Charles University in Prague
Faculty of Mathematics and Physics

Weak Pigeonhole Principle, and Randomized Computation

Emil Jeřábek

Ph.D. Thesis
Prague, 2005

Abstract

We study the extension of the theory S^1_2 by instances of the dual (onto) weak pigeonhole principle for p-time functions, $dWPHP(PV)^x_{x^2}$. We propose a natural framework for formalization of randomized algorithms in bounded arithmetic, and use it to provide a strengthening of Wilkie's witnessing theorem for $S^1_2 + dWPHP(PV)$.

Then we show that $dWPHP(PV)$ is (over S^1_2) equivalent to a statement asserting the existence of a family of Boolean functions with exponential circuit complexity. Building on this result, we formalize the Nisan-Wigderson construction (conditional derandomization of probabilistic p-time algorithms) in a conservative extension of $S^1_2 + dWPHP(PV)$. We also develop in S^1_2 the algebraic machinery needed for implicit list-decoding of Reed-Muller error-correcting codes (including some results on a modification of Soltys' theory $\forall LAP$), and use it to formalize the Impagliazzo-Wigderson strengthening of the Nisan-Wigderson theorem.

We construct a propositional proof system WF (based on a reformulation of Extended Frege in terms of Boolean circuits), which captures the $\forall\Pi^b_1$-consequences of $S^1_2 + dWPHP(PV)$. As an application, we show that WF and G_2 p-simulate the Unstructured Extended Nullstellensatz proof system.

We also consider two theories which have explicit counting facilities in their language. The first one is the Impagliazzo-Kapron logic; we propose a modification of the theory, and prove a generalization of the Impagliazzo-Kapron soundness theorem to $\forall\exists$-consequences of the theory. The second one is a feasible theory of approximate counting, formulated in a variant of Kleene's 3-valued logic. We introduce the theory, and prove a witnessing theorem for its existential consequences.

Acknowledgements

I would like to thank my supervisor Jan Krajíček for fruitful inspiration, careful guidance, neverending enthusiasm, patience, and support; Pavel Pudlák and Jiří Sgall for clarifying discussions and enlightment; Vítězslav Švejdar, Josef Mlček, and Antonín Sochor, for introducing me into the beautiful land of logic; numerous friends and colleagues, among others Libor Běhounek, Marta Bílková, Stefano Cavagnetto, Petr Cintula, Jiří Hanika, Rostislav Horčík, Antonina Kolokolova, Kerry Ojakian, Petr Pajas, Michal Peliš, and Michal Rössler, for mathematical discussions, and nonmathematical diversion and entertainment; my parents, for encouragement and support during my studies. I acknowledge financial support under grant project LN00A056 of MŠMT ČR, and grant EAA1019401 of GA AV ČR.

Contents

Chapter 1

Introduction

Variants of the pigeonhole principle are ubiquitous in complexity theory and related areas of mathematical logic. On one hand, *PHP* serves as a canonical example of a tautology for proving lower bounds on lengths of propositional proofs (e.g., Haken's lower bound for resolution [13], or Ajtai's lower bound for constant-depth Frege [1]); on the other hand, weak pigeonhole principle (provable in $I\Delta_0 + \Omega_1$) can be used to formalize certain counting arguments in bounded arithmetic (e.g., the existence of infinitely many primes [35], or Ramsey-type combinatorics [32]).

Our main object of study will the theory $S^1_2 + dWPHP(PV)$: the dual weak pigeonhole principle for polynomial-time functions. Whereas the usual weak pigeonhole principle says that there is no injection from $2n$ to n, the dual weak pigeonhole principle states that there is no surjection from n onto $2n$; in bounded arithmetic, the dual version is the weaker one of the two notions. The first important result on $S^1_2 + dWPHP(PV)$ was A. Wilkie's witnessing theorem (see [20]): $\forall\Sigma^b_1$-consequences of the theory are witnessable by randomized polynomial-time computable (multi)functions. The theory was later studied by J. Krajíček [21], and its model theory was investigated by N. Thapen [47, 48].

Probabilistic algorithms have attracted a lot of attention recently, concentrated on derandomization efforts. Results in this direction, also known as hardness-randomness tradeoffs, show that there are efficient deterministic simulations of randomized algorithms, if there exist uniform families of Boolean functions with large circuit complexity (see e.g. [31, 15, 46]); there are also some results showing the converse ([18]). This is one of our motivations for studying the dual weak pigeonhole principle, as Wilkie's witnessing theorem suggests that $dWPHP(PV)$ is connected with probabilistic computation.

After presenting some background information in chapter 2, we turn our attention in chapter 3 to the question of definability of randomized algorithms in $S^1_2 + dWPHP(PV)$. Typical witnessing theorems, such as the Buss' theorem, have the form of an equivalence: e.g., a function is Σ^b_i-definable in S^i_2 *iff* it is in $FP^{\Sigma^P_{i-1}}$. Wilkie's witnessing theorem only shows one direction of such an equivalence; we would like to complement it by the converse direction.

Notice that we cannot hope to Σ^b_1-define *all* randomized p-time algorithms in $S^1_2 + dWPHP(PV)$: by [48], this would imply that ZPP has a complete language, which is known to fail for a suitable relativized class ZPP^A. We thus isolate a special subclass of randomized algorithms; the idea is, instead of merely demanding the success probability of the algorithm to be large, we require to have a polynomial-time function which proves that the probability of failure is small by mapping the set of all random witnesses onto several copies of the set of bad witnesses. Such a condition is easily expressible in the language of arithmetic. We show that this approach is natural and well-behaved: we formalize in S^1_2 or $S^1_2 + dWPHP(PV)$ usual properties of randomized algorithms, like success amplification, simulation by circuits, or closure under composition. As a concrete example, we also present and analyze a formalized version of the Rabin-Miller primality testing algorithm. We prove that the witnessing functions in Wilkie's theorem are definable in $S^1_2 + dWPHP(PV)$, which shows that we have picked the right class of probabilistic algorithms.

In chapter 4 we will consider Boolean functions with exponential (worst-case or average-case) circuit complexity within bounded arithmetic. Exponentially hard functions are closely connected with probabilistic algorithms, by the derandomization results mentioned above. We show that they are also connected to the dual weak pigeonhole principle. It is not hard to see that $S^1_2 + dWPHP(PV)$ can formalize Shannon's counting argument, which implies that functions with exponential circuit complexity exist [40]; we will prove a strong converse to this statement: $S^1_2 + \neg dWPHP(PV)$ implies that the circuit complexity of all functions is bounded by a (nonstandard) constant. In particular, we obtain the following characterization: $dWPHP(PV)$ is (over S^1_2) equivalent to the existence of exponentially hard Boolean functions in an arbitrary large number of variables.

Since we have probabilistic algorithms and exponentially hard functions in $S^1_2 + dWPHP(PV)$, the natural question is whether we can carry out some derandomization in this theory (or its variant). As we will see, the answer is positive. We will work in a conservative extension of $S^1_2 + dWPHP(PV)$ with an additional symbol α, which stands for a family of hard Boolean functions. We formalize the construction of the Nisan-Wigderson pseudorandom generator; as a result we get that any definable probabilistic algorithm (in the sense of chapter 3) can be simulated by a p-time function with the oracle α, if the Boolean functions given by α have exponential *average-case* circuit complexity. (We will need to formalize variants of Chernoff's inequality and Stirling's bound in the process; we moved these technical details into the appendix.)

