AC^0 maps on strings and then use some form of Herbrand theorem (see also the remark at the end of Chapter 6). In this chapter we shall prove a witnessing theorem for V_1^0 by applying the general witnessing theorems from Chapter 3 to model $K(F_{tree}, G_{tree})$. The result corresponds to the one obtained by the classical approach combined with the switching lemma (Theorem 10.1.2).

Two prominent types of formulas $A(x)$ to consider witnessing theorems for are:

(a) $\forall X < x \exists Y < x \Sigma_0^{1,b}$,

(b) $\forall X < x \exists Y < x \forall Z < x \Sigma_0^{1,b}$.

We shall prove a witnessing theorem for the first type of formulas for V_1^0 in this chapter, and for the second type of formulas for theory $Q_2V_1^0$ in Chapter 16. It will be clear that the proofs work identically for both theories and both witnessing theorems apply equally well to V_1^0 and $Q_2V_1^0$. We use the witnessing theorems to give true statements of each form unprovable in the respective theory. More examples of witnessing theorems for stronger theories (PV, S_2^1, and T_2^1) are given in Part VII. All these examples show, we believe, that forcing with random variables provides an alternative framework for proving witnessing theorems.

There are three other types of formulas $A(x)$:

(c) $\forall X < x\Sigma^{1,b}_0$ (i.e. $s\Pi^{1,b}_1$)

(d) $\forall X < x\exists y < x\forall Z < x\Sigma^{1,b}_0$

(e) $\exists Y < x\forall Z < x\Sigma^{1,b}_0$

that are very interesting from the point of view of proof complexity. But independence for sentences of this form cannot be proved using $K(F_{tree}, G_{tree})$ because, as we shall show in Section 12.2, the model preserves all true $s\Pi^{1,b}_1$ sentences. To avoid such a preservation property is a key issue if one hopes to use models of bounded arithmetic for lengths-of-proofs lower bounds. We shall return to this topic in Part V.

In Section 12.3 we note that the $s\Pi^{1,b}_1$-elementarity, while bad for proof complexity, allows to prove a circuit lower bound for AC^0 circuits computing parity. This argument corresponds to the classical proof via the switching lemma.

12.1 Witnessing $\forall X < x\exists Y < x\Sigma^{1,b}_0$-formulas

Let $A(x)$ be a $\forall X < x\exists Y < x\Sigma^{1,b}_0$ formula of the form

$$\forall X < x\exists Y < xB(x, X, Y)$$

with B a $\Sigma^{1,b}_0$ formula. Families F_{tree} and G_{tree} are closed under definitions by open formulas (Lemma 10.2.7) and so we may apply Lemma 3.3.3 to deduce the following statement for $K(F_{tree}, G_{tree})$.

Lemma 12.1.1 *Assume $[\![A(n)]\!] = 1_{\mathcal{B}}$. Then for every $\Delta \in G_{tree}$ and every standard $\epsilon > 0$ there is $\Gamma \in G_{tree}$ such that*

$$\mu([\![\Delta < n \to (\Gamma < n \wedge B(n, \Delta, \Gamma))]\!]) \ > \ 1 - \epsilon \ .$$

Applying Corollary 11.1.2 to $B(n, \Delta, \Gamma)$ and then Lemma 2.2.1 we entail the next statement.

Lemma 12.1.2 *Assume $[\![A(n)]\!] = 1_{\mathcal{B}}$. Then for every $\Delta \in G_{tree}$ and every standard $\epsilon > 0$ there is $\Gamma \in G_{tree}$ such that*

$$Prob_{(T,\omega)\in\Omega_{tree}}[\ (T, \omega) \in \langle\langle \Delta < n \to (\Gamma < n \wedge B(n, \Delta, \Gamma))\rangle\rangle\ ] \ > \ 1 - \epsilon \ .$$

Applying the lemma to a specific random variable Δ will give us the witnessing theorem. Δ is simply the projection on the second coordinate:

$$\Delta((T,\omega)) \ := \ \omega \ .$$

If the samples were just ω's and not pairs (T,ω) this would be simply the identity map (that represents in a sense the uniform distribution on the sample space). It is easy to see that indeed $\Delta \in G$; it is given by a tuple $\hat{\beta} = (\beta_i)_{i<n}$ where β_i is a tree querying $\omega_i =_? 0$ and labeling the two resulting leafs by the respective answers 0 or 1 to the query. Also note that $\langle\langle \Delta < n \rangle\rangle = \Omega_{tree}$.

Theorem 12.1.3 *Assume $[\![A(n)]\!] = 1_{\mathcal{B}}$. Then for every standard $\epsilon > 0$ there is $\Gamma \in G_{tree}$ such that*

$$Prob_{(T,\omega)\in\Omega_{tree}}[\ \Gamma((t,\omega)) < n \wedge B(n,\omega,\Gamma((T,\omega)))\ ] \ > \ 1-\epsilon \ .$$

In particular, this holds if theory V^0_1 proves $\forall x A(x)$.

Using compactness, Lemma 10.2.2 and the fact that in the definition of cut $\mathcal{M}_n$ and of subsequent structures we have not used any property of n except that it is non-standard, we may formulate a finitary corollary of Theorem 12.1.3.

Corollary 12.1.4 *Assume that theory V^0_1 proves $\forall x A(x)$. Then for every standard $\epsilon > 0$ there is a collection $W = (W_i)_{i<m}$ of trees W_i of depth less than $m - m^{1/2}$ such that for all $m \in \mathbf{N}$ large enough:*

$$Prob_{\omega\in\{0,1\}^m}[\ B(m,\omega,W(\omega))\ ] \ > \ 1-\epsilon \ .$$

A specific formula $A(x)$ for which the witnessing theorem yields an unprovability from V^0_1 (and which is interesting in connection with theory $Q_2V^0_1$ defined in Chapter 13) expresses the existence of a set Y counting parity of the given set X (in a bit more general situation with a family of sets rather than one this will be called a 2-cover in Section 13.2). The formula is defined by taking for $B(x,X,Y)$ the formula to be denoted $Par(x,X,Y)$:

$$0 \notin Y \wedge \forall y < x(y+1 \in Y \equiv (y \in X \oplus y \in Y))$$

where $C \oplus D$ abbreviates $C \not\equiv D$. The intended meaning of the formula is that $y \in Y$ is true iff the number of $z < y$ such that $z \in X$ is odd.

It is now easy to see that either Theorem 12.1.3 or directly Corollary 12.1.4 entail the following unprovability result.

Theorem 12.1.5 *Sentence $\forall x \forall X < x \exists Y < x Par(x, X, Y)$ is true but unprovable in theory V^0_1.*

12.2 Preservation of true $s\Pi^{1,b}_1$-sentences

We are now going to show that true sentences $\forall x A(x)$ for formulas $A(x)$ of one of the forms (c), (d) or (e) listed in the introduction to Chapter 12 are valid in model $K(F_{tree}, G_{tree})$. The key case is (c), the other two are its corollaries.

Theorem 12.2.1 *Assume sentence $\forall x A(x)$ is true where $A(x)$ is $s\Pi^{1,b}_1$ formula. Then $\forall x A(x)$ is valid in $K(F_{tree}, G_{tree})$.*

Proof :

Assume $A(x)$ is of the form $\forall X < x B(x, X)$ where B is a $\Sigma^{1,b}_0$ formula. It suffices to show that

$$[\![\forall x < m B(x, \Gamma)]\!] \;=\; 1_{\mathcal{B}}$$

for any $m \in \mathcal{M}_n$ and $\Gamma \in G_{tree}$.

By Corollary 11.1.3 there is an open $L^2_n(F_{tree}, G_{tree})$-formula $C(x)$ such that

$$[\![\forall x < m \; (B(x, \Gamma) \;\equiv\; C(x))]\!] \;=\; 1_{\mathcal{B}}$$

and the counting measure of the symmetric difference

$$\langle\!\langle \alpha < m \to B(\alpha, \Gamma) \rangle\!\rangle \;\triangle\; \langle\!\langle \alpha < m \to C(\alpha) \rangle\!\rangle$$

is infinitesimal for any $\alpha \in F_{tree}$.

Because $\forall x A(x)$ is true, $\langle\!\langle \alpha < m \to B(\alpha, \Gamma) \rangle\!\rangle = \Omega_{tree}$. As C is open we may deduce that $[\![\alpha < m \to C(\alpha)]\!] = 1_{\mathcal{B}}$, for any α. This proves the theorem.

q.e.d.

We shall use the theorem to deduce the preservation for formulas of types (d) or (e) too. We state it in two separate corollaries as their proofs are different.

Corollary 12.2.2 *Assume sentence $\forall x A(x)$ is true where $A(x)$ is a formula of the form $\forall X < x \exists y < x \forall Z < x \Sigma^{1,b}_0$.*

Then $\forall x A(x)$ is valid in $K(F_{tree}, G_{tree})$.

Proof :

Let $A(x)$ be a formula of the form $\forall X < x \exists y < x \forall Z < x C(x, X, y, Z)$ where C is a $\Sigma_0^{1,b}$ formula. If $\forall x A(x)$ is true so is the formula $\forall x A^*(x)$ where $A^*(x)$ is

$$\forall X < x \forall Z^* < x^2 \exists y < x \; C^*(x, X, y, Z^*)$$

where we think of Z^* as of an array of sets $Z^*_y < x$, one for each $y < x$, and $C^*(x, X, y, Z^*)$ says that $C(x, X, y, Z^*_y)$ holds.

This is a formula of the form to which Theorem 12.2.1 applies, so $\forall x A^*(x)$ is valid in $K(F_{tree}, G_{tree})$. The validity of $\forall x A(x)$ follows noting that the bounded collection scheme is valid in the model. We state only the claim we need.

Claim: *The sentence*

$$\forall x, X \; [\; (\forall y < x \exists Z < x \neg C(x, X, y, Z)) \rightarrow (\exists Z^* < x^2 \forall y < x \neg C^*(x, X, y, Z^*)) \;]$$

is valid in $K(F_{tree}, G_{tree})$.

The claim is proved using the collection scheme in $\mathcal{M}$.

q.e.d.

Corollary 12.2.3 *Assume sentence* $\forall x A(x)$ *is true where* $A(x)$ *is a formula of the form* $\exists Y < x \forall Z < x \Sigma_0^{1,b}$.

Then $\forall x A(x)$ *is valid in* $K(F_{tree}, G_{tree})$.

Proof :

Let $A(x)$ be a formula of the form $\exists Y < x \forall Z < x \; D(x, Y, Z)$ where D is a $\Sigma_0^{1,b}$ formula. If $\forall x A(x)$ is true then there exists a binary relation R on $\mathbf{N}$ (i.e. we have a symbol for it in L_n) such that for each $u \in \mathcal{M}$ set R_u witnesses the truth of $\exists Y < u \forall Z D(u, Y, Z)$. Here we again think of R as encoding an (infinite, this time) array of sets. In other words, sentence

$$\forall x (R_x < x \wedge \forall Z < x D(x, R_x, Z))$$

is true. But this sentence is $s\Pi_0^{1,b}$ and hence by Theorem 12.2.1 it is valid in $K(F_{tree}, G_{tree})$.

Clearly, the validity of this sentence implies (in predicate logic only) the validity of $\forall x A(x)$.

q.e.d.

12.3 Circuit lower bound for parity

In Boolean complexity theory one of the well-known results states an exponential lower bound on constant-depth circuits computing parity. It was proved gradually by Ajtai[1], Furst et.al.[33], Yao[107] and Hastad[39].

Theorem 12.3.1 *For every $d \geq 1$ there is an $\epsilon > 0$ such that for $k >> 0$, any depth d circuit computing parity of k bits must have the size at least 2^{k^ϵ}.*

We shall observe in this section that the lower bound can be proved by using structure $K(F_{tree}, G_{tree})$. Of course, this is not surprising because the heart of Theorem 12.3.1 is the switching lemma (Theorem 10.1.2) which plays also a key role in the analysis of $K(F_{tree}, G_{tree})$. However, it is not uninteresting to see that the $s\Pi^{1,b}_1$-elementarity of $K(F_{tree}, G_{tree})$ over the standard model, which renders the structure useless for lengths of proofs lower bounds, allows on the other hand to prove a circuit lower bound.

Basic facts from descriptive complexity theory imply that a set $\mathcal{U}$ of strings (i.e. of bounded sets) is definable by constant depth circuits of a subexponential size iff there is an L_n $\Sigma^{1,b}_0$-formula $A(x, X)$ (with no parameters) such that for all $m \in \mathbf{N}$ and $V \subseteq [m]$ it holds:

$$V \in \mathcal{U} \cap \{0,1\}^m \text{ iff } A(m, V) \textit{ is true.}$$

Take for $\mathcal{U}$ the set of sets of odd cardinality and assume for the sake of contradiction that there is a formula $A(x, X)$ as above. If that is so then it holds

$$\forall x \forall X < x \forall y < x \; [\; A(x, X) \not\equiv A(x, X \oplus \{y\}) \;]$$

where $X \oplus \{y\}$ is an abbreviation for the set resulting from X by adding y if $y \notin X$ or by removing it if $y \in X$. This is a $s\Pi^{1,b}_1$ sentence and thus by Theorem 12.2.1 we may conclude that it is also valid in $K(F_{tree}, G_{tree})$. We bring the assumption to a contradiction by showing that, in fact, for some $\alpha \in F_{tree}$

$$(*) \qquad [\![\alpha < n \wedge (\; A(n, \Delta) \equiv A(n, \Delta \oplus \{\alpha\}) \;)]\!] \;=\; 1_{\mathcal{B}} \; .$$

Here $\Delta \in G_{tree}$ is the random variable considered in Theorem 12.2.1, the projection on the second coordinate.

By Corollary 11.1.3 there is $\gamma \in F_{tree}$ such that $[\![\gamma = 1]\!] = [\![A(n, \Delta)]\!]$. Using this element define α by extending the tree S defining γ as follows. Let ℓ be a leaf of S and let $i_\ell \in [n]$ is the first element of $[n]$ such that $\omega_i =_? 0$ is not queried on the path to ℓ. Such an i exists as the depth of S is less than n. The tree computing α is the tree S with all ℓ labeled by i_ℓ.

It is not difficult to see that with this α the equality $(*)$ holds.

Part IV

$AC^0(2)$ world

Chapter 13

Theory $Q_2V_1^0$

13.1 Q_2-quantifier and theory $Q_2V_1^0$

Modular counting quantifiers were considered first in bounded arithmetic by Paris and Wilkie [85]. We shall consider just one of them, the parity quantifier.

The new quantifier will be denoted Q_2. It can be used only as a bounded quantifier: if A is a formula then we can form a new formula $Q_2y < t(\overline{x})A$. The intended meaning is: $Q_2y < t(\overline{x})A$ is true iff there is an odd number of $y < t(\overline{x})$ for which A is true.

The axioms governing the behavior of Q_2 are:

1. $\neg Q_2y < 0A(y)$.
2. $(Q_2y < tA(y) \wedge A(t)) \ \rightarrow \ \neg \ Q_2y < (t+1)A(y)$.
3. $(Q_2y < tA(y) \wedge \neg A(t)) \ \rightarrow \ Q_2y < (t+1)A(y)$.

We shall denote by $Q_2\Sigma_0^{1,b}$, $Q_2\Sigma_\infty^b$ and similar the corresponding classes of bounded formulas allowing also the Q_2 quantifier. In particular, $Q_2\Sigma_0^{1,b}$ is the class of bounded L^2-formulas without second-order quantifiers while $Q_2\Sigma_\infty^b$ is the class of first-order bounded formulas.

Theory $Q_2V_1^0$ is defined quite analogously to V_1^0. The axioms of $Q_2V_1^0$ are:

- Axioms of PA^- from Section 6.

- Axioms 1.-3. above for the Q_2-quantifier.
- Comprehension axioms for all $Q_2\Sigma^{1,b}_0$-formulas.
- Induction axioms for all $Q_2\Sigma^{1,b}_0$-formulas.

Note that it is well-known (as a consequence of Theorem 12.3.1) that the Q_2-quantifier cannot be defined by a $\Sigma^{1,b}_0$-formula, even with extra parameters.

13.2 Interpreting Q_2 in structures

The truth values of sentences $\exists y A(y)$ and $\forall y A(y)$ are determined by the truth values of all instances $A(\alpha)$, computing in $\mathcal{B}$ the supremum and the infimum of the set of values, respectively. However, it is difficult to see how one could define $[\![Q_2 y < \beta A(y)]\!]$ coherently from values of its instances $[\![\alpha < \beta \wedge A(\alpha)]\!]$, even if all these values were equal to $1_{\mathcal{B}}$. For example, the family F_{alg} from which α is drawn is infinite and there is a bijective correspondence between the family and the family minus one element.

In interpreting the Q_2 quantifier we take an indirect approach. We notice that the quantifier can be defined by a $\Sigma^{1,b}_0$-formula (in the sense to be described shortly) as long as the class of relations of the structure is rich enough. We will illustrate this on the following informal example (a formal definition for the specific case of $K(F_{alg}, G_{alg})$ will appear in Section 15.2).

Assume $R(x_1, \ldots, x_k, y) \in \mathcal{M}$ is a relation on $m \in \mathcal{M}_n$. We shall call any relation $S(x_1, \ldots, x_k, y) \in \mathcal{M}$ on m a 2-cover of R iff the following holds: For $u_1, \ldots, u_k < m$ and $v < m$

$$S(\overline{u}, v) \text{ iff } |\{w < v \mid R(\overline{u}, w)\}| \textit{ is odd}\,.$$

This makes a good sense in $\mathcal{M}$ where we can count the size of finite sets but in a general structure such definition would not work. However, it is easy to see that the condition is equivalent to the validity of three other conditions, all first-order expressible:

1. $(\forall z < y \neg R(\overline{x}, z)) \rightarrow \neg S(\overline{x}, y)$
2. $S(\overline{x}, y) \rightarrow (S(\overline{x}, y+1) \equiv \neg R(\overline{x}, y))$
3. $\neg S(\overline{x}, y) \rightarrow (S(\overline{x}, y+1) \equiv R(\overline{x}, y))$

Note that if we had the Q_2 quantifier we could define the 2-cover explicitly:

$$S(\overline{x}, y) \equiv Q_2 z < y R(\overline{x}, z) .$$

The point is, however, that the same equivalence can be used to interpret the Q_2 quantifier in any structure (and in front of any $Q_2\Sigma_0^{1,b}$ formula) as long as any $\Sigma_0^{1,b}$-definable relation R has a 2-cover on any m.

This construction is performed formally in Chapter 15. The 2-covers are defined there, essentially, by defining the parity of a set of Boolean values but only of an $\mathcal{M}$-finite set and in Boolean algebra $\mathcal{A}$ and not $\mathcal{B}$.

We note that in this way one can define a theory for the computational complexity class $AC^0(2)$ in the language of V^0 (i.e. without the Q_2 quantifier), having the same $\Sigma_0^{1,b}$-consequences as $Q_2V_1^0$ does. This approach is taken in [29].

Chapter 14

Algebraic model

We shall construct in this chapter a particular model $K(F_{alg}, G_{alg})$ based on algebraic decision trees.

14.1 Family F_{alg}

Recall that we have a fixed non-standard number n. Our sample space will be simple this time: $\Omega := \{0,1\}^n$.

Definition 14.1.1 *Let $\mathbf{F}_2[x_1, \ldots, x_n]$ be the ring of polynomials over $\mathbf{F}_2$ in $\mathcal{M}$. We shall denote by $\mathbf{F}_2^{low}[x_1, \ldots, x_n]$ the family of polynomials from $\mathbf{F}_2[x_1, \ldots, x_n]$ of degree bounded above by some term of the form $n^{1/t}$, for a non-standard $t > \mathbf{N}$.*

In other words, the degree of polynomials in $\mathbf{F}_2^{low}[x_1, \ldots, x_n]$ is subpolynomial in n. Family $\mathbf{F}_2^{low}[x_1, \ldots, x_n]$ is a subset of $\mathcal{M}$ but not a definable one.

To define the family F_{alg} forming the first order part of the model we have to define (in $\mathcal{M}$) first an auxiliary concept of an algebraic decision tree.

Definition 14.1.2 *A decision tree T over $\mathbf{F}_2^{low}[x_1, \ldots, x_n]$ is a finite binary tree whose inner nodes (i.e. non-leafs) are labeled by queries of the form $p(\overline{x}) =_? 0$, for $p(\overline{x}) \in \mathbf{F}_2^{low}[x_1, \ldots, x_n]$. The two outgoing edges from such a node are labeled* yes *and* no*, respectively.*

The depth of a tree, denoted $dp(T)$, is the length of the longest path from the root to a leaf.

The degree of the tree, denoted $deg(T)$, is the maximum degree of a polynomial that is queried in T.

A labeling of T is a map $\ell \in \mathcal{M}$ assigning to leafs of T values in $\mathcal{M}_n$.

An algebraic tree is a algebraic labeled decision tree.

Let $(T, \ell) \in \mathcal{M}$ be an algebraic tree over $\mathbf{F}_2^{low}[x_1, \ldots, x_n]$. Any $\omega \in \Omega$ induces an evaluation $x_i := \omega_i$ and thus determines a unique path through T ending in a leaf that we shall denote $T(\omega)$. In this way (T, ℓ) defines a function

$$\omega \in \Omega \to \ell(T(\omega)) \in \mathcal{M}_n \ .$$

Definition 14.1.3 *Family F_{alg} consists of all functions*

$$\alpha \ : \ \Omega \to \mathcal{M}_n$$

from $\mathcal{M}$ such that there is a labeled decision tree $(T, \ell) \in \mathcal{M}$ over ring $\mathbf{F}_2^{low}[x_1, \ldots, x_n]$ such that

- *$\alpha(\omega) = \ell(T(\omega))$, for all $\omega \in \Omega$, and*
- *both the depth $dp(T)$ and the degree $deg(T)$ of T are bounded above by a term of the form $n^{1/t}$, for some $t > \mathbf{N}$.*

Note that a bound to $deg(T)$ of the form $n^{1/t}$, $t > \mathbf{N}$, follows already from the fact that $(T, \ell) \in \mathcal{M}$ and that all query polynomials are in $\mathbf{F}_2^{low}[x_1, \ldots, x_n]$.

There is an alternative presentation of family F_{alg} that will be useful to keep in mind.

Lemma 14.1.4 *For every $\alpha \in F_{alg}$ there is a sequence of polynomials*

$$(p_0, \ldots, p_{u-1}) \in \mathcal{M}$$

from $\mathbf{F}_2^{low}[x_1, \ldots, x_n]$ (as the sequence is in $\mathcal{M}$ there is a common bound $n^{1/s}$, $s > \mathbf{N}$, for degrees of p_is) for some $u \in \mathcal{M}_n$ such that:

1. *Equations*
   - *(a)* $\sum_{i<u} p_i = 1$
   - *(b)* $p_i \cdot p_j = 0$, *for* $i \neq j \in u$

hold identically on Ω.

2. *All labels of* α *are from* $u = \{0, \ldots, u-1\}$.

3. *For all* $\omega \in \Omega$ *and* $i < u$*:*

$$\alpha(\omega) = i \quad \textit{iff} \quad p_i(\omega) = 1 \ .$$

On the other hand, for all sequences $(p_0, \ldots, p_{u-1}) \in \mathcal{M}$ *of polynomials from* $\mathbf{F}_2^{low}[x_1, \ldots, x_n]$ *of the length* $u \in \mathcal{M}_n$ *obeying condition 1. there is* $\alpha \in F_{alg}$ *for which conditions 2. and 3. also hold.*

Proof :

Let (T, ℓ) be a tree defining α. For any leaf L in T define polynomial q_L as follows: label any yes-edge on the path to L leaving a query node $p =_? 0$ by $1-p$ and any no-edge by p, and take for q_L the product of the labels of the edges on the path. Note that $q_L \in \mathbf{F}_2^{low}[x_1, \ldots, x_n]$ as its degree is bounded above by $deg(T) \cdot dp(T)$.

Clearly $q_L(\omega) = 1$ iff ω determines the path to L. Hence $\sum_L q_L$ is identically 1 on Ω and any two such polynomials attached to two different leafs are incompatible (i.e. at most one is non-zero).

Define p_i to be the sum of q_L for all leafs L labeled by i. This proves the first parts of the lemma.

For the opposite direction, starting with the sequence $(p_0, \ldots, p_{u-1}) \in \mathcal{M}$, define α searching for a non-zero p_i by binary search, querying

$$\sum_{e \le i \le f} p_i =_? 0$$

for suitable $0 \le e < f < u$. The required bounds to degrees of polynomials is and to the depth of the tree defining α are obvious as $u \in \mathcal{M}_n$.

q.e.d.

Given polynomials $(p_i)_{i \in u}$ satisfying equations 1. of Lemma 14.1.4 we shall denote the function α they define in the sense of the lemma as

$$\alpha \ := \ \sum_{i \in u} p_i \cdot i^* \ .$$

This is to be understood as a formal notation not as an actual sum in some algebraic expression. In particular, i^* is a symbol attached to an element $i \in \mathcal{M}_n$, not an element of any ring.

14.2 Family G_{alg}

Family G_{alg} is introduced quite analogously to the way families G_{rud} and G_{tree} were introduced earlier.

Definition 14.2.1 *Let $m \in \mathcal{M}_n$ and let $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1}) \in \mathcal{M}$ be an $\mathcal{M}$-tuple of elements of F_{alg}.*

For any $\alpha \in F_{alg}$ define $\hat{\beta}(\alpha) : \Omega \to \mathcal{M}_n$ by

$$\hat{\beta}(\alpha)(\omega) = \begin{cases} \beta_{\alpha(\omega)}(\omega) & \text{if } \alpha(\omega) < m \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 14.2.2 *Let $m \in \mathcal{M}_n$ and let $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1}) \in \mathcal{M}$ be an m-tuple of elements of F_{alg}.*

Then function $\hat{\beta}$ from Definition 14.2.1 maps F_{alg} into F_{alg}.

Proof :

As $\hat{\beta} \in \mathcal{M}$, the induction in $\mathcal{M}$ implies that there is a maximal depth of a tree computing one of β_i, $i < m$. This depth is therefore bounded above by n raised to an infinitesimal. The depth of the tree computing any $\alpha \in F$ is bounded by a term of the same form, and hence so is the depth of the tree computing $\hat{\beta}(\alpha)$.

q.e.d.

Definition 14.2.3 *Family G_{alg} consists of all random variables Θ computed by some $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1})$, where $m \in \mathcal{M}_n$ and $\hat{\beta} \in \mathcal{M}$ is an m-tuple of elements of F_{alg}.*

In other words, for Θ computed by $\hat{\beta}$ the function Θ_ω has the graph $\{(i, \beta_i(\omega)) \mid i < m\}$ (and is zero outside $[m]$).

Note that both families F_{alg} and G_{alg} are compact.

14.3 Open comprehension and open induction

The proof that open induction is valid in the structure $K(F_{rud}, G_{rud})$ goes through here literally, just replacing Boolean decision trees with algebraic trees. We just state the two relevant statements, for a later reference.

The first one is analogous to Lemmas 8.2.1 and 10.2.7.

Lemma 14.3.1 **(1)** *Let $A(x)$ be an open $L^2_n(F_{alg}, G_{alg})$-formula with the only free variable x.*

Then there is $t > \mathbf{N}$ such that for all $i \in \mathcal{M}_n$ the membership in $\langle\langle A(i)\rangle\rangle$ is computed by an algebraic tree with the leafs labeled by 1 *(for* yes*) or* 0 *(for* no*) of the depth and the degree bounded above by $n^{1/t}$. In particular, the membership is computed by an element of F_{alg}.*

(2) *Let $s > \mathbf{N}$ and $w := n^{1/s}$. Assume $(\alpha_1, \ldots, \alpha_w) \in \mathcal{M}$ is an w-tuple of elements of F_{alg}. Then there is $\beta \in F_{alg}$ computing simultaneously all α_i in the tuple (i.e. it computes the w-tuple of the values).*

(3) *Families F_{alg} and G_{alg} are closed under definition by cases by open $L^2_n(F_{alg}, G_{alg})$-formulas.*

The next lemma is proved (using Lemma 14.3.1 in place of Lemma 8.2.1) identically as Lemmas 8.2.3 and 8.3.2.

Lemma 14.3.2 *Open comprehension and open induction are valid in $K(F_{alg}, G_{alg})$.*

Chapter 15

Quantifier elimination and the interpretation of Q_2

In this chapter we show that any formula of the form

$$\mathbf{Q}y < t(\overline{x})A(\overline{x}, y)$$

where A is open and $\mathbf{Q}$ is one of quantifiers $\exists$, $\forall$, or Q_2 is equivalent in $K(F_{alg}, G_{alg})$ on any interval $[0, m)$ to an open formula (with free variables $\overline{x}$); that is, the equivalence is valid in $K(F_{alg}, G_{alg})$. For the Q_2 quantifier this is done by defining it suitably in $K(F_{alg}, G_{alg})$, and by verifying that its required properties become valid in $K(F_{alg}, G_{alg})$ under the interpretation. For the ordinary first order quantifiers this requires a bit of work; the elimination of such quantifiers utilizes the Razborov-Smolensky approximation method from circuit complexity theory.

15.1 Skolemization and the Razborov - Smolensky method

Recall Definitions 9.2.1 and 9.2.2. As F_{alg} is closed under definition by cases by open $L_n(F_{alg}, G_{alg})$-formulas and families F_{alg} and G_{alg} are compact, Theorem 3.5.2 guarantees the existence of $\gamma \in F_{alg}$ such that

$$[\![\exists y < m B(\alpha, y)]\!] = [\![\gamma < m \wedge B(\alpha, \gamma)]\!]$$

for any formula B. But, as discussed in Section 9.2, this is insufficient for a useful quantifier elimination, and we will proceed differently.

Theorem 15.1.1 *Structure $K(F_{alg}, G_{alg})$ is Skolem closed.*

Proof :

The general strategy of the proof, as expressed in Claim 1, is identical to the proof of Theorem 11.1.1. For the sake of the completeness of the presentation of the proof we repeat this common part here too.

We need to show that for any open $L^2_n(F_{alg}, G_{alg})$-formula $B(x, y)$ (without a loss of generality we assume that we have just one variable x) and for any $m \in \mathcal{M}_n$ there is a $\Theta \in G_{alg}$ that is a Skolem function for $\exists y B(\overline{x}, y)$ on m. Assume we have an m-tuple $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1}) \in \mathcal{M}$ of elements of F_{alg} such that for all $i < m$:

$$[\![\beta_i < m \wedge B(i, \beta_i)]\!] = [\![\exists y < m B(i, y)]\!] \ .$$

Then we would like to take for the Skolem function the $\Theta \in G$ computed by $\hat{\beta}$. Of course, $\Theta(i) = \beta_i$ and so Θ behaves as a Skolem function for $x = 0, \ldots, m-1$ but that is not enough. For example, it may happen that there is another m-tuple $\hat{\gamma} = (\gamma_0, \ldots, \gamma_{m-1}) \in \mathcal{M}$ of elements of F behaving as $\hat{\beta}$ for all $i < m$:

$$[\![\beta_i < m \wedge B(i, \beta_i)]\!] = [\![\gamma_i < m \wedge B(i, \gamma_i)]\!]$$

but

$$\langle\langle \gamma_i < m \wedge B(i, \gamma_i) \rangle\rangle \setminus \langle\langle \beta_i < m \wedge B(i, \beta_i) \rangle\rangle$$

are non-empty. The only information we have about such difference sets is that their individual counting measures in Ω are infinitesimal.

By a careful choice of $\hat{\beta}$ we need to exclude the possibility that there is an $\alpha \in F$ with $[\![\alpha < m]\!] = 1_{\mathcal{B}}$ defining a distribution on $i < m$ so that the errors of individual β_i's combine into a set with standard positive probability.

For any $\hat{\beta}$ of the form as above define:

$$E_i \ := \ \langle\langle \exists y < m B(i, y) \rangle\rangle \setminus \langle\langle \beta_i < m \wedge B(i, \beta_i) \rangle\rangle \ .$$

Claim 1: *Assume that the counting measure in Ω of $\bigcup_{i<m} E_i \in \mathcal{A}$ is infinitesimal. Then $\Theta \in G$ computed by $\hat{\beta}$ is a Skolem function for $\exists y < m B(x, y)$ on m.*

Assume for the sake of contradiction that for some $\alpha \in F$:

$$[\![\Theta(\alpha) < m \wedge B(\alpha, \Theta(\alpha))]\!] < [\![\exists y < m B(\alpha, y)]\!] \ .$$

Applying the observation at the beginning of the section (i.e. Theorem 3.5.2) we get $\gamma \in F$ such that

$$[\![\Theta(\alpha) < m \wedge B(\alpha, \Theta(\alpha))]\!] < [\![\gamma < m \wedge B(\alpha, \gamma)]\!] \ .$$

Therefore

$$U \ := \ \langle\langle \gamma < m \wedge B(\alpha, \gamma) \rangle\rangle \setminus \langle\langle \Theta(\alpha) < m \wedge B(\alpha, \Theta(\alpha)) \rangle\rangle$$

has a non-infinitesimal counting measure in Ω. But clearly

$$U \subseteq \bigcup_{i<m} \{\omega \in \Omega \mid \alpha(\omega) = i\} \cap E_i \subseteq \bigcup_{i<m} E_i$$

which is a contradiction. This proves the claim.

We will find $\hat{\beta}$ such that the error sets E_i will satisfy the hypothesis of Claim 1. We have:

$$\langle\langle \exists y < m B(i, y) \rangle\rangle \ = \ \bigcup_{j<m} \langle\langle B(i, j) \rangle\rangle \ .$$

By Lemma 14.3.1 $\langle\langle B(i,j) \rangle\rangle$ are computed by 0/1-valued trees, elements $\rho_{i,j} \in F_{alg}$. In fact, there is $t > \mathbf{N}$ such that both the degree and the depth of all $\rho_{i,j}$, $i, j < m$, are bounded above by $n^{1/t}$.

Claim 2: *For all $i, j < m$ there is a polynomial $q_{i,j} \in \mathbf{F}_2^{low}[x_1, \ldots, x_n]$ of degree bounded by $n^{2/t}$ such that:*

$$\omega \in \langle\langle B(i,j) \rangle\rangle \quad \textit{iff} \quad q_{i,j}(\omega) = 1 \ .$$

The claim follows from Lemma 14.1.4.

Now comes a key point where we borrow an idea due to Razborov [91]. Fix $r := n^{1/s}$ for some $s > \mathbf{N}$ to be specified later. For each $i < m$ and an arbitrary r-tuple (an element of $\mathcal{M}$) of sets

$$J_{i,1}, \ldots, J_{i,r} \subseteq m = \{0, \ldots, m-1\}$$

put:

$$q_i \ := \ 1 \ - \ \Pi_{v \leq r} (1 - \sum_{j \in J_{i,v}} q_{i,j}) \ .$$

The degree of q_i is bounded by $r \cdot n^{2/t} \leq n^{(2/t)+(1/s)}$ independently of the choice of sets $J_{i,v}$, so $q_i \in \mathbf{F}_2^{low}[x_1, \ldots, x_n]$.

Claim 3: *There is a choice (in $\mathcal{M}$) of sets $J_{i,v} \subseteq m$ for $i < m$ and $v \leq r$ such that the counting measure of each set:*

$$\langle\langle \exists y < mB(i,y) \rangle\rangle \setminus \{\omega \in \Omega \mid q_i(\omega) = 1\}$$

is at most 2^{-r}.

Work in $\mathcal{M}$. As in [91] pick the sets independently and uniformly at random from subsets of m. For any particular $\omega \in \langle\langle \exists y < mB(i,y) \rangle\rangle$ the probability that $\sum_{j \in J_{i,v}} q_{i,j}(\omega) \neq 1$ is $1/2$, so the probability that this happens for all $v \leq r$ is 2^{-r}. If one of the sums equals to 1 then $q_i(\omega) = 1$. In other words, the probability of an error on ω is 2^{-r}. An averaging argument yields the required sets $J_{i,v}$.

Polynomial q_i computes (with a small error) whether or not the set $\langle\langle \exists y < mB(i,y) \rangle\rangle$ is non-empty. To find a particular $j < m$ with $\omega \in \langle\langle B(i,j) \rangle\rangle$ we simulate binary search by a depth $\log(m)$ tree β_i, with the query polynomials constructed as in Claim 3. The degree of those polynomials is bounded above by $n^{2/t+1/s}$, so $\beta_i \in F$.

Take $\Theta \in G_{alg}$ computed by this particular m-tuple $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1})$.

Claim 4: *The function Θ is a Skolem function for $\exists y B(x,y)$ on m.*

By the construction the error set

$$\langle\langle \exists y < mB(i,y) \rangle\rangle \setminus \langle\langle \Theta(i) < m \wedge B(i, \Theta(i)) \rangle\rangle$$

has the counting measure $\leq m \cdot 2^{-r}$ for each $i < m$. Hence the union of these sets has the counting measure $\leq 2^{-r} m^2$. As $m \in \mathcal{M}_n$, $m^2 \leq 2^{n^{1/u}}$ for some $u > \mathbf{N}$. Hence we want to pick r such that:

$$2^{-r} 2^{n^{1/u}} = 2^{n^{1/u} - n^{1/s}}$$

is infinitesimal. Taking $r := n^{1/s}$ for s such that $u > s > \mathbf{N}$ will do. This proves the claim and hence also the lemma.

q.e.d.

15.2 Interpretation of Q_2 in front of an open formula

We shall now interpret the Q_2 quantifier in front of an open formula. That is, for any $m \in \mathcal{M}_n$ and an open $L^2_n(F_{alg}, G_{alg})$-formula $A(\overline{x}, y)$ and any $\overline{\alpha}, \beta \in F_{alg}$ we define the truth value

$$[\![Q_2 y < \beta A(\overline{\alpha}, \beta)]\!] \ .$$

Recall that for an open formula $A(\overline{x}, y)$ and $\overline{\alpha}, \beta \in F_{alg}$:

$$[\![A(\overline{\alpha}, \beta)]\!] = \langle\langle A(\overline{\alpha}, \beta)\rangle\rangle / \mathcal{I} \ .$$

Moreover, for any $m \in \mathcal{M}_n$ there is a relation $\Gamma \in G_{alg}$ such that if $\langle\langle \overline{\alpha}, \beta < m\rangle\rangle = \Omega$ then

$$\langle\langle A(\overline{\alpha}, \beta) \equiv \Gamma(\overline{\alpha}, \beta)\rangle\rangle = \Omega \ .$$

Also, for such $\overline{\alpha}, \beta$,

$$\langle\langle A(\overline{\alpha}, \beta)\rangle\rangle = \bigcup_{\overline{i}, j < m} \langle\langle \overline{\alpha} = \overline{i}\rangle\rangle \cap \langle\langle \beta = j\rangle\rangle \cap \langle\langle A(\overline{i}, j)\rangle\rangle \ .$$

By Lemma 14.1.4 we may assume that Γ is given by array $(p_{\overline{i},j})_{\overline{i},j<m}$ of polynomials from $\mathbf{F}_2^{low}[x_1, \ldots, x_n]$:

$$\langle\langle \Gamma(\overline{i}, j)\rangle\rangle = \{\omega \in \Omega \mid p_{\overline{i},j}(\omega) = 1\} \ .$$

The following definition is the specialization of the notion of 2-cover from Section 13.2 for the model.

Definition 15.2.1 *Let $A(\overline{x}, y)$, $m \in \mathcal{M}_n$ and Γ be as above. The 2-cover of Γ on m is the relation $\Delta \in G$ on m defined by the array:*

$$(\sum_{k<j} p_{\overline{i},k})_{\overline{i},j<m} \ .$$

In particular,

$$\langle\langle \Delta(\overline{i}, j)\rangle\rangle = \{\omega \in \Omega \mid \sum_{k<j} p_{\overline{i},k}(\omega) = 1\} \ .$$

Definition 15.2.2 *Let $A(\overline{x}, y)$ be an open $L^2_n(F_{alg}, G_{alg})$-formula and $\overline{\alpha}, \beta \in F_{alg}$. Take any $m \in \mathcal{M}_n$ such that $\langle\langle \overline{\alpha}, \beta < m \rangle\rangle = \Omega$ and let $\Gamma \in G_{alg}$ be the relation representing A on m as above. Let Δ be the 2-cover of Γ on m.*

Then define:

$$[\![Q_2 y < \beta A(\overline{\alpha}, y)]\!] \ := \ [\![\Delta(\overline{\alpha}, \beta)]\!] \ .$$

The following lemma follows directly from the definition; note that while Γ may not be unique as an array of polynomials, the corresponding polynomials in two such arrays will compute an identical function over $\mathbf{F}_2$.

Lemma 15.2.3 *For any open $L^2_n(F_{alg}, G_{alg})$-formula A and any $\overline{\alpha}, \beta \in F_{alg}$ the definition of $[\![Q_2 y < \beta A(\overline{\alpha}, y)]\!]$ does not depend on the particular choice of $m \in \mathcal{M}_n$ or $\Gamma \in G_{alg}$.*

Furthermore, the instances of the axioms of the Q_2-quantifier for formula A from Section 13.1 are valid in $K(F_{alg}, G_{alg})$.

15.3 Elimination of quantifiers and the interpretation of the Q_2-quantifier

Theorem 15.1.1 provides an elimination of bounded existential or universal quantifiers in front of an open formula, while for the Q_2 quantifier this is achieved simply by its definition (Definition 15.2.2). Doing this in parallel, by induction on the number of quantifiers in a formula, we achieve simultaneously to eliminate quantifiers in $Q_2\Sigma^b_\infty$ $Q_2 L^2_n(F_{alg}, G_{alg})$-formulas and to define the Q_2 quantifier in front of any $Q_2\Sigma^b_\infty$ $Q_2 L^2_n(F_{alg}, G_{alg})$-formula. Let us summarize this into a lemma suitable for a later use.

Lemma 15.3.1 *Let $A(\overline{x})$ be a $Q_2\Sigma^b_\infty$ $Q_2 L^2_n(F_{alg}, G_{alg})$-formula with free variables $\overline{x}$. Let $m \in \mathcal{M}_n$ be arbitrary.*

Then there is an open $L^2_n(F_{alg}, G_{alg})$-formula $A'(\overline{x})$ with the same free variables as A such that:

$$[\![\forall \overline{x} < m A(\overline{x}) \equiv A'(\overline{x})]\!] \ = \ 1_{\mathcal{B}} \ .$$

In other words, $K(F_{alg}, G_{alg})$ admits bounded quantifier elimination.

Similarly to Corollary 11.1.3 we note here for a future reference a consequence of the construction underlying the quantifier elimination (and the definition of Q_2).