As proved by Impagliazzo and Wigderson [15], the assumption of average-case hardness in the Nisan-Wigderson theorem can be weakened to worst-case hardness. This result can be formalized in our setting as well, and we show it in section 4.3. We follow the approach of [46], which reduces hardness amplification to list-decoding of error-correcting codes. This strategy is much simpler than the original sequence of papers culminating in [15], but still its formalization in S^1_2 presented some nontrivial difficulties. We had to formalize a variant of a factorization algorithm for bivariate polynomials over finite fields, including Gaussian elimination for function fields (we used a modification of M. Soltys' theory $\forall LAP$ [42, 44] to deal with linear algebra issues).

In chapter 5, we study the strength of $S^1_2 + dWPHP(PV)$ in terms of propositional proof complexity. By a well-known construction [9, 22], arithmetical $\forall\Pi^b_1$-sentences can be translated to sequences of propositional formulas. (Almost) every theory T has an associated proof system P: this roughly means that T proves the consistency of P, and propositional translations of Π^b_1-formulas provable in T have polynomial-size proofs in P. In a sense, P is a non-uniform version of the $\forall\Pi^b_1$-fragment of T. We construct a propositional proof system, called *WF*, which corresponds to the theory $S^1_2 + dWPHP(PV)$, thus answering an open problem of [21]. To simplify the presentation, we first construct a proof system operating with Boolean circuits, p-equivalent to the Extended Frege proof system; this makes explicit the folklore idea that Extended Frege is essentially "*P/poly*-Frege". We also show that *WF* p-simulates the Unstructured Extended Nullstellensatz proof system from [7].

The last chapter 6 deals with two theories of arithmetic, which are only loosely connected by having explicit counting in their language. In section 6.1 we will consider the Impagliazzo-Kapron logic [14]. This theory was created as a framework for formalization of cryptographic reasoning; it was intended as a tool for incorporating other more specialized deductive systems, rather than a final product. Impagliazzo and Kapron prove a soundness theorem, which gives cryptographic interpretation to ($\forall \to \forall$)-consequences of their theory (more precisely: implications between Π^1_1-formulas, whose first-order kernel is Π^0_1). We propose a modification of the theory, intended to overcome some inconsistencies in application of the theory in the original paper [14]. Then we generalize the Impagliazzo-Kapron soundness theorem to $\forall\exists$-consequences of the theory.

In section 6.2 we will introduce a theory of approximate counting. Exact counting is computationally difficult: by results of Toda [50], $\#P$ is stronger than the polynomial-time hierarchy. In contrast to this, approximate counting can be done within PH, it can be efficiently realized by randomized algorithms; vice versa, randomized algorithms are easily definable using approximate counting. As we have already seen, it is possible to simulate some special kinds of approximate counting using $dWPHP$ (or other variants of the pigeonhole principle), but this approach has its drawbacks: it is not guaranteed to work in general, it employs a lot of ad hoc arguments, and it increases the quantifier complexity (which is a problem if we have restricted induction, such as in S^1_2). We want something more systematic, and more explicit; we thus introduce approximate counting quantifiers.

Unlike exact counting quantifiers, it is far from clear what should be the “right” axioms governing approximate counting. Worse yet, it is a serious problem how to model the inexactness of the approximate counting quantifiers, or in other words, how to prevent them from being exact. We attempt to solve this difficulty by working in an extension of Kleene’s 3-valued logic instead of the classical predicate calculus. It is quite unusual to use non-classical logics in bounded arithmetic (apart from the intuitionistic logic, which however has completely different motivation), but we believe it is the natural thing to do in this situation. One way to think about it is to consider the computational aspect of the problem: probabilistic complexity classes like BPP or AM are more naturally presented as classes of promise problems, rather than classes of languages. In a sense, it separates the syntactical description (e.g., a randomized Turing machine) from extra universal conditions imposed on its behaviour (e.g., for every input, the probability of acceptance is either very high or very low).

The contents of chapters 3, 5, and a part of chapter 4 of this thesis were already published as [16].

Chapter 2

Preliminaries

2.1 Computational complexity

We assume the reader is familiar with elementary notions from complexity theory. We review here some basic concepts mainly to fix the notation; missing details can be found in any textbook on computational complexity, such as [33].

Class P consists of predicates computable on a *deterministic* Turing machine in polynomial time; the corresponding class of polynomial-time computable functions is denoted FP. NP is the class of predicates computable in polynomial time on a *nondeterministic* Turing machine. Equivalently, a language L is in NP if there is a relation $P(x, y)$ computable in time polynomial in $|x|$ such that

$$x \in L \quad \text{iff} \quad \exists y\, P(x, y).$$

For any language class C, we define coC to be the class of complements of languages from C.

Relativized classes P^A and NP^A consist of predicates computable on a deterministic resp. nondeterministic polynomial-time Turing machine with oracle access to a language A. The *polynomial-time hierarchy* is defined by $\Sigma_0^P = P$, $\Sigma_{i+1}^P = NP^{\Sigma_i^P} = \bigcup\{NP^A;\, A \in \Sigma_i^P\}$, $\Pi_i^P = co\Sigma_i^P$, and $PH = \bigcup_i \Sigma_i^P$.

We will also often refer to *randomized complexity classes*. A language L is in the class BPP, if there exists a polynomial-time computable relation $R(x, y)$, and a polynomial $p(n)$ such that

$$x \in L \Rightarrow \Pr_y\bigl(R(x, y) \bigm| |y| \le p(|x|)\bigr) \ge 2/3,$$
$$x \notin L \Rightarrow \Pr_y\bigl(R(x, y) \bigm| |y| \le p(|x|)\bigr) \le 1/3.$$

If the second condition holds in the stronger form

$$x \notin L \Rightarrow \Pr_{|y| \leq p(|x|)}(R(x,y)) = 0,$$

the language L is in the class *RP*. We define *ZPP* $=$ *RP* $\cap$ *coRP*.

Another, maybe more intuitive, way to define randomized classes is to consider *randomized Turing machines.* A randomized TM is essentially a nondeterministic TM; we define a probability measure on the set of possible computation paths such that individual nondeterministic choices are independent and uniform (i.e., each of the two possible branchings is taken with probability 1/2).

In this setting, $L \in$ *BPP* iff there is a randomized TM $M(x)$ which runs in time polynomial in $|x|$, such that

$$x \in L \Rightarrow \Pr(M(x) \text{ accepts}) \geq 2/3,$$
$$x \notin L \Rightarrow \Pr(M(x) \text{ accepts}) \leq 1/3.$$

As above, $L \in$ *RP* if in addition $M(x)$ always rejects for $x \notin L$. Apart from the definition *ZPP* $=$ *RP* $\cap$ *coRP*, there are other characterizations of *ZPP*. First, $L \in$ *ZPP* iff there is a randomized TM which *always* computes the correct answer to $x \in L$?, and whose *expected* running time is polynomial in $|x|$. Second, $L \in$ *ZPP* iff there is a (worst-case) polynomial-time randomized TM $M(x)$ which either rejects, or correctly outputs 0 or 1 according to the characteristic function of L, and $\Pr(M(x) \text{ rejects}) \leq 1/2$.