Corollary 15.3.2 *For any $Q_2\Sigma^b_\infty$-formula $A(x_1,\dots,x_k)$, possibly with parameters from F_{alg} or G_{alg}, and for any $m \in \mathcal{M}_n$ and arbitrarily small but non-standard $s > \mathbf{N}$ there is an open $L^2_n(F_{alg}, G_{alg})$-formula $B(x_1,\dots,x_k)$ such that*

$$[\![\forall x_1,\dots,x_k < m\ (A(x_1,\dots,x_k)\ \equiv\ B(x_1,\dots,x_k))]\!]\ =\ 1_{\mathcal{B}}$$

and such that for any $\alpha_1,\dots,\alpha_k \in F_{alg}$ the counting measure of the symmetric difference

$$\langle\!\langle \bigwedge_i \alpha_i < m \to A(\alpha_1,\dots,\alpha_k)\rangle\!\rangle \ \triangle\ \langle\!\langle \bigwedge_i \alpha_i < m \to B(\alpha_1,\dots,\alpha_k)\rangle\!\rangle$$

is less than $2^{-n^{1/s}}$. In particular, it is infinitesimal.

15.4 Comprehension and induction for $Q_2\Sigma^{1,b}_0$-formulas

Model $K(F_{alg}, G_{alg})$ admits bounded quantifier elimination (Lemma 15.3.1) and open comprehension and open induction are valid in it (Lemma 14.3.2). Hence Theorem 9.3.1 yields the following.

Theorem 15.4.1 *Comprehension and induction for $Q_2\Sigma^{1,b}_0$ $L_n(F_{alg}, G_{alg})$-formulas are valid in $K(F_{alg}, G_{alg})$.*

In particular, theory $Q_2V^0_1$ is valid in $K(F_{alg}, G_{alg})$.

Chapter 16

Witnessing and independence in $Q_2V_1^0$

We have considered witnessing theorems, independence results, and definability in theory V_1^0 in Chapter 12. As an example we have proved there a witnessing theorem for $\forall X < x \exists Y < x \Sigma_0^{1,b}$ formulas. The proof proceeds by analyzing model $K(F_{tree}, G_{tree})$. It should be clear that structure $K(F_{alg}, G_{alg})$ can be used for an essentially identical proof to provide a quite analogous witnessing of $\forall X < x \exists Y < x \Sigma_0^{1,b}$ formulas in $Q_2V_1^0$.

We shall prove in Section 16.1 a witnessing theorem for the second case considered in the introduction to Chapter 12, the $\forall X < x \exists Y < x \forall Z < x \Sigma_0^{1,b}$ formulas. Again it should be obvious that similar argument applies to V_1^0 and $K(F_{tree}, G_{tree})$ as well.

In Section 16.2 we note that the proofs of the preservation theorems for $K(F_{tree}, G_{tree})$ from Section 12.2 apply also to $K(F_{alg}, G_{alg})$.

16.1 Witnessing $\forall X < x \exists Y < x \forall Z < x \Sigma_0^{1,b}$-formulas

Let $A(x)$ be a $\forall X < x \exists Y < x \forall Z < x \Sigma_0^{1,b}$ formula of the form

$$\forall X < x \exists Y < x \forall Z < x B(x, X, Y, Z)$$

with B a $\Sigma_0^{1,b}$ formula. Families F_{alg} and G_{alg} are closed under definitions by open formulas (Lemma 14.3.1) and so we may apply Lemma 3.3.3 to deduce the following statement for $K(F_{alg}, G_{alg})$.

Lemma 16.1.1 *Assume* $[\![A(n)]\!] = 1_{\mathcal{B}}$. *Then for every* $\Delta \in G_{alg}$ *and every standard* $\epsilon > 0$ *there is* $\Gamma \in G_{alg}$ *such that for every* $\Theta \in G_{alg}$*:*

$$\mu([\![\Delta < n \to (\Gamma < n \wedge (\Theta < n \to B(n, \Delta, \Gamma, \Theta)))]\!] \ > \ 1 - \epsilon \ .$$

Let us take for Δ the identity on $\Omega = \{0,1\}^n$, $\Delta(\omega) = \omega$. Applying Corollary 15.3.2 to $B(n, \Delta, \Gamma, \Theta)$ and then Lemma 2.2.1 we entail the next statement.

Theorem 16.1.2 *Assume* $[\![A(n)]\!] = 1_{\mathcal{B}}$. *Then for every standard* $\epsilon > 0$ *there is* $\Gamma \in G_{alg}$ *such that for all* $\Theta \in G_{alg}$*:*

$$Prob_{(\omega \in \Omega}[\ \omega \in \langle\langle \Gamma < n \wedge (\Theta < n \to B(n, \Delta, \Gamma))\rangle\rangle\ ] \ > \ 1 - \epsilon \ .$$

In particular, this holds if theory $Q_2V_1^0$ *proves* $\forall x A(x)$.

We may derive a finitary version of this theorem analogously to Theorem 12.1.4, using compactness and the fact that we have not used any property of n except that it is non-standard.

Corollary 16.1.3 *Assume that theory* $Q_2V_1^0$ *proves* $\forall x A(x)$. *Then for every standard* $\epsilon, \delta > 0$ *and* $m \in \mathbf{N}$ *there is a collection* $p = (p_i)_{i<m}$ *of polynomials over* $\mathbf{F}_2[x_0, \ldots, x_{m-1}]$ *of degree less than* n^δ *such that for any other collection* $q = (q_i)_{i<m}$ *of polynomials over* $\mathbf{F}_2[x_0, \ldots, x_{m-1}]$ *of degree less than* n^δ *the following holds for all* m *large enough:*

$$Prob_{\omega \in \{0,1\}^m}[\ B(m, \omega, p(\omega), q(\omega))\ ] \ > \ 1 - \epsilon \ .$$

Here $p(\omega)$ abbreviates $(p_i(\omega))_{i<m}$.

We now give an example of an independence result for a formula of the quantifier complexity considered in the witnessing theorem. We will formulate it as a statement about $K(F_{alg}, G_{alg})$ (and use Theorem 16.1.2) rather than a finitary statement. But it will be clear that it is possible to use in the place of n a number parameter, getting a finitary statement (with a finitary independence proof in which the role of Theorem 16.1.2 is taken up by Corollary 16.1.3).

We shall assume n has the form $n = \binom{m}{2}$, for some $m \in \mathcal{M}_n$, and we will interpret subsets of $[n]$ as defining undirected graphs on $[m]$ (without loops). In particular, the sample space Ω is now interpreted as a set of graphs ω on

m vertices. With this understanding we will write $(u,v) \in X$ for $u,v \in [m]$, meaning that the edge (u,v) is in graph X.

Let $A(n,X)$ be a $\Sigma_0^{1,b}$ formula

$$X \subseteq [n] \ \wedge \ \forall u,v,w \in [m] \ (((u,v) \in X \wedge (v,w) \in X) \rightarrow (u,w) \in X) \ .$$

The formula says that X is transitive.

The true $\forall X \exists Y \forall Z \Sigma_0^{1,b}$ sentence which fails in $K(F_{alg}, G_{alg})$ (and hence is independent of $Q_2V_1^0$) asserts that any graph has a transitive closure:

$$\forall X \exists Y \forall Z \ [\ X \subseteq Y \wedge A(n,Y) \wedge ((X \subseteq Z \wedge A(n,Z)) \rightarrow Y \subseteq Z) \ ]$$

(we left the bound $\subseteq [n]$ for X, Y, Z out). Denote this sentence $Closure(n)$.

Theorem 16.1.4 *In $K(F_{alg}, G_{alg})$ it holds that $[\![Closure(n)]\!] = 0_{\mathcal{B}}$.*

In particular, $Q_2V_1^0$ does not prove the existence of a transitive closure of a graph.

We will not give a proof of the theorem here using the witnessing theorem as a different but simpler one can be given. The statement is included as an example of a sentence of the right quantifier complexity.

16.2 Preservation of true $s\Pi_1^{1,b}$-sentences

In Section 12.2 we have shown that all true sentences $\forall x A(x)$ are valid in model $K(F_{tree}, G_{tree})$, as long as formula $A(x)$ has of one of the following three forms:

(c) $\forall X < x \Sigma_0^{1,b}$ (i.e. $s\Pi_1^{1,b}$)

(d) $\forall X < x \exists y < x \forall Z < x \Sigma_0^{1,b}$

(e) $\exists Y < x \forall Z < x \Sigma_0^{1,b}$

The key case was (c), the preservation in the other two cases are its corollaries. The preservation for case (c) (Theorem 12.2.1) was proved using Corollary 11.1.3, a consequence of the proof of bounded quantifier elimination for $K(F_{tree}, G_{tree})$. For structure $K(F_{alg}, G_{alg})$ we have the analogous statement in Corollary 15.3.2, and the same argument works.

The preservation for cases (d) and (e) were then derived using a collection scheme in $\mathcal{M}$ and the richness of language L_n respectively, see Corollaries 12.2.2 and 12.2.3, and identical arguments apply for $K(F_{alg}, G_{alg})$. We shall thus only state the theorem.

Theorem 16.2.1 *Assume sentence $\forall x A(x)$ is true where $A(x)$ has on of the three forms (c), (d) or (e) listed above. Then $\forall x A(x)$ is valid in $K(F_{alg}, G_{alg})$.*

Part V

Towards proof complexity

Chapter 17

Propositional proof systems

In this part we survey a few basic definitions and facts from proof complexity and model theory of bounded arithmetic. The material discussed in this and the next two chapters is mostly well-known and the reader may find a lot more in the literature suggested in the Introduction. In particular, the known material mentioned in this chapter, and indeed in whole Part V, can be found in [55].

We also present only the definitions and the facts we shall use later, not other related material, and we do not include specific examples illustrating general definitions. Although the exposition is brief, it is fairly complete and should be sufficient for understanding the arguments in the later parts of the book.

We start in this chapter by introducing the proof systems we shall be concerned with in later chapters. DeMorgan language for propositional logic has constants 1 (true) and 0 (false), unary connective $\neg$ (negation) and binary connectives $\vee$ (disjunction) and $\wedge$ (conjunction). We shall tacitly assume that the language of any proof system defined in this chapter contains the DeMorgan language.

17.1 Frege and Extended Frege systems

Definition 17.1.1 (Cook-Reckhow[30]) *A* Frege rule *is a* $k+1$*-tuple of formulas* $A_0, \ldots, A_k$ *in atoms* $p_1, \ldots, p_n$ *written as:*

$$\frac{A_1, \ldots, A_k}{A_0} \ ,$$

such that any truth assignment $a : \{p_1, \ldots, p_n\} \to \{0,1\}$ *satisfying all formulas* $A_1 \ldots, A_k$ *satisfies also* A_0 *(this condition is called the soundness of the rule). A Frege rule in which* $k = 0$ *is called a* Frege axiom scheme.

An instance of the rule is obtained by a simultaneous substitution of arbitrary formulas B_i *for all* p_i.

Definition 17.1.2 (Cook-Reckhow[30]) *Let* F *be a finite collection of Frege rules. A* Frege proof *(an* F*-proof briefly) of formula* D *from formulas* $C_1, \ldots, C_u$ *is a finite sequence* $B_1, \ldots, B_k$ *of formulas such that* $B_k = D$*, and such that every* B_i *is either one of* $C_1, \ldots, C_u$*, or is inferred from some earlier* B_j*'s* $(j < i)$ *by a rule of* F.

F *is implicationally complete if and only if any* D *can be* F*-proved from any set* $\{C_1, \ldots, C_u\}$ *if every truth assignment satisfying all* C_i*'s satisfies also* D.

F *is a* Frege proof system *if and only if it is implicationally complete.*

The next class of proof systems augments Frege systems by the ability to abbreviate formulas by new atoms. We use the connective $\equiv$ in its definition. If $\equiv$ is not in the language of F we use any fixed DeMorgan formula defining it.

Definition 17.1.3 (Cook-Reckhow[30]) *Let* F *be a Frege system. An* extended Frege proof *is a sequence of formulas* $A_1, \ldots, A_k$ *such that every* A_i *is either obtained from some previous* A_j*'s by an* F*- rule or has the form:*

$$q \equiv B$$

with the following conditions satisfied:

1. *Atom* q *does appear neither in* B*, nor in any* A_j *for* $j < i$.

2. *Atom* q *does not appear in* A_k.

A formula of this form is called an extension axiom*,* q *is called an* extension atom.

An extended Frege system EF *is the proof system whose proofs are extended Frege proofs.*

The ability to introduce an extension axiom in EF is sometimes called the "Extension rule" but it is not a Frege rule in the earlier sense.

It is a well-known fact that the definition of Frege systems is very robust. As long as lengths of proofs are concerned the strength of a Frege system does not depend on its language or particular format of proofs (from many variants considered in proof theory).This means that a proof of a DeMorgan formula in one Frege system can be translated (by a polynomial-time algorithm even) into an at most polynomially longer proof of the same formula in any other Frege system. Same relations hold among different Extended Frege systems.

17.2 Language with connective $\oplus$ and constant depth Frege systems

The next definition is a particular case of a more general definition extending DeMorgan language by modular counting connectives $MOD_{a,i}$, see [55, 18].

Definition 17.2.1 *$\oplus$ is a propositional connective of unbounded arity such that:*

$$\oplus(p_1, \ldots, p_k) \quad \textit{is true} \qquad \textit{iff} \qquad |\{j \mid p_j \quad \textit{true}\}| \text{ is odd} \quad .$$

$\oplus$-axioms is the formula

$$\neg \oplus (\emptyset)$$

together with the axiom schemes (one for each $k \geq 1$)

$$\oplus(A_1, \ldots, A_k) \equiv$$

$$[(\oplus(A_1, \ldots, A_{k-1}) \wedge \neg A_k) \vee (\neg \oplus (A_1, \ldots, A_{k-1}) \wedge A_k)] \ .$$

We shall study Frege systems and their subsystems that have either DeMorgan language or DeMorgan language with $\oplus$. Thus we shall make the provision that for a Frege system in DeMorgan language we use name F while a Frege system in the DeMorgan language augmented by the $\oplus$ connective will be named $F(\oplus)$.

This extension of the language is not so interesting for unrestricted Frege systems as there is a quadratic size DeMorgan formula defining $\oplus$ and F polynomially proves the translation of the $\oplus$-axioms.

However, having $\oplus$ in the language is a substantial difference if proofs are restricted to formulas of bounded depth. This is because any constant depth

DeMorgan formula defining parity of n bits must have the size exponential in n, cf. Theorem 12.3.1 of Ajtai [1], Furst, Saxe and Sipser [33], Yao [107] and Hastad [39].

The restriction to constant depth means that only a constant number of alternations between disjunctions and conjunctions are allowed in any formula; the depth of propositional formulas is analogous to the quantifier complexity of first order formulas. One may define Frege systems as operating with conjunctions and disjunctions of unbounded arity; the depth is then simply the depth of the formula. However, it seems easier to define the depth of a formula in a less straightforward way rather than modifying the definition of Frege systems.

The unbounded $\bigvee$ and $\bigwedge$ we sometimes use in formulas should be understood as being built from binary $\vee$ and $\wedge$, respectively.

Definition 17.2.2 *The depth of a formula A in the DeMorgan language augmented by $\oplus$, denoted $dp(A)$, is defined inductively as follows:*

1. *The depth of constants and atoms is* 0.
2. $dp(\neg A) := 1 + dp(A)$.
3. $dp(\oplus(\emptyset) := 0$ *and*
   $dp(\oplus(A_1, \ldots, A_k)) := 1 + max_{i \leq k} dp(A_i)$.
4. *If $B(q_1, \ldots, q_k)$ is a formula built from atoms using only $\vee$ and if none of formulas $A_1, \ldots, A_k$ has $\vee$ as the top connective, then for $C = B(A_1, \ldots, A_k)$ we put:*

$$dp(C) := 1 + max_{i \leq k} dp(A_i) \ .$$

   Analogously for $\wedge$.

One can get a somewhat cleaner definition by requiring that $\neg$ is applied only to atoms, and giving all literals depth 0. However, we will be concerned with proofs using only bounded depth formulas but not in the particular bounds on the depth, and hence such nuances of the definition are irrelevant for our purposes.

Note that DNF formulas have depth at most 3. The lower bound $d \geq 3$ in the next definition is to guarantee that the set of depth d DeMorgan tautologies is coNP-complete.

Definition 17.2.3 *For fixed $d \geq 3$, F_d is the subsystem of a Frege system F in DeMorgan language whose proofs may contain only formulas of depth at most d.*

Similarly, $F_d(\oplus)$ is the subsystem of a Frege system $F(\oplus)$ in DeMorgan language augmented by $\oplus$ whose proofs may contain only formulas of depth at most d.

Note that it seems that we cannot directly compare the strength of, say, F and F_d as the latter proof system cannot in principle prove tautologies of depth higher than d. Similarly, comparing F with $F(\oplus)$ appears cumbersome as the class of tautologies in the language with $\oplus$ is bigger than in the DeMorgan language only.

In fact, these and many other issues that come up when developing theory of proof systems can be successfully addressed. In particular, there is a general definition of a *propositional proof system* that subsumes the definitions of Frege systems and Extended Frege systems and their depth d fragments.

We will not explain this material here; the reader may find it in details in [30, 55] or briefly in Part VIII.

Chapter 18

An approach to lengths-of-proofs lower bounds

We now outline a general strategy for proving lower bounds for lengths-of-proofs, showing that it is sufficient to construct a certain model of bounded arithmetic. Stronger a proof system we aim at is, a model of a stronger theory we need to construct. This reduction uses essentially just the compactness of first order logic and provability of the soundness of a proof system in the corresponding bounded arithmetic theory, and goes back to Ajtai [2]. Here we describe this approach modified to allow our Boolean valued structures.

18.1 Formalization of the provability predicate

Let us start with a bit more general discussion. Assume that for every $k \in \mathbf{N}$ there is a string w_k of length at most $s(k)$, where function $s(k)$ is subexponential:

$$s(k) < 2^{k^{o(1)}} \ .$$

The string can be encoded by a set $W_k \subseteq s(k)$ and the whole sequence $\{W_k\}_{k\in\mathbf{N}}$ by a binary relation $W(x, y)$ such that

$$W_k = \{m \in \mathbf{N} \mid W(k, m)\} \ .$$

Assume that $P(X)$ is an NP-property of strings (finite sets). By Fagin's theorem $P(X)$ can be defined by an $s\Sigma^{1,b}_1$ L_n-formula

$$\exists Y A(X,Y)$$

where A is a $\Sigma^{1,b}_0$-formula, with the quantifiers of A and the length of the string Y bounded polynomially in terms of the length of the string X (we can take just unary Y as there is a pairing function in L_n). Hence there is a binary relation $V(x,y)$ such that each

$$V_k = \{m \in \mathbf{N} \mid V(k,m)\}$$

is a subset of $s(k)^{O(1)}$ and whenever W_k has the property $P(X)$ then V_k witnesses the validity of $P(W_k)$: $A(W_k, V_k)$ holds (language L_n has names for W and V, so this is a sentence of the language).

Assume now that W_k has the property $P(X)$ for infinitely many $k \in \mathbf{N}$. By induction in $\mathcal{M}$ it follows that W_n satisfies $P(X)$ also for some non-standard $n \in \mathcal{M}$ (in fact, for cofinaly many of them in $\mathcal{M}$). Similarly, if W_k had the property $P(X)$ for all $k \in \mathbf{N}$, W_n would satisfy $P(X)$ also for all $n \in \mathcal{M}$.

Now we apply this general argument to the provability predicate. Let P denote any of the proof systems defined earlier: F, $F(\oplus)$, EF, F_d or $F_d(\oplus)$.

Assume that $T(x,y)$ and $R(x,y)$ are two binary relations such that for all $k \in \mathbf{N}$ both sets

$$T_k = \{m \in \mathbf{N} \mid T(k,m)\}$$

and

$$R_k = \{m \in \mathbf{N} \mid R(k,m)\}$$

are subsets of $s(k)$ for some subexponential function $s(k) < 2^{k^{0(1)}}$, and such that for some $d \geq 1$ and for infinitely many $k \in \mathbf{N}$ it holds:

1. T_k encodes a DeMorgan formula of depth at most d, and
2. R_k encodes a P-proof of T_k.

Then for some non-standard $n \in M$ the statement

$$(*) \qquad R_n \text{ is a } P\text{-proof of a depth } d \text{ formula } T_n$$

is true in $\mathcal{M}$. This follows from the general discussion as the properties of being a DeMorgan formula or a P-proof are polynomial-time properties and can be defined in the way as above.

In fact, this general argument referring to Fagin's theorem is not quite adequate for our purpose as one needs to be able to prove various properties of these predicates. What we need is that the properties can be defined following the usual syntax of logic and in a way that their basic properties hold and are provable in a weak bounded arithmetic theory. It is well-known how such a formalization of logic can be performed (e.g. [55, 9.3]) and we shall not repeat it here. Further we may assume without a loss of generality that the encoding of proofs comes with an auxiliary structure (a witness to the $s\Sigma_1^{1,b}$-formula) so that (*) is expressible as

$$Fla_d(n, T_n) \wedge Prf_P(n, R_n, T_n) \ ,$$

a conjunction of instances of two $\Sigma_0^{1,b}$ formulas

$$Fla_d(x, Y)$$

defining depth d formulas and

$$Prf_P(x, X, Y)$$

defining the provability predicate. In these formula X and Y are bounded by an L_n-term in x; function $s(k)$ was subexponential in k and hence there is a function symbol in L_n for it. In particular, both R_n and T_n are subsets of $\mathcal{M}_n$.

Lemma 18.1.1 *The statement*

$$Fla_d(n, T_n) \wedge Prf_P(n, R_n, T_n)$$

is valid in any L_n-closed structure $K(F, G)$.

Proof :

The statement is an instance of a true $\forall\Sigma_\infty^{1,b}$ sentence. Hence it is valid in any L_n-closed structure by Lemma 5.5.1.

q.e.d.

18.2 Reflection principles

To formalize a reflection principle for a proof system P we need a formula defining the satisfiability relation. For general formulas this can be defined by both $s\Pi^{1,b}_1$ and $s\Sigma^{1,b}_1$ formulas but not by a $\Sigma^{1,b}_0$ formula. When the depth of formulas is bounded above by some standard d we can construct a $\Sigma^{1,b}_0$ definition

$$Sat_d(x, Z, Y) \ .$$

It says that a truth assignment Z satisfies a depth $\leq d$ formula Y, x bounding both Z and Y.

The construction of such a formula can be done in a way that V^0_1 proves its basic properties corresponding to Tarski's conditions for a truth definition. This can be found in [55].

Definition 18.2.1 *For $d \geq 1$ define formula*

$$Ref_{P,d}(x, X, Y, Z)$$

to be the implication

$$(Fla_d(x, Y) \wedge Prf_P(x, X, Y)) \ \rightarrow \ Sat_d(x, Z, Y) \ .$$

Ref_P denotes the universal closure of $Ref_{P,3}(x, X, Y, Z)$.

Parameter d is chosen to be 3 in the definition of Ref_P as DNF tautologies have depth at most 3, all proof systems considered so far can operate with depth 3 formulas and also all hard tautologies we deal with are DNF.

Sentence Ref_P is the universal closure of a $s\Pi^{1,b}_1$ formula and it is true (for the proof systems P we have defined). Hence we know, by Theorems 12.2.1 and 16.2.1, that it is valid in structures $K(F_{tree}, G_{tree})$ and $K(F_{alg}, G_{alg})$. However, for proof complexity is key that this validity also follows differently.

Theorem 18.2.2 *Theory V^0_1 proves $Ref_{F_d,d}$ and theory $Q_2V^0_1$ proves $Ref_{F_d(\oplus),d}$, for all $d \geq 3$.*

This theorem is well-known and we shall not reprove it here. The idea behind its proof is that, given a truth assignment Z, one verifies by induction on the number of steps ℓ in a depth d proof X of a formula Y that the i-th step in X is satisfied by Z, for $i = 1, \ldots, \ell$.

18.3 Three conditions for a lower bound

Now we put the ideas from the preceding sections together.

Theorem 18.3.1 *Let $d \geq 3$ and let T_k be depth d tautologies of size $k^{O(1)}$, for all $k \in \mathbf{N}$. Further let P be any of the proof systems of Chapter 17.*

Assume that for an arbitrary choice of non-standard n there is a structure $K(F,G)$ (built over $\mathcal{M}_n$) satisfying the following three conditions:

(1) $K(F,G)$ is L_n-closed.

(2) $Ref_{P,d}$ is valid in $K(F,G)$.

(3) There is a truth assignment $\Gamma \in G$ to atoms of T_n that falsifies the formula in $K(F,G)$, i.e. $Sat_d(n,\Gamma,\neg T_n)$ is valid.

Then there is a standard $\epsilon > 0$ such that for no $k \in \mathbf{N}$ large enough has T_k a P-proof of size less than 2^{k^ϵ}.

Proof :

We give a brief sketch of the proof (which is well-known).

Assume for the sake of a contradiction that for infinitely many $k \in \mathbf{N}$ there are P-proofs R_k of T_k of a subexponential size $2^{k^{o(1)}}$. As in Section 18.1, there is a non-standard n such that (for relations R and T with names in L_n) statement

$$Fla_d(n,T_n) \wedge Prf_P(n,R_n,T_n)$$

is true in $\mathcal{M}$. Hence by Lemma 18.1.1 and condition (1) it is also valid in $K(F,G)$.

It follows by condition (2) that

$$\forall Z Sat_d(n,Z,T_n)$$

is valid in $K(F,G)$. But by Condition (3) $Sat_d(n,\Gamma,\neg T_n)$ is valid for some $\Gamma \in G$. It follows by properties of Sat_d corresponding to Tarski's conditions that also

$$\neg Sat_d(n,\Gamma,T_n)$$

is valid. But that is a contradiction.

q.e.d.

Hence in order to prove that no R_n could exist in $\mathcal{M}_n$ and thus to prove an exponential lower bound to the size of P-proofs of formulas T_k, $k \in \mathbf{N}$, it suffices to construct a structure $K(F, G)$ satisfying the three conditions. In fact, it would suffice to have $[\![Sat_d(n, \Gamma, T_n)]\!] < 1_{\mathcal{B}}$.

Chapter 19

PHP principle

Many combinatorial principles (and, more generally, statements about finite structures) can be formalized both in first order logic and in propositional logic. They are very often expressible by $\Sigma_0^{1,b}$ formulas and instances of these can be translated into propositional ones quite easily (replacing bounded existential and universal quantifiers by big disjunctions or conjunctions). In fact, the propositional translation can be defined for more general formulas, including any of the three forms (c), (d) and (e) defined in the introduction to Chapter 12. In this translation a universally true formula is translated into a sequence of tautologies, one tautology for every instance of the universally quantified number variable.

The topic of propositional translations is very well developed and quite subtle at places. We shall not give a general definition but simply present the formalization for the pigeonhole principle (PHP) only. The reader may find detailed expositions in [55, 29].

19.1 First-order and propositional formulations of PHP

Let $R(x, y)$ be a second order variable for a binary relation symbol. The $\Sigma_0^{1,b}$ formula $PHP(x, R)$ is the disjunction of the following three formulas:

1. $\exists u \in [x+1] \forall v \in [x] \neg R(u, v)$.

2. $\exists v, v' \in [x] \exists u \in [x+1]$ $(v \neq v' \wedge R(u, v) \wedge R(u, v'))$.

3. $\exists u, u' \in [x+1] \exists v \in [x]\ (u \neq u' \wedge R(u,v) \wedge R(u',v))$.

A relation R violating the first two properties for some $x := k$ is a graph of a map of $[k+1]$ into $[k]$, and if R would violate also the third condition it would be an injective map. But that is impossible. Hence $\forall x \forall R\ PHP(x,R)$ is a true statement.

Now we formulate PHP propositionally.

Definition 19.1.1 *Let $k \geq 1$ be arbitrary. For any $i \in [k+1]$ and $j \in [k]$, let p_{ij} be a propositional variable. The formula PHP_k, formed from atoms p_{ij}, is the following one:*

$$\bigvee_{i \in k+1} \bigwedge_{j \in k} \neg p_{ij} \ \vee \bigvee_{i \in k+1} \bigvee_{j_1 \neq j_2 \in k} (p_{ij_1} \wedge p_{ij_2}) \vee \bigvee_{i_1 \neq i_2 \in k+1} \bigvee_{j \in k} (p_{i_1 j} \wedge p_{i_2 j})\ .$$

Again a falsifying assignment $p_{ij} := a_{ij} \in \{0,1\}$ to PHP_k would define the graph of an injective function from $[k+1]$ into $[k]$. Hence formulas PHP_k, $k \geq 1$, are tautologies. Also note that it is a DNF formula and its depth is 3.

A landmark result in proof complexity is Ajtai's proof [2] that formulas PHP_k cannot be proved by polynomial size proofs in constant depth Frege systems F_d (in DeMorgan language). He proved it using the approach of Chapter 18, constructing a suitable model (classical, not Boolean valued) of $I\Delta_0(R)$ where PHP fails for some n. The lower bound has been subsequently strengthened to an exponential one, cf. [78, 87].

Theorem 19.1.2 (Ajtai[2],[78, 87])
Theories $I\Delta_0(R)$ and V^0_1, even augmented by any function of a subexponential growth, do not prove $\forall x PHP(x,R)$.

No constant depth Frege system F_d (in DeMorgan language) proves tautologies PHP_k, $k \geq 1$, in a subexponential size.

In the next part we shall give another construction of a suitable model implying the lower bound. Although a form of a switching lemma is used in both constructions there is a subtle difference. In [2] one expands by (essentially) a model-theoretic forcing a cut in $\mathcal{M}$ by a new relation $R \subseteq [n+1] \times [n]$ and no new numbers are added (i.e. no new pigeons or new holes). In our construction the relation is a specific random variable known in advance but we add many new pigeons and new holes that were not in $\mathcal{M}_n$.

19.2 Three conditions for F_d and $F_d(\oplus)$ lower bounds for PHP

We now formulate the model-theoretic criterion for lower bounds for the constant depth proof systems and for the particular formulas PHP_k only.

Theorem 19.2.1 *Let $d \geq 3$ be arbitrary. Assume that for an arbitrary choice of a non-standard n there is a structure $K(F,G)$ (built over $\mathcal{M}_n$) satisfying the following three conditions:*

(1) $K(F,G)$is L_n-closed.

(2) Theory V^0_1 is valid in $K(F,G)$.

(3) There is a truth assignment $\Gamma \in G$ to atoms of PHP_n that falsifies PHP_n in $K(F,G)$, i.e. $Sat_3(n,\Gamma,\neg PHP_n)$ is valid.

Then there is a standard $\epsilon > 0$ such that for no $k \in \mathbf{N}$ large enough has PHP_k an F_d-proof of size less than 2^{k^ϵ}.

If a structure exists satisfying conditions (1), (3) and

(2') Theory $Q_2V^0_1$ is valid in $K(F,G)$.

then there is a standard $\epsilon > 0$ such that for no $k \in \mathbf{N}$ large enough has PHP_k an $F_d(\oplus)$-proof of size less than 2^{k^ϵ}.

Proof :

The theorem follows from Theorem 18.3.1, using Theorem 18.2.2 to replace condition (2) there by the conditions that the structure is a model of V^0_1 or $Q_2V^0_1$, respectively.

q.e.d.

We should now realize troubles lying ahead. We have built earlier the tree model $K(F_{tree},G_{tree})$ and the algebraic model $K(F_{alg},G_{alg})$ with the hope to mimic their constructions and define a structure $K(F,G)$ satisfying the conditions in the theorem. But the structure $K(F,G)$, unlike the tree model and the algebraic model, cannot preserve true $s\Pi^{1,b}_1$ sentences as PHP is one of them and we want to violate it. This is actually one of the reasons why we

have analyzed the validity of induction in $K(F_{tree}, G_{tree})$ and $K(F_{alg}, G_{alg})$ in the explicit way we did; a simpler verification would note that the induction axioms are $s\Pi_1^{1,b}$ formulas and appeal to the preservation property.

A structure $K(F, G)$ useful for lengths-of-proofs lower bounds cannot be an $s\Pi_1^{1,b}$-elementary extension of $\mathcal{M}_n$.

Part VI

Proof complexity of F_d and $F_d(\oplus)$

Chapter 20

A shallow PHP model

We will define a second-order structure $K(F_{php}, G_{php})$ in Chapter 21 in which V^0_1 will be valid but the sentence $PHP(n, R)$ will fail (will have the truth-value $0_{\mathcal{B}}$) for a particular interpretation of R by an element of G_{php}. Theorem 19.1.2 will follow as a corollary of Theorem 19.2.1.

Structure $K(F_{php}, G_{php})$ is constructed analogously as is the structure $K(F_{tree}, G_{tree})$ but it is based on a different concept of a tree: the PHP-tree. In this chapter we introduce this key concept and investigate its properties using a structure $K(F^0_{php}, G^0_{php})$, a rudimentary version of future $K(F_{php}, G_{php})$, based on shallow PHP-trees.

20.1 Sample space Ω^0_{php} and PHP-trees

We begin with the definition of the sample space for the shallow tree model.

Definition 20.1.1 *The sample space Ω^0_{php} consists of all $\omega \subseteq [n+1] \times [n]$ such that*

1. *ω is a graph of a partial injective function from $[n+1]$ into $[n]$, and*
2. *$|\omega| = n$ (i.e. ω is surjective).*

We will often write $\omega(i, j)$ or $\omega(i) = j$ instead of $(i, j) \in \omega$.

In the next section we plan to interpret R by the identity function on Ω^0_{php}; this element of the future G^0_{php} will be denoted Δ^0. That is, Δ^0 is the random variable

$$\Delta^0 \ : \ \omega \in \Omega^0_{php} \ \rightarrow \ \Delta^0_\omega := \omega \ .$$

In other words, for elements α, β of future F^0_{php} the atomic statement $R(\alpha, \beta)$ is interpreted by $\Delta^0(\alpha) = \beta$, and it holds at sample ω iff

$$\omega(\alpha(\omega)) \;=\; \beta(\omega) \; .$$

We wish to define $K(F^0_{php}, G^0_{php})$ such that $PHP(n, \Delta^0)$ will fail in the structure, even:

$$[\![PHP(n, \Delta^0)]\!] \;=\; 0_{\mathcal{B}} \; .$$

Note that the chosen interpretation of R forces that we cannot simply take for F^0_{php} the cut $\mathcal{M}_n$. This is because for any $i \in [n+1]$ and $j \in [n]$ the equality $\Delta^0(i) = j$ holds for a fraction $\frac{1}{n+1}$ samples ω (i has n options where to be mapped by ω and also the option of being outside the domain of ω) and hence

$$[\![\Delta^0(i) = j)]\!] = 0_{\mathcal{B}} \; .$$

In other words, we need to add new holes where pigeons $i \in [n+1]$ will map to, and this forces also new pigeons.

These considerations lead to the following auxiliary concept of a PHP-tree, taken from [7]. Families F^0_{php} and G^0_{php} will be defined using this notion in a way quite analogous to $K(F_{rud}, G_{rud})$.

Definition 20.1.2 *A* **PHP-tree** T *is a finite tree whose inner nodes are labeled either by*

$$i \rightarrow ?$$

for some $i \in [n+1]$*, or by*

$$? \rightarrow j$$

for some $j \in [n]$*. If a node is labeled by* $i \rightarrow ?$ *it has* n *outgoing edges labeled by all* $j \in [n]$*: Edge* j *corresponds to the case when* $\omega(i, j)$ *holds. If a node is labeled by* $? \rightarrow j$ *it has* $(n+1)$ *outgoing edges labeled by all* $i \in [n+1]$*, with edge* i *corresponding to the case* $\omega(i, j)$*.*

Any $\omega \in \Omega^0_{php}$ determines a unique path in T. This path, however, needs not to finish in a leaf. It may happen that a query $i \rightarrow ?$ is posed along the path such that $i \notin dom(\omega)$. If a leaf is reached we shall denote it $T(\omega)$, otherwise $T(\omega)$ is undefined.

The depth $dp(T)$ of T is defined as always: the maximal number of edges on a path from the root to a leaf.

Lemma 20.1.3 *Let T be a PHP-tree of depth d. Then*

$$Prob_{\omega\in\Omega^0_{php}}[\, T(\omega) \;\; \textit{is undefined}\,] \;\leq\; O(d/n)\ .$$

Proof :

The lemma is proved by induction on n and the depth d of T. The phrase "induction on n" means the following: We have defined PHP-trees using n as the a priori parameter. But equally well we could have used another parameter m, starting with the sample space consisting of partial bijections between $[m+1]$ and $[m]$ of size m, yielding "PHP-trees over m". The induction argument infers the lemma from a similar statement for $n-1$ and $d-1$ in place of n and d.

In particular, denote by $p(d,m)$ the probability (as computed in $\mathcal{M}$) that T is undefined on a sample (also over m) for PHP-trees of depth d over m. Then clearly:

$$p(1,m) = \frac{1}{m+1}$$

and

$$p(d,m) = \frac{1}{m+1} + \frac{m}{m+1}\cdot p(d-1,m-1)\ .$$

Hence $p(d,n) < O(\frac{d}{n})$.

q.e.d.

Assign to any vertex v in a PHP-tree T a subset $Path_T(v) \subseteq [n+1]\times[n]$ as follows. If $v_0,\ldots,v_k$ is the path from the root v_0 to $v = v_k$ then

$$Path_T(v) \;:=\; \{(i_0,j_0),\ldots,(i_{k-1},j_{k-1})\}$$

where at node v_t $(t = 0,\ldots,k-1)$ the tree queries either $i_t \rightarrow ?$ and the path follows the edge j_t, or it queries $? \rightarrow j_t$ and follows the edge i_t.

Now observe that if the path in T defined by a sample $\omega \in \Omega^0_{php}$ gets to a node v, then necessarily $Path_T(v) \subseteq \omega$. In particular, no sample ω ever gets to a node v for which $Path_T(v)$ is not a partial injective function. In other words, because we let the trees compute only on samples from Ω^0_{php}, we could have defined the PHP-trees differently: At node v the tree would not branch according to all i's or j's but only to those edges (i,j) such that neither i nor j occurs in $Path_T(v)$.

In particular, if $v = T(\omega)$ is defined then $Path_T(v)$ is a partial injective map of size at most $dp(T)$.

A **labeled PHP-tree** (T, ℓ) is a PHP-tree T whose leafs are labeled by elements of $\mathcal{M}_n$. It computes a partial function from Ω^0_{php} into $\mathcal{M}_n$: The function maps ω to the label of $T(\omega)$, if $T(\omega)$ is defined.

We will need to talk about predicates on Ω^0_{php}, i.e. Boolean function on Ω^0_{php}, being computed by a 0/1 - valued PHP-tree. However, the function defined by a PHP-tree is only partially defined and so we need to use a bit more relaxed notion.

Definition 20.1.4 *Let f be a Boolean function with domain Ω^0_{php} and (T, ℓ) a PHP-tree labeled by 0 or 1.*

Then (T, ℓ) **represents** *f iff*

$$\ell(T(\omega)) \;=\; f(\omega)$$

whenever $T(\omega)$ is defined.

Consider propositional formulas built form variables p_{ij}, $i \in [n+1]$ and $j \in [n]$. A **map-term** is a conjunction of the form

$$p_{i_1 j_1} \wedge \ldots \wedge p_{i_k j_k}$$

where all $i_1, \ldots, i_k$ are different and also all $j_1, \ldots, j_k$ are different. We shall write the map-terms usually without the $\wedge$ sign simply as $p_{i_1 j_1} \ldots p_{i_k j_k}$.

A map-term as above corresponds to a partial injective map

$$a := \{(i_1, j_1), \ldots, (i_k, j_k)\}$$

from $[n+1]$ into $[n]$ in the following sense: an assignment $\omega \in \Omega^0_{php}$ satisfies the map-term iff $a \subseteq \omega$.

A k**-disjunction** is a disjunction of map-terms each of which has size at most k. Such a disjunction is either true or false on any assignment from Ω^0_{php} and hence defines a total Boolean function on Ω^0_{php}. We will also represent such a function by a 0/1 - valued PHP-tree; this is meant in the precise sense of Definition 20.1.4. In particular, if f is a k-disjunction and (T, ℓ) a 0/1 -valued tree representing f then the tree computes the same value as f whenever it is defined.

Having a 0/1-valued PHP-tree (T, ℓ) of depth d we define a d-disjunction $D_{(T,\ell)}$ as follows: include a term $p_{i_1 j_1} \dots p_{i_d j_d}$ for each set $\{(i_1 j_1), \dots, (i_d, j_d)\}$ that equals to $Path_v$ for some leaf v labeled by 1. The following simple statement will be useful in the later construction of Skolem functions.

Lemma 20.1.5 *(T, ℓ) represents the function defined by $D_{(T,\ell)}$. $D_{(T,\ell)}$ equals to 0 at those samples where (T, ℓ) is undefined.*

Note that if (T, ℓ) represents f then this disjunction $D_{(T,\ell)}$ may not be equivalent to f.

20.2 Structure $K(F^0_{php}, G^0_{php})$ and open comprehension and open induction

Functions in the family F^0_{php} will be defined by the labeled trees and hence will be only partially defined. This is a key feature of the structure. For such partially defined α, β we put

$$\langle\langle \alpha = \beta \rangle\rangle \ := \ \{\omega \in \Omega^0_{php} \mid \text{ both } \alpha(\omega) \text{ and } \beta(\omega) \text{ are defined and equal}\}$$

and as before

$$[\![\alpha = \beta]\!] \ := \ \langle\langle \alpha = \beta \rangle\rangle / \mathcal{I} \ .$$

Similarly for other relations $R(\alpha_1, \dots, \alpha_k)$: if any argument α_i is undefined at ω then ω does not get into $\langle\langle R(\alpha_1, \dots, \alpha_k)\rangle\rangle$ and hence does not count for the truth value $[\![R(\alpha_1, \dots, \alpha_k)]\!]$. Note that with this definition, for example

$$\langle\langle \alpha = \alpha \rangle\rangle \neq \Omega^0_{php} \ .$$

We need to see how this conforms with the earlier set-up where random variables were defined everywhere. We also need to have the equality axioms valid in all structures.

In general this could be a fatal problem. However, elements of F^0_{php} will be undefined only on an infinitesimal fraction of Ω^0_{php}. Hence we can imagine that the functions are defined on the whole sample space, giving arbitrary values to the samples in the original region of undefinability. This arbitrary extension to total functions does change the $\langle\langle \dots \rangle\rangle$ values of the atomic sentences as defined above, but only by sets of infinitesimal counting measure.

Hence the truth values $[\![\ldots]\!]$ of atomic sentences, and therefore of all sentences, remain unchanged.

In the next definition and in the subsequent text we will therefore talk freely about partially defined random variables.

Now we are ready to define, in a complete analogy with the definition of F_{rud} in Section 7, family F^0_{php}.