The constants 2/3, 1/3, etc., in the definitions are somewhat arbitrary. Assume we have an *RP*-machine M which accepts $x \in L$ with probability $\alpha(n)$, where $n = |x|$. We can construct another machine M' which repeats the computation of M k-times, and accepts if at least one of the iterations accepts. The running time of M' is still polynomial if $k = n^{O(1)}$, and the probability of accepting for $x \in L$ is amplified to $1 - (1 - \alpha(n))^k$. It follows that the definition of *RP* does not change, if we replace the constant 2/3 by any function $\alpha(n)$ such that $n^{-c} \leq \alpha(n) \leq 1 - 2^{-n^c}$ for some constant c.

A similar construction works for *BPP* as well; in this case, the machine M' will take a majority vote. The probability of error of M' will be again exponentially small in k, due to the following theorem.

2.1.1 Theorem (Chernoff's inequality) *Let $X_1, \dots, X_k$ be independent random variables with values from $[0,1]$. Put $X = \frac{1}{k}\sum_i X_i$. Then*

$$\Pr(X - \operatorname{E} X \geq a) \leq e^{-2ka^2}$$

for any $a \geq 0$. □

Obviously, $RP \subseteq NP$. M. Sipser [41] has shown how to approximate probabilities within PH. As a result, $BPP \subseteq \Sigma_2^P \cap \Pi_2^P$, and in fact, $BPP \subseteq ZPP^{NP}$ [31].

We will need few notions from circuit complexity.

2.1.2 Definition A *Boolean circuit* C with n inputs and m outputs over a set B of Boolean connectives is a labelled acyclic graph C, such that each node is labelled either by an input variable $x_0, \dots, x_{n-1}$, in which case it has in-degree 0, or by a k-ary connective from B, in which case it has in-degree k. Additionally, for each $j < m$ there is exactly one node of C labelled by an output variable y_j.

The circuit computes a function $C \colon 2^n \to 2^m$ in an obvious way (where we identify $\{0,1\}^n = 2^n$).

Any language L can be identified with a family of Boolean functions $L_n \colon 2^n \to 2$, $n \in \omega$, where L_n is the restriction of the characteristic function of L to 2^n.

2.1.3 Definition *Circuit complexity* $C(f)$ of a function $f \colon 2^n \to 2^m$ is the minimal size of a circuit which computes f.

A language L is in $P/poly$, if it is computable by a family of polynomial-size circuits, i.e., there is a constant c such that $C(L_n) \leq n^c$ for all sufficiently large n.

A computation of a deterministic Turing machine can be simulated by a circuit of depth proportional to the time of the computation, and width proportional to the space. In particular, $P \subseteq P/poly$. Moreover, a similar simulation is known to hold for randomized Turing machines, thus $BPP \subseteq P/poly$.

We will sometimes write p-time, p-size, etc., instead of polynomial-time, polynomial-size.

2.2 Bounded arithmetic

The research of bounded arithmetic was initiated by R. Parikh [34], who defined the theory now known as $I\Delta_0$. An extension $I\Delta_0 + \Omega_1$ was later studied by J. Paris and A. Wilkie; a milestone in this line of research was S. Buss' book [5], which introduced the theory S_2 (and its fragments) as a conservative extension of $I\Delta_0 + \Omega_1$. We will briefly recapitulate Buss' theories in the sequel; more details can be found in [5, 20, 12].

2.2.1 Definition The first-order language L consists of a constant 0, unary functions S, $\lfloor \frac{\cdot}{2} \rfloor$, $|\cdot|$, binary functions $+$, $\cdot$, $\#$, and binary predicate $\leq$. The intended meaning of $|x|$ is the number of digits in the binary representation of x, i.e., $|x| = \lceil \log_2(x+1) \rceil$. The meaning of the *smash function* is $x \# y = 2^{|x||y|}$.

We let $x < y$ abbreviate $x \leq y \wedge x \neq y$, and for every natural number n, we define its unary numeral as $\overline{n} = \underbrace{S(\cdots(S}_{n}(0))\cdots)$.

2.2.2 Definition *Bounded quantifiers* are the abbreviations

$$\exists x \leq t\, \varphi = \exists x\, (x \leq t \wedge \varphi),$$
$$\forall x \leq t\, \varphi = \forall x\, (x \leq t \rightarrow \varphi),$$

where t is a term with no occurrence of x. A bounded quantifier of the form $\exists x \leq |t|$ or $\forall x \leq |t|$ is called *sharply bounded.*

A formula is *bounded* if it contains only bounded quantifiers; the set of all bounded formulas in the language L will be denoted by Σ^b_∞ (the symbol Δ^0_0 is usually reserved for bounded formulas in the language L_{PA} of Peano arithmetic). A formula consisting of i alternating (possibly empty) blocks of bounded quantifiers in front of a sharply bounded formula is called a *strict Σ^b_i-formula* if the first block is existential, or *strict Π^b_i-formula* if the first block is universal.

The classes of Σ^b_i and Π^b_i-formulas are defined similarly, except that we allow arbitrary sharply bounded quantifiers to intervene in the quantifier prefix. Up to logical equivalence, Σ^b_i and Π^b_i-formulas are closed under $\wedge$ and $\vee$; this is often incorporated directly in the definition.

A Σ^b_i-formula is called Δ^b_i in a theory T (shortly $\Delta^b_i(T)$), if it is in T equivalent to a Π^b_i-formula.

The hierarchy of bounded formulas is tightly connected with the polynomial-time hierarchy.

2.2.3 Theorem *For any $i > 0$, the class of predicates Σ^b_i-definable in the standard model of arithmetic $\mathbb{N}$ coincides with Σ^P_i.* □

2.2.4 Definition The theory *BASIC* in the language L has the following 32 open axioms.

$$x \leq y \rightarrow x \leq S(y),$$
$$x \neq S(x),$$
$$0 \leq x,$$

$$
\begin{gathered}
x < y \to S(x) \leq y, \\
\overline{2}x = 0 \to x = 0, \\
x \leq y \lor y \leq x, \\
x \leq y \land y \leq x \to x = y, \\
x \leq y \land y \leq z \to x \leq z, \\
|0| = 0, \\
x \neq 0 \to |\overline{2}x| = S(|x|) \land |S(\overline{2}x)| = S(|x|), \\
|\overline{1}| = \overline{1}, \\
x \leq y \to |x| \leq |y|, \\
|x \# y| = S(|x| \cdot |y|), \\
0 \# x = \overline{1}, \\
x \neq 0 \to \overline{1} \# (\overline{2}x) = \overline{2}(\overline{1} \# x) \land 1 \# S(\overline{2}x) = \overline{2}(\overline{1} \# x), \\
x \# y = y \# x, \\
|x| = |y| \to x \# z = y \# z, \\
|x| = |y| + |z| \to x \# w = (y \# w) \cdot (z \# w), \\
x \leq x + y, \\
x < y \to S(\overline{2}x) < \overline{2}y, \\
x + y = y + x, \\
x + 0 = x, \\
x + S(y) = S(x + y), \\
(x + y) + z = x + (y + z), \\
x + z \leq y + z \to x \leq y, \\
x \cdot 0 = 0, \\
x \cdot S(y) = x \cdot y + x, \\
x \cdot y = y \cdot x, \\
x \cdot (y + z) = x \cdot y + x \cdot z, \\
x \neq 0 \to (x \cdot y \leq x \cdot z \equiv y \leq z), \\
x \neq 0 \to |x| = S(|\lfloor \tfrac{x}{2} \rfloor|), \\
x = \lfloor \tfrac{y}{2} \rfloor \equiv (y = \overline{2}x \lor y = S(\overline{2}x)).
\end{gathered}
$$