Definition 20.2.1 *F^0_{php} is the set of all partial functions $\alpha : \Omega^0_{php} \to \mathcal{M}_n$ such that there is a labeled PHP-tree (T, ℓ) satisfying:*

- *$\alpha(\omega)$ is defined iff $T(\omega)$ is defined, for all $\omega \in \Omega^0_{php}$.*
- *$\alpha(\omega) = \ell(T(\omega))$, for all $\omega \in \Omega^0_{php}$ such that $T(\omega)$ is defined.*
- *The depth $dp(T)$ of T satisfies $dp(T) < n^{1/t}$, for some $t > \mathbf{N}$.*

By Lemma 20.1.3 each α is indeed undefined on at most an infinitesimal fraction of Ω^0_{php}.

Analogously with structure $K(F_{rud}, G_{rud})$ we shall interpret any m-tuple $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1}) \in \mathcal{M}$ of elements of F^0_{php} for an $m \in \mathcal{M}_n$ as defining a function on F^0_{php}

$$\hat{\beta}(\alpha) \ : \ \Omega^0_{php} \to \mathcal{M}_n$$

by the condition:

$$\hat{\beta}(\alpha)(\omega) = \begin{cases} \beta_{\alpha(\omega)}(\omega) & \text{if } \alpha(\omega) < m \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to check that any such $\hat{\beta}$ maps F^0_{php} into F^0_{php}. Thus we can define G^0_{php} analogously as a few times before.

Definition 20.2.2 *Family G^0_{php} consists of all random variables Θ computed by some $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1})$, where $m \in \mathcal{M}_n$ and $\hat{\beta} \in \mathcal{M}$ is a m-tuple of elements of F^0_{php}.*

The following lemma is obvious; it states properties of $K(F_{rud}, G_{rud})$ established in Lemmas 7.3.1.

Lemma 20.2.3 *The model $K(F_{php}, G_{php})$ has the following three properties:*

1. *F^0_{php} is*
   (a) *L_n-closed, and*
   (b) *closed under definitions by cases by open $L^2_n(F^0_{php}, G^0_{php})$-formulas.*
2. *Both F^0_{php} and G^0_{php} are compact.*

Before the next lemma recall the specific concept of a 0/1-tree representing a predicate on Ω^0_{php} from Definition 20.1.4. The statement is analogous to Lemma 8.2.1 but we shall sketch the proof, because we use the notion of representing a function rather than computing it as in Lemma 8.2.1.

Lemma 20.2.4 *Let $A(y)$ be an open $L^2_n(F^0_{php}, G^0_{php})$-formula with the only free variable x. Then there is $t > \mathbf{N}$ such that for all $i \in \mathcal{M}_n$ the membership in $\langle\langle A(i)\rangle\rangle$ is represented by a PHP-tree with the leafs labeled by* 1 *(for* yes*) or* 0 *(for* no*) of the depth bounded above by $n^{1/t}$. In particular, the membership predicate in $\langle\langle A(i)\rangle\rangle$ is represented by an element of F^0_{php}.*

Proof :

Proceed by the logical complexity of the formula. Take atomic formula $R(\alpha_1, \ldots, \alpha_k, y)$. The tree β representing $\langle\langle R(\alpha_1, \ldots, \alpha_k, i)\rangle\rangle$ is the tree α_1 to the leafs of which we attach α_2 etc., and at the bottom decide of the relation holds or not. Note that β is defined iff all α_j are, so the sample where β is undefined are also not in $\langle\langle R(\alpha_1, \ldots, \alpha_k, i)\rangle\rangle$.

Conjunction $B(y) \wedge C(y)$ corresponds to composing the trees β_i and γ_i attached to the respective formulas. The negation is treated simply: If we have trees β_i for $B(y)$ then trees β'_i for $\neg B(y)$ are obtained by switching the labels 0 and 1 of the leafs. But note that β and β'_1 are undefined at the same samples and hence this does not correspond to the complement of $\langle\langle B(i)\rangle\rangle$ in $\mathcal{A}$.

q.e.d.

With Lemmas 20.2.3 and 20.2.4 in hand the verification that open comprehension and open induction are valid in $K(F^0_{php}, G^0_{php})$ is done identically as for structure $K(F_{rud}, G_{rud})$ in Sections 8.2 and 8.3.

Lemma 20.2.5 *Open comprehension and open induction are valid in structure $K(F^0_{php}, G^0_{php})$.*

20.3 The failure of PHP in $K(F^0_{php}, G^0_{php})$

Now we come to the key specific property of the structure.

Lemma 20.3.1

$$[\![PHP(n, \Delta^0)]\!] \;=\; 0_{\mathcal{B}} \;.$$

Proof :

We need to verify that all three disjuncts of $PHP(n, \Delta^0)$ get value $0_{\mathcal{B}}$. This is quite straightforward to verify for the second and the third disjuncts. The second disjunct is:

$$\exists v, v' \in [n] \exists u \in [n+1] \; (v \neq v' \wedge \Delta^0(u) = v \wedge \Delta^0(u) = v') \;.$$

We need to verify that for any α, β, β' from F^0_{php} the set of samples

$$\langle\!\langle \alpha \in [n+1] \wedge \beta \in [n] \wedge \beta' \in [n] \wedge \beta \neq \beta' \wedge \Delta^0(\alpha) = \beta \wedge \Delta^0(\alpha) = \beta' \rangle\!\rangle$$

has an infinitesimal counting measure. But that is obvious as all samples are partial functions and hence assign to a pigeon at most one hole.

Similarly the third disjunct

$$\exists u, u' \in [n+1] \exists v \in [n] \; (u \neq u' \wedge \Delta^0(u) = v \wedge \Delta^0(u') = v)$$

gets value $0_{\mathcal{B}}$: Any sample is a partial injective function, i.e. in any hole sits just one pigeon.

Checking that the first disjunct in $PHP(n, \Delta^0)$

$$\exists u \in [n+1] \forall v \in [n] \; \Delta^0(u) \neq v$$

gets $0_{\mathcal{B}}$ too is a bit less straightforward. To check this means checking that for any $\alpha \in F^0_{php}$ it holds

$$[\![\alpha \in [n+1] \rightarrow \Delta^0(\alpha) \in [n]]\!] \;=\; 1_{\mathcal{B}} \;.$$

Assume that α is determined by a labeled PHP-tree (T, ℓ). The element $\Delta^0(\alpha)$ is computed by (T', ℓ'), a labeled PHP-tree defined as follows:

1. If the label $i = \ell(v)$ of a leaf v of T satisfies $i \in [n+1]$ extend T at v by a depth one tree: Query $i \rightarrow ?$, and label any new leaf corresponding to an edge $j \in [n]$ by j.

2. If $\ell(v) \notin [n+1]$ do nothing.

We need to show that the set

$$\langle\langle \alpha \in [n+1] \wedge \Delta^0(\alpha) \notin [n] \rangle\rangle$$

has an infinitesimal counting measure.

It follows from the definition of Δ^0 that this set equals to the set of samples ω for which $\Delta^0(\alpha)$ is undefined. But this set has an infinitesimal counting measure by Lemma 20.1.3.

q.e.d.

We leave it to the reader to verify that a similar argument proves that in $K(F^0_{php}, G^0_{php})$ the function Δ^0 is surjective:

$$[\![\forall y \in [n] \exists x \in [n+1] \ \Delta^0(x) = y]\!] \ = \ 1_{\mathcal{B}} \ .$$

Chapter 21

Model $K(F_{php}, G_{php})$ of V_1^0

We are going to modify the structure $K(F^0_{php}, G^0_{php})$ from the previous chapter to a new structure $K(F_{php}, G_{php})$. This new structure will be a model of V_1^0.

The construction of $K(F_{php}, G_{php})$ is quite analogous to the construction of $K(F_{tree}, G_{tree})$ from $K(F_{rud}, G_{rud})$: it is based on what could be called deep PHP-trees. And as before, the main purpose of the more complicated definition is to get bounded quantifier elimination. For this we will utilize a switching lemma from [78, 87], and we start by describing it.

21.1 The PHP switching lemma

In this section we shall recall the switching lemma involving PHP-trees, developed in [78, 87] for the purpose of improving Ajtai's original, mere superpolynomial, lower bound [2] in Theorem 19.1.2. The lemma is more involved than the basic example of Hastad's switching lemma (Theorem 10.1.2) as it concerns truth assignments where there is a great deal of dependence between values of propositional variables (in Theorem 10.1.2 these are completely independent).

We need first to define the space of partial truth assignments (restrictions) used in the switching lemma. Assume $D \subseteq [n+1]$ and $R \subseteq [n]$ are two sets of equal size. Denote by $Res(D, R)$ the set of all partial injective maps from $[n+1]$ into $[n]$ with domain D and range R. If $|D| = m$ then clearly there are $m!$ such maps.

Now we are ready to state the switching lemma in the form we shall use it.

Theorem 21.1.1 ([78, 87]) *Let $1 \leq m \leq n$ be arbitrary. Let $1 \leq a \leq b$. Assume a Boolean function f on Ω^0_{php} is defined by an a-disjunction.*

Pick a subset $D \subseteq [n+1]$ and a subset $R \subseteq [n]$ both of size m, uniformly at random, and then pick a restriction $\rho \in Res(D,R)$, also uniformly at random.

Then the probability that the restricted function $f \upharpoonright \rho$ is not on the set of truth assignment $\{\omega \in \Omega^0_{php} \mid \rho \subseteq \omega\}$ representable by a PHP-tree of depth at most b is bounded above by

$$[\ \frac{(n+1-m)^4 a^3)}{m}\ ]^{b/2}\ .$$

We will be picking m of the form $n - n^\delta$, $\delta > 0$ a standard rational. But note that contrary to Theorem 10.1.2 we cannot pick δ arbitrarily close to 1, even for a of the form n raised to an infinitesimal; we need $\delta < \frac{1}{4}$ (we will use $1/5$).

21.2 Structure $K(F_{php}, G_{php})$

The definition of $K(F_{php}, G_{php})$ is analogous to that of $K(F_{tree}, G_{tree})$ but there are differences. This is due to the remark after Theorem 21.1.1 and to the fact that it is not so straightforward to represent the set of restrictions $Res(D,R)$ by a tree. The formalism we define bellow could have been used in the definition of $K(F_{tree}, G_{tree})$ as well.

Definition 21.2.1 *For $k \geq 0$ define a* level k chain*, or briefly a k-chain, as follows. The 0-chain is the triple $(\emptyset, \emptyset, \emptyset)$.*

For $k \geq 1$, a k-chain T is a $(k+1)$-tuple

$$(D_i, R_i, \rho_i)\ _{0 \leq i \leq k}$$

such that:

- $D_0 = R_0 = \rho_o = \emptyset$.
- $D_1 \subseteq [n+1]$ *and* $R_1 \subseteq [n]$ *are sets of size* $n - n^{1/5}$ *(rounded down), and* $\rho_1 \in Res(D_1, R_1)$.
- *For* $i = 2, \ldots, k$, $D_i \subseteq [n+1] \setminus \bigcup_{0 \leq j < i} D_j$ *and* $R_i \subseteq [n] \setminus \bigcup_{0 \leq j < i} R_j$ *are sets of size* $n^{(1/5)^{i-1}} - n^{(1/5)^i}$.

- *For $i = 1, \ldots, k$, $\rho_i \in Res(D_i, R_i)$.*

We make the provision that if the size of ρ_k should drop below 10 then D_k, R_k and ρ_k are of maximal possible sizes, i.e. $R_k = [n]$, D_k is $[n+1]$ without one pigeon and $|\rho_k| = n$.

Denote by $\rho(T) := \bigcup_{0 \leq j \leq k} \rho_j$. The support of T is the pair of sets $(\bigcup_{0 \leq j \leq k} D_j, \bigcup_{0 \leq j \leq k} R_j)$.

A partial ordering $\preceq$ on these chains is defined as follows: for a k-chain T and an ℓ-chain S it holds $T \preceq S$ iff T the initial part of S.

$Chain_k$ is the set of all k-chains.

Note that if T and S are both k-chains then $T \preceq S$ iff $T = S$. Also note that the complement of the support of a k-chain contains $n^{5^{-k}} + 1$ pigeons and $n^{5^{-k}}$ holes.

Denote by h_0 the minimal number such that the support of h_0-chains contains all pigeons except one and all holes. That is, the restriction $\rho(T)$ defined by an h_0-chain T has size n (and, in particular, is in Ω^0_{php}). Constant h_0 has the size around $O(\log(\log(n)))$.

Definition 21.2.2 *The sample space Ω_{php} is the set $Chain_{h_0}$.*

Note that any $\omega \in \Omega^0_{php}$ equals to $\rho(T)$ for exponentially many h_0-chains T. In a sense the samples in Ω_{php} are those in Ω^0_{php} but augmented with a particular history how they were obtained by a succession of restrictions (of specific forms).

The definition of the family of random variables F_{php} is now analogous to Definition 10.2.4.

Definition 21.2.3 *Family F_{php} consists of all functions $\alpha \in \mathcal{M}$,*

$$\alpha : \Omega_{php} \to \mathcal{M}_n$$

for which there exists $k \in \mathbf{N}$, $t > \mathbf{N}$ and a tuple $(S_Z)_{Z \in Chain_k} \in \mathcal{M}$ of PHP-trees such that

- *Each S_Z is a PHP-tree.*
- *The depth of all S_Z in the tuple is bounded by $n^{1/t}$, for some common $t > \mathbf{N}$.*

- *The value of α on sample $T \in \Omega_{php}$ is computed by S_Z at $\omega := \rho(T)$, for $Z \preceq T$.*

The parameter k is called the level of α.

Note that for each $Z \in Chain_k$, $k \in \mathbf{N}$, Lemma 20.1.3 implies that S_Z fails to be defined for an infinitesimal fraction $O(\frac{n^{1/t}}{n^{5-k}})$ of $\{T \in \Omega_{php} \mid Z \preceq T\}$. Hence the region of undefinability of an element of F_{php} has an infinitesimal counting measure.

Note also that without a loss of generality we may assume that PHP-tree S_Z queries only pigeons/holes outside the support of Z (see the remark after Lemma 20.1.3).

We make the usual simple technical observation, with a proof analogous to the proof of Lemma 10.2.5.

Lemma 21.2.4 *Let $m \in \mathcal{M}_n$ and let $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1}) \in \mathcal{M}$ be an m-tuple of elements of F_{php}. For any $\alpha \in F_{php}$ define function $\hat{\beta}(\alpha) : \Omega_{php} \to \mathcal{M}_n$ by*

$$\hat{\beta}(\alpha)((T,\omega)) = \begin{cases} \beta_{\alpha((T,\omega))}((T,\omega)) & \text{if } \alpha((T,\omega)) < m \\ 0 & \text{otherwise.} \end{cases}$$

Then this function is in F_{php} too.

With the lemma in hand we can define family G_{php} in the same way as was defined G_{tree} or G^0_{php} before.

Definition 21.2.5 *Let $m \in \mathcal{M}_n$ and let $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1}) \in \mathcal{M}$ be an m-tuple of elements of F_{php}.*

For any $\alpha \in F_{php}$ define $\hat{\beta}(\alpha) : \Omega_{php} \to \mathcal{M}_n$ by

$$\hat{\beta}(\alpha)((T,\omega)) = \begin{cases} \beta_{\alpha((T,\omega))}((T,\omega)) & \text{if } \alpha((T,\omega)) < m \\ 0 & \text{otherwise.} \end{cases}$$

Family G_{php} consists of all random variables Θ computed by some $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1})$, where $m \in \mathcal{M}_n$ and $\hat{\beta} \in \mathcal{M}$ is a m-tuple of elements of F_{php}.

We conclude this section with a technical lemma useful for open comprehension and open induction, and later for the bounded quantifier elimination. It is again quite analogous to Lemma 10.2.7, and to Lemmas 20.2.3 and 20.2.4.

Lemma 21.2.6 (1) *Let $A(x)$ be an open $L^2_n(F_{php}, G_{php})$-formula with the only free variable x. Then there are $k \in \mathbf{N}$ and $t > \mathbf{N}$ such that for all $i \in \mathcal{M}_n$ the membership in $\langle\langle A(i)\rangle\rangle$ is represented by a $0/1$-valued element of F_{php} having level k and the depth bounded above by $n^{1/t}$.*

(2) *Let $s > \mathbf{N}$ and $w := n^{1/s}$. Assume $(\alpha_1, \ldots, \alpha_w) \in \mathcal{M}$ is an w-tuple of elements of F_{php}. Then there is $\beta \in F_{php}$ such that*

- *β is defined at a sample iff all α_i are defined.*
- *β computes simultaneously all α_i in the tuple (i.e. it computes the w-tuple of the values).*

(3) *Family F_{php} is closed under definition by cases by open $L^2_n(F_{php}, G_{php})$-formulas.*

21.3 Open comprehension, open induction, and failure of PHP

Recall from Chapter 7 that comprehension for open formula $A(x)$ on $[0, m)$ is witnessed by an m-tuple $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1})$, where β_i represents the predicate $\langle\langle A(i)\rangle\rangle$. Similarly, induction for $A(x)$ up to m is witnessed by a random variable computed by a tree that is built from trees for β_i in a form simulating the binary search. Hence Lemma 21.2.6 (Part (1)) gives us what we need to carry this proof in $K(F_{php}, G_{php})$ as well.

Lemma 21.3.1 *Open comprehension and open induction are valid in structure $K(F_{php}, G_{php})$.*

We also need to verify that PHP fails in $K(F_{php}, G_{php})$. But we have to first define a suitable map from G_{php}.

Definition 21.3.2 *Element $\Delta \in G_{php}$ is defined as follows: For sample $T \in \Omega_{php}$ put*

$$\Delta_T \ := \ \rho(T) \ .$$

That is, Δ is determined by the $(n+1)$-tuple $(\beta_i)_{i\in[n+1]}$ where β_i is a depth 1 tree querying $i \to ?$ at the root and labeling the leaf on the edge corresponding to $j \in [n]$ by j itself.

Note that $\Delta_T = \Delta^0_\omega$ for $\omega = \rho(T)$, so the definition of Δ just sheds the extra chain structure on T but is otherwise as in Δ^0.

Lemma 21.3.3

$$[\![PHP(n, \Delta)]\!] \;=\; 0_{\mathcal{B}} \;.$$

Proof :

The proof is essentially identical with that of Lemma 20.3.1; we just need to find our way in the bit more complicated structure $K(F_{php}, G_{php})$. Let us treat only the first disjunct of $PHP(n, R)$, the hardest case in the proof of Lemma 20.3.1.

We want to show

$$[\![\alpha \in [n+1] \to \Delta(\alpha) \in [n]]\!] \;=\; 1_{\mathcal{B}}$$

ie. that the set

$$(*) \qquad \langle\!\langle \alpha \in [n+1] \to \Delta(\alpha) \in [n] \rangle\!\rangle$$

differs from Ω_{php} by a set of an infinitesimal counting measure. Function α is undefined on at most an infinitesimal fraction of the sample space. If it is defined at a sample T and $\alpha(T) \notin [n+1]$, T gets into set (*). If $\alpha(T)$ is defined and equal to some $i \in [n+1]$ then either $\rho(T)(i) = j \in [n]$ and T gets into (*), or $\rho(T)$ is undefined. The latter happens with infinitesimal probability $(n^{5^{-k}} + 1)^{-1}$, if α is of level k.

q.e.d.

21.4 Bounded quantifier elimination

What remains is to construct suitable Skolem functions. The construction is analogous to that Skolem functions in $K(F_{tree}, G_{tree})$ but there some differences in the analysis of what is going on. This is discussed in detail later in Section 22.5 but some readers may prefer to read that exposition now.

Theorem 21.4.1 *Structure $K(F_{php}, G_{php})$ is Skolem closed.*

Proof :

We need to show that for any open first-order $L^2_n(F_{php}, G_{php})$-formula $B(x, y)$ and for any $m \in \mathcal{M}_n$ there is a $\Theta \in G_{php}$ that is a Skolem function for $\exists y B(x, y)$ on m.

We want to find an m-tuple $\hat{\beta} = (\beta_0, \ldots, \beta_{m-1}) \in \mathcal{M}$ of elements of F_{php} such that each of the sets:

$$E_i := \langle\langle \exists y < mB(i,y) \rangle\rangle \setminus \langle\langle \beta_i < m \wedge B(i,\beta_i) \rangle\rangle \ ,$$

$i = 0, \ldots, m-1$, has an infinitesimal counting measure in Ω_{php}. That is:

$$[\![\exists y < mB(i,y)]\!] \ = \ [\![\beta_i < m \wedge B(i,\beta_i)]\!] \ .$$

Claim 1: *Assume that the counting measure (in Ω_{php}) of each E_i is infinitesimal. Then $\Theta \in G_{php}$ computed by $\hat{\beta}$ is a Skolem function for $\exists y < mB(x,y)$ on m.*

To prove the claim assume for the sake of a contradiction that

$$[\![\exists y < m \ B(\alpha,y)]\!] \ > \ [\![\Theta(\alpha) < m \wedge B(\alpha, \Theta(\alpha))]\!]$$

for some α. Applying Part (3) of Lemma 21.2.6 and Lemma 3.3.2 we get γ such that

$$[\![\gamma < m \wedge B(\alpha,\gamma)]\!] \ > \ [\![\Theta(\alpha) < m \wedge B(\alpha, \Theta(\alpha))]\!]$$

and thus also

$$\mu([\![\gamma < m \wedge B(\alpha,\gamma)]\!]) > \ \epsilon + \mu([\![\Theta(\alpha) < m \wedge B(\alpha, \Theta(\alpha))]\!]) \ .$$

for some standard $\epsilon > 0$. By averaging there is i such that counting measure of

$$\langle\langle \alpha = i \wedge \gamma < m \wedge B(i,\gamma) \rangle\rangle$$

is bigger than that of

$$\langle\langle \beta_i < m \wedge B(i,\beta_i) \rangle\rangle$$

by the additive factor ϵ. But that is impossible by the hypothesis of the claim as

$$\langle\langle \alpha = i \wedge \gamma < m \wedge B(i,\gamma) \rangle\rangle \subseteq \langle\langle \exists y < mB(i,y) \rangle\rangle \ .$$

This proves the claim.

Hence we want to construct $\beta_i \in F_{php}$ satisfying the hypothesis of Claim 1. We have:

$$\langle\langle \exists y < mB(i,y) \rangle\rangle \ = \ \bigcup_{j<m} \langle\langle B(i,j) \rangle\rangle \ .$$

By Part (1) of Lemma 21.2.6 the membership in sets $\langle\langle B(i,j)\rangle\rangle$ are represented by 0/1-valued functions $\xi_{i,j}$ from F_{php} such that all $\xi_{i,j}$ have the same standard level k and depth bounded above by $n^{1/t}$, some common non-standard t .

Claim 2: *For any $i < m$, $0 \leq u < v < m$ and $s > \mathbf{N}$ there are level $k+1$ elements $\kappa_{i,u,v} = (R_W^{i,u,v})_{W \in Chain_{k+1}}$ of F_{php} of depth at most $n^{\frac{1}{s}}$ such that:*

- *The probability that for a random $W \in Chain_{k+1}$ tree $R_W^{i,u,v}$ does not represent a predicate containing*

$$(T \in \langle\langle \exists u \leq y \leq v \ B(i,y)\rangle\rangle \ \wedge \ W \preceq T)$$

is bounded above by

$$2^{-n^{1/s}} \ .$$

We shall use Theorem 21.1.1 to prove this key claim. We fix parameters i, u, v, s satisfying the restrictions of the claim (and do not otherwise reflect these parameters in the forthcoming notation). Also put $a := n^{1/t}$and $b := n^{1/s}$.

Let $(S_Z^j)_Z$, for $u \leq j \leq v$ and $Z \in Chain_k$, be the depth $\leq a$ trees computing $\xi_{i,j}$. Let D_Z^j be the disjunction of all map-terms corresponding to paths in S_Z^j ending with label 1 (for yes). Note that each D_Z^j is an a-DNF formula.

The element $\kappa_{i,u,v} \in F_{php}$ will be computed by trees $(R_W)_{W \in Chain_{k+1}}$, the trees having the depth at most b. We are now going to argue, using Theorem 21.1.1, that there exists such trees R_W such that the element $\kappa_{i,u,v}$ defined by them satisfies the requirement of the claim.

Fix $Z \in Chain_k$. Let D_Z be the disjunction $\bigvee_{u \leq j \leq v} D_Z^j$. It is an a-DNF formula that has variables corresponding to the complement of the support of Z. That is, there are $n^{5^{-k}}+1$ possible pigeons and $n^{5^{-k}}$ possible holes. By its definition the formula defines correctly for all T such that $Z \preceq T$, with an infinitesimal probability of an error, the predicate:

$$(T \in \bigcup_{u \leq j \leq v} \langle\langle B(i,j)\rangle\rangle \ .$$

Take a random $(k+1)$-chain W, $Z \preceq W$. By Theorem 21.1.1 here will exists a PHP-tree R_W such that

- R_W represents $D_Z \downarrow \rho(W)$, and
- R_W has the depth at most b,

with the probability of failing to do so for a random W at most

$$[\ \frac{(n^{5^{-(k+1)}}+1)^4 a^3}{n^{5^{-k}} - n^{5^{-(k+1)}}}\ ]^{b/2} < 2^{-n^{1/s}}\ .$$

If there exists a suitable R_W, we conclude the argument by a reasoning as explained in the example at the end of the preceding section. Namely, we argue that for any α, $\kappa_{i,u,v}$ represents a predicate containing $\langle\langle u \leq \alpha \leq v \wedge B(i,\alpha)\rangle\rangle$.

Fixing Z and ignoring as before the $u \leq \alpha \leq v$ part, the predicate

$$\langle\langle B(i,\alpha)\rangle\rangle \cap \{T \mid Z \preceq T\}$$

is represented by a PHP-tree $\Gamma(\alpha)$, where Γ is the map defined by the m-tuple $(S^j_Z)_{j<m}$. So it suffices to verify that for $Z \preceq T$:

$$\Gamma(\alpha)(T) = 1 \ \rightarrow\ \kappa_{i,u,v}(T) = 1$$

if they are both defined. If $\Gamma(\alpha)(T) = 1$ then $\alpha(T)$ is defined and equal to some $j < m$, and $S^j_Z(T)$ is defined and equal to 1. So $D^j_Z(T) = 1$ and $D_Z(T) = 1$, and $\kappa_{i,u,v}(T) = 1$ too.

But by the estimate above suitable trees R_W do exist for all but a fraction of at most $2^{-n^{1/s}}$ chains W. This proves the claim.

To define elements β_i finding a j such that $T \in \langle\langle B(i,j)\rangle\rangle$ if it exists (with only an infinitesimal probability of failing) we simulate the binary search using functions $\kappa_{i,u,v}$. This requires to compute $\log(m)$ of such functions on a path in the search, and the whole binary search can be performed by an element β_i of F_{php} (as in Lemma 10.2.7, Part.2). The depth of the trees computing β_i is bounded above by $\log(m) \cdot b$ and that obeys the restrictions for a $(k+1)$-level element of F_{php}.

By Claim 2 each $\kappa_{i,u,v}$ makes an error with probability less than $2^{-n^{1/s}}$. There is at most m such pairs $u \leq v$ used in a binary search and there are m different instances i. Hence the counting measure of the set of samples where we can make an error is less than

$$2^{-n^{1/s}} \cdot m^2\ .$$

By taking s non-standard but small enough this quantity can be made infinitesimal.

q.e.d.

The theorem yields, via Theorem 9.2.3, the bounded quantifier elimination.

Corollary 21.4.2 *Model $K(F_{php}, G_{php})$ admits bounded quantifier elimination.*

21.5 PHP lower bound for F_d: a summary

Proof of Theorem 19.1.2.

By Corollary 21.4.2 and Lemma 21.3.1 the structure $K(F_{php}, G_{php})$ is a model of V^0_1. By Lemma 21.3.3 the PHP principle fails in $K(F_{php}, G_{php})$; its truth-value is $0_{\mathcal{B}}$. This proves the first part of the theorem.

The second part, the lower bound for F_d-proofs of PHP_n, follows from the first part by Theorem 19.2.1.

Chapter 22

Algebraic PHP model?

This chapter is devoted to the problem of a construction of an L_n-closed model of theory $Q_2V_1^0$ in which PHP would fail. The existence of such a structure would imply via Theorem 19.2.1 the affirmative solution of the following problem.

Problem 22.0.1 *Is it true that for every $d \geq 3$ there is an $\epsilon > 0$ such that proofs of the tautologies PHP_k for $k \geq 1$ in a constant depth Frege system $F_d(\oplus)$ in DeMorgan language with the parity connective $\oplus$ must have the size at least 2^{k^ϵ}?*

In particular, is it true that the theory $Q_2V_1^0$, even augmented by any function of a subexponential growth, does not prove $\forall x PHP(x, R)$?

The problem to prove a lower for $F_d(\oplus)$ is open for quite some time. Neither Ajtai's method [2] so successful for F_d nor its later improvements by Krajíček, Pudlák and Woods [78] and Pitassi, Impagliazzo and Beame [87] (or any other method for that matter) were able to tackle the problem. On the other hand, already before Ajtai's paper Razborov [91] invented a simple method for proving lower bounds for constant depth circuits in DeMorgan language with the parity connective (this was subsequently simplified and generalized by Smolensky [100]). Thus right after Ajtai's paper researchers attempted to modify the Razborov-Smolensky method to a proof complexity setting in order to prove lower bounds for $F_d(\oplus)$. The problem is often branded as the hardest problem (in proof complexity) among those deemed doable (however, it was repeatedly described as doable over the span of more than twenty years).

No lower bounds for the proof system are known but there are several relevant results nevertheless. In particular, Krajíček [57] proves an exponential lower bound for a subsystem of the proof system that extends both constant depth Frege systems and polynomial calculus, Maciel and Pitassi [80] give a quasipolynomial simulation of the proof system by a depth 3 subsystem that can use also a threshold connective, and Impagliazzo and Segerlind [46] prove that constant depth Frege systems with counting axioms modulo a prime p do not polynomially simulate polynomial calculus over $\mathbf{F}_p$. Additional bibliographical information related to the problem can be found in these papers. Constant depth Frege systems and various algebraic proof systems are also surveyed in Buss et.al. [18].

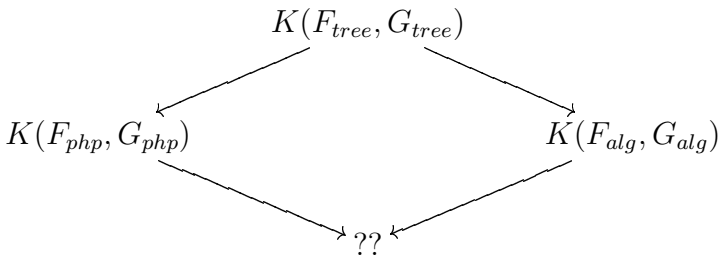
In earlier chapters we have developed a uniform approach to the construction of the tree model, the algebraic model and the PHP-tree model, and one may hope that it would give us a hint how to solve the "equation"

$$\frac{K(F_{php}, G_{php})}{K(F_{tree}, G_{tree})} = \frac{?}{K(F_{alg}, G_{alg})}$$

and construct a model needed for a solution of the problem.

One may be under the impression (as this author originally was) that solving this equation should be simple and straightforward, and that it will suffice to go where the flow of the method will lead us. In retrospect this appears distinctively over-optimistic.

We have not found the right pair of a sample space and a family of random variables that would define the model we want. But even through failed attempts one learns a few things. Perhaps the most relevant is that rather than to look at the problem as the equation above, more useful seems to be to consider the diagram



and think how to complete it.

It became clear that the key issue is to come up with the right notion of an "algebraic PHP-tree". Such trees will define random variables analogously to trees in earlier structures. However, these random variables have to be only partially defined. This appears to be a relevant insight and we discuss it in detail in Section 22.5.

Before we get to that we shall review in earlier sections some material from proof complexity of algebraic proof systems that may be relevant to the eventual construction of the wanted structure.

22.1 Algebraic formulation of PHP and relevant rings

Definition 22.1.1 *Let $\mathbf{F}_2[x_{ij}]$, with $i \in [n+1] \wedge j \in [n]$, be the ring of polynomials over $\mathbf{F}_2$ in the indicated variables x_{ij}.*

S is the ring $\mathbf{F}_2[x_{ij}]$ factored by the ideal generated by all polynomials $x_{ij}^2 - x_{ij}$. In particular, S is an $\mathbf{F}_2$-vector space whose basis is formed by multilinear monomials.

We say that a polynomial has **low** *degree if the degree is bounded above by some term of the form $n^{1/t}$, for a non-standard $t > \mathbf{N}$.*

We shall denote by S^{low} the family of polynomials from S of low degree. (Family S^{low} is a subset of $\mathcal{M}$ but not a definable one.)

We shall sometimes abuse the terminology and call also a number from $\mathcal{M}_n$ **low** if is bounded above by some term of the form $n^{1/t}$, for a non-standard $t > \mathbf{N}$.

Definition 22.1.2 ([8]) *The $(\neg PHP_n)$-system is the system of polynomial equations over S consisting of equations:*

- $x_{i_1 j} \cdot x_{i_2 j} = 0$, *for each* $i_1 \neq i_2 \in [n+1]$ *and* $j \in [n]$.
- $x_{ij_1} \cdot x_{ij_2} = 0$, *for each* $i \in [n+1]$ *and* $j_1 \neq j_2 \in [n]$.
- $1 - \sum_{j \in n} x_{ij} = 0$, *for each* $i \in [n+1]$.

The left-hand sides of these equations are denoted $Q_{i_1,i_2;j}$, $Q_{i;j_1,j_2}$ and Q_i respectively. We denote by $(\neg PHP_n)$ also the set of these polynomials.

Note that by multiplying $Q_i = 0$ by x_{ij} and using $Q_{i;j,j'} = 0$ for $j' \neq j$ we derive $x_{ij} - x_{ij}^2 = 0$. We introduce one more ring.

Definition 22.1.3 *If $\bar{r} = (r_\ell)_\ell$ are additional variables (different from all x_{ij}) then $S\tilde{[}\bar{r}]$ is the ring $\mathbf{F}_2[\bar{x}, \bar{r}]$ factored by all polynomials $x_{ij}^2 - x_{ij}$ and all polynomials $r_\ell^2 - r_\ell$.*

We shall use the same notation $S\tilde{[}\bar{r}]$ for rings possibly differing in tuples of $\bar{r}$; this is always clear from the context. Note that $S\tilde{[}\bar{r}]$ is not $S[\bar{r}]$; it is its quotient by polynomials $r_\ell^2 - r_\ell$ (hence the notation with $\tilde{}$). Every element of $S\tilde{[}\bar{r}]$ is represented by a canonical multilinear polynomial over $\mathbf{F}_2$ and hence have a well-defined degree.

We would like to take for the sample space the set of maximal ideals in S (or at least in S^{low}) in $\mathcal{M}$ containing $(\neg PHP_n)$ and not containing 1. But as $(\neg PHP_n)$ is unsolvable in $\mathcal{M}$ no such ideal exists.

22.2 Nullstellensatz proof system NS and designs

The following definition makes sense over any commutative ring but we shall specialize (here and in later definitions) to $S\tilde{[}\bar{r}]$ only.

Definition 22.2.1 (Beame et.al.[8]) *Let $\mathcal{F} = \{f_i \mid i = 1, \ldots, k\}$ be a set of polynomials from $S\tilde{[}\bar{r}]$ and $f_0 \in S\tilde{[}\bar{r}]$. A Nullstellensatz proof (NS-proof) of f_0 from $\mathcal{F}$ is a k-tuple of polynomials g_i, $i = 1, \ldots, k$, such that*

$$g_1 \cdot f_1 + \ldots + g_k \cdot f_k \;=\; f_0$$

holds in $S\tilde{[}\bar{r}]$. The degree of the refutation is the maximum of $deg(g_i) + deg(f_i)$, $i \leq k$.

An NS-proof of 1 is called an NS-refutation of $\mathcal{F}$.

The notion of design was invented by P. Pudlák[1] for the special case of counting principles and used first in Beame et.al.[8]. A general definition of designs has been studied by Buss [16]. We specialize to ring S only (and, as in the whole book, to $\mathbf{F}_2$).

[1] In the electronic Prague - San Diego seminar that ran in mid 90s.

Definition 22.2.2 *Let $t \geq 0$ be a parameter. Let $\mathcal{F}$ be a set of polynomials from S of degree $\leq t$. A degree t design for $\mathcal{F}$ is a map*

$$D \ : \ S \to \mathbf{F}_2$$

satisfying the following three conditions:

- *$D(0) = 0$ and $D(1) = 1$.*
- *D is a linear map.*
- *$D(g \cdot f) = 0$ for any $f \in \mathcal{F}$ and $g \in S$ such that $deg(f) + deg(g) \leq t$.*

The raison d'etre for the definition is the following simple theorem (see [8, Sec.5], [18, Sec.4]) or [16, Thm.3]). It follows by applying a design to both sides of a potential NS-refutation, yielding contradiction $0 = 1$.

Theorem 22.2.3 *Let $t \geq 0$ be a parameter. Let $\mathcal{F}$ be a set of polynomials from S of degree $\leq t$.*

If there is a degree t design for $\mathcal{F}$ then $\mathcal{F}$ does not have a degree $\leq t$ Nullstellensatz refutation in S.

In fact, the existence of a degree t design in the theorem is not only sufficient but also necessary; this follows by the duality of linear programming as shown by Buss [14]. Theorem 22.2.3 was used in the proof of the next statement (the original result is for the so called modular counting principles, the PHP lower bound follows via known and simple reductions).

Theorem 22.2.4 ([8, 18]) *Any Nullstellensatz refutation of $\neg PHP_n$) over $\mathbf{F}_2$ must have the degree at least $n^{\Omega(1)}$.*

22.3 A reduction of $F_d(\oplus)$ to NS with extension polynomials

We can represent constants 0 and 1 from S by propositional constants 0 and 1, and monomials in variables x_{ij} by conjunctions of variables occurring in them. If

$$f = \sum_{u=1,\ldots,v} \Pi_{(i,j)\in J_u} x_{ij}$$

is a degree d polynomial from S then statement $f = 0$ is expressible by a depth 2 propositional formula

$$\neg \bigoplus (\bigwedge_{(i,j)\in J_1} x_{ij}, \ldots, \bigwedge_{(i,j)\in J_v} x_{ij}) \ .$$

The size of the formula is bounded by the number of monomials of degree d, i.e. by $n^{O(d)}$. In particular, any degree d NS-refutation of a set of polynomials f_i from S can be turned into a constant depth $F(\oplus)$-proof of $\bigvee_i \neg(f_i = 0)$ of size $n^{O(d)}$.

Buss et.al. [18] found a way how to augment Nullstellensatz so that also the opposite implication holds. Here the ring $S\tilde{[}\bar{r}]$ enters the picture.

Definition 22.3.1 (Buss et.al.[18]) *Let $\bar{g} = g_1, \ldots, g_m$ be polynomials from $S\tilde{[}\bar{r}]$ and let $h \geq 1$ be a fixed number. Take new variables r_{iu}, $i = 1, \ldots, m$ and $u = 1, \ldots, h$ not occurring in any of g_j.*

The polynomials $E_{i,\bar{g}}$:

$$g_i \cdot \Pi_{u\leq h}(1 - \sum_{j\leq m} r_{ju} g_j),$$

one for each $i \leq m$, are called extension polynomials of accuracy h corresponding to $\bar{g}$. The variables r_{iu} are called extension variables corresponding to $\bar{g}$.

To understand the meaning of the extension polynomials consider polynomial

$$f(\bar{x}, \bar{r}) := \Pi_{u\leq h}(1 - \sum_{j\leq m} r_{ju} g_j).$$

If all $g_i = 0$ then $f(\bar{x}, \bar{r}) = 1$. On the other hand, if $f(\bar{x}, \bar{r}) = 1$ then - assuming that all extension polynomials corresponding to $\bar{g}$ are zero - necessarily all $g_i = 0$. In this way the polynomial $1 - f(\bar{x}, \bar{r})$, which is of degree at most $h \cdot (1 + \max_i deg(g_i))$, represents the disjunction $\bigvee_i (g_i = 1)$. If we represented this disjunction straightforwardly by $1 - \Pi_i(1 - g_i)$ we would get a polynomial of a possibly much bigger degree $\sum_i deg(g_i)$.

Definition 22.3.2 (Buss et.al.[18]) *Let $\mathcal{E}$ be a set of extension polynomials $S\tilde{[}\bar{r}]$. We call the set $\mathcal{E}$* levelled *if:*

1. *All variables occurring in $\mathcal{E}$ are either $\bar{x}$ or extension variables of some polynomial in $\mathcal{E}$.*

2. *If $\mathcal{E}$ contains some extension polynomial, $E_{i,g_1,\ldots,g_m}$, then $\mathcal{E}$ must contain all companion extension polynomials $E_{1,\bar{g}}, E_{2,\bar{g}}, \ldots, E_{m,\bar{g}}$.*

3. *$\mathcal{E}$ can be decomposed into levels $\mathcal{E} = \mathcal{E}_1 \dot{\cup} \ldots \dot{\cup} \mathcal{E}_\ell$ in such a way that for any polynomial $E_{i,g_1,\ldots,g_m} \in \mathcal{E}_j$ its companion polynomials $E_{1,\bar{g}}, \ldots, E_{m,\bar{g}}$ also belong to $\mathcal{E}_j$, and extension variables corresponding to $\bar{g}$ do not occur in any other polynomial from $\mathcal{E}_1 \dot{\cup} \ldots \dot{\cup} \mathcal{E}_j$.*

The minimal ℓ for which such a decomposition is possible is called the depth *of $\mathcal{E}$.*

Now we state the reduction of $F_d(\oplus)$ to NS augmented by the extension polynomials. We consider only the case of $(\neg PHP_n)$ but the reduction applies to a general system of polynomial equations.

Theorem 22.3.3 (Buss et.al.[18, Thm.6.7(1)]) *For every $d \geq 3$ there is a constant $c \geq 1$ such that for any $n \geq 2$ and $h \geq 1$ the following holds.*

If there exists an $F_d(\oplus)$-proof of PHP_n with s inferences then there exists a levelled set $\mathcal{E}$ of extension polynomials with accuracy h such that $|\mathcal{E}| \leq s^c$, the depth of $\mathcal{E}$ is at most $d+c$, and $(\neg\mathsf{PHP}_n) \cup \mathcal{E}$ has a Nullstellensatz refutation in $S\tilde{[}\bar{r}]$ (where $\bar{r}$ are all extension variables) of degree at most $(\log s)(h+1)^c$.

The theorem is proved by an elementary but lengthy proof-theoretic argument which we shall not repeat here. The idea is to translate the formulas occurring in a proof into low degree polynomials as they are introduced in the proof, and transform the proof step by step into a Nullstellensatz refutation. Whenever a big disjunction (or conjunction) should be translated, rather than using the straightforward translation one introduces suitable extension polynomials and uses the low degree translation outlined before Definition 22.3.2. The resulting set of extension polynomials is levelled because the extension polynomials introduced when translating a depth ℓ formula will be in level ℓ.

22.4 A reduction of polynomial calculus PC to NS

The Nullstellensatz is a proof system for proving that the ideal generated by some polynomials is trivial. One can take the closure properties of ideals as inference rules and define a bit different proof system, the so called

polynomial calculus PC. We shall now recall its definition (just for the ring S).

Definition 22.4.1 (Clegg et.al. [24])
Let $f_1, \ldots, f_k, g$ be polynomials over S. A polynomial calculus proof (PC-proof) of g from $f_1, \ldots, f_k$ is a sequence of polynomials $h_1, \ldots, h_t$ from S such that $h_t = g$, and for each $u \leq t$:

1. *Either h_u is one of $f_1, \ldots, f_k$,*
2. *or $h_u = h_v + h_w$ for some $v, w < u$,*
3. *or $h_u = h_v \cdot x_{ij}$, for some $v < u$ and x_{ij} a variable.*

The degree of the PC-proof is $max_{u \leq t} deg(h_u)$.

We write $f_1, \ldots, f_k \vdash_d g$ to indicate that there is a PC-proof of g from $f_1, \ldots, f_k$ of degree at most d.

Theorem 22.2.4 provided strong degree lower bound for NS refutations of $(\neg PHP_n)$. Razborov [93] proved a similar lower bound for PC. Note that a degree d NS refutation can be turned easily into a degree d PC refutation. The opposite is not true in general but Razborov's construction actually shows that as long as proofs from $(\neg PHP_n)$ are concerned PC is no stronger than NS (cf. [93, Remark 3.12]).

Theorem 22.4.2 (Razborov[93]) *There is no degree $n/2$ PC-proof of 1 from $(\neg PHP_n)$ in S.*

For any $f \in S$, if f has a degree $d \leq n/2$ PC-proof from $(\neg PHP_n)$ then it has also a degree d NS-proof from $(\neg PHP_n)$.

Having such a strong degree lower bound it would seem that we can substitute in the role of samples for the wishful but non-existent maximal ideals maximal subsets of S, containing $(\neg PHP_n)$, closed under degree $n/2$ PC-derivations but not containing 1. Such sets do exist and their intersections with S^{low} are, in fact, ideals in S^{low}. However, again no such set is definable in $\mathcal{M}$ and, in particular, the set of such samples would not be in $\mathcal{M}$, violating a key requirement of forcing with random variables.

We now outline Razborov's construction in some detail as it provides an explicit description of a certain vector space basis that may turn out to

be eventually useful. We follow Razborov [93] and, in parts, a simplified treatment by Impagliazzo, Pudlák and Sgall [45]. Our notation attempts to adhere to these two papers.

We shall define first several ambient vector spaces (everything is over $\mathbf{F}_2$) and describe a certain vector space basis. The definition of the pigeon dance and the basis Δ_t is due to Razborov [93], the basis C_t was explicitly defined by Impagliazzo et.al. [45].

Definition 22.4.3 *1. $\hat{S}$ is the ring S factored by the ideal generated by all polynomials $Q_{i_1,i_2;j}$ and $Q_{i;j_1,j_2}$ from $(\neg\mathsf{PHP}_n)$.*

For $f \in S$, $\hat{f} \in \hat{S}$ denotes the corresponding element of the quotient.

2. Map *is the set of partial injective maps a from $[n+1]$ into $[n]$. Each $a \in \mathsf{Map}$ defines a monomial $x_a := \Pi_{i \in dom(a)} x_{ia(i)}$. In particular, $\emptyset \in \mathsf{Map}$ and $x_\emptyset = 1$.*

   For $t \geq 0$, put $\mathsf{Map}_t := \{a \in \mathsf{Map} \mid |a| \leq t\}$.

3. *For $t \geq 0$, $\hat{S}_t$ is the set of elements of $\hat{S}$ of degree at most t.*

   $T_t := \{x_a \mid a \in \mathsf{Map}_t\}$ is the vector space basis of $\hat{S}_t$.

4. *V_t is the set of polynomials $\hat{f} \in \hat{S}$ such that $f \in S$ has a degree $\leq t$ PC-proof (equivalently, by Theorem 22.4.2, a degree $\leq t$ NS-proof) from $(\neg\mathsf{PHP}_n)$.*

5. *Let $a \in \mathsf{Map}$. A pigeon dance of a is the following non-deterministic process: Take the smallest pigeon $i_1 \in dom(a)$ and move it any currently unoccupied hole bigger than $a(i_1)$. Then take second smallest pigeon $i_2 \in dom(a)$ and move it any currently unoccupied hole bigger than $a(i_2)$, etc. We say that the pigeon dance is defined on a if this process can be completed for all pigeons in $dom(a)$.*

6. Map^* *is the set of all maps $a : [n+1] \to [n] \cup \{0\}$ that is injective on $dom(a) \setminus a^{(-1)}(0)$. $\mathsf{Map}^*_t := \{a \in \mathsf{Map}^* \mid |a| \leq t\}$.*

   For $a \in \mathsf{Map}^*$ denote by a^- the map a restricted to domain $dom(a) \setminus a^{(-1)}(0)$. In particular, $a^- \in \mathsf{Map}$.

   For $a \in \mathsf{Map}^*$ denote by x_a the element of $\hat{S}$ represented by $x_{a^-} \cdot Q_{i_1} \cdot \ldots \cdot Q_{i_k}$, where $\{i_1, \ldots, i_k\} = a^{(-1)}(0)$.

7. *$B_t \subseteq \hat{S}$ is the set of all elements x_a for $a \in \mathsf{Map}^*$ such that a pigeon dance is defined on a^-. In particular, $1 \in B_0$.*

8. *Put $C_t := B_t \setminus T_t$ and $\Delta_t := B_t \cap T_t$. In particular, Δ_t is the set of all elements x_a for $a \in \mathsf{Map}$ such that a pigeon dance is defined on a.*

Theorem 22.4.4 (Razborov[93], Impagliazzo et.al. [45])

(1) *For $0 \leq t \leq n/2$, as vector spaces*

$$\hat{S}_t \;=\; V_t \oplus \mathbf{F}_2\Delta_t \;.$$

and

$$V_t \;=\; \mathbf{F}_2 C_t \;.$$

In particular, $1 \notin V_t$ and Δ_t is a basis of the vector space $\hat{S}_t/V_t$.

(2) *Let $x_c \in T_t$ and assume that its (unique) expression in the basis $C_t \cup \Delta_t$ is*

$$\sum_{x_a \in X} \alpha_a x_a \;+\; \sum_{x_b \in Y} \alpha_b x_b$$

for some $X \subseteq C_t$ and $Y \subseteq \Delta_t$, and α's non-zero coefficients from $\mathbf{F}_2$.

Then for all $x_a \in X$ and $x_b \in Y$, $dom(a) \subseteq dom(c)$ and $dom(b) \subseteq dom(c)$.

(3) *Let $f_0, f_1 \in \hat{S}$, $deg(f_0) + deg(f_1) \leq t$ and $t \leq n/4$. If $f_0 \in V_t$, also $f_0 \cdot f_1 \in V_t$.*

The first statement of Part 1 of the lemma is from Razborov [93] (Claims 3.4 and 3.11 there, showing that Δ_t is a basis of $\hat{S}_t/V_t$). An alternative proof of that and the second statement in Part 1 are in Impagliazzo et.al. [45] (Proposition 3.8 and the proof of Theorem 3.9 there).

Part 2 just states the property of the reduction process from Razborov [93] (the proof of Claim 3.4 there, repeated in the proof of Proposition 3.8 in Impagliazzo et.al. [45]) that finds the expression of x_c in terms of basis $C_t \cup \Delta_t$. We now give a brief sketch of this reduction. This will prove Part 2 of the theorem and, in fact, it also shows that $C_t \cup \Delta_t$ spans $\hat{S}_t$ and that Δ_t spans $\hat{S}_t/V_t$ (the linear independence needs an additional argument). The reduction process also yields Part (3).

Define a partial ordering $x_c \preceq x_d$ for terms $x_c = x_{i_1 j_1} \ldots x_{i_k j_k}$ and $x_d = x_{u_1 v_1} \ldots x_{u_\ell v_\ell}$ from T_t by:

- Either $k < \ell$, or
- $k = \ell$ and for the largest w such that $j_w \neq v_w$ it holds that $j_w < v_w$.

The statement of Part 2 of the lemma is proved by induction on $\preceq$. Let $x_c = x_{i_1 j_1} \dots x_{i_k j_k}$ with $i_1 < \dots i_k$. Assume $x_c \notin \Delta_t$ (otherwise there is nothing to prove) and that the statement holds for all terms $\prec$-smaller than x_c. Rewrite x_c using axiom Q_{i_1} as

$$x_{i_2 j_2} \dots x_{i_k j_k} \; + \sum_{j_1^* < j_1} x_{i_1 j_1^*} \dots x_{i_k j_k} \; + \sum_{j_1^* > j_1} x_{i_1 j_1^*} \dots x_{i_k j_k}$$

(where we automatically delete terms with $j_1^* \in \{j_2, \dots, j_k\}$). The first term as well as the terms in the first summation are all $\prec$-smaller than x_c and their domain is included in $dom(c)$. Hence the statement for them follows by the induction hypothesis.

The terms in the last summation can be thought off as all possible moves of pigeon i_1 in the attempted dance of c. Now rewrite each this term analogously w.r.t. pigeon i_2, etc.. But as $x_c \notin \Delta_t$ the pigeon dance cannot be completed and this rewriting will eventually produce only terms $\prec$-smaller than x_c.

22.5 The necessity of partially defined random variables

In this section we leave the specifics related to Problem 22.0.1 and pause to reflect on what are the differences in the construction of Skolem functions in $K(F_{php}, G_{php})$ from those in structures $K(F_{tree}, G_{tree})$ and $K(F_{alg}, G_{alg})$. One may expect to tackle similar issues in the eventual construction of the wanted model of $Q_2 V_1^0$ in which PHP fails.

Some differences in the constructions of Skolem functions in these structures were a priori necessary as Skolem functions in $K(F_{tree}, G_{tree})$ and $K(F_{alg}, G_{alg})$ are in a sense "too good": They imply not only bounded quantifier elimination but also the preservation of true $s\Pi_1^{1,b}$-sentences. This is a property that no model that is hoped to have proof complexity applications can have, as we have discussed in Part V.

The qualification "too good" means that structures $K(F_{tree}, G_{tree})$ and $K(F_{alg}, G_{alg})$ have Skolem functions providing bounded quantifier elimination

in the following very strong sense. Let $B(x,X)$ be a $\Sigma^{1,b}_0$-formula. Then for any $m \in \mathcal{M}_n$ and any Γ there is an open formula $C(x)$ such that not only it holds that

$$(1) \qquad [\![\forall x < m\ B(x,\Gamma) \equiv C(x)]\!] \ = \ 1_{\mathcal{B}}$$

but even for any α

$$(2) \qquad \langle\!\langle \alpha < m \to B(\alpha,\Gamma)\rangle\!\rangle \triangle \langle\!\langle \alpha < m \to C(\alpha)\rangle\!\rangle \ \in \ \mathcal{I}\ .$$

In other words, the symmetric difference has an infinitesimal counting measure. Property (2) is what implies the preservation of true $s\Pi^{1,b}_1$-sentences and what therefore had to fail in $K(F_{php}, G_{php})$, and must so in any other model with proof complexity ambitions. Recall briefly the argument from Section 12.2: Let $A(x)$ be of the form $\forall X < xB(x,X)$, formula $B(x,X)$ as above. Assuming that $\forall x A(x)$ is true in $\mathbf{N}$ then for all m and any Γ it holds that

$$\langle\!\langle \alpha < m \to B(\alpha,\Gamma)\rangle\!\rangle \ = \ \Omega$$

and hence, by (2),

$$\Omega \ \setminus \ \langle\!\langle \alpha < m \to C(\alpha)\rangle\!\rangle \ \in \ \mathcal{I}$$

which implies, as $C(x)$ is open, $[\![\alpha < m \to C(\alpha]\!] = 1_{\mathcal{B}}$ and hence, by (1), $[\![\alpha < m \to B(\alpha,\Gamma)]\!] = 1_{\mathcal{B}}$ too. Note for a record that a completely analogous argument shows that the existence of Skolem functions with properties (1) and (2) implies that for any $\Sigma^{1,b}_0$-sentence D with parameters it holds that

$$[\![D]\!] \ = \ \langle\!\langle D\rangle\!\rangle \ / \ \mathcal{I}\ .$$

Namely, by (1) $[\![D]\!] \ = [\![D']\!]$ for some quantifier free sentence D', for which therefore holds $[\![D']\!] = \langle\!\langle D'\rangle\!\rangle / \mathcal{I}$, and (2) thus implies the observation.

Let us try next to understand in general terms where was any room in the construction of $K(F_{php}, G_{php})$ for the failure of property (2).

The switching lemma for PHP-trees (Theorem 21.1.1) is more subtle and has a harder proof than the original switching lemma in the plain Boolean case (Theorem 10.1.2). This is, however, not the only difference between the analysis of structures $K(F_{tree}, G_{tree})$ and $K(F_{php}, G_{php})$. The additional difficulty stemmed from the fact that random variables involved in the definition of $K(F_{php}, G_{php})$ were only partially defined. Hence there were suddenly two sources of imprecision in the construction of Skolem functions: One had to do with the fact that the switching lemma works only with some (albeit

exponentially small) error, the other one had to do with the regions of undefinability of random variables in question. The former property is shared by the Razborov-Smolensky approximation method; it too introduces an exponentially small error. As it turned out, it is exactly the latter property, the seemingly innocent fact that we allow partially defined random variables what caused the failure of property (2) above.

The idea of the constructions of Skolem functions in Theorems 11.1.1 and 15.1.1 is the same although its technical implementations are different in both cases (the use of the switching lemma and the Razborov-Smolensky approximation, respectively). Also the analysis why the construction actually works is the same in both cases (based on bounding the total size of the error sets). The construction of Skolem functions for $K(F_{php}, G_{php})$ was again formally same as before (utilizing the PHP switching lemma, i.e. Theorem 21.1.1) but what was different was the analysis why it actually worked.

Let us discuss this point in some detail and again in general terms, as it may shed light on the wanted construction. First an example. As pointed out earlier property (2) of Skolem functions implies, in particular, that for an open formula $C(x)$ the truth value $[\![\exists x < m\ C(x)]\!]$ is equal to $\langle\langle \exists x < m\ C(x)\rangle\rangle/\mathcal{I}$. Even this corollary of (2) is no longer true in $K(F_{php}, G_{php})$. Take the formula $\exists x \in [n+1]\ \Delta(x) \notin [n]$. Clearly $\langle\langle \exists x \in [n+1]\ \Delta(x) \notin [n]\rangle\rangle = \Omega_{php}$ but by the analysis in Lemma 21.3.3 $[\![\exists x \in [n+1]\ \Delta(x) \notin [n]]\!] = 0_{\mathcal{B}}$. The set $\langle\langle \exists x \in [n+1]\ \Delta(x) \notin [n]\rangle\rangle$ equals to $\bigcup_{i\in[n+1]}\langle\langle \Delta(i) \notin [n]\rangle\rangle$. By the definition $\Delta(i)$ is the depth 1 PHP tree querying $i \to ?$ and with the leafs labeled by $j \in [n]$. Hence $\langle\langle \Delta(i) \in [n]\rangle\rangle$ is represented by the same tree with all labels changes to 1 (yes). Then also $\langle\langle \Delta(i) \notin [n]\rangle\rangle$ is represented by the same tree but now with all labels 0 (no). Hence the eventual Skolem function looks at a disjunction of no's and cannot find any witness for the formula.

Let us now briefly repeat the analysis of the earlier constructions and point out where we needed to be more subtle in the case of Skolem function for $K(F_{php}, G_{php})$ than before. In the constructions of a Skolem function for an open formula $B(x, y)$ on m we seek first to find β_i such that

$$[\![\exists y < mB(i, y)]\!] = [\![\beta_i < m \wedge B(i, \beta_i)]\!]$$

and then take for the Skolem function the function determined by the tuple $(\beta_i)_{i<m}$. For this to work we need to pick β_i in such a way that no α can find, given a sample, with more than an infinitesimal probability an i such

that the sample is in the error set

$$E_i \ := \ \langle\langle \exists y < mB(i,y) \rangle\rangle \setminus \langle\langle \beta_i \wedge B(i,\beta_i) \rangle\rangle \ .$$

In the proofs of Theorems 11.1.1 and 15.1.1 this is achieved by finding β_i such that each E_i has not only an infinitesimal counting measure but even exponentially small measure. This then guarantees that their union has an infinitesimal counting measure and that is what is aimed at.

In the case of $K(F_{php}, G_{php})$ this analysis would have run into a problem. Each β_i is undefined on a set whose counting measure is infinitesimal but not exponentially small. The region of undefinability of each β_i may have the measure $\Omega(n^{-\Omega(1)})$ so already $n^{O(1)}$ of these sets can sum up to a set covering the whole sample space (and m can be even much bigger than that).

Fortunately a more careful analysis (that would work in earlier cases too but was not needed) showed that this can be circumvented. Recall the idea here on a simple example, an existential sentence. Let $C(y)$ be an open formula (possibly with parameters) and $m \in \mathcal{M}_n$. Assume we want to find β (a 0/1-valued PHP-tree) that signals the validity of $\exists y < m \, C(y)$; that would be the first step, analogously with earlier constructions of Skolem functions, in a construction of a Skolem constant γ such that:

$$[\![\exists y < m \ C(y)]\!] \ = \ [\![\gamma < m \wedge C(\gamma)]\!] \ .$$

Consider 0/1-valued PHP-trees S_j representing $\langle\langle C(j) \rangle\rangle$, for $j < m$. Let D_j be the disjunction of map-terms corresponding to paths in S_j labeled by 1, and let $D := \bigvee_{j<m} D_j$. Assume PHP-tree R represents D. We would like to argue that β defined by R works for our purposes.

The first argument that comes to mind, quite as in earlier constructions of Skolem functions, is to argue that R represents $\bigcup_{j<m} \langle\langle C(j) \rangle\rangle$. We ignore the sample (an infinitesimal fraction of all samples) where β is undefined. If $\beta(T)$ is defined and $\beta(T) = 1$, then $D(T) = 1$ and hence also $D_j(T) = 1$ for some $j < m$. By Lemma 20.1.5 also $S_j(T) = 1$ and hence $T \in \langle\langle C(j) \rangle\rangle$. Note that this in itself does not necessarily implies anything like

$$[\![\beta = 1]\!] \ \leq \ [\![\exists y < m \ C(y)]\!] \ .$$

For this one needs to complete the construction of γ in the binary tree fashion employed in the earlier constructions. We will not discuss this case here in detail (see Section 21.4) as the second case is more interesting for our general discussion.

In the second case $\beta(T)$ is defined but equal to 0. Then $D(T) = 0$ and all $D_j(T) = 0$. However, $D_j(T) = 0$ does not imply that also $S_j(T) = 0$ (which would yield the wanted $T \notin \langle\langle C(j)\rangle\rangle$). It only implies that either $S_j(T) = 0$ or T is in the region of undefinability of S_j. And we have no a priori control over how large is the union of these individual regions; it can be the whole sample space.

The argument that did work was a slightly more subtle. Instead of arguing that β represents $\bigcup_{j<m}\langle\langle C(j)\rangle\rangle$ we argued that for any α, β represents a predicate containing $\langle\langle \alpha < m \wedge C(\alpha)\rangle\rangle$.

Ignoring the $\alpha < m$ part, $\langle\langle C(\alpha)\rangle\rangle$ is represented by a PHP-tree $\Gamma(\alpha)$, where Γ is the map defined by the m-tuple $(S_j)_{j<m}$. So it suffices to verify that

$$\Gamma(\alpha)(T) = 1 \ \rightarrow \ \beta(T) = 1$$

if they are both defined. If $\Gamma(\alpha)(T) = 1$ then $\alpha(T)$ is defined and equal to some $j < m$, and $S_j(T)$ is defined and equal to 1. So $D_j(T) = 1$ and $D(T) = 1$, and $\beta(T) = 1$ too. The point is however, that the set of T for which $\Gamma(\alpha)$ or β are undefined is of an infinitesimal measure.

In other words, we relied in Section 21.4 on a general fact that if we have PHP-trees α and S_j for $j < m$, all of depth at most d, and we attach trees S_j arbitrarily to the leafs of α then the region of undefinability of the resulting tree is not the sum of the regions of undefinability of the individual trees. It is much smaller, its counting measure is only $O(2d/n)$. This is due to the key Lemma 20.1.3.

In the light of the constructions of the earlier structures and the above discussion we may search for a structure constructed as follows. The sample space Ω is formed by suitable set of some (not all) partial one-to-one maps from $[n+1]$ to $[n]$. Note that all $\omega \in \Omega$ satisfy all equations defining $\hat{S}$.

The random variables underlying the structure would be defined by some algebraic trees. It seems that in the algebraic context the notation introduced after Lemma 14.1.4 is more natural. Consider an expression

$$\alpha \ := \ \sum_{i \in u} p_i \cdot i^*$$

where $u \in \mathcal{M}_n$, all $p_i \in \hat{S}^{low}$ satisfying in the ring

$$p_i \cdot p_j = 0 \quad , \text{for } i \neq j < u \ .$$

This defines a map from Ω to $\mathcal{M}_n$: $\alpha(\omega)$ equals to the unique $i < u$ such that $p_i(\omega) = 1$ if it exists, and is undefined if all $p_i(\omega) = 0$. Hence α is possibly only partially defined. Denote by $\alpha(\omega) \uparrow$ the fact that α is undefined at ω.

The crucial condition to arrange, analogous to key Lemma 20.1.3, is that

$$\mathrm{P}rob_{\omega\in\Omega}[\alpha(\omega) \uparrow] \quad \textit{if infinitesimal} \ .$$

One needs to arrange a few simple closure properties of family F in order show that open comprehension and open induction are valid in the structure, and that the construction of Skolem functions as in $K(F_{alg}, G_{alg})$ goes through.

There is a natural choice for a map Δ, a second order object in the structure, that would violate PHP in the structure. Take

$$\Delta \ := \ (\beta_1, \ldots, \beta_{n+1})$$

where

$$\beta_i \ := \ \sum_{j\in[n]} x_{ij} \cdot j^* \ .$$

This is quite analogous to $K(F_{php}, G_{php})$. The requirement is then that F is closed under Δ.

All these conditions seem to require a fine balancing to arrange at the same time.

Part VII

Polynomial-time and higher worlds

Chapter 23

Relevant theories

We shall turn in this part to theories adequate for polynomial time or stronger computational models. These include Cook's theory PV, Buss's theory S_2 and, in particular, its subtheories S^1_2, T^1_2, and theory $BT = (S^1_2 + dWPHP(\Delta_1))$. We will use mostly the first order formulation of these theories (with the exception of theories U^1_1 and V^1_1). We give definitions of the theories but otherwise do not recall much of a background that is not directly used later. The reader may find a much more detailed exposition in [55].

Some of the first order theories are presented in the second order formalism in [29].

23.1 Theories PV and $Th_\forall(L_{PV})$

Cook's theory PV (standing for Polynomially Verifiable) was a first theory that we today include in the collection of bounded arithmetic theories. Its original definition in [27] presented PV as an equational theory but we shall follow later presentations and define the theory as first order (this is often called PV_1 but we will abuse the notation slightly and stick just with PV).

Roughly speaking the theory uses a language with a name for every polynomial time algorithm and its axioms codify how these algorithms are built one from another. The names for polynomial time algorithms are introduced following Cobham's [25] characterization of polynomial time functions on $\mathbf{N}$. This characterization says that the class of polynomial time function is the minimal class of functions $\mathcal{C}$ such that

1. constant 0 and functions $s_0(x)$, $s_1(x)$ and $x\#y$ are in $\mathcal{C}$

where

$$s_0(x) := 2x \quad , \quad s_1(x) := 2x+1 \quad \text{and} \quad x\#y := 2^{|x|\cdot|y|}$$

(with $|x|$ being the bit length of number x) and

2. $\mathcal{C}$ is close under permutation of arguments of functions and under compositions, and

3. $\mathcal{C}$ is closed under limited recursion on notation.

The limited recursion on notation defines a new function $f(\overline{x}, y)$ from functions g, h_0, h_1, ℓ if it holds:

(a) $f(\overline{x}, 0) = g(\overline{x})$.

(b) $f(\overline{x}, s_i(y)) = h_i(\overline{x}, y, f(\overline{x}, y))$, for $i = 0, 1$.

(c) $f(\overline{x}, y) \leq \ell(\overline{x}, y)$.

The language L_{PV} of PV has names for the functions from 1., as well as names for functions introduced by 2. and 3., and relation $\leq$ (and equality which is included automatically). Axioms of PV are universal formulas saying how a function was introduced, and the scheme of induction for open formulas. For example, if f was introduced by the limited recursion on notation as above, PV has as axioms the four formulas in (a), (b) and (c).

We remark that any open induction axiom can be also formulated as a universal formula using a function (with a name in L_{PV}) simulating the binary search (see [55] for details). Thus PV is a universal theory.

Note that the open induction suffices to prove that if functions f and f' were introduced by the limited recursion on notation from two sets of functions g, h_0, h_1, ℓ and g', h'_0, h'_1, ℓ' respectively, such that $g = g'$, $h_i = h'_i$ and $\ell = \ell'$, then also $f = f'$.

Theory $Th_\forall(L_{PV})$ is axiomatized by all true universal sentences in L_{PV}. Of course, we have no explicit description what these sentences are but the theory is useful in connection with witnessing theorems.

23.2 Theories S_2^1, T_2^1 and BT

The language of Buss's theory T_2 extends language L_{PA} (see Section 5.1) by function symbols

$$|x| \ , \ \lfloor \frac{x}{2} \rfloor \ \text{ and } \ x\#y \ .$$

The theory is axiomatized by a set of axioms (called BASIC) defining "basic" properties of the symbols in the language, and by the scheme of induction IND for bounded formulas in the language (as in Chapter 6).

BASIC naturally extends set PA^- from Chapter 6 by adding axioms concerning the new function symbols. We shall not spell them out (cf. [11, 55]). In fact, one often takes for a convenience L_{PV} as the language. Then also all axioms of PV are included into BASIC and the theory is denoted $T_2(PV)$.

Buss [11] considered also two other forms of induction; the LIND scheme (Length IND):

$$A(0,\overline{y}) \wedge \forall x(A(x,\overline{y}) \rightarrow A(x+1,\overline{y})) \rightarrow \forall x A(|x|,\overline{y})$$

and the PIND scheme (Polynomial IND):

$$A(0,\overline{y}) \wedge \forall x(A(\lfloor \frac{x}{2} \rfloor,\overline{y}) \rightarrow A(x,\overline{y})) \rightarrow \forall x A(x,\overline{y}) \ .$$

He proved that when the schemes are accepted for all bounded formulas they both define (over BASIC) theories equivalent to T_2. S_2 is the theory based on $PIND$ for bounded formulas. Hence $T_2 = S_2$.

However, fragments of these theories obtained by restricting the schemes IND and PIND to bounded formulas of certain complexity, are not known to be equivalent (and are not expected to be).

The relevant hierarchies Σ_i^b and Π_i^b, $i = 0, 1, \ldots$, of bounded formulas is defined completely analogously with the arithmetical hierarchy Σ_i^0, with the roles of bounded and unbounded quantifiers in the definition of the latter hierarchy taken by sharply bounded quantifiers $\exists y < |t|$ and $\forall y < |t|$ (the variables bounded above by the length of a term) and by bounded quantifiers respectively. In particular, Σ_0^b formulas contain only sharply bounded quantifiers (similarly as Σ_0^0 contain only bounded ones) and Σ_1^b include all formulas resulting from Σ_0^b formulas by prefixing a bounded existential quantifier while Π_1^b include all formulas resulting from Σ_0^b formulas by prefixing a bounded universal quantifier.

Theories T_2^i and S_2^i are then subtheories of T_2 and S_2 obtained by restricting the schemes IND and PIND, respectively, to Σ_i^b formulas only.

Denote $S_2^i(PV)$ the theory with L_{PV} as its language. A key fact proved by Buss (cf.[11, 55]) is that $S_2^1(PV)$ is conservative over PV for all Σ_1^b formulas. More results about the theories will be recalled when needed.

23.3 Theories U_1^1 and V_1^1

Theories U_1^1 and V_1^1 have the same language as V_1^0. Theory V_1^1 extends V_1^0 by accepting the scheme of induction IND for all $\Sigma_1^{1,b}$ formulas (recall these formulas from Section 5.1).

Theory U_1^1 extends V_1^0 by accepting instead the scheme of polynomial induction PIND for all $\Sigma_1^{1,b}$ formulas. It is known that U_1^1 is contained in V_1^1.

I shall say more about the theories when we need it but now let us recall only one simple model-theoretic correspondence between V_1^1 and S_2^1. Having a model $\mathbf{A} = (A, \mathcal{A})$ of V_1^1, with A the numbers and $\mathcal{A}$ the bounded sets of the model, we may construct a model B of S_2^1 as follows: the elements (numbers) of B are represented by $\mathcal{A}$, thinking of a set as of a string of bits defining a number. One can then define in $\mathbf{A}$ the interpretation of the language of S_2^1 on B and the structure will be a model of S_2^1.

On the other hand, having a model B of S_2^1 take for A the sharply bounded numbers of B, i.e. the lengths of elements of B, and for $\mathcal{A}$ the (necessarily bounded) subsets of B coded in B. The resulting structure will satisfy V_1^1. See [55] for details.

Chapter 24

Witnessing and conditional independence results

In this chapter we give new proofs to some of the known witnessing theorems and conditional independence results in bounded arithmetic. We take, for the sake of examples, the following three important results, but it will be clear how to reprove in the same way other results of this sort. The three results are:

(1) $PV \neq S^1_2$, assuming that $NP \not\subseteq P/poly$ (cf.[77]).

(2) $S^1_2 \neq T^1_2$, assuming that $P^{NP} \neq P^{NP}[O(\log n)]$ (cf. [53]).

(3) S^1_2 does not prove the weak pigeonhole principle for polynomial time functions, assuming that RSA is secure or assuming that strong collision free hash families exists (cf. [60, 75]).

The qualification *new* means that statements of Section 2.2 and Chapter 3 replace the use of proof theory in the original arguments. A price worth to pay for this is that we need to modify the complexity assumptions from the worst-case hardness to the average-case hardness.

We formulate the respective witnessing theorems only as parts of proofs of the independence results. One reason is that they are immediate consequences of the general facts from Chapter 3. Another reason for not trying to formulate them universally is that in the coming constructions sample spaces vary from case to case and are always tailored to the particular independence results.

It will be clear that analogous results for higher theories S^i_2 and T^i_2 with $i > 1$ can be proved identically by simply allowing the random variables in the particular L-closed families to be sampled by algorithms of the type used in the base case $i = 1$ but with an access to an oracle from an appropriate level of the polynomial time hierarchy.

We will treat the three statements in reverse order, from simpler constructions to harder ones. Recall that L_{PV} is the language of theory PV, having a name for every polynomial-time algorithm (cf. Section 23.1).

The definitions of the models follow the original first order version of Section 1.3. In particular, the cut $\mathcal{M}_n$ makes no (explicit) appearance.

24.1 Independence for S^1_2

Let $WPHP(C, a)$ be the existential L_{PV}-formula formalizing the weak pigeonhole principle:

- *Either C is not a circuit computing a map from $\{0,1\}^{|a|+1}$ into $\{0,1\}^{|a|}$ or there are $u \neq v \in \{0,1\}^{|a|+1}$ such that $C(u) = C(v)$.*

For a number k let RSA_k denotes the set of all pairs (g, N) such that

- N is a product of two primes (denoted p and q) of length k (i.e. $|N| = 2k$).
- $1 < g < N$ and g is coprime to $(p-1)(q-1)$.

By an RSA function based on such a pair (g, N) we mean function

$$x < N \to g^x \bmod N \ .$$

The following theorem alludes to statement (3).

Theorem 24.1.1 *Assume that there is a standard $\epsilon > 0$ such that no polynomial time algorithm can find the secret key for more than $(1 - \epsilon)$ fraction of RSA functions based on a pair from RSA_k, for all k large enough.*

Then S^1_2 (even augmented by $Th_{\forall_1}(L_{PV})$) does not prove the sentence

$$\forall C, a \; WPHP(C, a) \ .$$

Proof :

Consider the set of pairs $RSA_{n/2}$, n our non-standard parameter, as the sample space. Let F_{PV} be the family of polynomial time computable functions with domain $RSA_{n/2}$.

Let $\alpha \in F_{PV}$ be a function that from sample $(g, N) \in RSA_{n/2}$ computes a circuit $C_{g,N}$ computing the map $x \in \{0,1\}^{n+1} \longrightarrow g^x \bmod N \in \{0,1\}^n$. Note that the statement "any $C_{g,N}$ is a circuit with $n+1$ inputs and n outputs" is $K(F_{PV})$-valid as it is a true universal statement (Lemma 1.4.2). For the same reason PV is valid in the structure.

Witnessing claim: *Assume* $\forall C, a\ WPHP(C, a)$ *is* $K(F_{PV})$*-valid. Then for any standard* $\epsilon > 0$ *there is a* $\beta \in F_{PV}$ *such that:*

$$\mu([\![\beta(g, N) \text{ is a pair } u \neq v \in \{0,1\}^{n+1} \text{ s.t. } g^u \equiv g^v \bmod N]\!]) \ \geq\ 1 - \epsilon\ .$$

In particular,

$$Prob_{(g,N)}[\ \beta(g, N) \text{ is a pair } u \neq v \in \{0,1\}^{n+1} \text{ s.t. } g^u \equiv g^v \bmod N\ ] \ \geq\ 1-\epsilon\ .$$

If the sentence was valid then after instantiating the universal quantifiers by α for C and by 2^n for a Lemma 3.3.2 guarantees the existence of the required β, for any standard $\epsilon > 0$. The inequality for the probability follows then by Lemma 2.2.1.

However, a pair of $u \neq v$ provided by the Claim gives $w := u - v \neq 0$ such that $g^w \equiv 1 \bmod N$, and it is well-known that having such w is enough to break the particular RSA. Hence, using β, we could do it for at least a $(1-\epsilon)$ fraction of all parameters from RSA_n, any standard $\epsilon > 0$. That contradicts the hypothesis of the theorem.

The theorem follows by Lemma 1.4.1 noting that S^1_2 is $\forall\exists$-conservative over PV (by [11]) which is $K(F_{PV})$-valid as it is a true universal theory (Lemma 1.4.2). The same argument shows that we can allow $Th_{\forall_1}(L_{PV})$ as extra axioms.

q.e.d.

The alternative hypothesis in statement (3) about hash families can be treated analogously (cf. [60]).

We leave it to the reader to verify that one can similarly prove, assuming the existence of strong one-way permutations, that S^1_2 does not prove the following principle for polynomial time functions:

- *Any injective map* $f : \{0,1\}^n \longrightarrow \{0,1\}^n$ *must be onto.*

24.2 S^1_2 versus T^1_2

Next we turn to statement (2). Let Ξ be a family of counter-example functions (cf. Section 2.2), one for each formula $\forall z \leq t(x)A(x,z)$, A an open L_{PV}-formula and $t(x)$ an L_{PV}-term.

We shall denote by $FP^{\Xi}[q(n)]$ the class of functions computable by a polynomial time oracle machine that is allowed at most $q(n)$ queries to a function from Ξ.

Note that a query to an NP-oracle $A(u) = \exists v \leq t(u)B(u,v)$ can be answered if we have an access to a counter-example function for $\neg A(u)$. Hence the class $FP^{\Xi}[q(n)]$ contains the bounded query classes $FP^{NP}[q(n)]$ of [17, 106] and, in fact, also the bounded query classes with witness-oracles $FP^{NP}[wit, q(n)]$ of [20, 53].

Theorem 24.2.1 *Assume that there is a P^{NP} function such that no bounded query $P^{NP}[O(\log n)]$ algorithm can compute the predicate on more than $(1-\epsilon)$ fraction of all inputs, some $\epsilon > 0$.*

Then $S^1_2 \neq T^1_2$.

Proof :

Let F be the family of functions from $\bigcup_c FP^{\Xi}[c \cdot \log(n)]$ with domain $\Omega := \{0,1\}^n$, some nonstandard n, where the constant c in the union ranges over standard $c \geq 1$.

Claim 1: *S^1_2 is $K(F)$-valid.*

F is an L_{PV}-closed family, so PV is $K(F)$-valid. The claim follows (via Lemma 1.4.1) from the fact that S^1_2 is axiomatized over PV by the sharply bounded function minimization scheme $\#FM(C,a)$:

- *If C is a circuit computing on $\{0,1\}^{|a|}$ a function whose values are numbers $< |a|$ then the minimal value is assumed on some $u \in \{0,1\}^{|a|}$,*

cf.[23, 38].

Clearly the minimal value $t < |a|$ can be found by binary search with $\log |a| = O(\log n)$ (as a priori $|a| = n^{O(1)}$) queries to an NP-oracle, and then a suitable u can be found by one call to a counter-example function for formula $\forall z(|z| \leq |a|)(f(z) \geq t+1)$. Hence the principle $\forall C, a\ \#FM(C,a)$ is witnessed by a function in F and so is, by Lemma 1.4.2, $K(F)$-valid (substituting a witnessing function for the existential quantifier in an $\forall\exists\forall$-sentence transforms it into a true universal sentence).

Witnessing claim 2: *Assume T^1_2 is $K(F)$-valid. Then for any P^{NP} function $f(x)$ there is $\beta \in F$ such that:*

$$\Pr_{\omega}[\beta(\omega) = f(\omega)] \geq 1 - \epsilon \ ,$$

where ω ranges over Ω.

By [11] the graph $f(x) = y$ of any P^{NP} function can be defined by an $\exists\forall$ L_{PV}-formula $\exists z \forall t A(x, y, z, t)$ such that T^1_2 proves that it is well-defined:

$$\forall x \exists y \exists z \forall t A(x, y, z, t) \ .$$

If T^1_2 were valid in $K(F)$, so would be this formula. Assume that it is so. By Lemma 3.3.3, for any standard $\epsilon > 0$, there are $\beta, \beta' \in F$ such that

$$\mu([\![\forall t A(\mathrm{id}_\Omega, \beta, \beta', t)]\!]) \ \geq \ 1 - \epsilon \ .$$

Lemma 2.2.2, as F is closed under counter-example functions, then implies the inequality from the claim.

The theorem now follows from the claim, using the hypothesis that there exists f and $\epsilon > 0$ such that the inequality from the claim cannot hold for any β.

q.e.d.

We remark that predicates in $P^N[O(\log n)]$ are exactly those in L^{NP} (logspace with an NP-oracle), cf. [17, 106] or [55].

24.3 PV versus S^1_2

Finally we turn to statement (1).

Theorem 24.3.1 *Assume that no polynomial size circuit can find a satisfying assignment for more than a $(1-\delta)$ fraction of satisfiable formulas of size k, for some $\delta > 0$ and all $k > 0$ large enough.*

Then PV (even $Th_{\forall_1}(L_{PV})$) does not prove S^1_2.

Proof :

Let $Sat(x,y)$ be an open L_{PV} formula formalizing that x *is a propositional formula and* y *is a satisfying assignment for* x.

We will use it only for formulas x of size n (i.e. encoded by size n strings), and we think of truth assignments y as being encoded by strings of size n too, but we suppress the parameter n in the formulas.

The sample space Ω of a structure to be defined shortly is the set of all n-tuples

$$\omega = (\omega_1, \ldots, \omega_n)$$

where all ω_i are different satisfiable formulas of length n. In particular, samples are n^2-tuples of bits.

Before we define the family F of random variables of the model we need to define a certain formula and a specific counter-example function. Let $f(x,y)$ be a polynomial time function whose domain is $(\{0,1\}^n)^n \times (\{0,1\}^n)^n$, i.e. both $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ are n-tuples of strings of length n, and which is defined as:

$$f(x,y) \ := \ \max_{0 \le i \le n} \bigwedge_{j \le i} Sat(x_j, y_j) \ .$$

The sentence Max_f is

$$\forall x \exists y \forall z \ f(x,y) \ge f(x,z)$$

(with x, y and z restricted to strings of size n^2, as above).

Claim 1: *The sentence* Max_f *is provable in* $S^1_2(PV)$.

The sentence is an instance of the Σ^b_1-LENGTH-MAX principle that is available in S^1_2. Alternatively, note that the natural proof by induction uses Σ^b_1-LIND which is known to be equivalent to S^1_2. See [11, 55] for details.

Let $h(x)$ be a counter example function for formula

$$\forall y \ \neg Sat(x,y)$$

(again implicitly restricted to size n formulas). That is, $h(x)$ is a satisfying assignment for formula x if any exists at all.

Using h we can define a specific counter example function h^* for the formula

$$\forall z \ f(x,y) \ge f(x,z) \ .$$

Function $h^*(x,y)$ is computed by a polynomial time algorithm that calls h as an oracle: On input (x,y) (of size n^2) computes $i := f(x,y)$ and outputs z which is equal to y except that y_{i+1} is replaced by $h(x_{i+1})$.

Now we are ready to define F: it is the smallest class of functions defined on Ω that contains all PV functions and $h^*(\mathrm{id}_\Omega, y)$. In particular, F is L_{PV}-closed and it is closed under h^*. Hence PV is $K(F)$-valid (by Lemma 1.4.2).

Note that any function in F is defined by a standard term of $L_{PV} \cup \{h^*(\mathrm{id}_\Omega, y)\}$ and any such term contains at most a standard number of occurrences of h^*.

To show that $S^1_2(PV)$ is not valid in $K(F)$, and hence PV does not prove $S^1_2(PV)$, it suffices to demonstrate that the sentence Max_f is not $K(F)$-valid. This follows from the next claim, using the hypothesis of the theorem.

Witnessing claim 2: *Assume that the sentence Max_f is $K(F)$-valid. Then for any standard $\delta > 0$ there is a circuit of size n^e, some standard $e \in \mathbf{N}$, that finds a satisfying assignment for at least a fraction $(1-\delta)$ of satisfiable formulas of size n.*

To prove the claim assume for the sake of a contradiction that the sentence is $K(F)$-valid. As F is closed under h^*, a counter-example function for formula

$$\forall z \; f(\mathrm{id}_\Omega, y) \geq f(\mathrm{id}_\Omega, z)$$

the assumption together with Lemma 2.2.2 imply that for an arbitrary standard $\epsilon > 0$, the value of which we shall fix later, there exists element $\beta \in F$ such that

$$\Pr_{\omega \in \Omega}[f(\omega, \beta(\omega)) \text{ is maximal possible }] \;\geq\; 1 - \epsilon \; .$$

Assume that the term defining β contains c calls to $h^*(\mathrm{id}_\Omega, y)$, some fixed standard $c \geq 1$.