2.2.5 Definition Let Γ be a set of formulas. We define the following axiom schemata, where formulas φ are taken from Γ.

- *Induction* Γ-*IND*:

$$\varphi(0) \land \forall u \leq x \, (\varphi(u) \to \varphi(S(u))) \to \varphi(x).$$

- *Polynomial induction* Γ-*PIND*:

$$\varphi(0) \wedge \forall u \leq x\,(\varphi(\lfloor \tfrac{u}{2} \rfloor) \rightarrow \varphi(u)) \rightarrow \varphi(x).$$

- *Length induction* Γ-*LIND*:

$$\varphi(0) \wedge \forall u \leq |x|\,(\varphi(u) \rightarrow \varphi(S(u))) \rightarrow \varphi(|x|).$$

- *Minimization* Γ-*MIN*:

$$\varphi(x) \rightarrow \exists y \leq x\,(\varphi(y) \wedge \forall u < y\,\neg\varphi(u)).$$

- *Maximization* Γ-*MAX*:

$$\varphi(0) \rightarrow \exists y \leq x\,(\varphi(y) \wedge \forall u \leq x\,(u > y \rightarrow \neg\varphi(u))).$$

- *Length minimization* Γ-*LENGTH-MIN*:

$$\varphi(x) \rightarrow \exists y \leq x\,(\varphi(y) \wedge \forall u \leq x\,(|u| < |y| \rightarrow \neg\varphi(u)).$$

- *Length maximization* Γ-*LENGTH-MAX*:

$$\varphi(0) \rightarrow \exists y \leq x\,(\varphi(y) \wedge \forall u \leq x\,(|u| > |y| \rightarrow \neg\varphi(u))).$$

For any $i \geq 1$, we define a chain of theories $S^i_2 = \mathit{BASIC} + \Sigma^b_i$-*PIND*, and $T^i_2 = \mathit{BASIC} + \Sigma^b_i$-*IND*. The unions of these theories are $S_2 = \mathit{BASIC} + \Sigma^b_\infty$-*PIND*, and $T_2 = \mathit{BASIC} + \Sigma^b_\infty$-*IND*.

Basic relations between these schemata are summarized below.

2.2.6 Theorem (Buss [5, 6]) *Let $i \geq 1$.*

(i) $S^i_2 \subseteq T^i_2 \subseteq S^{i+1}_2$, *thus* $S_2 = T_2$,

(ii) S^{i+1}_2 *is* $\forall\Sigma^b_{i+1}$*-conservative over* T^i_2,

(iii) S^i_2 *is over BASIC equivalent to* Σ^b_i-*LIND*, Π^b_i-*PIND*, Π^b_i-*LIND*, Σ^b_i-*LENGTH-MIN*, Σ^b_i-*LENGTH-MAX*, Π^b_{i-1}-*LENGTH-MIN*, Π^b_{i-1}-*LENGTH-MAX*,

(iv) T^i_2 *is over BASIC equivalent to* Π^b_i-*IND*, Σ^b_i-*MIN*, Σ^b_i-*MAX*, Π^b_{i-1}-*MIN*, Π^b_{i-1}-*MAX*,

except that for $i = 1$, one needs to use $\Delta^b_1(S^1_2)$ instead of Π^b_0 in items (iii) and (iv). □

Theories S^i_2 and T^i_2 can be *relativized.* $L(\alpha)$ denotes the language L expanded by a new function or relation symbol α. We define $\Sigma^b_i(\alpha)$ and $\Pi^b_i(\alpha)$ as before, but allowing α to appear in atomic subformulas. The theories $S^i_2(\alpha)$ and $T^i_2(\alpha)$ have the respective induction schema expanded to the new language, but do not contain any other axioms involving α.

S^1_2 is a sequential theory: we can encode finite sequences of numbers in such a way that basic operations on sequences (like concatenation) are Δ^b_1-definable and well-behaved in S^1_2. We will denote by $(w)_i$ the Δ^b_1-definable function which extracts the ith element from a sequence w, and $lh(w)$ will denote the length of a sequence w. Having sequence coding, we can introduce another useful schema.

2.2.7 Definition *Sharply bounded collection* $BB\Gamma$ is the schema

$$\forall i \le |x|\, \exists v \le y\, \varphi(i,v) \to \exists w\, \forall i \le |x|\, \varphi(i,(w)_i),$$

for every formula $\varphi \in \Gamma$.

2.2.8 Theorem *Let $i \ge 1$.*

- (Buss [5]) S^i_2 *proves* $BB\Sigma^b_i$.
- (Ressayre [39]) $S^i_2 + BB\Sigma^b_{i+1}$ *is* $\forall\Sigma^b_{i+1}$*-conservative over* S^i_2.

□

Rather than using the basic Buss' language of bounded arithmetic, it will be more convenient for us to work in a richer language of theory PV, introduced by Cook [9]. The language of PV contains function symbols for all polynomial-time algorithms; it is based on the following characterization of FP.

2.2.9 Definition Let $S_0(x) = 2x+1$ and $S_1(x) = 2x+2$. A function f is defined from functions g, h_0, h_1, and b by *limited* (or *bounded*) *recursion on* (*binary*) *notation*, if

$$\begin{aligned} f(\vec{x},0) &= g(\vec{x}), \\ f(\vec{x},S_i(y)) &= h_i(\vec{x},y,f(\vec{x},y)), \qquad i=0,1, \\ f(\vec{x},y) &\le b(\vec{x},y). \end{aligned}$$

2.2.10 Theorem (Cobham [8]) *The closure of constant* 0*, functions* S_0*,* S_1*,* $\#$*, and projections under composition and limited recursion on notation is exactly the class of polynomial-time computable functions.* □

Cook's equational theory PV, and Cobham's result were originally formulated for functions on binary strings, but we present it here for number functions to be consistent with other theories of bounded arithmetic. We identify binary strings and natural numbers by the following bijection: given a number x, the corresponding string is the binary representation of $x+1$ without the leading digit 1.