Let Ω_0 and Ω_1 be the sets of $(c+1)$-tuples and of $(n-c-1)$-tuples of satisfiable formulas of size n. Hence $\Omega = \Omega_0 \times \Omega_1$. An averaging argument shows that there is an element $b \in \Omega_1$, a tuple of $(n-c-1)$ satisfiable formulas, such that

$$\Pr_{(\omega_1,\ldots,\omega_{c+1}) \in \Omega_0}[f((\omega_1,\ldots,\omega_{c+1}, b), \beta((\omega_1,\ldots,\omega_{c+1}, b))) \text{ is maximal possible }]$$

$$\geq \; 1 - \epsilon \; .$$

The rest of the argument is similar (although not identical) to the argument from [77], see also [55, L.10.2.2]; we give its brief sketch.

The term computing $\beta((\omega_1,\ldots,\omega_{c+1},b))$ asks for $h^*((\omega_1,\ldots,\omega_{c+1},b),y)$, for up to c different values of y. Let V_0 be the set of satisfiable formulas of size n that do not occur among b. For a c-element subset $Q \subseteq V_0$ and a formula $a \in V_0 \setminus Q$ we say that Q *helps* a if for some ordering $a_1,\ldots,a_{c+1}$ of $Q \cup \{a\}$ term β computes a satisfying assignment for a without asking (via h^*) for $h(a)$. In other words, knowing satisfying assignments for formulas from Q and also for all formulas in the tuple b allows us to compute one also for formula a.

There is the complication that term β works correctly for at least a fraction of $(1-\epsilon)$ of all $(c+1)$ tuples of different formulas from V_0 but not for all. Call such a $(c+1)$-tuple *good* if β works correctly for it, and call a formula a *good* if it occurs in at least one good tuple. All good tuples can draw elements only from good formulas. Hence if $1-\delta'$ is the proportion of good formulas in V_0 it must hold

$$1-\epsilon \leq (1-\delta')^c < 1-\delta' \ ,$$

i.e. $\epsilon > \delta'$. Hence taking ϵ to be the δ from Claim 2 will guarantee that there is at least a fraction of $(1-\delta)$ good formulas in V_0. We will construct a polynomial size circuit computing a satisfying assignment for all good formulas as well as for formulas from b; this is at least a fraction of $(1-\delta)$ of all satisfiable formulas.

Let V_1 be the good formulas in V_0. Let m_1 be the size of V_1. There are $\binom{m_1}{c}$ possible sets Q while there are $\binom{m_1}{c+1}$ sets $Q \cup \{a\}$. Hence there is some Q_1 that helps at least

$$\frac{\binom{m_1}{c+1}}{\binom{m_1}{c}} \ = \ \frac{(m_1-c)}{c+1}$$

of formulas $a \in V_1$.

Collect into a future advice the $(n-c-1)$ formulas occurring in b, the c formulas from Q_1 and also satisfying truth assignments for all these $(n-1)$ formulas. This is $(n-1)2n$ bits of advice so far.

Now proceeds analogously as before but with set $V_2 := V_1 \setminus Q_1$ in place of V_1. Let m_2 be the cardinality of V_2. Find set $Q_2 \subseteq V_2$ of c formulas that help at least a fraction of $\frac{(m_2-c)}{c+1}$ of formulas from V_2. Add this set of c formulas together with some satisfying assignments for them ($2cn$ bits in total) to the advice, and put for the next step $V_3 := V_2 \setminus Q_2$.

The cardinalities of the sets $V_1, V_2, \ldots$ decrease exponentially and in $O(n)$ steps get below some fixed (standard) bound, say below c. Add to the advice also the remaining formulas and some satisfying assignments to them.

The total advice has size $O(n^2)$. The polynomial time algorithm using this advice (i.e. a polynomial size circuit) that finds a satisfying assignment for all good formulas and also for formulas in the advice works as follows. Given a, first check if it is one of the formulas in the advice. If so, output the assignment encoded in the advice. Otherwise attempt to find a satisfying assignment to a using term β to all tuples obtained from all possible orderings of sets $Q_i \cup \{a\}$, $i = 1, 2, \ldots$ completed to an n-tuple by b. It follows form the definition of sets Q_i that if a is good, the algorithm succeeds.

q.e.d.

24.4 Transfer principles

The topic of this section does not really fit with the rest of the chapter (or into any other one) and would warrant its own chapter. However, it is a bit peripheral in relation to our main line of investigation and we shall discuss it here just briefly. It may be possibly useful in some future constructions.

Consider a structure $K(F)$ defined by a family of random variables F on some sample space Ω. An element $\alpha \in F$ can be viewed from two different perspectives: as an element of the structure $K(F)$ or as an element of the ambient model $\mathcal{M}$, a function defined on Ω obeying some particular restrictions defining F. Thus properties of elements of $K(F)$ can be translated into properties of (some) elements of $\mathcal{M}$, and vice versa. This observation was first used, I believe, by Takeuti [101] who considered Boolean valued models of set theory. Below we note one specific example of such a translation to illustrate this phenomenon.

Let us consider structure $K(F_{PV})$, with F_{PV} being the family of polynomial time functions defined on $\{0,1\}^n$ (see Section 3.2 or Section 24.1 in this Chapter).

Let Q be a propositional proof system in the sense of Cook and Reckhow [30], see Chapter 27. Let $\alpha_Q \in F_{PV}$ be Q restricted to $\{0,1\}^n$. In particular,

$$[\![\alpha_Q \text{ is a formula }]\!] = 1_{\mathcal{B}}$$

but even more:

$$[\![Taut(\alpha_Q)]\!] = 1_{\mathcal{B}}$$

where $Taut(x)$ is a bounded universal formula $\forall y(|y| \leq |x|)Sat(x,y)$, with $Sat(x,y)$ being an open L_{PV}-formula defining that relation "y is a truth assignment satisfying formula x" (as in the proof of Theorem 24.3.1).

To see this note that

$$[\![Taut(\alpha_Q)]\!] = \bigwedge_{\beta} [\![|\beta| \leq |\alpha_Q| \rightarrow Sat(\alpha_Q, \beta)]\!]$$

and that for any β

$$\mathrm{P}rob_{\omega\in\{0,1\}^n}[\ |\beta(\omega)| \leq |\alpha_Q(\omega)| \rightarrow Sat(\alpha_Q(\omega), \beta(\omega))\ ] \ = \ 1$$

as only tautologies get into the range of Q.

Now let P be another proof system (represented in $K(F_{PV})$ by its symbol in L_{PV}) and assume

$$[\![P \vdash \alpha_Q]\!] = 1_{\mathcal{B}}$$

that is

$$[\![\exists z P(z) = \alpha_Q]\!] = 1_{\mathcal{B}}\ .$$

Using the properties of witnessing quantifiers it follows that for every standard $\epsilon > 0$ there is a polynomial time function γ such that

$$\mathrm{P}rob_{\omega}[\ P(\gamma(\omega)) = Q(\omega)\ ] \ > \ 1 - \epsilon\ .$$

In other words, γ translates Q-proofs into P-proofs of the same formula, i.e. it is a polynomial simulation of Q by P (albeit with an error ϵ). Let us call such a function an *approximate p-simulation*, leaving the quantification via ϵ out of this discussion (the parameter ϵ could be avoided if we used a compact family, e.g. functions computable in subexponential time, in place of F_{PV}, cf. Section 3.5).

A function $\sigma \in F_{PV}$ may satisfy the equation

$$[\![Taut(\sigma)]\!] = 1_{\mathcal{B}}$$

even if σ is not a proof system in the Cook and Reckhow sense. Such a σ may make errors but these should be hard to detect. In particular, it must hold that for all polynomial time β

$$\mathrm{P}rob_{\omega}[\ |\beta(\omega)| \leq |\sigma(\omega)| \rightarrow Sat(\sigma(\omega), \beta(\omega))\ ] \ > \ 1 - \epsilon$$

for all standard $\epsilon > 0$. A polynomial time function σ satisfying this condition may be called a *pseudo proof system*.

Thus properties of formulas and their proofs can be translated, via $K(F_{PV})$, into properties of pseudo proof systems and their approximate p-simulations, and vice versa.

Chapter 25

Pseudorandom sets and a Löwenheim-Skolem phenomenon

Let $K(F)$ be an L-structure with Ω its sample space. There seems to be a natural concept of a substructure of $K(F)$: a structure $K(F_0)$ with the same sample space but with possibly a smaller, L-closed family of random variables $F_0 \subseteq F$. As the sample space of both structures is the same, so is the Boolean algebra $\mathcal{B}$, and the atomic sentences using parameters from F_0 get the same truth-values. This mirrors well the classical concept of a substructure.

There is, however, another possibility how to create structures that are in a good sense substructures of $K(F)$. Instead of taking a smaller family of random variables, take a smaller sample space $\Omega_0 \subseteq \Omega$ (we need to maintain the general requirement for the method that Ω_0 is infinite) but the same family F.

Let us use the following notation: $\mathcal{A}_0$, $\mathcal{B}_0$ are the resulting Boolean algebras, $[\![\ldots]\!]_0$ the truth values in $\mathcal{B}_0$, and μ_0 will be the (standard) measure on $\mathcal{B}_0$. As the Boolean algebras of truth-values are different in each case we cannot compare the truth-values of $L(F)$-sentences in the two structures. But we can compare their measures.

Assume the space Ω_0 is chosen in such a way that:

(*) *For any atomic formula* $A(x_1, \ldots, x_k)$ *and any* $\alpha_1, \ldots, \alpha_k \in F$ *it holds:*

$$\mu([\![A(\alpha_1, \ldots, \alpha_k)]\!]) \;=\; \mu_0([\![A(\alpha_1, \ldots, \alpha_k)]\!]_0) \;.$$

We will say that language L is closed under definition by cases by open formulas if every open L-formula is equivalent in $\mathbf{N}$ (recall that always $L \subseteq L_{all}$) to an atomic one. Such an equivalence is a universal statement and hence will be valid in any L-closed structure by Lemma 1.4.2.

In the next lemma we use the notation just introduced.

Lemma 25.0.1 *Assume L is closed under definition by cases by open formulas. Assume $\Omega_0 \subseteq \Omega$, $\Omega_0 \in \mathcal{M}$ and infinite, and F is an L-closed family. Assume also that the hypothesis (*) above is satisfied.*

Then for any $L(F)$-sentence A it holds:

$$\mu([\![A]\!]) \;=\; \mu_0([\![A]\!]_0) \;.$$

In particular,

$$[\![A]\!] = 1_{\mathcal{B}} \;\; \textit{iff} \;\; [\![A]\!]_0 = 1_{\mathcal{B}_0} \;.$$

Proof :

The statement is proved by induction on the logical complexity of the sentence. For atomic sentences this is the hypothesis (*) and for open sentences this follows from the assumption that L is closed under definition by cases by open formulas, i.e. any open formula is equivalent to an atomic one.

Now assume A has the form $\exists x B(x)$, where B is an $L(F)$-formula. Family F is closed under definition by cases by open formulas (this follows from the corresponding assumption about L) and hence by Lemma 3.3.2 the following holds: For any $k \in \mathbf{N}$ there is $\alpha_k \in F$ such that

$$\mu([\![A]\!]) \;\geq\; \mu([\![B(\alpha_k)]\!]) \;>\; \mu([\![A]\!]) \;-\; \frac{1}{k} \;.$$

Hence $\mu([\![A]\!])$ is the limit of values $\mu([\![B(\alpha_k)]\!])$. By induction assumption

$$\mu([\![B(\alpha_k)]\!]) = \mu_0([\![B(\alpha_k)]\!]_0) \;.$$

It follows that $\mu([\![A]\!]) \leq \mu_0([\![A]\!]_0)$. Identically one proves the opposite inequality.

The universal quantifier is treated analogously.

q.e.d.

Now we link these considerations with the concept of a pseudorandom distribution from computational complexity theory. We will restrict the discussion to language L_{PV}, sample space $\Omega = \{0,1\}^n$ and to the family F_{PV} of random variables computed by (standard) polynomial-time algorithms.

On the other hand we need to consider a bit more general form of Ω_0; instead of mere subsets $\{0,1\}^n$ we need to allow Ω_0 to be a probabilistic distribution on $\{0,1\}^n$. Such a distribution assigns to elements of $\{0,1\}^n$ arbitrary probabilities (defined in $\mathcal{M}$) summing up to 1 on the whole space. In defining $\mathcal{B}_0$ from $\mathcal{A}_0$ counting measure is replaced by the new measure. In the text below we assume this more general definition of μ_0 and $[\![\ldots]\!]_0$.

The following two definitions are weaker versions of the standard definitions, cf. [35]. The weakening is that for our purposes it suffices to consider indistinguishability w.r.t uniform polynomial time algorithm while in the standard definitions one needs to allow randomized polynomial time algorithms or even non-uniform polynomial time algorithms (i.e. polynomial size circuits).

Definition 25.0.2 *Families of probabilistic distributions* $(\Delta_k)_k$ *and* $(\Gamma_k)_k$ *on* $\{0,1\}^k$, $k \in \mathbf{N}$, *are called* computationally indistinguishable, *denoted*

$$(\Delta_k)_k \equiv_c (\Gamma_k)_k \ ,$$

if the following condition holds.

For any unary predicate $P \subseteq \{0,1\}^*$ *that is computable in polynomial time and for any* $c \in \mathbf{N}$ *we have:*

$$\mid Prob_{x \in \Delta_k}[x \in P] \ - \ Prob_{y \in \Gamma_k}[y \in P] \mid \ < \ k^{-c}$$

for all $k \in \mathbf{N}$ *large enough.*

The probabilities in the definition are computed according to the respective distributions.

Definition 25.0.3 *Family of probabilistic distributions* $(\Delta_k)_k \subseteq \{0,1\}^k$, $k \in \mathbf{N}$, *is called* pseudorandom *iff*

$$(\Delta_k)_k \equiv_c (\{0,1\}^k)_k \ .$$

where $(\{0,1\}^k)_k$ *denotes the family of uniform distributions.*

Lemma 25.0.4 *Assume $(\Delta_k)_k \subseteq \{0,1\}^k$ is a pseudorandom family of probabilistic distributions. Take for Ω_0 the distribution Δ_n on $\{0,1\}^n$, and $\Omega := \{0,1\}^n$ (i.e. the uniform distribution). Let $L = L_{PV}$ and $F = F_{PV}$.*

Then (using the notation above) for all $L_{PV}(F_{PV})$-sentences A it holds

$$\mu([\![A]\!]) \;=\; \mu_0([\![A]\!]_0) \;.$$

In particular,

$$[\![A]\!] = 1_{\mathcal{B}} \;\; \textit{iff} \;\; [\![A]\!]_0 = 1_{\mathcal{B}_0} \;.$$

Proof :

The lemma follows from Lemma 25.0.1 once we verify the hypothesis (*). To do this note that any atomic $L_{PV}(F_{PV})$ sentence $A(\alpha_1, \ldots, \alpha_k)$ defines a polynomial time predicate on Ω: $\omega \in \langle\langle A(\alpha_1, \ldots, \alpha_k)\rangle\rangle$.

By the definition of the pseudorandomness it follows that the counting measure and the measure in the sense of Δ_n of

$$\langle\langle A(\alpha_1, \ldots, \alpha_k)\rangle\rangle$$

differ at most infinitesimally. Hence

$$\mu([\![A(\alpha_1, \ldots, \alpha_k)]\!]) = \mu_0([\![A(\alpha_1, \ldots, \alpha_k)]\!]_0) \;.$$

q.e.d.

It is not known how to define a pseudorandom distribution efficiently, in the sense to be described shortly. But we note a simple consequence of the Chernoff's inequality, using the fact that the set of standard polynomial time algorithm is included in an $\mathcal{M}$-definable family of algorithms of arbitrarily small non-standard cardinality.

Lemma 25.0.5 *For any $s > \mathbf{N}$ there is an infinite distribution $\Omega_0 \subseteq \{0,1\}^n$, $\Omega_0 \in \mathcal{M}$, of cardinality*

$$|\Omega_0| \leq n^s$$

and such that for any $P \subseteq \{0,1\}^n$ definable by a (standard) polynomial time algorithm and for any $c \in \mathbf{N}$ we have:

$$\mid \mathit{Prob}_{x \in \Omega_0}[x \in P] \;-\; \mathit{Prob}_{y \in \{0,1\}^n}[y \in P] \mid \;<\; n^{-c} \;.$$

The set Ω_0 defines a structure and satisfies the properties stated in Lemma 25.0.4. But note that Lemma 25.0.5 does not suggest how to define Ω_0 efficiently, it only asserts its existence.

There is a conjectural way how to find such distributions in a way one can list its elements by a polynomial time algorithm or, more precisely, one can sample the distribution in polynomial time. This uses a now accepted hypothesis (some even say "standard" hypothesis) about the existence of a pseudorandom number generator (or, equivalently, a one-way function). This conjecture is crucial for the foundation of cryptography based on computational complexity, cf.[35].

We will formulate the next definition only for $\{0,1\}^n$. Note that because we talk again only about uniform algorithms the next definition is weaker than the standard one too.

Definition 25.0.6 *A polynomial time function $g : \{0,1\}^m \to \{0,1\}^n$ is a* pseudorandom number generator *if $m < n$ and the distribution giving to $\omega \in \{0,1\}^n$ the probability*

$$\frac{|g^{(-1)}(\omega)|}{2^m}$$

is a pseudorandom distribution on $\{0,1\}^n$. In other words,

$$\mid Prob_{x\in\{0,1\}^m}[g(x) \in P] \; - \; Prob_{y\in\{0,1\}^n}[y \in P] \mid \; < \; n^{-c}$$

holds for any polynomial time predicate P on $\{0,1\}^n$ and any $c \in \mathbf{N}$.

A pseudorandom generator g defines a distribution on $\{0,1\}^n$ that can be sampled as follows: pick $u \in \{0,1\}^m$ uniformly at random and take sample $\omega := g(u)$. This can be used to define a structure $K(F_g)$ as follows:

- The sample space is $\{0,1\}^m$.
- Family F_g consists of functions $\beta : \{0,1\}^m \to \mathcal{M}_n$ that have the form

  $$\beta \; = \; \alpha \circ g$$

  for some $\alpha \in F_{PV}$.

It is then clear that this structure has the same properties as the structure from Lemma 25.0.4. We state just the latter property.

Lemma 25.0.7 *Assume g is a pseudorandom generator. Then any L_{PV} sentence A is valid in $K(F_{PV})$ iff it is valid in $K(F_g)$.*

Pseudorandom generators are conjectured to exists for any $m \leq n^\epsilon$, arbitrary standard $\epsilon > 0$. The sample space of $K(F_g)$ has then the size at most 2^{n^ϵ}, i.e. at most

$$2^{(\log |\Omega|)^\epsilon} .$$

In the non-uniform construction of sample space Ω_0 in Lemma 25.0.5 the size is even much smaller, less then

$$(\log |\Omega|)^s ,$$

arbitrarily small but non-standard s. It thus seems coherent to interpret these constructions as a Löwenheim-Skolem type phenomenon.

Chapter 26

Sampling with oracles

In this chapter we define two fairly simple models of PV, a first order $K(F_{oracle})$ and its second order expansion $K(F_{oracle}, G_{oracle})$. The latter structure allows for a natural interpretation of structural complexity results about random oracle. It is also one of the structures we shall try to tweak a bit in Part VIII.

26.1 Structures $K(F_{oracle})$ and $K(F_{oracle}, G_{oracle})$

Take a non-standard $n > \mathbf{N}$ and put $h := 2^{n^n}$ (we only need that h is bigger than any 2^{n^c}, any $c \in \mathbf{N}$). The sample space is

$$\Omega_{oracle} \ := \ \{0,1\}^h \ .$$

The family of random variables F_{oracle} consists of all function

$$\alpha \ : \ \Omega_{oracle} \to \mathcal{M}$$

such that there is a (standard) polynomial time oracle machine M^Q such that for all samples $\omega \in \Omega_{oracle}$:

$$\alpha(\omega) \ := \ M^{\omega}(1^{(n)}) \ .$$

As the machine is polynomial time any potential oracle query has the length bounded above by n^c, some $c \in \mathbf{N}$, and hence all samples ω contain answers to all such queries.

Lemma 26.1.1 *F_{oracle} is L_{PV}-closed and closed under definitions by cases by open $L_{PV}(F_{oracle})$ formulas.*

An immediate corollary of the lemma and Lemma 1.4.2 is this.

Corollary 26.1.2 *$Th_\forall(L_{PV})$ is valid in $K(F_{oracle})$. In particular, PV is valid in the structure.*

Now we will put a second order structure, a family of functions G_{oracle}, on top of $K(F_{oracle})$.

An element $\Theta \in G_{oracle}$ is a map

$$\Theta \ : \ F_{oracle} \to F_{oracle}$$

determined by a polynomial time oracle machine N^Q and defined by

$$\Theta(\alpha)(\omega) \ := \ N^\omega(\alpha(\omega)) \ .$$

It is clear that $\Theta(\alpha)$ is computed (on input $1^{(n)}$ by the composition of machines M^Q computing α with the machine N^Q.

Note that any L_{PV} function symbol has a natural interpretation in G_{oracle}: a PV function f computed by a polynomial time machine N is represented by the element of G_{oracle} determined by N (which asks no oracle queries). In this sense the following holds.

Lemma 26.1.3 *There is a family $G_{PV} \subseteq G_{oracle}$ such that PV is valid in the structure $K(F_{oracle}, G_{PV})$.*

26.2 An interpretation of random oracle results

We will demonstrate in this section that structural complexity results of the form that something happens in the relativized world with probability 1 for a random oracle can be naturally interpreted in $K(F_{oracle}, G_{oracle})$.

One can imagine a random oracle as a subset of $\mathbf{N}$ to which any number gets with probability $\frac{1}{2}$. This is sound if one studies (as it is usually the case) a behavior of an algorithm of a limited time or space complexity on inputs of a bounded size (and hence the size of oracle queries is bounded

too). This view of random oracles obviously generalizes to viewing a random oracle as a random element of $\{0,1\}^m$ for a non-standard m. A more general approach considers measurable sets of oracles (i.e. subsets of the power set of $\mathbf{N}$) and the results talk about measures of sets of oracles having a certain property. This actually also relates to viewing random oracles as a random element of $\{0,1\}^m$ as the counting measure is a non-standard refinement of the Lebesgue measure. We will skip details of these comments and appeal just to the intuition of the reader.

A typical result in this area is the following.

Theorem 26.2.1 (Bennett-Gill [9]) *For a random oracle $A \subseteq \mathbf{N}$, with probability 1 it holds:*

$$P^A \neq NP^A .$$

Let $R^Q(x,y)$ be a binary relation defined by a polynomial time oracle machine $N^Q(x,y)$. Assume for simplicity of formulas that there is a constant $c \in \mathbf{N}$ such that for any oracle A:

$$R^A(x,y) \rightarrow |y| \leq |x|^c .$$

Let R^Q be a relation witnessing Theorem 26.2.1; that is, no polynomial time oracle machine M^Q satisfies

$$\forall x \ (\exists y N^A(x,y)) \rightarrow N^A(x, M^A(x))$$

with a non-zero probability for a random oracle $A \subseteq \mathbf{N}$.

Take $\Theta \in G_{oracle}$ determined by N^Q. The random oracle result has the following interpretation in $K(F_{oracle}, G_{oracle})$.

Theorem 26.2.2 *The open formula $\Theta(x,y) = 1$ has no Skolem function in $K(F_{oracle}, G_{oracle})$. In particular, for all $\Gamma \in G_{oracle}$*

$$[\![\forall x, y \ (\Theta(x,y) = 1 \rightarrow \Theta(x, \Gamma(x)) = 1)]\!] = 0_{\mathcal{B}} .$$

We remark that it is natural to define a non-uniform version of structure $K(F_{oracle}, G_{oracle})$ where the role of polynomial time oracle machines is taken by decision trees (as in $K(F_{rud}, G_{rud})$) of depth $\leq n^c$, arbitrary $c \in \mathbf{N}$.

It is not without interest to note that this non-uniform version can be seen as a "short" version of $K(F_{rud}, G_{rud})$. Namely, if we denote the length of samples in both cases ℓ then the depth of the trees as well as the length

of the values used in $K(F_{rud}, G_{rud})$ is $\ell^{o(1)}$ while in the non-uniform version of $K(F_{oracle}, G_{oracle})$ it is only $(\log\log\ell)^{O(1)}$. (The double log is because of our generous bound 2^{n^n} in the definition of the sample space but 2^{n^t} would suffice for any non-standard t and hence the estimate should be more like $(\log\ell)^{o(1)}$.)

Part VIII

Proof complexity of EF and beyond

Chapter 27

Fundamental problems in proof complexity

In this last part of the book we look through the eyes of forcing with random variables on some problems about strong proof systems. We start by recalling in this chapter briefly the very general background of proof complexity and by spelling out a few open problems that we consider fundamental for further development. This exposition is included in order to motivate our choice of topics we study in later chapters. Details can be found in the references given in the Introduction (or in the text below).

Cook and Reckhow [30] defined a general **proof system** for propositional logic to be a polynomial time computable function P defined on $\{0,1\}^*$ whose range is exactly the set TAUT of propositional tautologies in DeMorgan language $0, 1, \wedge, \vee, \neg$ (see Chapter 17). Any string that P maps to a tautology τ is called a P-proof of τ. This definition subsumes usual calculi for propositional logic one encounters in textbooks and, in particular, those defined in Chapter 17: Given such a calculus interpret it as a function that maps a string that is a valid proof to the formula being proved, and all other strings to some fixed tautology (for example, to 1). Such a function will satisfy the Cook-Reckhow definition as in all logical calculi it is a recognizable by a p-time algorithm whether a string is a valid proof or not. Note that the fact that the range of the function constructed in this way from a propositional calculus is exactly the set TAUT is equivalent to the soundness and the completeness of the calculus.

The main question about proof systems is whether there exists one in

which all tautologies have a proof of size polynomially bounded in terms of the size of the tautology; such a proof system is called **p-bounded** (in some original texts it was called *super*). Cook and Reckhow [30] observed that the existence of a p-bounded proof system is equivalent to the statement that the computational complexity class NP is closed under complementation, i.e. that NP = coNP. It is widely believed that NP $\neq$ coNP (although perhaps not as widely as P $\neq$ coNP) and hence that for every proof system P there is an infinite set of tautologies whose shortest P-proofs cannot be bounded by a polynomial in their length (in fact, it is expected that such hard tautologies can be found for which the shortest P-proofs must have exponential size). In principle thus one can show that NP is not closed under complementation, and consequently that also P $\neq$ NP, by proving sufficiently strong and general lengths-of-proofs lower bounds[1]. The main problem of proof complexity is therefore:

Show that no p-bounded proof system exists (i.e. that NP $\neq$ coNP).

There are many proof systems utilizing ideas from logic, combinatorics, algebra or geometry, and superpolynomial lower bounds are known only for a few of them (see [55, 88, 67] for surveys). The best-understood among them is resolution, which is of independent interest for automated theorem proving. Problems to demonstrate lower bounds for stronger proof systems, based on more complicated mathematical ideas, have led to increasingly complex combinatorial problems. At present we cannot even rule out that a Frege system (see Definition 17.1.2) is p-bounded.

A specific place among lower bound methods has feasible interpolation. It is a method that arguably applies to the most varied class of proof systems. Here one shows that from a short proof of disjointness of the intersections of two NP sets with $\{0,1\}^n$ it is possible to extract a feasible algorithm that separates them. Shorter proofs leads to more efficient algorithms. Hence pairs of disjoint NP sets that are hard to separate yield formulas that need long proofs. Computational complexity theory does not offer such examples unconditionally (suitable examples follow from the conjectured existence of one-way functions) but sometimes there is a "monotone" version of the argument that needs only lower bounds for monotone Boolean circuits (or other "monotone" computational models), and such lower bounds are sometimes known. This method, and its variants, applies to resolution, cutting planes

[1] This approach to the P vs. NP problem is sometimes called Cook's program.

proof system, algebraic and geometric proof systems, and recently also to the OBDD proof system. However, various cryptographic assumptions (e.g. the security of the RSA) are known to imply that the method does not apply to strong proof systems (but see [70]; there can be found also references to earlier papers).

It seems therefore that combinatorics alone is inadequate to tackle strong proof systems. This may be reminiscent of Boolean complexity: The combinatorial problems arising there are very complicated and this hindered severely a progress during the last twenty years. But proof complexity is fortunate to have another - genuinely non-combinatorial - facet: links with bounded arithmetic theories. These theories capture the informal concept of "effective" reasoning in a way analogous to how polynomial time algorithms (and other restricted classes of circuits like AC^0, TC^0, etc.) capture the notion of a feasible algorithm. One can translate between proofs in these weak first-order systems and short proofs in commonly studied propositional proof systems; one can be thought of as a uniform version of the other. The upshot is that one can think of lengths-of-proofs lower bounds as about problems how to construct suitable models of particular bounded arithmetic. This is what we have formulated in Part V (Chapter 18)[2].

The existence of a p-bounded proof system (i.e. the NP vs. coNP problem) appears to be a very hard problem and it makes a good sense to consider also (presumably) easier problems which nevertheless retain the spirit of the main problem. The following three problems seem to me to be particularly stimulating:

(1) Prove the conjecture that NP $\neq$ coNP from a plausible computational hardness assumption.

(2) Prove a super-polynomial lengths-of-proofs lower bound for Extended Frege system EF.

(3) Decide whether or not there exists a p-optimal (or an optimal) proof system.

Concerning Problem (1) recall that there are several fundamental conjectures in computational complexity theory that do follow from computational

[2]There are also many direct connections of bounded arithmetic with algorithmic complexity, see [55, 29].

hardness assumptions about Boolean circuits. These assumptions do have a rather similar form:

- *Every Boolean circuit performing a particular, explicitly given, computational task has to be large.*

These include the conjectures that P $\neq$ NP, P = BPP (i.e. that the randomization does not increase the power of p-time algorithms), and the conjecture that a strong pseudorandom generator useful for cryptography exists. In particular, using the trivial inclusion of P $\subseteq$ P/poly we can reduce P $\neq$ NP to the hardness assumption that

- SAT cannot be decided by polynomial size Boolean circuits.

Using Nisan-Wigderson generators (cf. Section 29.4) Impagliazzo and Wigderson [47] proved that P = BPP assuming that

- Boolean circuits computing a complete language in the exponential time class E (time bound $2^{O(n)}$) must have the size at least $2^{\Omega(n)}$.

And finally, Hastad et.al.[40] proved that strong pseudorandom generators exists, assuming that a one-way function exists. In particular, we have a reduction of the existence of a strong pseudorandom generator to the following concrete hardness assumption:

- Factoring is intractable by polynomial size Boolean circuits. In particular, polynomial size circuits can factor only a subpolynomial fraction of integers that are products of two primes of comparable sizes.

It would thus be quite interesting (and mathematically appealing) to have also a reduction of the conjecture that NP $\neq$ coNP to an assumption of a similar form. See [66] for more discussion of this topic.

Problem (2) is significant for a variety of reasons (see [52]). Extended Frege system is considered to be a pivotal case in proof complexity research should we progress from proving lower bounds for (seemingly) weak systems and attack strong systems. There is also another reason having to due with the link of EF to Cook's theory PV (cf. Chapter 23 for PV). It is known that the consistency of P $\neq$ NP with PV would follow from *any* superpolynomial lower bound for Extended Frege proofs. Such a consistency would be a breakthrough as a large part of contemporary complexity theory can be formalized in PV (or in its mild extension).

Problem (3) raises a quite fundamental issue interesting on its own. If we cannot resolve the existence of a p-bounded proof system the next natural question is whether there exists a proof system that is in a sense the strongest, as good as any other proof system. The right (quasi)ordering of proof systems, called **p-simulation**, has been defined already in [30]: Proof system P p-simulates Q iff there exists a p-time function f such that for all $w \in \{0,1\}^*$ we have $P(f(w)) = Q(w)$. In other words, simulation f translates Q-proofs into P-proofs of the same formulas. As f is p-time it has also polynomial growth: the length of $f(w)$ is at most polynomially longer than that of w. If we keep the requirement of the polynomial growth but drop the p-time computability of f we talk of **simulation**. We say that P is **p-optimal** (resp. optimal) iff it p-simulates (resp. simulates) any other proof system Q. Starting with Krajíček and Pudlák[72] this question (and its non-uniform version) has been studied from many angles and links with various other topics in complexity theory and mathematical logic have been found (cf.[55]). In particular, the non-existence of a p-optimal proof system implies that EXP $\neq$ NEXP (exponential time with $2^{n^{O(1)}}$ bound) and hence also P $\neq$ NP, and Krajíček [63] has proved that at least one of the following three conjectures is true:

- There is a function $f : \{0,1\}^* \to \{0,1\}$ computable in E that has circuit complexity $2^{\Omega(n)}$.
- NP $\neq$ coNP.
- There is no p-optimal propositional proof system.

There is a problem about proof search closely related to Problem (3). Namely:

(3') Is there an optimal way of searching for proofs of unsatisfiability?

As mentioned earlier, proof complexity of resolution is related to automated theorem proving. In fact, known algorithms essentially search for a proof in resolution or in some of its variants (e.g. tree-like resolution). It is a quite interesting question whether one can define an algorithm searching for proofs in a stronger proof system that is "significantly better" than any algorithm yielding resolution proofs. In any case, searching for proofs is a classical topic and it is connected with an early history of computational complexity: a version of the P vs NP problem is a question posed by Gödel [34] about searching for proofs. There is also a formal link between problems (3) and

(3'): Krajíček and Pudlák [72] proved that a p-optimal proof system exists iff there exists a deterministic algorithm recognizing TAUT that has at most polynomial slow-down over any other such algorithm on inputs from TAUT.

Problem (3) has several other interesting facets. It has deep links with quantitative versions of Gödel's theorem (cf.[72]) and with NP search problems. Interestingly, Cook and Krajíček [28] defined a concept of a proof system with advice and showed that there is an optimal system in that class.

All this discussion aims at justifying the following problem as a worthwhile target of a research:

Prove a super-polynomial lower bound for EF from a plausible computational hardness hypothesis .

It is clearly a common weakening of Problems (1) and (2) and, in fact, there are also links (via proof complexity generators, cf. Chapter 29) to Problem (3).

Our strategy for an attack on this problem is clear: Using an approach analogous to Chapter 18 we want to find a structure (formed by a family of random variables) in which the soundness of EF is valid while some tautology is not. These two properties of the model are de facto properties of the family of random variables forming the model, and the statement that the family indeed has the properties will be the hypothesis to which the lengths-of-proofs lower bound for EF will reduce. Of course, the all-important qualification *plausible* hypothesis should not slip our mind.

Let us remark that in this respect forcing with random variables relates to feasible interpolation mentioned earlier, where a lower bound is also reduced to a computational hypothesis (see the earlier discussion).

Chapter 28

Theories for EF and stronger proof systems

In this chapter we recall the correspondence of EF with some bounded arithmetic theories. We shall consider both first-order and second-order theories as both possibilities are convenient for different constructions.

28.1 First-order context for EF: PV and S^1_2

For the definitions of theories PV and S^1_2 see Chapter 23. The relation between EF and PV discovered by Cook [27] was, in fact, the first example of a general correspondence between proof systems and bounded arithmetic theories (cf.[72]). In the following we briefly summarize the relation for the case of EF and PV; the same relation of EF to S^1_2 follows from the $\forall\Sigma^b_1$-conservativity of $S^1_2(PV)$ over PV proved by Buss [11], see also Section 23.2.

Let $A(x)$ be a Π^b_1-definition (see Section 23.2) in language L_{PV} of a coNP-set of numbers. Assume $A(x)$ has the form $\forall y(|y| \leq |x|^k \rightarrow B(x,y))$ with $B(x,y)$ an open L_{PV}-formula (defining thus a polynomial time predicate).

Fix length n to bound $|x|$ and construct a propositional formula $||A(x)||^n$ as in the proof of the NP-completeness of satisfiability: the formula has n atoms $p_1,\ldots,p_n$ for bits of an x, $m = n^k$ atoms $q_1,\ldots,q_m$ for bits of a potential y, and also atoms $r_1,\ldots,r_s$ for $s = n^{O(1)}$ bits of values on nodes of a fixed circuit C_n computing from $\overline{p}$, $\overline{q}$ the truth value of predicate $B(x,y)$. Formula $||A(x)||^n$ says, in a DNF form, that if $\overline{r}$ are correctly computed by circuit C_n from inputs $\overline{p}$, $\overline{q}$ then the output of the computation is 1.

Having any b of length n with bits $b(1), \ldots, b(n)$ denote by $||A(x)||^n(b)$ the propositional formula with $b(i)$ substituted for p_i, and with the remaining atoms $\overline{q}$ and $\overline{r}$ left unsubstituted. Clearly then b satisfies $A(x)$ iff $||A(x)||^n(b)$ is a tautology.

The relation between EF and PV is as follows:

(1) If PV proves $\forall x A(x)$ then tautologies $||A(x)||^n(b)$ have polynomial size EF-proofs (and these are definable from $1^{(n)}$ in PV).

(2) PV proves the soundness of EF and for any another proof system Q, if PV proves also the soundness of Q then EF polynomially simulates Q (and the simulation is definable in PV).

Crucially, the first property has a form of a converse that underlines our approach to lower bounds (Chapter 18). We shall formulate this in quite some detail in the next section in the second-order formalism (as this is the one used in Chapter 18 and that we shall use later on).

28.2 Second-order context for EF: VPV and V_1^1

The second-order theory V_1^1 has been defined in Section 23.3. Theory VPV (defined in [29]) is also second-order and has an analogous relation to V_1^1 as does PV have to S_2^1 (this goes under the name RSUV isomorphism, see [55, 29] or the model-theoretic explanation in Section 23.3).

The first-order part of the language of theory VPV is L_{PV}. The second-order part includes symbols for all polynomial time functionals $f(\overline{x}, \overline{X})$ whose arguments are both numbers and bounded sets (i.e. strings) and values can be either numbers or strings too. The restriction *polynomial time* means here polynomial in the length of the string inputs. The axioms for VPV include definitions for all of these functions based on Cobham's Theorem, quite analogous to the definition of PV in Section 23.1. VPV proves (as PV does) the induction scheme for open formulas. It is a universal theory.

The property that is a form of converse to (1) from the previous section is provided by Theorem 18.3.1. But we want to generalize it a bit. Namely, in Chapter 18 we have considered only constant depth formulas (denoted T_k there); here we want to allow any formulas or, in fact, circuits. The reason

for considering only constant depth formulas there was that in the theories corresponding to constant depth Frege systems one cannot prove that an arbitrary formula can be evaluated on an arbitrary truth assignment. In other words, in models of such theories we cannot talk about validity of general formulas. In VPV and V^1_1 any circuit can be evaluated (and in U^1_1 any formula). Hence we may allow T_k to be arbitrary circuits. By Fagin's theorem the satisfiability of circuits can be defined by both $\Sigma^{1,b}_1$- and $\Pi^{1,b}_1$-formulas. Let Sat denote in the next theorem a $\Sigma^{1,b}_1$-definition of satisfiability for which its basic properties can be proved in VPV (cf.[55]).

The theorem stated for EF and VPV and generalized as just explained is as follows.

Theorem 28.2.1 *Let T_k be circuits of size $k^{O(1)}$, for all $k \in \mathbf{N}$. Assume that for an arbitrary choice of non-standard n there is a structure $K(F,G)$ (built over $\mathcal{M}_n$) satisfying the following three conditions:*

(1) $K(F,G)$ is L_n-closed.

(2) VPV is valid in $K(F,G)$.

(3) There is a truth assignment $\Gamma \in G$ to atoms of T_n that falsifies the circuit in $K(F,G)$, i.e. $Sat(n, \Gamma, \neg T_n)$ is valid.

Then there is a standard $\epsilon > 0$ such that for no $k \in \mathbf{N}$ large enough has T_k an EF-proof of size less than 2^{k^ϵ}.

The proof is the same as that of Theorem 18.3.1 (outlined in Chapter 18), using the fact that VPV proves the soundness of EF.

28.3 Stronger proof systems

By a *strong* proof system different people mean different things. Sometimes it is simply meant to be a proof system for which we are unable, at the present time, to demonstrate that it is not p-bounded. In this respect there are countless strong systems. But that appears to be a rather ad hoc criterion.

The choice of EF as the first example of a strong proof system is motivated by several considerations none of which is really convincing on its own but which together seem to make a good case for EF.

First, many weaker proof systems operate not with arbitrary circuits but with some restricted class of circuits. It may thus happen that an eventual lower bound proof for the weaker system will somehow utilize the circuit restriction. EF on the other hand is p-equivalent to a proof system CF, a kind of a Frege system operating with unrestricted circuits[1]. Thus a lower bound for EF cannot use any specific circuit properties.

Second, while we do have several particular proof systems and several classes of proof systems that we think may be stronger than EF (i.e. may p-simulate EF but not be p-simulated by it), they all can be p-simulated by a "simple" extension of EF by a p-time set of extra axioms. The particular proof systems possibly stronger than EF include WF of [49] (see also Section 30.1), a proof system corresponding to S^1_2 extended by the dual weak PHP for p-time functions, or an algebraic proof system UENS (the abbreviation stands for "unstructured extended Nullstellensatz") of [18] which can be p-simulated by WF but it is not known if EF p-simulates it too. The classes of proof system that may be stronger than EF are, for example, quantified propositional calculi G_i of [73] corresponding to theories T^i_2 or implicit proof systems like iEF of [64] corresponding even to $S^1_2 + Exp$ or stronger theories. In fact, any first-order theory T capable of formalization of logic and having a p-time set of axioms (e.g. PA or ZFC) can be interpreted as a proof system: a proof of tautology τ is any T-proof of the formula in the language of T formalizing that "τ is a tautology". But an arbitrary proof system Q can be p-simulated by EF if we add translations $||Ref_Q||^n$, $n \geq 1$, of the reflection principle for Q (here we use the first-order notation for translations). This set of tautologies is p-time and sparse. The corresponding theory would be an extension of PV by a true $\forall\Pi^b_1$-sentence (cf.[72]).

Third, in the propositional world EF has a similar role as in the first-order world has a "sufficiently strong base theory". Some constructions (axiomatizations, p-simulations, constructions of short proofs, etc.) require that a proof system is sufficiently strong and EF functions very smoothly (due to its correspondence with PV) in this respect. It is true that one can often take for the base system something much weaker (frequently even resolution) but it may make the particular construction in question more cumbersome.

Fourth, EF has also the technical advantage of being p-equivalent to a variety of other interesting proof systems, including Extended Resolution ER,

[1]This is often claimed as a folklore but the definition of CF, a Frege system for circuits, is not entirely straightforward and has been properly done only recently by Jeřábek [49].

Substitution Frege SF (and other variants of Frege) or a quantified system G_1^*.

All in all, we believe that the choice of EF as a system personificating "a strong proof system" is a honest one (i.e. no hidden restrictions) but not introducing overly abstract notions.

See [72, 55] for more detailed expositions of the correspondence between general proof systems and bounded arithmetic theories.

Chapter 29

Proof complexity generators: definitions and facts

We have chosen the model-theoretic approach (Chapter 18 and Theorem 28.2.1) as a framework in which to think about lower bounds for EF. However, an all-important ingredient is to also have examples of specific candidate tautologies that could be plausibly conjectured to be hard for EF. The experience with lower bound proofs for weaker proof systems shows that one needs specific, combinatorially transparent, formulas to work with. From the model-theoretic perspective this is not surprising: a lower bound is, in effect, a construction of a model where the formula is not valid, so in order to hope to describe such a model we better understand what the formula means. Of course, one may expect that a random 3DNF (with n variables and $c \cdot n$ clauses, suitable c) is hard for any proof system, and for some specific systems this has indeed been established (see e.g. Chvatal and Szemeredi [22]). However, lower bounds for random formulas are obtained by verifying that in a lower bound proof discovered originally for some different and specific formula only such properties of the formula are used that are also shared by the random one.

Unfortunately the formulas that worked for weak proof systems are demonstrably easy to prove in EF. The foremost example is the PHP principle we have seen earlier (Chapter 19). However, PHP has short proofs in EF (see [30]) and, in fact, even in F (see [12]). Similar remark applies to all combinatorial principles considered earlier, in particular to a variety of the so called counting principles.

A class of tautologies that is provably hard for all non-optimal proof

systems are tautologies formalizing reflection principles. If P is a non-optimal proof system and Q is a proof system that P does not simulate (cf. Chapter 27 for these notions) then formulas $||Ref_Q||^n$ formalizing the soundness of Q w.r.t. proofs of length at most n are demonstrably hard for P. This goes back to [27, 72] and is exposed in detail in [55]. The drawback of these formulas is that one derives their hardness for P from another lengths-of-proofs assumption, namely that Q cannot be simulated by P.

A very interesting class of tautologies expressing the disjointness of suitable examples of disjoint NP pairs was considered in connection with feasible interpolation, see the discussion in Chapter 27. EF and stronger proof systems do not admit feasible interpolation assuming that RSA is secure, cf.[72] and, in fact, the argument from [72] shows that some of these tautologies do have short proofs in EF and hence, assuming their inseparability by polynomial time sets, the infeasibility of interpolation for EF is deduced.

Fortunately there is yet another class of formulas proposed as candidates for hard tautologies. These formulas are defined using specific maps called **proof complexity generators**. The resulting formulas, the so called **τ-formulas**, could be conceivably hard even in very strong proof systems, including EF. In fact, for all we know at present some τ-formulas may be hard for all proof systems. The τ-formulas have been defined by Krajíček [60] and independently by Alekhnovich et.al. [5], and there is now an emerging theory (see [61, 95, 62, 95, 63, 65, 68, 69]); the introductions to [62] or [95] offer a comprehensive discussion of motivations, aims and background (from a bit different perspectives though).

The rest of the chapter is devoted to an exposition of a part of this theory. Section 29.1 introduces the construction of τ-formulas and the definition of the hardness of a map, a proof complexity generator. Then we discuss three specific maps deemed at present as possibly hard for strong proof systems: the truth-table function (Sections 29.2, 29.3 and 30.1), the Nisan-Wigderson generator, and the gadget generator (Section 29.5).

29.1 τ-formulas and their hardness

We shall consider maps

$$g : \{0,1\}^n \to \{0,1\}^m$$

computed by a family of polynomial size circuits $\{C_n\}_n$, and we will assume that $m = m(n)$ is an injective function of n and that $m = m(n) > n$ (the injectivity of $m(n)$ implies that any string is in the range of at most one circuit C_n and that is convenient). Such maps will be called P/*poly*, p-stretching.

As the domain $\{0,1\}^n$ of C_n is smaller than $\{0,1\}^m$, the range of C_n $Rng(C_n)$ is a proper subset of $\{0,1\}^m$. Hence there are strings $b \in \{0,1\}^m \setminus Rng(C_n)$ and for any such b define the τ**-formula** $\tau(C_n)_b(x)$ as:

$$b \not\equiv C_n(x)$$

expressing that $b \notin Rng(C_n)$ (and hence also expressing that $b \notin Rng(g)$ as $Rng(g) \cap \{0,1\}^m = Rng(C_n)$) in the sense that

$$\tau(C_n)_b \text{ is a tautology iff } b \notin Rng(C_n) .$$

Formula $\tau(C_n)_b(x)$ is a disjunction:

$$\bigvee_{i \leq m} b_i \not\equiv C_{n,i}(x)$$

where b_i is the i-th bit of b and $C_{n,i}$ is the (sub)circuit of C_n computing the i-th bit of its output. The variables of $\tau(C_n)_b$ are n variables $x_1, \ldots, x_n$ for bits of a potential element x of $\{0,1\}^n$, and also $n^{O(1)}$ many auxiliary variables used in order to define the computation of C_n on x. When circuits C_n are thought as canonically determined by g we shall denote the formulas simply $\tau(g)_b$.

The main aim is to find maps g for which the resulting τ-formulas are hard to prove in as strong proof systems as possible. To measure the hardness we make, following [95], the following definition.

Definition 29.1.1 *Let g be a P/poly, p-stretching map and let P be a proof system. When all $\tau(g)_b$ require super-polynomial (resp. exponential) size P-proofs we say that g is* **hard** *(resp.* **exponentially hard***) proof complexity generator for P.*

Note that the size of the τ-formulas is polynomial in m which is polynomial in n as well because of our stipulation that g is P/*poly*. However, the length of the formulas would remain polynomial in m even if we allowed m to be super-polynomially bigger than n and the circuits computing g to be

polynomial in m rather than in n. In addition, the property "$b \notin Rng(g)$" can be expressed by a tautology even for maps g whose output bits are defined by non-uniform NP $\cap$ coNP conditions (in terms of the input bits). In fact, we could again allow the non-deterministic time to be polynomial in m rather than in n.

We now observe that the hardness can be interpreted as a sort of a hitting set property (following [62]).

Definition 29.1.2 *Let g be a P/poly p-stretching map computed by circuits $\{C_n\}_n$, and let P be a proof system. The* **resultant** *of g with respect to P, denoted Res_g^P, is the class of all NP/poly-sets $A \subseteq \{0,1\}^*$ such that for some definition of A:*

$$y \in A \text{ iff } \exists z(|z| \leq |y|^k)B(y,z)$$

$B(y,z)$ a P/poly relation, the proof system P admits polynomial-size proofs of the propositional statements

$$||\forall x, z\ (B(y,z) \rightarrow C_n(x) \neq y)||^{m(n)}$$

(with the bounds to the lengths of x and z implicitly polynomial in n).

Lemma 29.1.3 ([62]) *Let g be a P/poly p-stretching map computed by circuits $\{C_n\}_n$. Let $P \supseteq EF$ be a proof system. Then the following two conditions are equivalent:*

1. *There exists $t \geq 0$ such that for infinitely many $n \geq 1$ and $b \in \{0,1\}^{m(n)}$ the formula $\tau(C_n)_b$ has a P-proof of size $\leq |\tau(C_n)_b|^t = |b|^{O(t)}$.*
2. *The resultant Res_g^P contains an infinite set.*

Note also that a P/*poly* p-stretching map g whose range would intersect all infinite NP-sets would be a proof complexity generator hard for any proof system P. This is because for any P and $k \geq 1$ the set of strings b for which the corresponding τ-formula has a P-proof of length at most m^k is in NP.

Similarly, if $Rng(g)$ intersected all NP-sets of density at least $1/2$ it would still follow that for any P and any $k \geq 1$ there is at least one $b \notin Rng(g)$ for m large enough for which $\tau(g)_b$ cannot be proved by a P-proof of size less than m^k. This is because the ambient space $\{0,1\}^m$ is at least twice as large as $Rng(g) \cap \{0,1\}^m$.

In the next few sections we shall introduce three candidates for hard proof complexity generators. Let us remark that it is not true that any potential strong pseudo-random number generator g (SPRNG) will make a good candidate to be a hard proof complexity generator (by a SPRNG we mean a function obeying Definition 25.0.6 but with respect to non-uniform adversaries; see a remark before Definition 25.0.6). For example, assume f is a one-way permutation and g is a SPRNG constructed from f by adding its hard bit. Assuming that EF proves shortly that f is injective it is easy to see that it also proves shortly all τ-formulas from g. This applies, for example, to RSA as shown in [72].

However, there is some use for SPRNGs in this context nevertheless. As [5] noted, if $n < m/2$ and g is a SPRNG, then map

$$h \ : \ (x^1, x^2) \in \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$$

defined by the bit-wise sum

$$h(x^1, x^2) \ := \ g(x^1) \oplus g(x^2)$$

will be a hard proof complexity generator for any proof system that admits feasible interpolation. The proof of this fact is by contradiction: the existence of a short proof of some τ-formula $\tau(h)_b$ together with feasible interpolation would allow to define a P/*poly* set C^* disjoint with $Rng(g)$ but of measure at least $1/2$, contradicting the pseudo-randomness of g. The construction of C^* is quite analogous to the argument in the construction of C^* in the proof of Theorem 29.2.3 below.

29.2 The truth-table function

The truth-table function is the most prominent example of a proof complexity generator.

Definition 29.2.1 *Let $k \geq 1$ be a parameter and let $s = s(k) \leq 2^{(1-\Omega(1))k}$ be a function of k.*

The **truth-table function $\mathbf{tt}_{s,k}$** *takes as an input a description of a circuit C with k inputs and of size at most s, and outputs the 2^k bits of the truth table of the Boolean function C computes.*

A size s circuit is encoded by $O(s \log s) < 2^k$ bits. Hence $\mathbf{tt}_{s,k}$ maps $n := O(s \log s)$ bit strings to $m := 2^k > n$ bit strings.

$\mathbf{tt}_{s,k}$ *is, by definition, equal to zero at inputs that do not encode a size* $\leq s$ *circuit with* k *inputs.*

A basic observation is that the τ-formula $\tau(\mathbf{tt}_{s,k})_b$ expresses that $b \in \{0,1\}^{2^k}$ is the truth-table of a Boolean function which cannot be computed by a circuit of size at most s. That is, the τ-formula expresses a circuit lower bound for b.

This interpretation of the formulas allows us to make two observations.

Theorem 29.2.2 *(1) Assume there is a proof system* P *and* $s(k) \geq 2^{\Omega(k)}$ *such that* $\mathbf{tt}_{s,k}$ *is not hard for* P*. Then BPP* $\subseteq$ *NP.*

(2) Assume there is a proof system P *and* $s(k) \geq k^{\omega(1)}$ *such that* $\mathbf{tt}_{s,k}$ *is not hard for* P*. Then NEXP* $\not\subseteq$ *P/poly.*

Proof :

For (1) consider[1] the following process: Guess a pair (b, π), where b is the truth-table of a function outside of the range of $\mathbf{tt}_{s.k}$ (i.e. with exponential circuit complexity) and π is a polynomial size (i.e. $2^{O(k)}$) P-proof of $\tau(\mathbf{tt}_{s,k})_b$. By [83, 47] such a function b can be used for derandomization of BPP.

Statement (2) follows as [43] proved (by derandomizing MA) that the existence of an NP-way (considered there as a form of NP-natural proofs) of certifying a superpolynomial circuit complexity of a function implies that NEXP $\not\subseteq$ P/*poly*. The hypothesis of (2) provides such a way of certification.

q.e.d.

The statement can be interpreted as saying that it is not to be expected that it can be simple to exhibit a proof system for which the truth-table function is not a hard proof complexity generator. Note also that while it is easy to prove by a counting argument that there exists a function b of a superpolynomial circuit complexity, i.e. $b \notin Rng(\mathbf{tt}_{s,k})$ for $s \geq k^{\omega(1)}$, we do not know how to prove this shortly (in polynomial size) for any particular b in the whole of Mathematics (as formalized e.g. by ZFC).

We shall now make an observation in the opposite direction; we describe a fairly large class of proof systems for which the truth-table function is hard. By a strong pseudo-random generator (SPRNG) in the next statement we mean the same thing as at the end of Section 29.1. The proof of the theorem uses a trick invented by Razborov [92].

[1] See [62]; the observation is attributed to R. Impagliazzo there.

Theorem 29.2.3 *Assume that there exists a SPRNG. Let $s(k) \geq k^{\omega(1)}$. Then the truth-table function $\mathbf{tt}_{s,k}$ is hard for any proof system that admits feasible interpolation.*

Proof :

Assume $\mathbf{tt}_{s,k}$ is not hard for P and let $b \in \{0,1\}^{2^k}$ be a function such that P admits polynomial size proof of

$$\tau(\mathbf{tt}_{s,k})_b \ .$$

Put $t := s/3$ and let x be a 2^k-tuple of atoms. Then P also proves in polynomial size the disjunction

$$\tau(\mathbf{tt}_{t,k})_x \ \vee \ \tau(\mathbf{tt}_{t,k})_{b \oplus x}$$

as two circuits of size $\leq t$ computing some function x and $b \oplus x$ respectively can be combined into a circuit of size $\leq s$ computing b.

By feasible interpolation for P there is a circuit $C(y)$ with 2^k inputs and of size $2^{O(k)}$ defining a subset $C^* \subseteq \{0,1\}^{2^k}$ that separates two sets:

$$A \ := \ \{x \in \{0,1\}^{2^k} \mid x \in Rng(\mathbf{tt}_{t,k})\}$$

and

$$B \ := \ \{x \in \{0,1\}^{2^k} \mid x \oplus b \in Rng(\mathbf{tt}_{t,k})\}$$

in the sense that

$$A \subseteq C^* \ \text{ and } \ C^* \cap B = \emptyset \ .$$

If C^* contains at most half of $\{0,1\}^{2^k}$ put $D(y) := \neg C(y)$, otherwise put $D(y) := C(y \oplus b)$. It follows that if y satisfies D then it is a truth-table of a function that cannot be computed by a circuit of size $\leq t$. In particular, it cannot be computed by a polynomial size circuit.

Such a circuit D is a P/*poly*-natural property against P/*poly* in the sense of Razborov-Rudich's natural proofs [96]. But by the main theorem of [96] the existence of such a property implies that no SPRNG exists. That is a contradiction.

q.e.d.

The hardness of $\mathbf{tt}_{s,k}$ is also known to follow for a proof system if a version of the weak pigeonhole principle with $2^{n^{\Omega(1)}}$ pigeons and n holes is hard for the proof system, cf. [92]. But this condition does not apply even to constant depth Frege systems.

29.3 Iterability and the completeness of $\mathbf{tt}_{s,k}$

For the next definition and for Theorem 29.3.2 we will assume that h is a P/*poly* map that stretches inputs of length n to outputs of length $n+1$ (that is the most general context for Definition 29.3.1 and Theorem 29.3.2). We shall denote by h_n the restriction of h to $\{0,1\}^n$.

Definition 29.3.1 ([62]) *Let $s(n) \geq 1$ be a function. Function h is $s(n)$-* **iterable** *for proof system P iff all disjunctions of the form*

$$\tau(h_n)_{b_1}(x^1) \vee \ldots \vee \tau(h_n)_{b_t}(x^1, \ldots, x^t)$$

require P-proofs of size at least $s(n)$. Here $t \geq 1$ is arbitrary, and $b_1, \ldots, b_t$ are $(n+1)$-tuples of variables and constants such that:

1. *x^i are disjoint n-tuples of atoms, for $i \leq t$.*
2. *$b_1 \in \{0,1\}^{n+1}$ (i.e. no variables in b_1).*
3. *variables occurring in b_i are among $x^1, \ldots, x^{i-1}$, for $i \leq t$.*

If map h satisfies a variant of the definition where b_i are allowed to be any circuits (with $n+1$ outputs) obeying the restrictions 1.-3. it is called $s(n)$- **pseudo-surjective** *for P.*

Note that the $s(n)$-pseudosurjectivity with super-polynomial (resp. exponential) $s(n)$ implies the (exponential) iterability which in turn implies the (exponential) hardness, the latter being the iterability condition with $t = 1$.

The disjunction in the definition can be interpreted as follows (see [62, Sec.3] for a more extensive commentary including an interpretation using an interactive computation among Student and Teacher of [76]). Assume that the disjunction is a tautology. Then it may be that already the first disjunct $\tau(h_n)_{b_1}(x^1)$ is a tautology, meaning that the string b_1 is outside of the range of h_n. If not, and $a^1 \in \{0,1\}^n$ is such that $h_n(a^1) = b_1$, then $b_2(a^1)$ is the next candidate for a string being outside of the range of h_n, etc.. Because the disjunction is a tautology we must find in this process a string outside of the range of h_n in at most t rounds.

The importance of the concept stems from the next theorem.

Theorem 29.3.2 ([62]) *Assume that proof system P simulates resolution. Then the following two statements hold:*

1. *There exists a h (exponentially) iterable for P iff for any $0 < \delta < 1$, the truth table function $\mathbf{tt}_{s,k}$ with $s = 2^{\delta k}$ is (exponentially) iterable for P too.*

2. *There exists a h that is exponentially iterable for P iff there is $c \geq 1$ such that for $s = k^c$ the truth table function $\mathbf{tt}_{s,k}$ is exponentially iterable for P.*

The same holds with iterability replaced by pseudo-surjectivity.

In other words, if we replace hardness by iterability then the truth-table function is the "hardest" function.

29.4 The Nisan-Wigderson generator

We have pointed out at the end of Section 29.1 that not all maps that are considered as plausible candidates for strong pseudo-random generators (SPRNGs) are also good candidates for hard proof complexity generators. However, the argument outlined there why some SPRNGs are not hard from the proof complexity point of view does not seem to apply to the classic Nisan-Wigderson generator (shortly NW-generator), cf. [83]. In fact, the NW-generator was proposed as a possibly hard proof complexity generator in [5] and taken up in [62], although the motivations and, more importantly, the choice of parameters in the two proposals were different.

An NW-generator is defined by a 0-1 matrix A and a Boolean function f. In computational complexity theory the term *NW-generator* usually refers to the original construction of [83], i.e. where A is the so called (d, ℓ) design (see below) and f is a suitably hard function. Here we shall use the term more loosely, meaning any map defined in the following manner.

Let $n < m$ and let A be an $m \times n$ 0-1 matrix with ℓ ones per row. $J_i(A) := \{j \leq [n] |\ A_{ij} = 1\}$. Let $f : \{0,1\}^\ell \to \{0,1\}$ be a Boolean function. Define function $NW_{A,f} : \{0,1\}^n \to \{0,1\}^m$ as follows: The i-th bit of the output is computed by f from the bits of the input that belong to $J_i(A)$. Matrix A is an (d, ℓ)-design if in addition the intersection of any two different rows $J_i(A) \cap J_k(A)$ has size at most d.

Note that the size of the τ-formula from $NW_{A,f}$ is $m^{O(1)}$ if f is computed by a circuit of size $m^{O(1)}$. In fact, that remains true even if f is computed in the non-uniform class $\mathrm{NTime}(m^{O(1)}) \cap \mathrm{coNTime}(m^{O(1)})$.

The idea that $NW_{A,f}$ forms a good proof complexity generator has been formulated in [5] in general terms, assuming that A has a suitable combinatorial property and that f is hard in some sense for the proof system in question. The combinatorial property used in [5] was formulated as an expansion requirement, using the following definition.

Definition 29.4.1 ([5, Def.2.1]) *Let A be an $m \times n$. A boundary of a set of rows $I \subseteq [m]$, denoted $\partial_A(I)$, is the set of $j \in [n]$ such that exactly one entry A_{ij} equals 1 for $i \in I$.*

Let $1 \leq r \leq m$, $\ell \leq n$ and $c > 0$ be any parameters. Matrix A is an (r, ℓ, c)-expander iff A has ℓ ones per row and for all $I \subseteq [m]$, $|I| \leq r$, $|\partial_A(I)| \geq c|I|$.

For various arguments in [5, 62] one needs r close to n and $c = \Omega(\ell)$. As shown in [5], a random ℓ-sparse matrix has the expansion property with suitable parameters but not an arbitrary $(\Omega(\ell), \ell)$-design.

The hardness of $NW_{A,f}$ was shown in [5] for expanding matrices A with $m \leq n^{2-\Omega(1)}$ and suitable f (e.g. parity function) for resolution (and with a different definition of the τ-formulas also for polynomial calculus and its combination with resolution PCR). This was then improved to $m = n^{3-\Omega(1)}$ and $\ell = O(1)$ in [62] and subsequently [95] has shown that such maps $NW_{A,f}$ are exponentially iterable for resolution (and for its extension $R(k)$ with small k). In fact, [95] also observed the argument in [62] already shows the exponential iterability too.

29.5 Gadget generators

Start with a polynomial-time function

$$f \ : \ \{0,1\}^{\ell} \times \{0,1\}^{k} \to \{0,1\}^{k+1}$$

and define a new function

$$g \ : \ \{0,1\}^{n} \to \{0,1\}^{n+1}$$

for

$$n \ := \ \ell + k \cdot (\ell + 1)$$

as follows:

- An input $x \in \{0,1\}^n$ is interpreted as one string v of length ℓ and followed $\ell+1$ strings u^s of length k, $s = 1, \ldots, \ell+1$.
- The output $y \in \{0,1\}^{n+1}$ is the concatenation of $\ell+1$ strings y^s, each of length $k+1$, defined by

$$y^s := f(v, u^s) \ .$$

Each y^s has one extra bit of length over that of u^s, so the first ℓ of the blocks swallow v and the last one produces the one extra bit of output we want. The string v is called *a gadget*.

Of course, for this to work we have to have a suitable gadget and a function f. We now present a simple example constructed in [69]. The gadget there is interpreted as a graph of a function h between $[k]$ and $[k+1]$ and the idea is that unless we can prove PHP in a proof system P we cannot rule out in P that h is a bijection and one may use it to stretch each block u^s to a block y^s longer by one bit.

Definition 29.5.1 *Let $k, t \geq 1$ be any parameters such that $t > k \cdot (k+1)$. Put $n := k \cdot (k+1+t)$ and $m := (k+1) \cdot t$. Hence $m > n$.*

Map $g_{k,t} : \{0,1\}^n \to \{0,1\}^m$ is defined as follows. Input string x of length n interpret as

$$x = (v, u^1, \ldots, u^t)$$

where

$$v = (v_{ij})_{i \in [k+1], j \in [k]} \quad \textit{and} \quad u^s = (u^s_j)_{j \in [k]}$$

for $s = 1, \ldots, t$.

The output string y of length m is defined as $y := (y^1, \ldots, y^t)$ where

$$y^s_i := \bigvee_{j \in [k]} (v_{ij} \wedge u^s_j)$$

for $s = 1, \ldots, t$.

Note that the map f in this particular construction is very simply defined: its output bits are defined by 2DNF formulas.

Theorem 29.5.2 ([69]) *Let $d \geq 2$ and assume $k \geq 1$ and $t = k^2 + k + 1$. Then, with $n := k \cdot (k+1+t)$ as above, the map*

$$g_{k,t} : \{0,1\}^n \to \{0,1\}^{n+1}$$

is an exponentially hard proof complexity generator for constant depth Frege systems F_d.

Proof :

Let $b := (b^1, \ldots, b^t) \in \{0,1\}^{n+1}$ be an arbitrary string, b^s' blocks of length $k+1$, and assume that we have an F_d-proof π of the τ-formula $\tau(g)_b$. Substitute in π everywhere for all variables u^s_j formulas

$$u^s_j(v) := \bigvee_{i\in[k+1]} v_{ij} \wedge b^s_i \ .$$

Denote the substituted proof π^v.

Let $\neg PHP^{k+1}_k(v)$ be the formula expressing that v defines a graph of a function violating the pigeonhole principle from $[k+1]$ into $[k]$ (not necessarily bijective):

$$\bigwedge_{i\in[k+1]} \bigvee_{j\in[k]} v_{ij} \ \wedge \ \bigwedge_{i\in[k+1]} \bigwedge_{j_1\neq j_2\in[k]} \neg v_{ij_1} \vee \neg v_{ij_2}$$

$$\wedge \bigwedge_{i_1\neq i_2\in[k+1]} \bigwedge_{j\in[k]} (\neg p_{i_1 j} \vee \neg p_{i_2 j}) \ .$$

It is not difficult to see that there is a size $n^{O(1)} = k^{O(1)}$ F_d-proof σ of

$$g_{k+1,k,t}(v, u^1(v), \ldots, u^t(v)) \neq b \ \rightarrow \ PHP^{k+1}_k(v) \ .$$

If we combine proofs π^v and σ we get a proof of $PHP_k(v)$. By [2, 78, 87] any such proof must have size exponential in k, and hence π^v, and so π too, must be exponential too.

q.e.d.

Quite generally, one can fix the function f to be the circuit-value function

$$CV_{\ell,k}(v,u)$$

that takes ℓ bits v describing a circuit C with k input bits and $k+1$ output bits and $u \in \{0,1\}^k$, and outputs the string $C(u)$. It may seem at first that the value of the parameter ℓ does not allow for a canonical choice. However, using a form of self-reducibility of the resulting generators [69] observed that it is possible to assume without a loss of generality that $\ell \leq k^2$ and, in fact, even $\ell \leq k^{1+\epsilon}$, any fixed $\epsilon > 0$.

Using as the gadget a map violating PHP results in a generator hard for any proof system where PHP is hard (see [69] for a discussion of algebraic proof systems). One may use other well-known tautologies hard for some proof systems to create a combinatorial situation that can be used as a good gadget. In Section 30.3 we will use as gadgets the data A, f defining an NW-generator but here we give just one more combinatorial example to illustrate the definition (but provably not useful for EF, though).

Assume that $v : [k^2] \to [k^3]$, $w : [k^3] \to [k]$ are two maps defined by their graphs (i.e. we have k^5 and k^4 variables v and w, respectively), and let $G = ([k^3], E)$ be an undirected graph without loops with vertex set $[n]$ and edges E ($(k^3)(k^3-1)/2$ variables). The Clique/Coloring principle says that it cannot happen simultaneously that:

1. v is an injective map.
2. The range of v is a clique in G.
3. w is a coloring of G.

This is obviously true as otherwise we could color a clique of size k^2 by k colors which is impossible.

Assume, however, that v, w and E do violate the principle. Then the composed map $w \circ v : [k^2] \to [k]$ is injective and hence violates the weak PHP. In other words, if the weak PHP has only long P-proofs then the generator resulting from this gadget will be hard for P. An example of a proof system for which PHP is easy while the Clique/Coloring principle is hard is the cutting planes proof system CP, cf.[88].

29.6 Optimal automatizer for the τ-formulas

Hirsch and Itsykson [41] introduced a concept of heuristic randomized proof systems that do make errors, but only a few, and a corresponding concept of an automatizer. We shall recall only the latter as it is simpler and quite natural in the context of proof complexity generators.

Let $L \subseteq \{0,1\}^*$ be a language and $D = (D_n)_{n \geq 1}$ a family of probabilistic distributions on $\{0,1\}^n \setminus L$. Such a pair is called a *distributional proving problem* in [41]. Informally, an automatizer for (D, L) is a probabilistic algorithm that accepts all of L and only a little of its complement, when the qualification "little" is defined via the family D.

Definition 29.6.1 (Hirsch and Itsykson[41]) *An automatizer A for a distributional proving problem (D, L) is a probabilistic algorithm that:*

1. *Gets as inputs $(x, d) \in \{0,1\}^* \times \mathbf{N}$.*
2. *On every input it either halts and outputs 1 or does not halt at all.*
3. *For every $(x, d) \in L \times \mathbf{N}$, $A(x, d) = 1$ with probability 1.*
4. *For any $n, d \geq 1$:*

$$Prob_{x \in D_n} [\ (Prob[\ A(x,d) = 1\ ] > 1/4)\ ]\ <\ 1/d$$

where the inner probability is computed over random choices of A and in the outer probability the x of length n is chosen according to D_n.

The time an automatizer takes on an input is defined to be the median time (the minimal time such that there is at least a fifty - fifty chance that the algorithm halts). An automatizer is **optimal** if it is at most polynomially slower than any other automatizer.

Theorem 29.6.2 (Hirsch and Itsykson [41]) *Let L be a recursively enumerable language and D a family of polynomial time samplable probability distributions on its complement.*

Then there is an optimal automatizer for the distributional proving problem (D, L).

In general, for an r.e. language, there may be a variety of distributions on the complement of L, and there does not seem to be a natural and universal criterion on how to select one. But in the case of proof complexity generators there does appear to be the most natural choice.

Namely, let g be a polynomial time function mapping $\{0,1\}^n$ into $\{0,1\}^m$, where $n \to m = m(n)$ is an injective function (and hence m determines n) as in Section 29.1. Define language L^g (obviously r.e.) by:

$$L^g\ :=\ \{0,1\}^* \setminus Rng(g)\ .$$

Now note that the function g itself provides us a with a (p-time samplable) distribution D^g on $\{0,1\}^* \setminus L^g$, i.e. on $Rng(g)$. That is, the distribution

D^g_m on $\{0,1\}^m \setminus L^g = \{0,1\}^m \cap Rng(g)$ gives to $w \in \{0,1\}^m \cap Rng(g)$ the probability:

$$\mathrm{P}rob_{x\in\{0,1\}^n}[g(x) = w] \ .$$

Theorem 29.6.2 in this context provides an optimal automatizer for accepting the valid $\tau(g)$-formulas. Each formula $\tau(g)_b$ is represented by b and hence L^g represents the set of valid $\tau(g)$-formulas.

Theorem 29.6.3 *Let g be a polynomial time function mapping $\{0,1\}^n$ into $\{0,1\}^m$, where $n \to m = m(n)$ is an injective function.*

Then there is an automatizer A that accepts all of $L^g \times \mathbf{N}$ with probability 1, for which

$$Prob_{x\in\{0,1\}^n}\ [(Prob[A(g(x), d) = 1] > 1/4)]\ <\ 1/d$$

and which is optimal among all automatizers for the distributional proving problem (D^g, L^g) determined by g.

Chapter 30

Proof complexity generators: conjectures

All three generators we have discussed in Chapter 29 are determined by some kind of data that can vary: an NW-generator is determined by a matrix A and a function f, a gadget generator by its gadget, and the truth table function $\mathbf{tt}_{s,k}$ by its size rate function function $s = s(k)$.

A generator with specific data may be hard for one proof system and easy for another one. For example, if f were the parity function then EF can prove all valid τ-formulas for the NW-generator by following the Gaussian elimination. Or the PHP-gadget is hard for F_d while easy for CP. With the current sorry state of circuit complexity lower bounds one has to take s very slow indeed to have a similar example (but $s(k) = 2k$ will do).

It is an interesting question if one can choose these data in some particular way so that the resulting generator would be hard for *all* proof systems. In this chapter we present various informal speculations as well as formal conjectures related to the hardness of these generators.

Theorem 29.3.2 makes in a sense the truth table function the most prominent among proof complexity generators. We will thus start our speculations in the first section with this function.

30.1 On provability of circuit lower bounds

Assume that a lower bound $s = s(k)$ (maybe even with an exponential s) is valid for SAT. Then $\mathbf{tt}_{s,k}$ will not be hard for all proof systems. Define a

proof system P which upon receiving a string b of length 2^k checks whether b is or is not equal to χ_{SAT_k}, the characteristic function of SAT restricted to inputs of length k. If so, then it accepts $\tau(\mathbf{tt}_{s,k})_b$ as a tautology, and if not it proceeds as, say, EF. The point is that deciding the required property of b can be done in polynomial time and hence P is indeed a proof system in the sense of Cook and Reckhow.

Same argument applies to all languages L in place of SAT that belong to the exponential class E. In fact, if L is in $NE \cap coNE$ one can still define in an analogous way a proof system Q for which $\mathbf{tt}_{s,k}$ will not be hard: this time Q expects that all non-deterministic witnesses for all values in the truth table b are part of a proof it is given. Each individual witness has size polynomial in 2^k and there are 2^k of them, so the collections of all required witnesses is of polynomial size (and can be verified as correct in polynomial time).

One may speculate that this is the *only* way how $\mathbf{tt}_{s,k}$ may fail to be hard for a proof system.

Possibility A: *Assume that* $\mathbf{tt}_{s,k}$ *is not hard for a proof system P. Then there is* $L \in NE \cap coNE$ *such that P admits polynomial size proofs of tautologies*

$$\tau(\mathbf{tt}_{s,k})_{\chi_{L_k}} \ .$$

An obvious consequence is the following observation. For a function $s = s(k)$ denote by $Size(s)$ the class of languages with circuit complexity at most $s(k)$.

Lemma 30.1.1 *Assume that Possibility A is true, and that* $s(k) \leq 2^{(1-\Omega(1))k}$. *Then* $NE \cap coNE \subseteq Size(s)$ *implies* $NP \neq coNP$.

Note, as a small test of Possibility A, that the implication

$$NE \cap coNE \subseteq Size(s) \ \rightarrow \ \mathrm{NP} \neq co\mathrm{NP}$$

is actually true, as $\mathrm{NP} = co\mathrm{NP}$ implies that $NE \cap coNE$ contains a language of any circuit complexity $2^{(1-\Omega(1))k}$ (as in the proof of Theorem 30.2.2).

Another speculation concerns a possibility to give a sufficient condition on a proof system guaranteeing that $\mathbf{tt}_{s,k}$ is hard for it.

Possibility B: *If a proof system P is not optimal then for all* $\delta > 0$ *the function* $\mathbf{tt}_{s,k}$ *with* $s(k) = 2^{\delta \cdot k}$ *is hard for P.*

The non-optimality of P implies (via known links between simulations and reflection principles) that P can think that another proof system Q is not sound, say proves 0. Without a loss of generality Q is EF augmented by a polynomial time set of formulas as extra axioms (cf.[55]). Now, a proof of 0 in, for example, EF itself is essentially the same thing as a description of a circuit without inputs and then two different "deductions" of its value leading to the opposite values. For EF the meaning of "a deduction" is just reasoning in resolution R, for Q stronger than EF it is R enhanced in a particular way (depending on Q). The idea behind Possibility B is that such a "contradictory" circuit might be transformed into a circuit (perhaps with an almost exponential blow-up) computing any predetermined function b. In particular, P cannot prove shortly $\tau(\mathbf{tt}_{s,k})_b$.

A small part of Possibility B follows from [62, Thm.5.2]: it is shown there that if EF does not simulate WF (a proof system possibly stronger that EF, introduced by Jeřábek [49]) then $\mathbf{tt}_{s,k}$ is not only hard but is even pseudo-surjective for EF.

Lemma 30.1.2 *Assume that Possibility B is true and that* $E \not\subseteq Size(2^{o(n)})$. *Then there exists an optimal proof system.*

Proof :

Let L be a complete language for E and $s(k)$ be some specific exponential lower bound $2^{\Omega(k)}$ to its circuit complexity (by the hypothesis of the lemma). Form a proof system P that is EF augmented by all instances of τ-formulas

$$\tau(\mathbf{tt}_{s,k})_{L_k} \ .$$

q.e.d.

We now turn to a particular problem seemingly having nothing to do with proof complexity. Consider set CS $\subseteq \{1\}^* \times \{0,1\}^*$ (for Circuit Size) consisting of all pairs

$$(1^{(s)} \ , \ b)$$

where b is a string of length 2^k defining a Boolean function in k variables of circuit complexity at most s, and $s \leq 2^k$.

Problem 30.1.3 *Is set CS NP-complete?*

This problem has been studied in a slightly different version by Kabanets and Cai [50], under the name "circuit minimization problem". The difference is that they do encode s in binary. That is, of course, irrelevant from the point of view of mutual p-reducibility. But [50] consider a notion of "natural reduction" (i.e. length determined) and then the unary and the binary encodings seem to matter.

Possibility C: *CS is NP-complete and moreover there are p-time functions f, g and h such that:*

1. *f is defined on 3CNF formulas φ and it is a p-reduction of 3SAT to CS.*

2. *g is defined on pairs (φ, a) of a 3CNF formula and a truth assignment and its value is a circuit, and it holds:*

$$\varphi(a) = 1 \ \rightarrow \ \mathbf{tt}_{s,k}(g(\varphi, a)) = \pi_2(f(\varphi)) \ ,$$

   where π_2 is the projection on the second coordinate ($f(\varphi)$ is a pair $(1^{(s)}, b)$).

3. *h is defined on pairs (C, b) of a circuit and a truth table and its value is a truth assignment, and the following holds. If f maps a 3CNF formula φ to pair $(1^{(s)}, b)$ then:*

$$\mathbf{tt}_{s,k}(C) = b \ \rightarrow \varphi(h(C, b)) = 1 \ .$$

In other words, maps g and h send witnesses to witnesses of the two NP-properties.

Define a proof system P as the extension of EF by all propositional translations (for all lengths $n \geq 1$) of statements 2. and 3. from Possibility C. Due to the strength of PV it seems likely that, if possibility C holds, statements 2. and 3. may be provable in PV. Hence, due to the relation of EF to PV, these extra axioms would be polynomially provable in EF and hence, in fact, P=EF would hold.

In the proof of the following lemmas we use known properties of the propositional $||\ldots||$-translation, cf.[55].

Lemma 30.1.4 *Assume Possibility C. Then there is a constant $c \geq 1$ such that every tautology φ of length n has a P-proof of size at most n^c from instances of some valid τ-formula $\tau(\mathbf{tt}_{s,k})_b$.*

Informally: circuit lower bounds are the hardest tautologies over a fixed proof system P.

Proof :

Let ψ be a tautology and assume w.l.o.g. that it is a 3DNF formula. Take φ to be the 3CNF of $\neg\psi$ and assume $|\varphi| \leq n$ and that atoms of φ are among $x_1, \ldots, x_n$.

An instance of one of the new axioms in P is

$$|| \ \varphi(x_1, \ldots, x_n) \rightarrow \mathbf{tt}_{s,k}(g(\varphi, x)) = b \ ||^n$$

where s, k and b are given by $\pi_2(f(\varphi)) \in \{0,1\}^{2^k}$ and $x = (x_1, \ldots, x_n)$.

A substitution instance of the axiom $\tau(\mathbf{tt}_{s,k})_b$ is $\mathbf{tt}_{s,k}(g(\varphi, x)) \neq b$, and hence $\neg\varphi$ follows.

q.e.d.

Lemma 30.1.5 *Assume Possibility C. Assume also that $\{\varphi_n\}_n$ is a sequence of tautologies constructed by a polynomial time algorithm from $1^{(n)}$.*

Then there is a language L in class E and $s = s(k)$ such that τ-formulas $\tau(\mathbf{tt}_{s,k})_{L_k}$ are valid and each φ_n has a polynomial size P-proof from instances of one of these τ-formulas.

Proof :

Composing the polynomial time construction of φ_n from $1^{(n)}$ with the map $\pi_2 \circ f$ gives a polynomial time algorithm computing the truth table b in the previous lemma from $1^{(n)}$. Hence $L \in$ E.

q.e.d.

Lemma 30.1.6 *Assume Possibility C. Then any propositional proof system Q can be simulated by a proof system P augmented by instances of valid τ-formulas $\tau(\mathbf{tt}_{s,k})_{L_k}$, for some L in class E and some functions s.*

Proof :

The statement follows form Lemma 30.1.5 using the fact that Q can be p-simulated by EF augmented by the propositional translations of the reflection principle for Q; these formulas are p-time construable.

q.e.d.

Lemma 30.1.5 also yields similarly the following statement.

Lemma 30.1.7 *Let $\delta > 0$ and take a subset CS' of CS consisting of those pairs $(1^{(s)}, b)$ such that $s \geq 2^{\delta k}$. Assume that Possibility C holds for CS'.*

Then there exists a language L in E requiring circuits of size $2^{\Omega(k)}$. (And hence $P = BPP$.

30.2 Razborov's conjecture on the NW generator and EF

Lower bound arguments for the hardness of the NW generator in resolution in [62, 95], as well as related results in earlier [5], use a strong expanding property of matrix A that is not guaranteed by being a mere (d, ℓ)-design, with parameters ℓ and d satisfying the constraints of the original construction in [83]. In fact, only recently Pich [86] demonstrated the hardness of the original NW generator from [83] for proof systems admitting feasible interpolation (which includes resolution). Nevertheless, Razborov [95] has made the following intriguing conjecture.

Conjecture 30.2.1 (A. A. Razborov[95, Conjecture 2])
Any NW-generator based on a matrix A which is a combinatorial design with the same parameters as in [83] and on any function f in $NP \cap coNP$ that is hard on average for P/poly, is hard for EF.

The parameters are not explicitly specified in [95] and the formulation "*with the same parameters*" requires some investigation into what the constraints actually are.

The conjecture is not valid for all parameters for which the [83] construction works (see [63] for an example). But it seems that the conjecture alludes to the following constraints on the parameters in the main construction of combinatorial designs in [83, L.2.5]:

$$d = \log(m), \quad \log(m) \leq \ell \leq m, \quad n = O(\ell^2) \ .$$

We shall interpret the phrase "*hard on average for P/poly*" from the conjecture as meaning that the hardness on average of f in the sense of [83] is exponential:

$$H_f(\ell) \geq 2^{\Omega(\ell)} \ . \tag{1}$$

The quantity $H_f(\ell)$ is defined to be the minimal s such that there is a circuit C of size s such that

$$\mathrm{P}rob_u[f(u) = C(u)] \geq 1/2 + 1/s$$

where $u \in \{0,1\}^\ell$. In addition, [83, L.2.4] requires that

$$H_f(\ell) \geq m^2 \ .$$

Putting this together and taking the maximal allowed value for m we can write the constrains as:

$$(2) \qquad m = 2^{\epsilon \cdot n^{1/2}} \ , \quad d = \epsilon \cdot n^{1/2} \ , \quad \ell = n^{1/2} \ ,$$

where $\epsilon > 0$ is a constant. But note that it would be valid to take also the minimal value $m := n + 1$ (as we shall do later in Chapter 31).

The conjecture requires that $f \in \mathrm{NP} \cap co\mathrm{NP}$. But note that if we relax this condition to:

$$(3) \qquad f \ \in \ NTime(m^{O(1)}) \cap coNTime(m^{O(1)}) \ .$$

the size of the τ-formula will remain $m^{O(1)}$ (by the discussion after Definition 29.1.1).

Consider (following [63]) instead of the conjecture the following **Statement (R)** which which states the conjecture for the specific parameters (1) and (2) above but poses on the function f only requirement (3).

(R) *Let g be an NW-generator based on an $m \times n$ matrix A that is an (d, ℓ) combinatorial design and on any function f such that the constrains in (1), (2) and (3) are satisfied. Then g is hard for EF.*

Statement (R) asserts a conditional lower bound for EF. However, [63] made a simple but a bit surprising observation that it, in fact, implies also unconditional lower bound for EF. We sketch its proof from [63] for the completeness of our presentation.

Theorem 30.2.2 ([63]) *Assume that Statement (R) is true. Then EF is not p-bounded.*

Proof :

Assume for the sake of a contradiction that EF is p-bounded and hence, in particular, NP = coNP.