2.2.11 Definition The language of PV, and PV-derivations are introduced by simultaneous induction. Basic function symbols are constant 0, unary functions S_0, S_1, Tr, and binary functions $\overline{\#}$, $\frown$, $Less$. (The intended meaning of $\frown$ is concatenation, $Tr(x) = \lfloor \frac{x-1}{2} \rfloor$, i.e., string x with its rightmost bit deleted, $Less(x,y)$ is x with $|y|$ rightmost bits deleted, and $x \mathbin{\overline{\#}} y$ is $|y|$ concatenated copies of x, where $|y|$ is the length of y as a string.) The following defining equations are derivable:

$$\begin{aligned}
Tr(0) &= 0, \\
Tr(S_i(x)) &= x, \qquad i = 0,1, \\
x \frown 0 &= x, \\
x \frown S_i(y) &= S_i(x \frown y), \\
x \mathbin{\overline{\#}} 0 &= 0, \\
x \mathbin{\overline{\#}} S_i(y) &= x \frown (x \mathbin{\overline{\#}} y), \qquad i = 0,1, \\
Less(x,0) &= x, \\
Less(x, S_i(y)) &= Tr(Less(x,y)), \qquad i = 0,1.
\end{aligned}$$

PV-derivations are closed under usual rules of equational logic, and the rule

$$\frac{\begin{array}{cc} f_1(\vec{x},0) = g(\vec{x}) & f_2(\vec{x},0) = g(\vec{x}) \\ f_1(\vec{x},S_0(y)) = h_0(\vec{x},y,f_1(\vec{x},y)) & f_2(\vec{x},S_0(y)) = h_0(\vec{x},y,f_2(\vec{x},y)) \\ f_1(\vec{x},S_1(y)) = h_1(\vec{x},y,f_1(\vec{x},y)) & f_2(\vec{x},S_1(y)) = h_1(\vec{x},y,f_2(\vec{x},y)) \end{array}}{f_1(\vec{x},y) = f_2(\vec{x},y)}$$

where f_1, f_2, g, h_0, and h_1 are PV-functions.

For every term t, there is a function symbol f_t, and a defining equation

$$f_t(\vec{x}) = t(\vec{x}).$$

Let g, h_0, h_1, b_0, and b_1 be PV-functions, and π_i PV-derivations of equations

$$Less(h_i(\vec{x},y,z), z \frown b_i(\vec{x},y)) = 0,$$

for $i = 0, 1$. Then there is a PV-function symbol $f := f_{g,h_0,h_1,b_0,b_1,\pi_0,\pi_1}$, and defining equations

$$\begin{aligned} f(\vec{x}, 0) &= g(\vec{x}), \\ f(\vec{x}, S_i(y)) &= h_i(\vec{x}, y, f(\vec{x}, y)), \qquad i = 0, 1. \end{aligned}$$

Original Cook's PV, as we just described it, is an equational theory; [24] introduced a first-order theory PV_1, which is conservative extension of PV. The theory PV_1 has the same language as PV, its axioms are equations derivable in PV, and the *PIND* schema for open formulas. The theory can be axiomatized by purely universal formulas (and, in fact, that is the way it was defined in [24]). By a slight abuse of language, we will *use the symbol PV to denote* PV_1 in the sequel, and we will also use the same symbol for the language of PV.

PV can be relativized to $PV(\alpha)$, similarly to $S^i_2(\alpha)$. In this case, the new function symbol α is also allowed in the inductive clauses for introduction of new function symbols in definition 2.2.11; this means that the language of $PV(\alpha)$ contains symbols for all polynomial-time *oracle* algorithms.

The hierarchy of $\Sigma^b_i(PV)$-formulas in the language of PV is defined similarly to 2.2.2, and we let $S^i_2(PV)$ be the extension of PV by the $\Sigma^b_i(PV)$-*PIND* schema. Notice that symbols from language L, being polynomial-time computable, have natural counterparts in the language of PV; in particular, $S^1_2(PV)$ is an extension of S^1_2.

On the other hand, PV-functions have well-behaved Δ^b_1-definitions in S^1_2. Under this interpretation, every $\Sigma^b_i(PV)$-formula is equivalent to a Σ^b_i-formula, thus $S^1_2(PV)$ is a definable (hence conservative) extension of S^1_2. For these reasons, we will usually ignore the distinction between S^1_2 and $S^1_2(PV)$, and use PV-functions freely to simplify the presentation. If the reader is unfamiliar with PV, she may simply identify PV-functions with functions Δ^b_1-definable in S^1_2.

Now we introduce the central topic of this thesis.

2.2.12 Definition Let f be a function. *Dual weak pigeonhole principle* for f is the formula

$$\forall a > 1\; dPHP(f)^a_{a^2},$$

where $dPHP(f)^a_b$ stands for

$$\exists v < b\, \forall u < a\; f(u) \neq v.$$

The schema $dWPHP(PV)$ is the dual weak pigeonhole principle for all PV-functions f (with parameters).

Similarly to other variants of the weak pigeonhole principle, $dWPHP(PV)$ is provable in T^2_2 [35, 28].

Notice that $dWPHP(PV)$ is finitely axiomatizable over PV: it is equivalent to its instance $dWPHP(eval)$, where $eval(C, u)$ is the PV-function which evaluates a Boolean circuit C on input u. The exact bound $b = a^2$ in the definition of $dWPHP(PV)$ is inessential, since the following are equivalent over S^1_2 (this is essentially due to [35]):

(i) $\forall a \, \exists b \; dPHP(PV)^a_b$,

(ii) $dWPHP(PV)$,

(iii) $\forall a > 0 \, \forall c \; dPHP(PV)^{a|c|}_{a(|c|+1)}$.

In particular, we will often use the principle with $b = 2a$.

The theory $S^1_2 + dWPHP(PV)$ was studied in [21] under the name BT (for "Basic Theory"). We prefer to use the explicit longer name, because we will also consider the $dWPHP(PV)$ schema over other base theories (such as PV).

Various witnessing theorems are indispensable tools in bounded arithmetic, and we will need them as well. We quote here two particularly important results, *Parikh's theorem*, and *Buss' witnessing theorem.*

2.2.13 Theorem (Parikh [34]) *Let T be a $\forall\Sigma^b_\infty$-axiomatizable extension of S^1_2. If T proves $\forall x \, \exists y \, \varphi(x, y)$, where φ is bounded, then there exists a term t such that*

$$T \vdash \forall x \, \exists y \le t(x) \, \varphi(x, y). \qquad \square$$

2.2.14 Theorem (Buss [5, 6]) *Assume that $S^1_2 \vdash \forall x \, \exists y \, \varphi(x, y)$, with $\varphi \in \Sigma^b_1$. Then there exists a PV-function f such that $PV \vdash \forall x \, \varphi(x, f(x))$.* □

We will often use the following corollary to Buss' witnessing theorem: S^1_2 is $\forall\Sigma^b_1$-conservative over PV.

A witnessing theorem for $S^1_2 + dWPHP(PV)$ was found by A. Wilkie; the result was first published in Krajíček's book [20].

2.2.15 Theorem (Wilkie) *Assume that*

$$S^1_2 + dWPHP(PV) \vdash \forall x \, \exists y \, \varphi(x, y),$$

where $\varphi \in \Sigma^b_1$. Then there exists a randomized p-time algorithm f such that

$$\Pr(\varphi(x, f(x))) \ge 2/3$$

holds for every natural number x. □

We also introduce some bits of notation. If M is a model of PV or S^1_2, $Log(M)$ denotes the cut $\{|a|^M;\ a \in M\}$. We will often use this notation outside the model-theoretical context, in which case $x \in Log$ is a shortcut for $\exists y\, x = |y|$. Similarly, $x \in LogLog$ means $\exists y\, x = ||y||$.