Claim: *There is a function $f : \{0,1\}^* \to \{0,1\}$ computable in $NE \cap coNE$ such that $H_f(\ell) \geq 2^{\Omega(\ell)}$.*

To see this note that the property of a string to be the lexicographically first string which is a truth table of a function on $\{0,1\}^\ell$ with $H_f(\ell) \geq 2^{\delta \cdot \ell}$ (some $0 < \delta < 1$) is in the polynomial time hierarchy PH and the function f it defines will be in E^{PH}.

But if NP = coNP then such an f is in $E^{\mathrm{NP} \cap co\mathrm{NP}} = NE \cap coNE$. This proves the claim.

Having such f we may invoke Statement (R) to deduce that EF is not p-bounded. That is a contradiction.

q.e.d.

We shall see in Chapter 31 (see remark after the proof of Lemma 31.2.1) that the original requirement from Conjecture 30.2.1 that f is in NP $\cap$ coNP is probably quite important.

We shall prove in Chapter 31 a uniform, bounded arithmetic, version of Conjecture 30.2.1. In fact, the model we construct there is not only a model of PV (a theory corresponding to EF) but even of theory $Th_\forall(L_{PV})$, the true universal theory in the language of PV, considered in Section 23.1. This theory corresponds in a sense to the class of all proof systems simultaneously (it proves the soundness of all proof systems).

This may cause one to contemplate a strengthening of Conjecture 30.2.1 in that the τ-formulas considered there are hard for all proof systems, not just for EF. We now note that the argument from Section 30.1 (before Possibility A) that the truth table is likely not hard for all proof system can be modified for the NW-generator too, as long as it has an exponential (or at least superpolynomial) number of output bits. This is similar to arguments in [96, 95].

Assume, as in (2) above, that m is exponential in n, say $m = 2^{n^\delta}$. Denote $k := n^\delta$, hence $n = k^{O(1)}$. The strings from $\{0,1\}^m$ can be identified with truth tables of Boolean functions with k inputs. The definition of the NW-generator implies that a function $b \in \{0,1\}^m = \{0,1\}^{2^k}$ in its range

is computed in a non-uniform NP $\cap$ *co*NP. In particular, if a proof system cannot prove shortly $\tau(NW_{A,f})_b$ it also cannot rule out shortly that b is in non-uniform NP $\cap$ *co*NP.

Arguing as in Section 30.1 it follows that if there is a function in NE $\cap$ coNE that is not in non-uniform NP $\cap$ *co*NP then any NW-generator with parameters as in (2) based on an NP $\cap$ *co*NP function (no matter how hard) cannot be hard for all proof systems.

30.3 A possibly hard gadget

We shall discuss a gadget that could possibly lead to a hard generator. A function defined analogously as the NW generator but based on a random sparse $(n+1) \times n$ matrix A and a random function f was proposed as a possibly hard proof complexity generator in [62] (see there for the motivation). The qualification sparse means that each row of A contains c ones where $1 \leq c \leq \log(n)$. A similar construction with an $n \times n$ matrix and c a constant was proposed in [36] as a one way function. We shall now use the idea of gadget generators to remove the randomness part.

The generator $\mathsf{nw}_{n,c}$ will be determined by two parameters $n, c \geq 1$ with $c \leq \log n$. The gadget will have $(n+1)^2$ bits. We shall interpret the first $(n+1)n$ bits as determining the $(n+1) \times n$ matrix A, and denote them accordingly A_{ij}. Then the next 2^c bits of the remaining $n+1$ bits define the truth table of function $f : \{0,1\}^c \to \{0,1\}$, and these bits will be denoted f_ϵ, where $\epsilon \in \{0,1\}^c$. The remaining bits of the gadget are not used (we prefer simple terms to optimal numbers). Of course, we could have encoded A by $O(nc \log n) \leq O(n(\log n)^2)$ bits but at the expense of a less transparent formulas defining the generator below.

The function $\mathsf{nw}_{n,c}$ sends k bits to $k+1$ bits where

$$k \ := \ (n+1)^2 + n((n+1)^2 + 1) \ .$$

An input $x \in \{0,1\}^k$ is interpreted as a tuple

$$x \ = \ (A, f, u^1, \ldots, u^t)$$

where (A, f) is the gadget ($(n+1)^2$ bits) and $u^i \in \{0,1\}^n$ for $i = 1, \ldots, t = (n+1)^2 + 1$.

The value $\mathsf{nw}_{n,c}(x)$ is defined to be the string $0^{(k+1)}$ if the following condition fails:

(1) *A has exactly c ones in each row.*

Otherwise $\mathsf{nw}_{n,c}(x) = y$, where y is a tuple

$$y \;=\; (v^1, \ldots, v^t)$$

with

$$NW_{A,f}(u^i) = v^i, \quad \text{for } i \in [t].$$

Note that $\mathsf{nw}_{n,c}$ is computed by a constant depth formula of size $O(n^{2+c})$: matrices A satisfying the condition (1) are defined by a CNF of size $O(n^{2+c})$, and the i-th bit of $NW_{A,f}(u)$ is defined, assuming (1), by a DNF formula

$$\bigvee_{J\subseteq[n],|J|=c} [\, (\bigwedge_{j\in J} A_{ij}) \;\wedge\; (\bigwedge_{\epsilon} (x(J)^\epsilon \to f_\epsilon)\,)\,]$$

where $\epsilon = (\epsilon_1, \ldots, \epsilon_c)$ ranges over $\{0,1\}^c$, and for $J = \{j_1 < \ldots < j_c\}$

$$x(J)^\epsilon \;:=\; x_{j_1}^{\epsilon_1} \wedge \ldots \wedge x_{j_c}^{\epsilon_c}$$

with $x^1 := x$ and $x^0 := \neg x$. This formula has size $O(c2^c n^c) \leq O(n^{2+c})$ as $c \leq \log n$.

The following statement appears to be consistent with our present knowledge.

Possibility D: *Function* $\mathsf{nw}_{n,c}$ *with* $c = \log n$ *is hard for any proof system.*

Note that the argument from the end of Section 30.2 that the NW generator is likely not hard for all proof systems does not seem to apply here.

It is easy to see that the PHP gadget generator from Section 29.5 is a special case of $\mathsf{nw}_{n,c}$ with $c = 1$ and can be easily embedded if $c > 1$. Hence we have, by Theorem 29.5.2:

Theorem 30.3.1 *For any* $1 \leq c \leq \log n$, *generator* $\mathsf{nw}_{n,c}$ *is exponentially hard for constant depth Frege systems* F_d.

30.4 Rudich's demi-bit conjecture

Let g be a P/*poly* proof complexity generator. Recall from Lemma 29.1.3 that g is hard for all propositional proof systems iff the complement of the range of g contains no infinite NP set.

A statement related to this property has been put forward as a conjecture by Rudich [98] in a different context, in a connection with his attempts to generalize the concept of the so called *natural proofs* of [96] (see [98] for this background).

Following Rudich[98] we define the following notion of the hardness of a function. Let h be map $h : \{0,1\}^n \to \{0,1\}^m$, with $m > n$. The **demi-hardness** of h is the minimal s such that there is a non-deterministic circuit C of size at most s which defines a subset C^* of $\{0,1\}^m \setminus Rng(h)$ of measure $|C^*|/2^m \geq 1/s$.

Conjecture 30.4.1 (Rudich[98, demi-bit conjecture])
There exists a P/poly map $g = (g_n)_n$, $g_n : \{0,1\}^n \to \{0,1\}^{n+1}$, and $\epsilon > 0$ such that the demi-hardness of g_n is at least 2^{n^ϵ}.

Rudich [98] proposes the subset sum generator of [44] as a candidate function to satisfy the conjecture (even a stronger statement, the super-bit conjecture, cf.[98]). This generator is defined as follows.

The function g takes as an input k numbers $a_1, \ldots, a_k$ each having $k+1$ bits, and k single bits $b_1, \ldots, b_k$. The output of g is the k-tuple $a_1, \ldots, a_k$ together with

$$\sum_i a_i b_i \pmod{2^{k+1}} ,$$

another $k+1$ bits. Function g thus stretches $n = k(k+2)$ bits into $n+1$ bits.

The generator has several very nice properties interesting from the point of view of cryptography (cf.[44]) but [98] does not offer any particular statement supporting the suggestion that it has also exponential demi-hardness (or even the stronger super-hardness).

Chapter 31

The local witness model

Our general aim is to construct models relevant to some of the conjectures discussed in Chapter 30, and eventually models that would prove some of them. From a logical point of view this should be more accessible for Razborov's Conjecture 30.2.1 than for the other conjectures as it concerns a function of the highest complexity. This translates into a higher quantifier complexity of the associated formalized statement and thus offers, in principle, more chances to manipulate things to our advantage.

The model we shall construct in this chapter, the local witness model, will yield the following statement. We shall formulate it now a bit informally; the formal version in Section 31.4 will quantify the parameters involved.

Theorem 31.0.2 (informal) *Let A be an $m \times n$ 0-1 matrix that is a $(\log m, n^{1/3})$ design in the sense of [83], and $n < m$. Let f be a Boolean function in $n^{1/3}$ variables that is a hard bit of a one way permutation. Assume R is an infinite NP-set.*

Then it is consistent with Cook's theory PV (and, in fact, with the true universal theory in the language of PV) that

$$Rng(NW_{A,f}) \cap R \neq \emptyset \ .$$

This gives a form of consistency of Razborov's conjecture (in fact, of a stronger statement) and also of Rudich's demi-bit conjecture. We shall discuss this in Section 31.4.

31.1 The local witness model $K(F_b)$

We shall continue using the notation from Section 29.4. In particular, $A \in \mathcal{M}_n$ will be an $m \times n$ 0-1 matrix with ℓ ones per row, in row i the ones are in columns from $J_i \subseteq [n]$. We shall fix $\ell := n^{1/3}$ and we will assume that A is $(\log m, \ell)$-design. We assume that $m > n$ and note that $A \in \mathcal{M}_n$ implies that $m < 2^{n^{1/t}}$, for some $t > \mathbf{N}$ (i.e. m is subexponential in n).

For any $a \in \{0,1\}^n$ and $J = \{j_1 < \ldots < j_\ell\} \subseteq [n]$ denote

$$a(J) \ := \ (a_{j_1}, \ldots, a_{j_\ell}) \ .$$

Further we have a function $f : \{0,1\}^\ell \to \{0,1\}$ of the form

$$f(u) \ := \ B(h^{(-1)}(u))$$

where h is a polynomial time permutation and B is a Boolean function. It is intended that h will be a one way permutation and B its hard bit but that is not needed for the definition of the model (but it is key for its properties).

The model will be parameterized by an element $b \in \mathcal{M}_n$, a string $b = (b_1, \ldots, b_m) \in \{0,1\}^m$, that satisfies

$$b \notin Rng(NW_{A,f}) \ .$$

Given such a string define the **sample space** Ω_b to be the set of all $(m+1)$-tuples

$$\omega \ = \ (a, v^1, \ldots, v^m)$$

where $a \in \{0,1\}^n$ and all $v^i \in \{0,1\}^\ell$, and ω satisfies the following condition:

- *For every $i \in [m]$: if $f(a(J_i)) = b_i$ then v^i is the unique string such that*
$$h(v^i) = a(J_i) \wedge B(v^i) = b_i \ .$$
If no such string exists then $v^i = 0^{(\ell)}$.

We think of v^i as of a witness to the fact that the i-th bit of $NW_{A,f}(a)$ has the "right" value b_i.

From Section 22.5 we know that when aiming at proof complexity we should expect to use partially defined random variables. The **family** F_b of partial random variables forming the local model is the family of all function

$\alpha \in \mathcal{M}_n$, $\alpha : \Omega_b \to \mathcal{M}_n$, satisfying the following condition. There is a non-uniform algorithm S running in time bounded by $2^{n^{1/t}}$, for some $t > \mathbf{N}$ (i.e. in subexponential time) that computes α on a sample $\omega = (a, v^1, \ldots, v^m)$ as follows:

1. S can read from the input a but not any of v^i's.

2. S can learn the value of v^i if it queries the parameter i. Upon such a query S is provided with the string v^i, if

$$h(v^i) = a(J_i) \wedge B(v^i) = b_i$$

   and continues with the computation.

   If v^i does not satisfy this condition the computation is aborted and $\alpha(\omega)$ is undefined.

3. There exists a constant $c \geq 1$ such that S on every input S queries at most c values of v^i.

   (We could have allowed c up to any number bounded above by $n^{1/t}$, some $t > \mathbf{N}$.)

Note that this type of a computation is essentially a computation with an oracle access to a counter-example function similarly as in Section 24.2 but allowing a partially defined counter-example function. The present formalism makes the "sources of undefinability" of α more transparent and better amenable to an analysis.

We want to show that PV and, in fact, the true universal theory in its language $Th_\forall(L_{PV})$ (see Section 23.1) is valid in $K(F_b)$ and that

$$(*) \qquad [\![\exists x NW_{A,f}(x) \;=\; b]\!] = 1_{\mathcal{B}} \; .$$

However, as we are dealing with only partially defined random variables we have to establish first an important property that each random variable from the family F_b is undefined only on an infinitesimal fraction of the sample space. This fact also plays a crucial role in establishing the equality $(*)$. This is analogous to Lemma 20.1.3 and it is the key property of F_b. In order to establish this fact we shall use the assumed hardness of the function f.

31.2 Regions of undefinability

Recall the definition of the hardness of a function, $H_f(\ell)$, from Section 30.2.1; it is the minimal s such that some circuit of size s computes f with the advantage over $1/2$ at least $1/s$. Using this concept we can formulate the key lemma.

Lemma 31.2.1 *Assume that the permutation h and the hard-bit predicate B are such that the function $f := B \circ h^{(-1)}$ they determine has the hardness*

$$H_f(\ell) \geq 2^{\ell^\epsilon}$$

for some constant $\epsilon > 0$. Let $\alpha \in F_b$ be arbitrary. Then the probability

$$Prob_{\omega \in \Omega_b}[\alpha \quad \textit{is undefined}\,]$$

is infinitesimal.

The rest of the section is devoted to the proof of the lemma.

Assume that we have a random variable α for which the probability in the lemma is at least (a standard positive) $\delta > 0$. Denote by $W \subseteq \{0,1\}^n$ the set of all $a \in \{0,1\}^n$ such that α is undefined on the unique sample starting with a (the sample is unique because the witnesses are unique). We shall denote such a sample $\omega(a)$, and the witnesses v^i appearing in it $v^i(a)$.

Let S be the algorithm computing α and making on each sample at most c queries about witnesses, as required in Section 31.1. If S aborted the computation on $\omega(a)$ and i was its last query about a witness it means that

$$f(a(J_i)) \neq b_i \ .$$

Hence we may interpret S as a non-uniform and interactive algorithm solving the following:

Task (T): *Given $a \in \{0,1\}^n$ find $i \in [m]$ such that the i-th bit of $NW_{A,f}(a)$ differs from b_i.*

By the interactive part of S we mean the ability to ask for a witness and get a correct one, if it exists. To be able to speak about such computations easier we shall say that "S asks the oracle for a witness" and that "the oracle sends a witness". In this terminology we can reformulate the behavior of S as follows:

- S, upon receiving an input $a \in \{0,1\}^n$, computes his first candidate solution $i_1 \in [m]$ for task (T).
- If i_1 solves (T) (this can be tested in polynomial time) the computation stops.
- If i_1 fails to solve (T) the oracle sends to S the witness $v^{i_1}(a)$, witnessing that $f(a(J_{i_1})) = b_{i_1}$.
- In the k-th step S computes a candidate solution $i_k \in [m]$ from a and from the $(k-1)$ witnesses received in earlier rounds.
- If i_k solves (T) the computation stops. Otherwise the oracle sends the correct witness $v^{i_k}(a)$, and the computation proceeds.

Assume that for a string $a \in W$ on which S succeeds in solving (T), S asks k queries and then the computation stops: the candidate solutions S produced were $i_1, \ldots, i_k$ and the last one i_k was correct. We shall call the k-tuple $(i_1, \ldots, i_k)$ the *trace of the computation* on a. In particular, $c \leq k$. As the witnesses are unique the trace determines also the oracle replies.

Claim 1: *There is a k-tuple $\overline{i} = (i_1, \ldots, i_k) \in [m]^k$ for some $k \leq c$ that is the trace of computations on at least a fraction of $\frac{2}{(3m)^k}$ of all inputs from W.*

Construct by induction on t a string $(i_1, \ldots, i_t) \in [m]^t$ such that the traces of at least $\frac{1}{3^{t-1}m^t}$ fraction of W start with this t-tuple. For $t = 1$ it is obvious.

For the induction step assume that we have constructed a t-tuple $(i_1, \ldots, i_t)$ satisfying the property. If $(i_1, \ldots, i_t)$ is the complete trace of at least $2/3$ of all inputs from W whose traces start with $(i_1, \ldots, i_t)$, it is already the wanted trace.

If it is not the case extend $(i_1, \ldots, i_t)$ by i_{t+1} so that for at least $1/(3m)$ of all inputs from W with traces starting with $(i_1, \ldots, i_t)$ the traces are not complete and will continue with $(i_1, \ldots, i_{t+1})$. It is possible to find such an i_{t+1} as at least a third of all computations with traces starting with $(i_1, \ldots, i_t)$ continue and there are m choices for i_{t+1}. This proves the claim.

Let us fix a trace $\overline{i} = (i_1, \ldots, i_k)$ satisfying the claim. For $u \in \{0,1\}^\ell$ and $v \in \{0,1\}^{n-\ell}$ define string $a(u,v) \in \{0,1\}^n$ by putting bits of u into positions J_{i_k} (in the natural order) and then fill the remaining $n - \ell$ positions by bits of v. The following claim follows by averaging.

Claim 2: *There is an $n-\ell$-tuple $e \in \{0,1\}^{n-\ell}$ such that for a fraction of at least $\delta \cdot \frac{2}{(3m)^k}$ of all $u \in \{0,1\}^\ell$ the string $a(u,e)$ is in W and its trace is $\overline{i}$.*

Fix one such an $(n-\ell)$-tuple e. The design property of A implies that there are, for any row $i \neq i_k$, at most $\log(m)$ bits from J_i not set by e. Thus there are at most m assignments w to bits in J_i not set by e. Any such w determines, together with e, an assignment to variables in J_i and hence a string in $\{0,1\}^\ell$; denote it z_w. Let Y_i be the set of all witnesses to

$$f(z_w) = b_i$$

for all such z_w (whenever a witness exists). The total bit size of each Y_i is $m \cdot \ell^{O(1)} = m^{O(1)}$, and there are $m-1$ of these sets, hence all this information has $m^{O(1)}$ bits.

Define an algorithm C that uses as advice the trace $\overline{i}$, the partial assignment e, and all $m-1$ sets Y_i. The total size of the advice is bounded above by

$$\log(m) \cdot c + (n-\ell) + m^{O(1)} \leq m^{O(1)} \ .$$

The aim of the algorithm is to compute the function f. Let U be those inputs u for which the trace of $a(u,e))$ either equals to $\overline{i}$ or starts with $\overline{i}$, and let b_0 be the majority value of f on the complement of U.

On input $u \in \{0,1\}^\ell$ C defines the string $a := a(u,e) \in \{0,1\}^n$ and starts the computation of S on a. If the first candidate solution S computes is different from i_1 then C halts and produces b_0 as the output. If the first candidate solution is i_1 the algorithm C reads from Y_{i_1} the right witness y^{i_1}. The uniqueness of witnesses for f implies that there is exactly one suitable string in Y_{i^1} and C can find it by testing the condition

$$h(y^{i_1}) = a(J_{i_1}) \wedge B(y^{i_1}) = b_{i_1} \ .$$

C proceeds analogously for steps $2, \ldots, k-1$: if any candidate solution computed is different from the particular element i_j in the trace, $j = 2, \ldots, k-1$, or if the computation of S halts before reaching the k-th step, C halts and outputs the value b_0. Otherwise, in each step, C retrieves from the sets Y_i the correct witness and proceeds with the simulation of S.

Assume that the computation evolved according to the trace $\overline{i}$ and C reaches the k-th step. If the k-th candidate solution is different from i_k, C again outputs b_0. But if it is i_k it outputs $1 - b_{i_k}$.

Claim 3: *The algorithm C computes f correctly on at least a fraction of*

$$1/2 \ + \ 2^{-\ell^\epsilon}$$

of inputs, and it is computed by a circuit of size at most 2^{ℓ^ϵ}.

The algorithm C outputs the bit b_0 in all cases except when the computation reaches the k-th step and the simulation of S produced i_k as its candidate solution. If the computation of S were to actually stop at that point then the value $1 - b_{i_k}$ in indeed equal to $f(u)$. If the computation were to continue then we have no information. But note that by the choice of $\overline{i}$ in Claim 1 the former case happens with probability at least $2/3$ and hence for at least a fraction of $\delta \cdot \frac{2}{(3m)^k}$ of all inputs $u \in \{0,1\}^\ell$. Because b_0 is the correct value of f for at least half of $u \notin U$, the overall advantage algorithm C has in computing f is at least $\delta \cdot \frac{1}{(3m)^k}$.

Now note that $m < 2^{n^{1/t}}$ for some $t > \mathbf{N}$, and hence

$$\delta \cdot \frac{1}{(3m)^k} \ > \ 2^{-\ell^\epsilon}$$

as $\epsilon > 0$ is standard.

The algorithm C uses advice of size $m^{O(1)}$ and needs the same time as S except when it needs to simulate a reply of the oracle and to find an appropriate witness in one of the sets Y_i. This is done at most $(c-1)$-times and takes $m^{O(1)}$ time each. Hence the total time is $m^{O(1)}$ which is also less than 2^{ℓ^ϵ}.

This establishes the claim.

Claims 3 contradicts the assumed hardness of f and hence the lemma follows.

q.e.d.

Let us make an observation relevant to Theorem 30.2.2. Assume we would want to allow f not only from NP $\cap$ *co*NP but from NE$\cap$coNE. In such a case each step in which C simulates the oracle and searches for a witness for would take time $2^{O(\ell)}$. Hence the size of C would be estimated by $O(m^{O(1)}2^{O\ell})$, and one gets no contradiction with the hardness of f.

31.3 Properties of the local model $K(F_b)$

Having the key Lemma 31.2.1 we are in a position to state and prove the required properties of $K(F_b)$. Let A, f and b be as in Section 31.1.

Theorem 31.3.1 *Assume the function f has an exponential hardness*

$$H_f(\ell) \ \geq \ 2^{\ell^\epsilon}$$

for some $\epsilon > 0$.

Then the true universal theory in the language of PV, theory $Th_\forall(L_{PV})$, is valid in $K(F_b)$, and so is therefore theory PV. Also

$$[\![\exists x NW_{A,f}(x) \ = \ b]\!] = 1_{\mathcal{B}} \ .$$

Proof :

$K(F_b)$ is clearly L_{PV}-closed and hence the validity of $Th_\forall(L_{PV})$ follows from the general Lemma 1.4.2.

To compute the truth value of the statement $\exists x NW_{A,f}(x)$ write it in full as

$$\exists x \ \forall i(i \in [m]) \ \exists y(|y| = \ell) \ B(x, i, y)$$

where $B(x, i, y)$ is the open PV-formula (with parameters A, b)

$$h(y) = x(J_i) \ \wedge \ B(y) = b_i \ .$$

Let $\alpha \in F_b$ be the function that is just the projection on the first coordinate of the sample:

$$\alpha((a, v^1, \ldots, v^m)) \ := \ a \ .$$

We want to show that

$$[\![\ \forall i(i \in [m]) \ \exists y(|y| = \ell) \ B(\alpha, i, y) \ ]\!] \ = \ 1_{\mathcal{B}} \ .$$

For this it is necessary to show that for each $\iota \in F_b$

$$[\![\ \iota \in [m] \ \rightarrow \ \exists y(|y| = \ell) \ B(\alpha, \iota, y) \ ]\!] \ = \ 1_{\mathcal{B}}$$

given ι take random variable $\gamma \in F_b$ defined on a sample $\omega = (a, v^1, \ldots, v^m)$ as follows:

$$\gamma(\omega) = \begin{cases} v^i & \text{if } \iota(\omega) = i \in [m] \wedge h(v^i) = a(J_i) \wedge B(v^i) = b_i \\ \mathsf{undefined} & \text{otherwise.} \end{cases}$$

The random variable γ can be clearly computed in the manner required for F_b, and by the key Lemma 31.2.1 it is undefined only at an infinitesimal fraction of samples. Hence

$$Prob_{\omega\in\Omega_b}[\ B(\alpha(\omega),\iota(\omega),\gamma(\omega))\ ]$$

is infinitesimaly close to 1, and the equality

$$[\![\ B(\alpha(\omega),\iota(\omega),\gamma(\omega))]\!] \ = \ 1_{\mathcal{B}}$$

is established.

q.e.d.

31.4 What does and what does not follow from $K(F_b)$

We shall now reformulate Theorem 31.0.2 formally, specifying the requirements on the parameters involved, and use Theorem 31.3.1 to prove it. This statement does not use any nonstandard objects.

Theorem 31.4.1 *Let $k \geq 1$ be a natural number parameter and $m = m(k)$ a function of k such that*

$$k \ < \ m(k) \ < \ 2^{k^{o(1)}} \ .$$

Let A_k be $k \times m$ 0-1 matrices that are $(\log m, k^{1/3})$ designs. Let f be an $NP \cap coNP$ function that is a hard bit of a one way permutation, and assume that it has an exponential hardness

$$H_f(\ell) \ \geq \ 2^{\ell^\epsilon}, \ \textit{some } \epsilon > 0 \ .$$

Let $g : \{0,1\}^* \to \{0,1\}^*$ be the map that is defined on $\{0,1\}^k$ as $NW_{A_k,f}$.

Assume that R is an infinite NP set that it satisfies the following lengths-condition:

- *There are infinitely many elements of R whose length equals to $m(k)$ for some $k \geq 1$.*

Then it is consistent with Cook's theory PV and, in fact, with the true universal theory $Th_\forall(L_{PV})$ in the language of PV, that

$$Rng(NW_{A,f}) \cap R \neq \emptyset \ .$$

Proof :

Let g be a map and R be an infinite NP set satisfying the hypothesis of the theorem. If R intersects the range of g there is nothing to prove. So assume that R and $Rng(g)$ are disjoint.

As R is infinite and satisfies the lengths-condition it follows by the overspill in $\mathcal{M}$ (see Chapter A) that R has a nonstandard element (in fact, there are cofinaly many of them) of some nonstandard length m of the form $m = m(n)$ for some nonstandard n. Moreover, b is not in the range of g. Fix one such non-standard element b, and parameters n and m.

Take n as the parameter defining the cut $\mathcal{M}_n$. Then $b \in \mathcal{M}_n$ and we may apply Theorem 31.0.2 and construct the Boolean valued structure $K(F_b)$ in which PV and $Th_\forall(L_{PV})$ are valid but also

$$[\![\ b \in Rng(g)\ ]\!] \ = \ 1_{\mathcal{B}} \ .$$

The membership of b in R is witnessed in $\mathcal{M}$ by a string of length $m^{O(1)}$ and thus this strings is in $\mathcal{M}_n$, and hence in $K(F_b)$, too. Hence

$$[\![\ b \in R\ ]\!] \ = \ 1_{\mathcal{B}}$$

and we have

$$[\![\ Rng(g) \cap R \ \neq \ \emptyset\ ]\!] \ = \ 0_{\mathcal{B}}$$

The theorem follows from Lemma 1.4.1.

q.e.d.

For the sake of an easier discussion to follow (and to avoid a confusion) let us formulate the statement from the theorem whose consistency was shown (for each R) as a statement of its own.

Statement (S): *Let $k \geq 1$ be a natural number parameter and $m = m(k)$ a function of k such that*

$$k \ < \ m(k) \ < \ 2^{k^{o(1)}} \ .$$

Let A_k be $k \times m$ 0-1 matrices that are $(\log m, k^{1/3})$ designs. Let f be an $NP \cap coNP$ function that is a hard bit of a one way permutation, and assume that it has an exponential hardness

$$H_f(\ell) \ \geq \ 2^{\ell^\epsilon}, \ \textit{some } \epsilon > 0 \ .$$

Let $g : \{0,1\}^* \to \{0,1\}^*$ be the map that is defined on $\{0,1\}^k$ as $NW_{A_k,f}$. Assume that R is an infinite NP-set which has infinitely many elements whose length equals to $m(k)$ for some $k \geq 1$.

Then

$$Rng(NW_{A,f}) \cap R \neq \emptyset \ .$$

Lemma 31.4.2 *Statement (S) implies Razborov's Conjecture 30.2.1 for all proof systems P (not just EF).*

Proof :

The NW-generator from the conjecture satisfies the properties of the map g required in Statement (S).

Let P be a proof system and $k \geq 1$ a constant. The set of all b for which the τ-formula $\tau(g)_b$ has a P-proof of length at most $|\tau(g)_b|^k$ is in NP and it is disjoint (by the soundness of P) from the range of g. It also trivially satisfies the lengths-condition.

That is a contradiction with Statement (S).

q.e.d.

We thus view Theorem 31.4.1 as relevant to Conjecture 30.2.1. In addition, theory PV corresponds to EF and theory $Th_\forall(L_{PV})$ corresponds to (the union of) all proof systems.

Theorem 31.4.1 seems to be also relevant (or, at least, related) to Rudich's demi-bit Conjecture 30.4.1. Take Statement (S) for $m := n+1$. In that case the lengths-condition is void. The cut $\mathcal{M}_n$ contains all numbers of the length subexponential in n, hence the theorem yields (as $m = n+1$) the consistency of Statement (S) with R from subexponential nondeterministic time as in Conjecture 30.4.1 (but the conjecture allows also nonuniform algorithms).

Statement (S) is weaker than Conjecture 30.4.1 in the sense that the latter claims the existence of a P/poly map while in (S) we have an NP $\cap$ *co*NP-map. On the other hand the conjecture is weaker as it does not rule out infinite sets R but only those having a subexponential measure.

Let us conclude this chapter by spelling out what does not follow from the construction. Assume for a moment that Conjecture 30.2.1 is not true or, more generally, that Statement (S) is not true. Let f be a hard NP $\cap$ *co*NP-function and denote by g the NW-generator based on it (with the parameters as in this chapter). Assume that g is not hard for a proof system P. By the overspill in $\mathcal{M}$ we get a nonstandard n, a string $b \in \mathcal{M}_n$ of length $m = m(n)$ that is not in the range of g, and also a P-proof $\pi \in \mathcal{M}_n$ of $\tau(g)_b$. The proof π will be present also in $K(F_b)$ and the soundness of P is valid there (the soundness is a true universal statement). But the statement the τ-formula expresses:

$$\forall x \exists i \in [m] \forall y^i \neg [\ h(y^i) = x(J_i) \wedge B(y^i) = b_i\ ]$$

and π proves is not valid in $K(F_b)$.

The reader may wonder why in this situation we cannot apply the approach of Chapter 30.2.1 and derive a contradiction, and thus proving Statement (S). The reason is the following. The variables of the τ-formula corresponds to (bits of) x and of all y^i. In the model $K(F_b)$ we have a truth assignment corresponding to a witness for $\exists x$ in

$$\exists x \forall i \in [m] \exists y^i\ [\ h(y^i) = x(J_i) \wedge B(y^i) = b_i\ ]$$

and also truth assignments corresponding to individual witnesses y^i but we do not have a string there that would collect all these assignments together. The principle asserting the existence of such a string is the so called sharply bounded collection scheme and, unfortunately, it is not provable in $Th_\forall(L_{PV})$ unless factoring is easy (cf.[31]). But such a string seems to be needed in order to apply the soundness of P.

What we need to arrange in the model is the following form of a reflection principle for P. Denote by $\varphi_i(y^i)$ the propositional translation of

$$\neg[h(y^i) = \alpha(J_i) \wedge B(y^i) = b_i]\ ,$$

a formula defined in $K(F_b)$. The substitution instance of π (substituting for x) is a proof of a disjunction

$$\bigvee_i \varphi_i(y^i)$$

in disjoint sets of variables. For each φ_i we have a falsifying assignment. This situation should not be possible in the constructed model of $Th_\forall(L_{PV})$.

The reader may still wonder if the construction cannot be modified to achieve this. I am wondering too; paper [71] presents (in classical terms) variants of the construction, considers quantitative aspects of relations between various parameters, and discusses possibilities how to extend the construction. Perhaps the most promising avenue is to try to employ the *feasible disjunction property* which can be assumed to hold, for the purpose of proving lower bounds, without a loss of generality for every proof system (see [71]).

Appendix

Appendix A

Non-standard models and the ultrapower construction

In this chapter we present the ultrapower construction of a non-standard model of the true arithmetic. The construction is quite elementary and intuitive, and we present it in a self-contained manner (proofs of all statements are included). Indeed, we also discuss a few other basic notions and terminology useful when dealing with non-standard models. All this material is included in order to help a reader with a less experience in mathematical logic to create a mental image of model $\mathcal{M}$ that forms the ambient universe for our constructions (Section 1.1).

For the presentation of the construction we fix the language to be L_{PA}, the language of Peano arithmetic (Section 5.1), having constants 0 and 1, two binary functions $x+y$ and $x \cdot y$, and the binary relation $x < y$. When talking about arithmetic in mathematical logic this is the usual choice of language. In Section 1.1 we opted for a much larger language L_{all} as the language of $\mathcal{M}$; we shall discuss the differences after Theorem A.0.11 but now we want to keep the whole picture as elementary as possible.

Each symbol from L_{PA} has a canonical interpretation in the set of natural numbers $\mathbf{N}$. We shall denote the resulting L_{PA}-structure also $\mathbf{N}$. The phrase **the true arithmetic** means $Th_{L_{PA}}(\mathbf{N})$, the L_{PA}-theory of $\mathbf{N}$, i.e. the set of all L_{PA}-sentences true in $\mathbf{N}$. The structure $\mathbf{N}$ is called the **standard model** of arithmetic.

Several general theorems of mathematical logic guarantee the existence of L_{PA}-structures that satisfy the same theory as $\mathbf{N}$, i.e. they are **elemen-**

tary equivalent to $\mathbf{N}$, but are not isomorphic to $\mathbf{N}$. Such structures are called **non-standard models** of true arithmetic. We shall later employ the ultrapower construction to describe one such model. But before we do so we will observe a few basic properties shared by all non-standard models of arithmetic.

Let $\mathbf{M}$ denote a non-standard model $(M, 0^{\mathbf{M}}, 1^{\mathbf{M}}, =^{\mathbf{M}}, \cdot^{\mathbf{M}}, <^{\mathbf{M}})$ of true arithmetic. It holds that $<$ is a strict linear ordering in $\mathbf{N}$ and this can be expressed by one L_{PA}-sentence, hence the same is true in $\mathbf{M}$. Analogous arguments yield all following statements. Constant 0 is $<$-minimal in $\mathbf{N}$, hence $0^{\mathbf{M}}$ is $<^{\mathbf{M}}$-minimal in $\mathbf{M}$ too. Constant 1 is bigger than 0 in $\mathbf{N}$ but there is no element between them, and thus $1^{\mathbf{M}}$ is bigger than $0^{\mathbf{M}}$, in $\mathbf{M}$ with no element of $\mathcal{M}$ in between.

Each positive element $0 < k \in \mathbf{N}$ is the value of a closed L_{PA} term s_k of the form

$$s_k \ := \ 1 + (1 + (1 + \ldots (1 + 1) \ldots)$$

with k occurrences of 1. Put also $s_0 := 0$. In $\mathbf{N}$ it is true for each such term s_k that s_k is smaller than s_{k+1} without an element in between. Hence again this will hold in $\mathbf{M}$ too about values $(s_k)^{\mathbf{M}}$ of the terms in the structure. Furthermore, for any $k, \ell \in \mathbf{N}$ it holds in $\mathbf{N}$:

$$s_k + s_\ell = s_{k+\ell} \ \text{ and } \ s_k \cdot s_\ell = s_{k \cdot \ell} \ .$$

These L_{PA}-sentences are valid in $\mathbf{M}$ too.

Consequently, the sequence of values of L_{PA}-terms in $\mathbf{M}$:

$$0^{\mathbf{M}}, 1^{\mathbf{M}}, (s_2)^{\mathbf{M}}, (s_3)^{\mathbf{M}}, \ldots$$

forms an initial segment of $\mathbf{M}$ (w.r.t. to the ordering $<^{\mathbf{M}}$) that is isomorphic to $\mathbf{N}$. They are called **standard numbers**. It is customary (in order to simplify the notation) to identify $\mathbf{N}$ with this initial segment and to denote $(s_k)^{\mathbf{M}}$ simply k; hence we think that $\mathbf{N} \subseteq \mathbf{M}$.

As we have assumed that $\mathbf{N}$ and $\mathbf{M}$ are not isomorphic there must be an element $n \in \mathbf{M} \setminus \mathbf{N}$. Each such element is called a **non-standard number**. As there can be nothing between the values of s_k and s_{k+1}, n is bigger than all standard numbers k.

Once we have one non-standard number n we can argue for the existence of many more. For example, $n + 1, n + 2, \ldots$ are non-standard. But there must also be an element m such that $m + 1 = n$ (as that is true in $\mathbf{N}$ about

all numbers bigger than 0 which n is in $\mathbf{M}$). It is sensible to denote it $n-1$. And there will be $n-2=(n-1)-1, n-3, \ldots$. Thus each non-standard number has around itself a block of numbers that is order-isomorphic to the integers $\mathbf{Z}$.

There will be also such a $\mathbf{Z}$-block around $n+n$ and it is easy to see that they will be disjoint as there are infinitely many elements of $\mathbf{M}$ between n and $n+n$. In the other direction there must be a unique element m such that $m+m=n$ or $(m+m)+1=n$; denote it $\lfloor \frac{n}{2} \rfloor$. The $\mathbf{Z}$-block around $\lfloor \frac{n}{2} \rfloor$ is again disjoint from the $\mathbf{Z}$-block around n. Further there will be elements with values $\lfloor \frac{3n}{4} \rfloor$ and $\lfloor \frac{3n}{2} \rfloor$, etc. The reader can verify in this elementary manner that the $\mathbf{Z}$-blocks of non-standard numbers are linearly ordered by $<^{\mathbf{M}}$ and that this ordering is dense, and has neither minimal nor maximal block.

Speaking of integers: integers $\mathbf{Z}$ and rational numbers $\mathbf{Q}$ are easy to construct from natural numbers $\mathbf{N}$ and the same construction applied to a non-standard model $\mathbf{M}$ will give non-standard integers and rationals. We called the latter $\mathbf{M}$-rationals in Section 1.1. $\mathbf{M}$-rational q that is positive but smaller than $\frac{1}{k}$ for any standard $k \in \mathbf{N}$, is called **infinitesimal**. For example, if $n \in \mathbf{M}$ is non-standard then $\frac{1}{n}$ is infinitesimal.

Elementary but very useful properties of non-standard models are the so called **Overspill** and **Underspill**.

Theorem A.0.3 (Overspill) *Let* $\mathbf{M}$ *be a non-standard model of true arithmetic and let* $\varphi(x)$ *be any* L_{PA}*-formula possibly with parameters from* $\mathbf{M}$*.*

Assume $\varphi(k)$ *holds in* $\mathbf{M}$ *for all standard numbers* k*. Then there is also a non-standard number* n *such that* $\varphi(n)$ *holds.*

Proof :

Assume $\varphi(x)$ has the form $\psi(x, b_1, \ldots, b_t)$ with $b_1, \ldots, b_t \in M$, where $\psi(x, y_1, \ldots, y_t)$ has no parameters.

Assume for the sake of contradiction that $\varphi(k)$ holds for all standard numbers k but for no non-standard number. Then in $\mathbf{M}$ the following L_{PA}-sentence holds:

$$\exists y_1, \ldots, y_t \, [\, \psi(0, y_1, \ldots, y_t) \;\wedge\; \forall x(\psi(x, y_1, \ldots, y_t) \rightarrow \psi(x+1, y_1, \ldots, y_t)) \\ \wedge\; \neg \forall x \psi(x, y_1, \ldots, y_t) \,] \,.$$

But that is impossible as the sentence can never hold in $\mathbf{N}$: it would violate induction.

q.e.d.

The reader can verify that analogous reasoning gives also the Underspill.

Theorem A.0.4 (Underspill) *Let* $\mathbf{M}$ *be a non-standard model of true arithmetic and let* $\varphi(x)$ *be any* L_{PA}*-formula possibly with parameters from* $\mathbf{M}$*.*

Assume $\varphi(n)$ *holds in* $\mathbf{M}$ *for arbitrarily small non-standard number* n*: For each non-standard* m *there is a non-standard* $n < m$ *such that* $\varphi(n)$ *holds.*

Then there is also a standard number k *such that* $\varphi(k)$ *holds.*

These two properties are shared by all non-standard models. But the two properties (1) and (2) of $\mathcal{M}$ from Section 1.1 (see also below) are not. To arrange these two properties we have to construct specific models. We now embark on one such construction, the ultrapower.

Let us denote by $\prod_\omega \mathbf{N}$ the set of all functions

$$f \ : \ \omega \to \mathbf{N} \ .$$

Here ω and $\mathbf{N}$ are the same sets but the role of arguments and values is quite different in the construction and we prefer to show this in the notation as well.

The universe of the future model will be "essentially" formed by $\prod_\omega \mathbf{N}$. The qualification essentially is, however, crucial. For the success of the construction it will be necessary to identify those functions f and g that agree on most but not necessarily all arguments. For example, if f and g are eventually equal, i.e. equal for large enough arguments, then we will identify them. In other words, whether or not two functions f and g denote the same element of the universe will depend only on whether or not the set

$$\langle\langle f = g \rangle\rangle \ := \ \{i \in \omega \mid f(i) = g(i)\}$$

is "large". Every cofinite set will be large but we need to define the "largeness" for all subsets of ω. The concept that formalizes this informal notion is the **non-principal ultrafilter**.

Definition A.0.5 *A set* $\mathcal{U} \subseteq \mathcal{P}(\omega)$ *of some subsets of* ω *is called a* **filter** *iff it satisfies the following three conditions:*

1. $\emptyset \notin \mathcal{U}$ *and* $\omega \in \mathcal{U}$.

2. *If* $I, J \in \mathcal{U}$ *then* $I \cap J \in \mathcal{U}$.

3. *If* $I \in \mathcal{U}$ *and* $J \supseteq I$ *then* $J \in \mathcal{U}$.

A filter that satisfies also

4. *For any* $I \subseteq \omega$ *either* $I \in \mathcal{U}$ *or* $\bar{I} := \omega \setminus I \in \mathcal{U}$

is an **ultrafilter**.

An ultrafilter that contains no finite set is called **non-principal**.

It is the fourth property that makes the notion of a non-principal ultrafilter quite non-transparent; the first three conditions are satisfied, for example, by the class of cofinite sets. Indeed, there is no similarly explicit example of a non-principal ultrafilter and its existence can be proved only from the Axiom of Choice. We shall leave these set-theoretic issues aside and simply take the existence of a non-principal ultrafilter $\mathcal{U}$ as given. The reader may consult [48] for details.