We denote the set of natural numbers by ω. We also borrow from set theory the convention $n = \{0, 1, \dots, n-1\}$, in particular a "function $f\colon a \to b$" is really $f\colon [0, a-1] \to [0, b-1]$. Ordered pairs and sequences will be denoted by angle brackets, such as $\langle x, y\rangle$. The symbol $f\colon a \twoheadrightarrow b$ means that f is a function from a *onto* b.

Many of our results are formalizations of known statements in fragments of bounded arithmetic, like PV or $S^1_2 + dWPHP(PV)$. To make the notation more compact, we indicate the theory by the symbol "$(T \vdash:)$", which can be read as "theory T proves:".

2.3 Propositional proof complexity

The fundamental notion of a general propositional proof system was defined by S. Cook and A. Reckhow in [10].

2.3.1 Definition Let $TAUT$ denote the set of all classical propositional tautologies (in de Morgan language, for concreteness). *Propositional proof system* is a polynomial-time computable function P such that $\mathrm{rng}(P) = TAUT$.

A proof system P *polynomially simulates* a proof system Q, in symbols $Q \leq_p P$, if there is a polynomial-time function f such that $Q = P \circ f$.

Usual textbook calculi fit into this definition as follows: the relation $R(\pi, \varphi)$ expressing that π is a proof of a formula φ is p-time decidable, we may thus construct a propositional proof system P obeying the Cook-Reckhow definition as

$$P(\langle \pi, \varphi\rangle) = \begin{cases} \varphi, & \text{if } R(\pi, \varphi), \\ \top, & \text{otherwise.} \end{cases}$$

Two important examples are the Frege and Extended Frege proof systems, defined in [10].

2.3.2 Definition A *Frege proof system* (F) in a finite basis of Boolean connectives $\mathcal{B}$ is given by a finite, sound, and implicationally complete set $\mathcal{R}$ of rules of the form

$$\frac{\psi_1(\vec{p}) \quad \cdots \quad \psi_k(\vec{p})}{\psi_0(\vec{p})},$$

where ψ_i are formulas over $\mathcal{B}$ in propositional variables $\vec{p}$. An *instance* of such a Frege rule is

$$\frac{\psi_1(\vec{\chi}) \ \cdots \ \psi_k(\vec{\chi})}{\psi_0(\vec{\chi})},$$

where $\vec{\chi}$ are $\mathcal{B}$-formulas. A *Frege proof* of a formula φ is a sequence of formulas $\varphi_0, \dots, \varphi_n$ such that $\varphi_n = \varphi$, and each φ_i is derived from some previous φ_j's by a Frege rule; i.e., there are $j_1, \dots, j_k < i$ such that

$$\frac{\varphi_{j_1} \ \cdots \ \varphi_{j_k}}{\varphi_i}$$

is an instance of a rule from $\mathcal{R}$.

2.3.3 Definition An *Extended* (or *Extension*) *Frege* proof system (EF) is defined as follows: an Extended Frege proof of a formula φ is a sequence of formulas $\varphi_0, \dots, \varphi_n$ such that $\varphi_n = \varphi$, and each φ_i is either derived from some $\varphi_{j_1}, \dots, \varphi_{j_k}$, $j_1, \dots, j_k < i$, by a Frege rule, or it is an *extension axiom*

$$q \equiv \psi,$$

where q is a propositional variable which has no occurrence in ψ, φ_j for $j < i$, or φ.

The set of propositional tautologies is definable by the Π^b_1-formula

$$Taut(\varphi) \equiv \forall x < 2^{|\varphi|}\, eval(\varphi, x) = 1.$$

If P is PV-function which defines a propositional proof system, the *consistency* and *reflection* principles for P are the $\forall\Pi^b_1$-sentences

$$Con(P) \equiv \forall \pi\, P(\pi) \neq \bot,$$
$$0\text{-}RFN(P) \equiv \forall \pi\, Taut(P(\pi)).$$

An important link between bounded arithmetic and propositional proof complexity is given by translation of bounded formulas into propositional logic [9, 22]. For any Π^b_1-formula φ, there is a canonically constructed sequence of propositional formulas $\{\|\varphi\|^n;\, n \in \omega\}$, such that $\forall x\, \varphi(x)$ is true in the standard model iff all $\|\varphi\|^n$ are tautologies.

Roughly, the construction goes as follows. First, we assign to every PV-function $f(x_1, \dots, x_k)$ and numbers $n_1, \dots, n_k$ a (polynomial-size) circuit $\{\!\{f\}\!\}^{\vec{n}}(\vec{p})$, which computes the restriction $f\colon 2^{n_1} \times \dots \times 2^{n_k} \to 2^{b(n_1, \dots, n_k)}$, where b is a bounding polynomial to f. The formula

$$\|f\|^{\vec{n}}(\vec{p}; \vec{r}; \vec{q})$$

expresses that the circuit $\{\!\{f\}\!\}^{\vec{n}}$ computes $\vec{r}$ on input $\vec{p}$, with $\vec{q}$ being the intermediate steps of the computation (there is an atom q_i for every node of the circuit).

Then we define Boolean formulas $\|\varphi(\vec{x})\|^{\vec{n}}(\vec{p};\vec{q})$ by induction on complexity of a $\Pi^b_1(PV)$-formula φ. (Atoms $\vec{p}$ correspond to the variables $\vec{x}$. Atoms $\vec{q}$ are auxiliary, you may think of them as being universally quantified; they arise from universal quantifiers of φ, and from the output and intermediate atoms $\vec{q}$, $\vec{r}$ of $\|f\|^{\vec{n}}$, for functions f appearing in φ). The induction steps are done in an obvious way, and for atomic formulas and their negations we have

$$\begin{aligned}
\|f(\vec{x}) = g(\vec{x})\|^{\vec{n}} &:= \|f\|^{\vec{n}}(\vec{p};\vec{r};\vec{q}) \mathbin{\&} \|g\|^{\vec{n}}(\vec{p};\vec{r'};\vec{q'}) \to \bigwedge_i (r_i \equiv r'_i),\\
\|\neg f(\vec{x}) = g(\vec{x})\|^{\vec{n}} &:= \|f\|^{\vec{n}}(\vec{p};\vec{r};\vec{q}) \mathbin{\&} \|g\|^{\vec{n}}(\vec{p};\vec{r'};\vec{q'}) \to \neg \bigwedge_i (r_i \equiv r'_i),\\
\|f(\vec{x}) \le g(\vec{x})\|^{\vec{n}} &:=\\
&\|f\|^{\vec{n}}(\vec{p};\vec{r};\vec{q}) \mathbin{\&} \|g\|^{\vec{n}}(\vec{p};\vec{r'};\vec{q'}) \to \bigwedge_i \big(r_i \mathbin{\&} \bigwedge_{j>i}(r_j \equiv r'_j) \to r'_i\big),\\
\|\neg f(\vec{x}) \le g(\vec{x})\|^{\vec{n}} &:=\\
&\|f\|^{\vec{n}}(\vec{p};\vec{r};\vec{q}) \mathbin{\&} \|g\|^{\vec{n}}(\vec{p};\vec{r'};\vec{q'}) \to \neg \bigwedge_i \big(r_i \mathbin{\&} \bigwedge_{j>i}(r_j \equiv r'_j) \to r'_i\big).
\end{aligned}$$

The following theorem is then a prototypical example of a connection between an arithmetical theory, and a propositional proof system.

2.3.4 Theorem (Cook [9])

(i) *If* $PV \vdash \varphi(x)$, $\varphi \in \Pi^b_1$, *then tautologies* $\|\varphi\|^n$ *have polynomial-time constructible proofs in* EF.

(ii) $PV \vdash 0\text{-}RFN(EF)$ □

More information on reflection principles and propositional translations can be found in chapter 9.3 of [20].

Chapter 3

Randomized computation in bounded arithmetic

The main purpose of the present section is to develop a framework for expressing (defining) a certain kind of probabilistic algorithms in bounded arithmetic. The concept of a definable randomized algorithm is introduced in section 3.1; in section 3.2 we study basic consequences of the definition, and in section 3.3 we present a strengthened version of Wilkie's witnessing theorem for $S^1_2 + dWPHP(PV)$.

3.1 Definable probabilistic algorithms

As described in the next definition, we deal with a slightly nonstandard class of randomized algorithms; this choice was motivated by two demands: (i) we want *ZPP*, *RP*, and *coRP* languages to fit in, (ii) we want to consider functions as well as predicates. Moreover, it is not natural for randomized algorithms to compute *univalued* functions, hence we allow also *multifunctions.* Formally, an n-ary partial multifunction (pmf) is just an $(n+1)$-ary relation; by an abuse of language, we write $F(\vec{x}) = y$ as a shorthand for "y is one of the possible values of $F(\vec{x})$". Also notice that we left out *BPP* algorithms; they would require a different treatment, which does not blend smoothly with our approach to *RP*-like algorithms (in particular, the concept of *BPP*-like multifunctions does not seem to make much sense).

3.1.1 Definition Let F be a partial multifunction, $\alpha\colon \omega \to [0,1]$, and M a randomized Turing machine. We say that *M is an α-PPTM (probabilistic polynomial-time Turing machine) for F* iff the following conditions are satisfied:

(i) The time of any computation of M is polynomial in the length of its input.

(ii) On any input x, either M computes a number y such that $F(x) = y$, or it rejects.

(iii) If $x \in \mathrm{dom}(F)$, the probability that the computation of M on input x rejects is bounded from above by $\alpha(|x|)$.

Let *MFRP* be the class of all partial multifunctions (pmf) computable by a 1/2-*PPTM*.

3.1.2 Remarks

- Trivial amplification shows that the definition of *MFRP* does not change, if we replace the constant 1/2 by any function $\alpha(n)$ such that $1 - n^{-c} \geq \alpha(n) \geq 2^{-n^c}$ for some $c > 0$.
- $L \in ZPP$ iff the characteristic function of L is in *MFRP*.
- $L \in RP$ iff $L = \mathrm{dom}(F)$ for some $F \in MFRP$ iff the function which is constantly 0 on L and undefined on its complement is in *MFRP*.
- An α-*PPTM* for F is also an α-*PPTM* for any pmf G such that $\mathrm{dom}(F) = \mathrm{dom}(G)$ and $F \subseteq G$.

Our next step is to formalize (a strengthened version of) this definition in bounded arithmetic. First we give an informal description. We take a PV-function $f(\vec{x}, w)$, which simulates the computation of M on input $\vec{x}$ and a string of random bits w. The machine may touch only a polynomial number of these random bits, we thus fix an explicit bound $w < r(\vec{x})$. The output of $f(\vec{x}, w)$ is either a number, or a special symbol "$*$", which corresponds to halting in a rejecting state (we may encode it as a number by putting "$*$" $= 0$, "n" $= n + 1$). Now we need to express the condition (iii). Assume that $F(\vec{x})$ is defined, and let us say that a random string w is *good*, if $f(\vec{x}, w) \neq *$, otherwise it is *bad*. We will consider an *onto* mapping $m\colon t \times r(\vec{x}) \twoheadrightarrow s \times Bad$, where Bad is the set of all bad random strings; such a mapping explicitly witnesses that the ratio of bad strings is at most t/s, hence (iii) holds with $\alpha = t/s$. A formal definition follows:

3.1.3 Definition Let T be a theory containing PV, and $t(\vec{x})$ and $s(\vec{x})$ any PV-functions. A *definable* t/s-*PPTM* consists of PV functions f and r such that T proves

$$(\bigstar)\quad \exists w < r(\vec{x})\ \ f(\vec{x},w) \neq * \to \exists\,\text{circuit } C\,\forall w < r(\vec{x})\,(f(\vec{x},w) = * \to \\ \to \forall i < s(\vec{x})\,\exists v < r(\vec{x})\,\exists j < t(\vec{x})\ \ C(v,j) = \langle w,i\rangle),$$

where the size of C is tacitly bounded by a polynomial in the length of $\vec{x}$. A t/s-*PPTM* is *uniformly witnessed* if the formula above holds with $C(v,j)$ replaced by $m(\vec{x},v,j)$, where m is a PV-function symbol.

A definable t/s-*PPTM* computes a pmf $F(\vec{x})$, defined by

$$F(\vec{x}) = y \quad \text{iff} \quad \exists w < r(\vec{x})\ \ f(\vec{x},w) = y \neq *.$$

(Notice that this is Σ^b_1. Condition ($\bigstar$) itself is $\forall\Sigma^b_3$ for general *PPTM*'s, and $\forall\Sigma^b_1$ for uniformly witnessed *PPTM*'s.) We will call such a function *definable* t/s-*MFRP*, or shortly t/s-*definable.* A definable *MFRP* is *weakly total* iff ($\bigstar$) holds with the condition "$\exists w < r(\vec{x})\ \ f(\vec{x},w) \neq * \to$" *dropped.*

3.2 Properties of definable *MFRP*

3.2.1 Observation Assuming $dWPHP(PV)$, a weakly total definable t/s-*MFRP* is total, provided $2t \leq s$.

Proof: If $F(\vec{x})$ were undefined, the circuit C from 3.1.3 would represent a surjective mapping of $t(\vec{x})r(\vec{x})$ onto $s(\vec{x})r(\vec{x})$, contradicting $dWPHP(PV)$. □

3.2.2 Lemma ($PV \vdash$:) *Let t, s and p be PV-functions such that $p(x) \geq 1$. Any tp/sp-definable MFRP F has a t/s-definition, which is uniformly witnessed and/or weakly total, whenever F is. (Hence the symbol t/s may be interpreted as a quotient.)*