Armed with $\mathcal{U}$ we define an equivalence relation on $\prod_\omega \mathbf{N}$ as follows:

$$f \sim g \text{ iff } \langle\langle f = g \rangle\rangle \in \mathcal{U} \ .$$

Lemma A.0.6 *Relation* $\sim$ *is an equivalence relation on* $\prod_\omega \mathbf{N}$.

Proof :

$\langle\langle f = f \rangle\rangle = \omega$ which is in $\mathcal{U}$ by condition 1. of Definition A.0.5, and clearly $\langle\langle f = g \rangle\rangle = \langle\langle g = f \rangle\rangle$. Hence $\sim$ is reflexive and symmetric.

Assume $\langle\langle f = g \rangle\rangle = I \in \mathcal{U}$ and $\langle\langle g = h \rangle\rangle = J \in \mathcal{U}$. Clearly $\langle\langle f = h \rangle\rangle \supseteq I \cap J$ and hence $\langle\langle f = h \rangle\rangle \in \mathcal{U}$ by conditions 2. and 3. of Definition A.0.5. That is, $\sim$ is transitive.

q.e.d.

For a function $f \in \prod_\omega \mathbf{N}$ denote by $[f]$ the $\sim$-equivalence class of f. The universe of the future model consists of these equivalence classes. We shall denote the universe, as well as the whole L_{PA}-structure on it to be defined shortly, $\mathbf{N}^*$ (the usual notation is $\prod_{\mathcal{U}} \mathbf{N}$).

To make $\mathbf{N}^*$ into a structure we need to interpret on it the symbols from language L_{PA}. We shall denote the interpretations by the superscript $*$. Put:

$$0^* := [c_0] \text{ and } 1^* = [c_1]$$

where c_0 and c_1 are functions on ω that are constantly 0 or 1, respectively. Next define:

$$[f] +^* [g] := [f + g]$$

and

$$[f] \cdot^* [g] := [f \cdot g] .$$

Finally:

$$f <^* g \text{ iff } \langle\langle f < g \rangle\rangle \in \mathcal{U} .$$

Lemma A.0.7 *The functions $+^*$ and $\cdot^*$ and the relation $<^*$ are well-defined on $\mathbf{N}^*$, i.e. the definitions do not depend on the choice of representants f and g from the equivalence classes $[f]$ and $[g]$.*

Proof :

We need to show that if $f \sim f'$ and $g \sim g'$ then

$$[f + g] = [f' + g'] \text{ and } [f \cdot g] = [f' \cdot g']$$

and

$$\langle\langle f < g \rangle\rangle \in \mathcal{U} \text{ iff } \langle\langle f' < g' \rangle\rangle \in \mathcal{U} .$$

Denote $I := \langle\langle f = f' \rangle\rangle$ and $J := \langle\langle g = g' \rangle\rangle$. Hence $I, J \in \mathcal{U}$.

We have

$$\langle\langle f + f' = g + g' \rangle\rangle \supseteq I \cap J .$$

As the right-hand side is in $\mathcal{U}$ by condition 2. of Definition A.0.5, and so is - by condition 3. - the left-hand side. That is, $[f + f'] = [g + g']$. The argument for the multiplication is identical.

To check the property for the ordering relation denote $A := \langle\langle f < g \rangle\rangle$ and $B := \langle\langle f' < g' \rangle\rangle$, and assume $A \in \mathcal{U}$. As $I, J, A \in \mathcal{U}$, also $I \cap J \cap A \in \mathcal{U}$ (condition 2.). But $B \supseteq I \cap J \cap A$ and so $B \in \mathcal{U}$ too (condition 3.). The opposite direction is analogous.

q.e.d.

Now we have our structure

$$\mathbf{N}^* \;=\; (\mathbf{N}^*, 0^*, 1^*, +^*, \cdot^*, <^*)$$

but we still need to establish that it is a model of true arithmetic, and that it is non-standard. For both tasks we need some criterion how to decide what is true and what is false in the structure. Such a criterion is provided by a simple and beautiful theorem of Loś. (The notation $\langle\langle \ldots \rangle\rangle$ is not a usual in this context; we have borrowed it from our earlier chapters.)

Theorem A.0.8 (Loś theorem) *Let $\varphi(x_1, \ldots, x_k)$ be an L_{PA} formula with all free variables among $x_1, \ldots, x_k$, and let $f_1, \ldots, f_k \in \prod_\omega \mathbf{N}$ be arbitrary. Denote by*

$$\langle\langle \varphi(f_1, \ldots, f_k) \rangle\rangle$$

the set of all $i \in \omega$ such that

$$\mathbf{N} \;\models\; \varphi(f_1(i), \ldots, f_k(i)) \;.$$

Then

$$\mathbf{N}^* \;\models\; \varphi([f_1], \ldots, [f_k]) \quad \textit{iff} \quad \langle\langle \varphi(f_1, \ldots, f_k) \rangle\rangle \in \mathcal{U} \;.$$

Proof :

This is proved by induction on the number of connectives and quantifiers in φ. To start the induction we need to establish the theorem for atomic formulas $t = s$ and $t < s$, where t and s are L_{PA}-terms. This follows readily from the following property of terms that is verified by induction on the number of function symbols occurring in the term, using the definition of the operations on $\mathbf{N}^*$.

Claim 1: *Let $t(y_1, \ldots, y_\ell)$ be an L_{PA}-term with variables among $y_1, \ldots, y_\ell$, and let $g_1, \ldots, g_\ell$ be arbitrary functions from $\prod_\omega \mathbf{N}$.*

Then

$$t([g_1], \ldots, [g_\ell]) = [t(g_1, \ldots, g_\ell)] \;.$$

If φ is a Boolean combination of smaller formulas we use for the induction step the following claim, a straightforward consequence of the definition of an ultrafilter.

Claim 2: *For any $I, J \subseteq \omega$:*

1. $I \cap J \in \mathcal{U}$ *iff* $I \in \mathcal{U} \wedge J \in \mathcal{U}$.
2. $I \cup J \in \mathcal{U}$ *iff* $I \in \mathcal{U} \vee J \in \mathcal{U}$.
3. $I \in \mathcal{U}$ *iff* $\bar{I} \notin \mathcal{U}$.

Let us see how this handles the Boolean connectives. Assume for instance that $\varphi = \psi \wedge \eta$. By induction hypothesis we have Loś theorem for both ψ and η. Then using Tarski's satisfiability conditions in a structure we compute:

$$\mathbf{N}^* \models \varphi([f_1], \ldots, [f_k])$$

iff

$$\mathbf{N}^* \models \psi([f_1], \ldots, [f_k]) \text{ and } \mathbf{N}^* \models \eta([f_1], \ldots, [f_k])$$

iff

$$\langle\langle \psi(f_1, \ldots, f_k) \rangle\rangle \in \mathcal{U} \text{ and } \langle\langle \eta(f_1, \ldots, f_k) \rangle\rangle \in \mathcal{U}$$

iff (by 1. of Claim 2)

$$\langle\langle \psi(f_1, \ldots, f_k) \rangle\rangle \cap \langle\langle \eta(f_1, \ldots, f_k) \rangle\rangle \in \mathcal{U}$$

iff

$$\langle\langle \varphi(f_1, \ldots, f_k) \rangle\rangle \in \mathcal{U} \ .$$

If φ is a disjunction or a negation analogous argument works, using the other two conditions from Claim 2. Note that the first two conditions hold also for a filter but the third one needs an ultrafilter.

Finally we need to treat the case when φ starts with a quantifier. Let $\varphi = \exists y \rho$ and assume first that

$$\mathbf{N}^* \models \exists y \rho([f_1], \ldots, [f_k], y)$$

and hence

$$\mathbf{N}^* \models \rho([f_1], \ldots, [f_k], [g])$$

for some $g \in \prod_\omega \mathbf{N}$. By induction hypothesis this is equivalent to

$$\langle\langle \rho(f_1, \ldots, f_k, g) \rangle\rangle \in \mathcal{U} \ .$$

Clearly

$$\langle\langle \rho(f_1, \ldots, f_k, g) \rangle\rangle \subseteq \langle\langle \exists y \rho(f_1, \ldots, f_k, y) \rangle\rangle$$

and hence

$$\langle\langle \exists y \rho(f_1, \ldots, f_k, y) \rangle\rangle = \langle\langle \varphi(f_1, \ldots, f_k) \rangle\rangle \in \mathcal{U}$$

too. This establishes one direction of the equivalence from Loś theorem.

The other direction is also simple. Assume

$$\langle\langle \exists y \rho(f_1, \ldots, f_k, y) \rangle\rangle \in \mathcal{U} \ .$$

Take a function g defined as follows: $g(i)$ equals to the minimal number $j \in \mathbf{N}$ such that

$$\mathbf{N} \ \models \ \rho(f_1(i), \ldots, f_k(i), j)$$

if it exists, and to 0 otherwise. Clearly then

$$\langle\langle \exists y \rho(f_1, \ldots, f_k, y) \rangle\rangle \ = \ \langle\langle \rho(f_1, \ldots, f_k, g) \rangle\rangle$$

and hence

$$\langle\langle \rho(f_1, \ldots, f_k, g) \rangle\rangle \in \mathcal{U}$$

too. The induction hypothesis then yields

$$\mathbf{N}^* \ \models \ \rho([f_1], \ldots, [f_k], [g])$$

which implies the wanted

$$\mathbf{N}^* \ \models \ \exists y \rho([f_1], \ldots, [f_k], y) \ .$$

The case of the universal quantifier is completely analogous (alternatively we can replace it by a combination of the existential one with two negations).

q.e.d.

Note that for φ an L_{PA} sentence without parameters we either have $\langle\langle \varphi \rangle\rangle = \omega$ or $\langle\langle \varphi \rangle\rangle = \emptyset$, the former when φ holds in $\mathbf{N}$ and the latter when it does not. This immediately yields the following statement.

Corollary A.0.9 $\mathbf{N}^*$ *is a model of true arithmetic.*

So far we used that $\mathcal{U}$ is an ultrafilter but not that it is non-principal. This is used in the next statement.

Lemma A.0.10 $\mathbf{N}^*$ *is non-standard.*

Proof :

The initial segment in $\mathbf{N}^*$ that is isomorphic to $\mathbf{N}$ is the sequence

$$[c_0], [c_1], [c_2], \ldots$$

where c_k is the constant function with the value $k \in \mathbf{N}$. This is easy to verify using Loś theorem.

Define function f by:

$$f(i) \ := \ i \ .$$

Then for each k the set $\langle\langle c_k = f \rangle\rangle$ is finite and therefore, by the assumption of non-principality, not in $\mathcal{U}$. Hence by Loś theorem $[f] \neq [c_k]$, for each k.

q.e.d.

It may be illustrative for the reader to define functions of various growth rates and compare the elements of the non-standard model they define. This is quite consistent with the idea of comparing functions by their asymptotic behavior.

It remains to establish for $\mathbf{N}^*$ the two properties (1) and (2) of $\mathcal{M}$ from Section 1.1. The properties formulated for $\mathbf{N}^*$ are:

(1) If a_k, $k \in \mathbf{N}$, is a countable family of elements of $\mathbf{N}^*$ then there exists a non-standard $t \in \mathbf{N}^*$ and a sequence $(b_i)_{i<t} \in \mathbf{N}^*$ such that $b_k = a_k$ for all $k \in \mathbf{N}$.

(2) If A_k, $k \in \mathbf{N}$, is a countable family of definable subsets of $\mathbf{N}^*$ such that $\bigcap_{i<k} A_i \neq \emptyset$ for all $k \geq 1$, then $\bigcap_k A_k \neq \emptyset$.

In property (1) we talk about a sequence of non-standard length as being an element of the model. Formally one needs to show how a single number can code a sequence of other numbers in a way that the ith element of the sequence can be defined (from the code and the index i) by a formula of the arithmetic language. This requires some work (and goes back to Gödel) and the reader can consult [37]. However, there is an informal picture that helps to understand these statements even to those not familiar with the development of formal arithmetic. This is based on the observation (which, when proved in detail, is as technical as the coding) that instead of numbers we can think of finite sets. It is well-known how the set universe interprets

natural numbers (and then goes on interpreting ordinals etc.) and in this sense we can interpret natural numbers in the universe of hereditary finite sets. But there is also an interpretation going in the opposite direction. For example, think of 0 as coding the empty set $\emptyset$ and a number $k = 2^{u_1} + \ldots 2^{u_t} > 0$ with $u_1 > \ldots > u_t \geq 0$ as coding the set whose elements are the sets coded by $u_1, \ldots, u_t$.

Everybody (hopefully) learns in school how pairs, sequences, functions, relations, etc. are represented by sets. Hence the interpretation of sets by numbers allows to transfer the same representation to the universe of numbers.

When the universe of numbers (i.e. a model of true arithmetic) contains also non-standard numbers it happens that some of them will code sequences whose length, while finite in the sense of the model (that is, the length is a number from the universe), will be non-standard. Similarly, a set coded in a non-standard model $\mathbf{M}$ may be finite but it may also be infinite. In the latter case its cardinality in the model is counted by some (necessarily non-standard) number of the model. Sets of both type are called $\mathbf{M}$-finite.

An $\mathbf{M}$-finite set is definable in the model but the opposite is not true; for example, the set of odd numbers is obviously definable but not $\mathbf{M}$-finite. It is easy to verify by induction that a definable set is coded by a number, is $\mathbf{M}$-finite, iff it is **bounded** in $\mathbf{M}$: there is a number in $\mathbf{M}$ bounding above all elements of the set.

I hope this helps the reader not familiar with the topic of formal arithmetic to understand the statements and the construction below. However, there is a limit how far one can travel with informal mental images. At some point it may be necessary to read a more formal treatment in a textbook.

Theorem A.0.11 *Both properties (1) and (2) hold for* $\mathbf{N}^*$.

Proof :

First observe that (1) follows from (2). To see this define in $\mathbf{N}^*$ sets A_k as follows; A_k contains all sequences (tacitly, $\mathbf{N}^*$-finite and coded in $\mathbf{N}^*$) that start with an initial segment $(a_0, a_2, \ldots, a_k)$.

Clearly $(a_0, a_2, \ldots, a_{k-1}) \in \bigcap_{i<k} A_i$ and hence the intersection is non-empty. Condition (2) then implies that the intersection of all sets A_k must be non-empty. In other words, there must be a sequence whose length is a non-standard number and whose k-th element is a_k, for all standard k.

To prove property (2) assume that A_k are definable subsets of $\mathbf{N}^*$ with non-empty finite intersections. Replacing A_k by $\bigcap_{i\leq k} A_i$ we may assume that

$$A_0 \supseteq A_1 \supseteq \dots \ .$$

Let $\alpha_k(x,y)$ be L_{PA}-formulas and $[f_k] \in \mathbf{N}^*$ parameters such that $\alpha_k(x,[f_k])$ defines A_k (several parameters can be included in one, using the pairing function).

The finite intersection property now means simply that for all k

$$\mathbf{N}^* \models \exists x \alpha_k(x,[f_k])$$

which is equivalent to

$$\langle\langle \exists x \alpha_k(x,f_k) \rangle\rangle \in \mathcal{U} \ .$$

Let $[g_k]$ be a witness to the existential quantifier:

$$\langle\langle \alpha_k(g_k,f_k) \rangle\rangle \in \mathcal{U} \ .$$

What we want is a function $h \in \prod_\omega \mathbf{N}$ such that for all k:

$$\langle\langle \alpha_k(h,f_k) \rangle\rangle \in \mathcal{U} \ .$$

We leave the task to devise a suitable function (using g_k's) as an exercise. (Hint: If you did not needed the non-principality of $\mathcal{U}$ something must be wrong.)

q.e.d.

Let us conclude this appendix by discussing the issue of the language of $\mathcal{M}$. In Section 1.1 we have chosen a much larger language L_{all} as the language of $\mathcal{M}$. This has the technical advantage that it contains all other languages that are needed in different constructions and we do not need to change the ambient model every time. The disadvantage is that L_{all} is huge; it has the cardinality of the continuum. In particular, it is not countable.

The $\aleph_1$-saturation that we invoked in Section 1.1 is a property of models that does imply properties (1) and (2) but for uncountable languages is stronger. On the other hand, the particular ultrapower with the index set ω we have presented here will give properties (1) and (2) for any language (but yields the $\aleph_1$-saturation only for countable).

It is this reason why we proved properties (1) and (2) for $\mathbf{N}^*$ rather than the $\aleph_1$-saturation (which is also true as the language L_{PA} is finite): the argument from the proof of Theorem A.0.11 would work identically for any language (a subset of L_{all}). In any case, as mentioned in Section 1.1, these two properties (1) and (2) are essentially the only consequences of the assumption of the $\aleph_1$-saturation of $\mathcal{M}$ we use in this book.

List of symbols, standard notation and conventions

We list first (in the order of appearance) the specific symbols used in the book, pointing out the sections or chapters where they are defined.

Section 1.1:

- L_{all}: language having symbols for all functions and relations on $\mathbf{N}$, the standard structure of natural numbers.
- $\mathcal{M}$ is an $\aleph_1$-saturated model of the theory of $\mathbf{N}$ in the language L_{all}.

Section 1.2.

- $\Omega \in \mathcal{M}$ is a sample space.
- $\mathcal{A} \in \mathcal{M}$ is the Boolean algebra of subsets Ω.
- $\mathcal{I} \subseteq \mathcal{A}$ is the ideal of sets having infinitesimal counting measure.
- $\mathcal{B}$ is the quotient algebra $\mathcal{A}/\mathcal{I}$.
- μ is the induced measure on $\mathcal{B}$ (Loeb's measure) with values in the standard reals $\mathbf{R}$.

Section 1.3.

- F denotes a family of random variables defined on Ω.
- $K(F)$: the Boolean valued model resulting from F.
- $[\![\ldots]\!]$ is the Boolean truth evaluation of sentences.

Section 1.4.

- $Th_\Gamma(L)$: the set of L-sentences in prefix class Γ true in $\mathbf{N}$.

Section 2.1.

- $d(a,b)$ is a metric on $\mathcal{B}$.

Section 3.2.

- F_{PV}: family of polynomial-time functions on $\{0,1\}^n$.

Chapter 4.

- A_{Sk}: a Skolemization of A
- $Sk[A; f_1, \ldots, f_k]$: a conjunction of universal Skolem axioms

Section 5.1.

- L^2: second order extension of language L
- L_{PA}: the language of PA
- $X < t$: bounding relation
- $\Sigma_0^{1,b}$, Σ_∞^b, $\Sigma_i^{1,b}$, $\Pi_i^{1,b}$, $s\Sigma_1^{1,b}$, $s\Pi_i^{1,b}$: various classes of bounded formulas

Section 5.2.

- $\mathcal{M}_n$: a cut in $\mathcal{M}$
- L_n and L_n^2: languages

Section 5.3.

- $K(F,G)$: a second-order structure

Chapter 6.

- PA^-, $I\Delta_0$, $I\Delta_0(R)$, V_1^0: various theories

- IND: the scheme of induction

Chapter 7.

- $dp(T)$: the depth of a tree
- F_{rud}, G_{rud}, $K(F_{rud}, G_{rud})$, $L_n(F_{rud}, G_{rud})$: rudimentary families of random variables, the rudimentary structure and language

Section 8.1.

- $\langle\langle \ldots \rangle\rangle$ notation

Section 10.2.

- Ω_{tree}, F_{tree}, G_{tree} and $K(F_{tree}, G_{tree})$: the sample space and the families of the tree model

Section 12.1.

- $Par(x, X, Y)$: a formula formalizing parity

Section 13.1.

- Q_2, $Q_2V_1^0$, $Q_2\Sigma_0^{1,b}$, $Q_2\Sigma_\infty^{1,b}$: counting quantifier, a resulting theory and classes of formulas

Section 13.2.

- 2-cover: a family of sets counting parity of sets in another family

Chapter 14.

- $deg(T)$: the degree of an algebraic decision tree
- F_{alg}, G_{alg}, $K(F_{alg}, G_{alg})$: families of random variables and the algebraic tree model
- $\mathbf{F}_2^{low}[x_1, \ldots, x_n]$: a ring of low degree polynomials
- $\sum_i p_i \cdot i^*$: alternative formalism for elements of F_{alg}

Section 16.1

- $Closure(n)$: a formula expressing the existence of the transitive closure of a graph

Chapter 17.

- F, EF: Frege and Extended Frege systems
- $\oplus$, $F(\oplus)$: parity connective and a Frege system in DeMorgan language with $\oplus$
- F_d and $F_d(\oplus)$: constant depth subsystems

Chapter 18.

- $Fla_d(x, Y)$, $Prf_P(x, X, Y)$, $Sat_d(x, Z, Y)$: $\Sigma_0^{1,b}$ formulas defining depth d formulas and the provability and the satisfiability predicates
- $Ref_{P,d}(x, X, Y, Z)$, Ref_P: reflection principles

Chapter 19.

- $PHP(x, R)$, PHP_m: first order and propositional formulas formalizing the pigeonhole principle
- g_m: a gadget generator

Chapter 20.

- $K(F^0_{php}, G^0_{php})$ and Ω^0_{php}: a structure based on shallow PHP-trees and its sample space

Chapter 21.

- $K(F_{php}, G_{php})$ and Ω_{php}: a structure and its sample space

Chapter 23.

- L_{PV}: a language
- Σ^b_i, Π^b_i: classes of bounded formulas

- PV, S_2, T_2, S_2^i, T_2^i, U_1^1, V_1^1, BASIC: various theories
- PIND, LIND: schemes of polynomial induction and length induction

Chapter 24.

- $WPHP(C, a)$: PHP for circuit C
- RSA_k: a particular sample space
- $\#FM(C, a)$: sharply bounded function minimization
- $Sat(x, y)$: a PV formula defining the satisfaction relation
- $\Sigma_1^b - LENGTH - MAX$: a maximization principle

Chapter 25.

- Ω_0: a distribution on a sample space
- $\mathcal{A}_0$, $\mathcal{B}_0$: the resulting Boolean algebras
- $[\![\ldots]\!]_0$: the truth values in $\mathcal{B}_0$
- μ_0: the measure on $\mathcal{B}_0$.
- $\equiv_c$: computational indistinguishability (weaker definition)

Chapter 26.

- Ω_{oracle}, $K(F_{oracle})$ and $K(F_{oracle}, G_{oracle})$: sample space and structures

We have used some **standard notation and conventions** including:

- $[n] := \{1, \ldots, n\}$ while we identify n with $\{0, \ldots, n-1\}$.
- Notation $A(x_1, \ldots, x_k)$ implies that all free variables of the formula are among $x_1, \ldots, x_k$.

Conventions specific for the book include:

- Letters ϵ and δ always stand for a positive standard rational number.
- When we talk about positive infinitesimals we shall always denote them $1/t$ or similarly, for t a nonstandard element of $\mathcal{M}$.

Finally we remark few facts that do follow from the definitions but that cannot harm to mention explicitly:

- Probabilities $Pr_{\omega\in\Omega}[\ldots]$ are evaluated in $\mathcal{M}$ and hence their values are $\mathcal{M}$-rationals.
- If $\mu(b) = 1$ for $b = X/\mathcal{I}$, $X \in \mathcal{A}$, then it follows that the counting measure of X is infinitesimally close to 1 and so bounded from below by any $1 - \epsilon$ ($\epsilon > 0$ is standard by the convention above).

Bibliography

[1] M. Ajtai, Σ_1^1 - formulae on finite structures, *Annals of Pure and Applied Logic*, **24**, (1983), pp.1-48.

[2] M. Ajtai, The complexity of the pigeonhole principle, in: *Proc. IEEE 29th Annual Symp. on Foundation of Computer Science*, (1988), pp. 346-355.

[3] M. Ajtai, Parity and the pigeonhole principle, in: *Feasible Mathematics*, Eds. S.R.Buss and P.J.Scott, (1990), pp.1-24. Birkhauser.

[4] M. Ajtai, The independence of the modulo p counting principles, in: *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, (1994), pp.402-411. ACM Press.

[5] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov, and A. Wigderson, Pseudorandom generators in propositional proof complexity, *Electronic Colloquium on Computational Complexity*, Rep. No.**23**, (2000). Ext. abstract in: *Proc. of the 41st Annual Symp. on Foundation of Computer Science*, (2000), pp.43-53.

[6] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*, Cambridge University Press, (2009).

[7] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi, P. Pudlák, and A. Woods: Exponential lower bounds for the pigeonhole principle, in: *Annual ACM Symp. on Theory of Computing*, (1992), pp.200-220.

[8] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi, and P. Pudlák: Lower bounds on Hilbert's Nullstellensatz and propositional proofs, *Proceedings of the London Mathematical Society*, **(3) 73**, (1996), pp.1-26.

[9] C. H. Bennett and J. Gill, Relative to a Random Oracle A, $P^A \neq NP^A \neq co - NP^A$ with Probability 1, *SIAM J. Comput.*, **10(1)**, (1981), pp.96-113.

[10] G. Boole, *The mathematical analysis of logic*, Barclay and Macmillan, Cambridge, (1847).

[11] S. R. Buss, *Bounded Arithmetic*. Naples, Bibliopolis, (1986).

[12] S. R. Buss, Polynomial size proofs of the propositional pigeonhole principle, *J. Symbolic Logic*, **52**, (1987), pp.916-927.

[13] S. R. Buss, Axiomatizations and conservation results for fragments of bounded arithmetic, in: *Logic and Computation*, Contemporary Mathematics **106**, (1990), pp.57-84. Providence, American Mathematical Society.

[14] S. R. Buss, Some remarks on the lengths of propositional proofs, *Archive for Mathematical Logic*, **34**, (1995), pp.377-394.

[15] S. R. Buss, Relating the bounded arithmetic and polynomial time hierarchies, *Annals of Pure and Applied Logic*, **75**, (1995), pp.67-77.

[16] S. R. Buss, Lower bounds on Nullstellensatz proofs via designs, in: Proof Complexity and Feasible Arithmetics, eds. S. Buss and P. Beame, American Mathematical Society, Providence RI, (1998), pp. 59-71.

[17] S. R. Buss, and L. Hay, On truth-table reducibility to SAT and the difference hierarchy over NP, *Proc. Structure in Complexity*, IEEE, (1988), pp.224-233.

[18] S. R. Buss, R. Impagliazzo, J. Krajíček, P. Pudlák, A. A. Razborov, and J. Sgall: Proof complexity in algebraic systems and bounded depth Frege systems with modular counting, *Computational Complexity*, **6(3)**, (1996/1997), pp.256-298.

[19] S. R. Buss, and J. Krajíček, An application of boolean complexity to separation problems in bounded arithmetic, *Proceedings of the London Mathematical Society*, **69(3)**, (1994), pp. 1-21.

[20] S. R. Buss, J. Krajíček, and G. Takeuti, On provably total functions in bounded arithmetic theories R^i_3, U^i_2 and V^i_2, in: *Arithmetic, Proof Theory and Computational Complexity*, eds.P. Clote and J. Krajíček, (1993), pp.116-161, Oxford Press.

[21] C. C. Chang, and J. Keisler, *Model theory*, Ser. Studies in Logic, North-Holland, (1977).

[22] V. Chvatal and E. Szemeredi, Many hard examples for resolution, *J. of ACM*, **35(4)**, pp.759-768, (1988).

[23] M. Chiari, and J. Krajíček, Witnessing functions in bounded arithmetic and search problems, *J. of Symbolic Logic*, **63(3)**, (1998), pp. 1095-1115.

[24] Clegg, Edmonds, and R. Impagliazzo, Using the Groebner basis algorithm to find proofs of unsatisfiability, in: *Proc. 28^{th} Annual ACM Symp. on Theory of Computing*, (1996), pp. 174-183. ACM Press.

[25] A. Cobham, The intrinsic computational difficulty of functions, in : *Proc. Logic, Methodology and Philosophy of Science*, ed. Y. Bar-Hillel, North-Holland, (1965), pp. 24-30.

[26] S. A. Cook, The complexity of theorem proving procedures, in: *Proc. 3^{rd} Annual ACM Symp. on Theory of Computing*, (1971), pp. 151-158. ACM Press.

[27] S. A. Cook, Feasibly constructive proofs and the propositional calculus, in: *Proc. 7^{th} Annual ACM Symp. on Theory of Computing*, (1975), pp. 83-97. ACM Press.

[28] S. A. Cook, and J. Krajíček, Consequences of the Provability of $NP \subseteq P/poly$, *J. of Symbolic Logic*, **72(4)**, (2007), pp. 1353-1371.

[29] S. A. Cook, and P. Nguyen, *Logical foundations of proof complexity*, (2010), Cambridge U. Press.

[30] S. A. Cook, and Reckhow, The relative efficiency of propositional proof systems, *J. Symbolic Logic*,**44(1)**, (1979), pp.36-50.

[31] S. A. Cook, and N. Thapen, The strength of replacement in weak arithmetic, preprint, *ACM Transactions on Computational Logic*, **Vol 7:4**, (2006).

[32] P. Erdös, Some remarks on the theory of graphs, *Bull. of the AMS*, **53**, (1947), pp.292-294.

[33] M. Furst, J. B. Saxe, and M. Sipser, M. Parity, circuits and the polynomial-time hierarchy, *Math. Systems Theory*, (1984), **17**: 13–27.

[34] K. Gödel, a letter to John von Neumann from 1956, reprinted e.g. in: *Arithmetic, Proof Theory and Computational Complexity*, eds.P. Clote and J. Krajíček, (1993), Oxford University Press.

[35] O. Goldreich, *Foundations of cryptography*, Vol.1, Cambridge University Press, (2001).

[36] O. Goldreich, Candidate one-way functions based on expander graphs, *Electronic Colloquium on Computational Complexity*, Rep. No.**90**, (2000).

[37] P. Hájek and P. Pudlák, Metamathematics of first-order arithmetic, *Perspectives in Mathematical Logic*, (1993), 460 p. Springer-Verlag.

[38] J. Hanika, *Search Problems and Bounded Arithmetic*, PhD.Thesis, Charles University, Prague, (2004).

[39] J. Hastad, Almost optimal lower bounds for small depth circuits. in: *Randomness and Computation*, ed. S.Micali, Ser.Adv.Comp.Res., (1989), **5**: 143-170. JAI Pres.

[40] J. Hastad, R. Impagliazzo, L. Levin, and M. Luby, A Pseudorandom Generator from any one-way function, *SIAM Journal on Computing*, **Vol 28**, (1999), pp.1364–1396.

[41] E. A. Hirsch and D. Itsykson, An optimal heuristic randomized semidecision procedures, with application to proof complexity, preprint (2009).

[42] W. Hodges, *Model theory*, Encyclopedia of Mathematics and Its Applications, Vol. **42**, Cambridge University Press, (1993).

[43] R. Impagliazzo, V. Kabanets, and A. Wigderson, In Search of an Easy Witness: Exponential Time vs. Probabilistic Polynomial Time, *J.Comp.Syst.Sci.*, **65(4)**, (2002), pp.672-694.

[44] R. Impagliazzo, M. Naor, Efficient cryptographic schemes provably as secure as subset sum, *J. of Cryptology*, **9(4)**, (1996), pp.199-216.

[45] R. Impagliazzo, P. Pudlak, and J. Sgall, Lower bounds for the polynomial calculus and the Groebner basis algorithm, *Comput. Complexity*, **8(2)**, (1999), pp.127-144.

[46] R. Impagliazzo, and N. Segerlind, Counting axioms do not polynomially simulate counting gates, in: *Proc. IEEE 42nd Annual Symp. on Foundation of Computer Science*, (2001), pp. 200-209.

[47] R. Impagliazzo, and A. Wigderson, P = BPP unless E has subexponential circuits: derandomizing the XOR lemma, in: *Proc. of the 29th Annual ACM Symposium on Theory of Computing*, (1997), pp. 220-229.

[48] T. Jech, *Set theory*, (2003). Springer.

[49] E. Jeřábek, Dual weak pigeonhole principle, Boolean complexity, and derandomization, *Annals of Pure and Applied Logic*, **129**, (2004), pp.137.

[50] V. Kabanets and J.-Y. Cai, Circuit minimization problem, Extended abstract in Proc. of the 32nd Annual ACM Symposium on Theory of Computing, pp.73-79, (2000). Also: Technical Report TR99-045, Electronic Colloquium on Computational Complexity.

[51] R. Kaye, *Models of Peano arithmetic*, Oxford Science Publication, Oxford, UK, (1991).

[52] J. Krajíček, A fundamental problem of mathematical logic, *Annals of the Kurt Gödel Society*, Springer-Verlag, *Collegium Logicum*, Vol.**2**, (1996), pp.56-64.

[53] J. Krajíček, Fragments of bounded arithmetic and bounded query classes, *Transactions of the A.M.S.*, **338(2)**, (1993), pp.587-598.

[54] J. Krajíček, On Frege and Extended Frege Proof Systems. in: "Feasible Mathematics II", eds. P. Clote and J. Remmel, Birkhauser, (1995), pp. 284-319.

[55] J. Krajíček, *Bounded arithmetic, propositional logic, and complexity theory*, Encyclopedia of Mathematics and Its Applications, Vol. **60**, Cambridge University Press, (1995).

[56] J. Krajíček, On methods for proving lower bounds in propositional logic, in: *Logic and Scientific Methods* Eds. M. L. Dalla Chiara et al., (Vol. 1 of Proc. of the Tenth International Congress of Logic, Methodology and Philosophy of Science, Florence (August 19-25, 1995)), Synthese Library, Vol.259, Kluwer Academic Publ., Dordrecht, (1997), pp.69-83.

[57] J. Krajíček, Lower bounds for a proof system with an exponential speed-up over constant-depth Frege systems and over polynomial calculus, in: Eds. I.Prívara, P. Růžička, 22nd Inter. Symp. *Mathematical Foundations of Computer Science* (Bratislava, August '97), Lecture Notes in Computer Science 1295, Springer-Verlag, (1997), pp.85-90.

[58] J. Krajíček, On the degree of ideal membership proofs from uniform families of polynomials over a finite field, *Illinois J. of Mathematics*, **45(1)**, (2001), pp.41-73.

[59] J. Krajíček, Extensions of models of PV, in: Logic Colloquium'95, Eds. J.A.Makowsky and E.V.Ravve, ASL/Springer Series *Lecture Notes in Logic*, Vol. **11**, (1998), pp.104-114.

[60] J. Krajíček, On the weak pigeonhole principle, *Fundamenta Mathematicae*, Vol.**170(1-3)**, (2001), pp.123-140.

[61] J. Krajíček, Tautologies from pseudo-random generators, *Bulletin of Symbolic Logic*, **7(2)**, (2001), pp.197-212.

[62] J. Krajíček, Dual weak pigeonhole principle, pseudo-surjective functions, and provability of circuit lower bounds", *J. of Symbolic Logic*, **69(1)**, (2004), pp.265-286.

[63] J. Krajíček, Diagonalization in proof complexity, *Fundamenta Mathematicae*, **182**, (2004), pp.181-192.

[64] J. Krajíček, Implicit proofs, *J. of Symbolic Logic*, **69(2)**, (2004), pp.387-397.

[65] J. Krajíček, Structured pigeonhole principle, search problems and hard tautologies, *J. of Symbolic Logic*, **70(2)**, (2005), pp.619-630.

[66] J. Krajíček, Hardness assumptions in the foundations of theoretical computer science, *Archive for Mathematical Logic*, **44(6)**, (2005), pp.667-675.

[67] J. Krajíček, Proof complexity, in: Laptev, A. (ed.), European congress of mathematics (ECM), Stockholm, Sweden, June 27–July 2, 2004. Zurich: European Mathematical Society, (2005), pp.221-231.

[68] J. Krajíček, Substitutions into propositional tautologies, *Information Processing Letters*, **101(4)**, (2007), pp.163-167.

[69] J. Krajíček, A proof complexity generator, in: *Proc. from the 13th Int. Congress of Logic, Methodology and Philosophy of Science (Beijing, August 2007)*, King's College Publications, London, ser. Studies in Logic and the Foundations of Mathematics. Eds. C.Glymour, W.Wang, and D.Westerstahl. To appear (preprint Dec.'07).

[70] J. Krajíček, A form of feasible interpolation for constant depth Frege systems, *J. of Symbolic Logic*, to appear (preprint March'09).

[71] J. Krajíček, On the proof complexity of the Nisan-Wigderson generator based on a hard NP $\cap$ *co*NP function, in preparation.

[72] J. Krajíček and P. Pudlák, Propositional proof systems, the consistency of first order theories and the complexity of computations, *J. Symbolic Logic*, **54(3)**, (1989), pp.1063-1079.

[73] J. Krajíček and P. Pudlák, Quantified Propositional Calculi and Fragments of Bounded Arithmetic, *Zeitschr. f. Mathematikal Logik u. Grundlagen d. Mathematik*, Bd. **36(1)**, (1990), pp. 29-46.

[74] J. Krajíček and P. Pudlák, Propositional provability in models of weak arithmetic, in: *Computer Science Logic (Kaiserlautern, Oct. '89)*, eds. E. Boerger, H. Kleine-Bunning and M.M. Richter, Lecture Notes in Computer Science **440**, (1990), pp. 193-210. Springer-Verlag.

[75] J. Krajíček and P. Pudlák, Some consequences of cryptographical conjectures for S^1_2 and EF", *Information and Computation*, Vol. **140 (1)**, (January 10, 1998), pp.82-94.

[76] J. Krajíček, P. Pudlák, and J. Sgall, Interactive Computations of Optimal Solutions, in: B. Rovan (ed.): *Mathematical Foundations of Computer Science* (B. Bystrica, August '90), Lecture Notes in Computer Science **452**, Springer-Verlag, (1990), pp. 48-60.

[77] J. Krajíček, P. Pudlák and G. Takeuti, Bounded arithmetic and the polynomial hierarchy, *Annals of Pure and Applied Logic*, **52**, (1991), pp.143–153.

[78] J. Krajíček, P. Pudlák,and A. Woods, An Exponential Lower Bound to the Size of Bounded Depth Frege Proofs of the Pigeonhole principle", *Random Structures and Algorithms*, **7(1)**, (1995), pp.15-39.

[79] P. A. Loeb, Conversion from nonstandard to standard measure spaces and applications in probability theory, *Trans. Am. Math. Soc.*, Vol.**211**, (1975), pp.113-122.

[80] A. Maciel and T. Pitassi, Towards lower bounds for bounded-depth Frege proofs with modular connectives, in: *Proof Complexity and Feasible Arithmetics*, P. Beame and S. Buss, eds., DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. **39**, pp. 195-227, American Mathematical Society, (1998).

[81] R. Mansfield, The theory of boolean ultrapowers, *Annals of Mathematical Logic*, **2(3)**, (1971), pp.297-323.

[82] D. Marker, *Model Theory, An Introduction*, Grad.Texts in Mathematics **217**, (2002), Springer.

[83] N. Nisan and A. Wigderson, Hardness vs. randomness, *J. Comput. System Sci.*, Vol.**49**, (1994), pp.149–167.

[84] R. Parikh, Existence and feasibility in arithmetic, *Journal of Symbolic Logic*, **36**, (1971), pp.494-508.

[85] J. Paris and A. J. Wilkie, Counting problems in bounded arithmetic, in: *Methods in Mathematical Logic*, LNM 1130, (1985), pp. 317-340. Springer-Verlag.

[86] J. Pich, Nisan-Wigderson generators in proof systems with forms of interpolation, preprint (2009).

[87] T. Pitassi, P. Beame, and R. Impagliazzo, Exponential lower bounds for the pigeonhole principle, *Computational complexity*, **3**, (1993), pp.97-308.

[88] P. Pudlák, The lengths of proofs, in: Handbook of Proof Theory, S.R. Buss ed., Elsevier, (1998), pp.547-637.

[89] H. Rasiowa and R. Sikorski, Algebraic treatment of the notion of satisfiability, *Fundamenta Mathematicae*, **40**, (1953), pp.62-65.

[90] H. Rasiowa and R. Sikorski, *The mathematics of metamathematics*, Panstwowe Wydaw. Naukowe, Warsaw, (1963).

[91] A. A. Razborov, Lower bounds on the size of bounded depth networks over a complete basis with logical addition, *Matem. Zametki*, **41(4)**, (1987), 598-607.

[92] A. A. Razborov, Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic, *Izvestiya of the R.A.N.*, **59(1)**, (1995), pp.201-224.

[93] A. A. Razborov, Lower Bounds for the Polynomial Calculus, *Computational Complexity*, **7(4)**, (1998), pp.291-324.

[94] A. A. Razborov, Resolution lower bounds for perfect matching principles, in: *Proc. of the 17th IEEE Conf. on Computational Complexity*, (2002), pp.29-38.

[95] A. A. Razborov, Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution, preprint, (May'03).

[96] A. A. Razborov and S. Rudich, Natural proofs, *J.Comp. Syst. Sci.*, **55(1)**, (1997), pp.24-35.

[97] S. Riis, *Independence in bounded arithmetic*, DPhil. Thesis, (1993), Oxford University.

[98] S. Rudich, Super-bits, demi-bits, and $\tilde{NP}/qpoly$-natural proofs, in: *Proc. of the 1st Int.Symp. on Randomization and Approximation Techniques in Computer Science*, LN in Comp.Sci., Springer-Verlag, Vol.**1269**, (1997), pp.85-93.

[99] D. Scott, A proof of the independence of the continuum hypothesis, *Mathematical Systems Theory*, **1**, (1967), pp.89-111.

[100] R. Smolensky, Algebraic methods in the theory of lower bounds for Boolean circuit complexity, in: *Proc. 19th Ann. ACM Symp. on Th. of Computing*, (1987), pp. 77-82.

[101] G. Takeuti, *Two applications of logic to mathematics*, (1978). Princeton Univ. Press.

[102] G. Takeuti and M. Yasumoto, Forcing in bounded arithmetic, in: Gödel'96, Ed. P. Hájek, LN in Logic **6**, (1996), pp.120-138.

[103] G. Takeuti and M. Yasumoto, Forcing in bounded arithmetic II., *J.Symbolic Logic*, **63**, (1998), pp.860-868.

[104] G. Takeuti and W. Zaring, *Axiomatic set theory*, Grad. Texts in Math. **8**, (1973). Springer.

[105] Em. Viola, Correlation bounds for polynomials over $\{0,1\}$, *ACM SIGACT news*, **40(1)**, 2009.

[106] K. W. Wagner, Bounded query classes, *SIAM J. Computing*, **19(5)**, (1990), pp.833-846.

[107] A. Yao, Separating the polynomial-time hierarchy by oracles, in: *Proc. 26th Ann. IEEE Symp. on Found. of Comp. Sci.*, (1985), pp. 1-10.

[108] D. Zambella, Notes on polynomially bounded arithmetic, *Journal of Symbolic Logic*, **61(3)**, (1996), pp.942-966.