Proof: Let $f(x,w)$ and $r(x)$ be as in definition 3.1.3. Put

$$r'(x) := r(x)\cdot p(x),$$
$$f'(x,w') := f(x,w'_1),$$

where we consider w' as a pair $[w'_0,w'_1]$, $w'_0 < p(x)$, $w' = w'_1 \cdot p(x) + w'_0$. Let $f'(x,w') \neq *$ for some $w' < r'(x)$. This means that $f(x,w'_1) \neq *$, hence there is a circuit C such that $C(v,j) = \langle w,i\rangle$ for some $j < t(x)\cdot p(x)$ and $v < r(x)$, whenever $i < s(x)\cdot p(x)$, $w < r(x)$ and $f(x,w) = *$. Define a new circuit C' by

$$C'(v',j') := \langle [i_0,w],i_1\rangle, \qquad \text{where } C(v'_1,[v'_0,j']) = \langle w,i\rangle.$$

(As above, we decompose $i = [i_0,i_1]$, $v' = [v'_0,v'_1]$, etc.) Given $i' < s(x)$ and $w' < r(x)\cdot p(x)$ such that $f'(x,w') = *$, we have $[w'_0,i'] < s(x)\cdot p(x)$

and $f(x, w'_1) = *$, hence there is $j < t(x) \cdot p(x)$ and $v < r(x)$ such that $C(v, j) = \langle w'_1, [w'_0, i'] \rangle$. Therefore $C'([j_0, v], j_1) = \langle [w'_0, w'_1], i' \rangle = \langle w', i' \rangle$, $j_1 < t(x)$, and $[j_0, v] < r'(x)$ as required. □

3.2.3 Lemma $(PV + BB\Sigma^b_1 \vdash:)$ *Let t, s and p be PV-functions such that $p(x) \geq 1$. Then any t/s-definable MFRP F has a $t^{|p|}/s^{|p|}$-definition, which is uniform and/or weakly total, if the original one was. (This lemma also holds in plain PV, if p is constant.)*

Proof: For any fixed numbers a and b, we may identify $w < a^{|b|}$ with a sequence $\langle w_k \rangle_{k<|b|}$ of numbers less than a, namely $w_k = \lfloor \frac{w}{a^k} \rfloor \bmod a$. Given f and r defining F, we put

$$r'(x) := r(x)^{|p(x)|},$$

$$f'(x, w) := \begin{cases} *, & \text{if } \forall k < |p(x)| \;\; f(x, w_k) = *, \\ f(x, w_k), & \text{if } k < |p(x)| \text{ is minimal such that } f(x, w_k) \neq *. \end{cases}$$

Clearly, $\exists w < r(x) \; f(x, w) = y$ iff $\exists w < r'(x) \; f'(x, w) = y$. Assume that C is a circuit satisfying (★). Define

$$C'(v, j) = \langle w, i \rangle, \qquad \text{where } C(v_k, j_k) = \langle w_k, i_k \rangle \text{ for each } k < |p(x)|.$$

Let $i < s(x)^{|p(x)|}$ and $f'(x, w) = *$, $w < r'(x)$. This means that for any k, $f(x, w_k) = *$, hence there are $v' < r(x)$ and $j' < t(x)$ such that $C(v', j') = \langle w_k, i_k \rangle$. By $BB\Sigma^b_1$ there are sequences v and j such that $C(v_k, j_k) = \langle w_k, i_k \rangle$ for any $k < |p(x)|$. Then $C'(v, j) = \langle w, i \rangle$. □

3.2.4 Corollary $(PV + BB\Sigma^b_1 \vdash:)$ *Assuming $s(x) \geq 1$ and $t(x)(|p(x)|+1) \leq s(x)|p(x)|$ for some p, any t/s-definable MFRP has a $1/q$-definition for any $q(x)$. (I.e., as in the real world, we can boost the probability of error from $1 - 1/poly(n)$ to $1/2^{poly(n)}$.)*

Proof: Straightforward induction shows that $(a+1)^b \geq a^b + ba^{b-1}$ for any $b \geq 1$, $b \in Log$, in particular $(|p(x)|+1)^{|p(x)|} \geq 2|p(x)|^{|p(x)|}$. This implies $s^{|p|}|p|^{|p|} \geq t^{|p|}(|p|+1)^{|p|} \geq 2t^{|p|}|p|^{|p|}$, hence $s^{|p|} \geq 2t^{|p|}$. Thus using lemmas 3.2.3 and 3.2.2, any t/s-definable MFRP has a $1/2$-definition, and also a $1/q$-definition by lemma 3.2.3 again, as $2^{|q|} > q$. □

3.2.5 Lemma $(PV + BB\Sigma^b_1 \vdash:)$ *Any $1/2$-definable MFRP has a uniformly witnessed $1/2$-definition.*

Proof: Let f and r be the 1/2-definition of F, and let $C \leq c(x)$ be the circuit size bound implicit in ($\bigstar$). Put $p(x) = 2^{|c(x)|}$ and define f' and r' as in the proof of lemma 3.2.3. Finally, define

$$m(x, v, j) := \langle w, i \rangle, \quad \text{where } eval(j, \langle v_k, 0 \rangle) = \langle w_k, i_k \rangle \text{ for each } k < |p(x)|.$$

(Here $eval(C, x)$ is the value computed by a Boolean circuit C on input x.) Assuming $C \leq c(x)$ is a circuit satisfying ($\bigstar$), the proof of lemma 3.2.3 shows that m witnesses that f' and r' form a $2^{|c|}/2^{|p|}$-definition of F (the third argument of m will be C for all w and i). Lemma 3.2.2 implies that F has a uniform 1/2-definition, because $2^{|p|} = 2 \cdot 2^{|c|}$. □

3.2.6 Definition Let $F(\vec{x})$ and $G(y)$ be partial multifunctions. We say that G is *composable* with F, if for all $\vec{x}$, y and y' such that $F(\vec{x}) = y$ and $F(\vec{x}) = y'$, $y \in \mathrm{dom}(G)$ iff $y' \in \mathrm{dom}(G)$. Similarly for $G(y_1, \ldots, y_n)$ and $F_1(\vec{x}), \ldots, F_n(\vec{x})$.

3.2.7 Remark There is a total multifunction F and a partial function G, both in *MFRP* (using no randomness at all, in fact), such that their composition $G \circ F$ is a constant partial function with an *NP*-complete domain (hence $G \circ F \notin \mathit{MFRP}$, unless $\mathit{NP} = \mathit{RP}$). Indeed, choose an *NP*-complete predicate $Q(x) \leftrightarrow \exists y\, (|y| \leq |x|^n \,\&\, R(x, y))$ with $R \in P$, and put

$$\begin{aligned} &F(x) = y \quad \text{iff} \quad y = 0 \text{ or } R(x, y-1),\ |y-1| \leq |x|^n, \\ &G(0) \text{ is undefined}, \\ &G(x+1) = 0. \end{aligned}$$

Clearly, G is a partial p-time function. Also $F \in \mathit{MFRP}$, because F contains the constant 0 function. However,

$$(G \circ F)(x) = \begin{cases} 0, & \text{if } Q(x), \\ \text{undefined} & \text{otherwise.} \end{cases}$$

This shows that dealing with a condition like 3.2.6 is unavoidable, if we want *MFRP* to be closed under composition (or even to formalize this in bounded arithmetic).

3.2.8 Lemma ($PV + BB\Sigma^b_1 \vdash$:) *Let $F_1(\vec{x}), \ldots, F_n(\vec{x})$, $G(y_1, \ldots, y_n)$ be 1/2-definable p.m.f., such that G is composable with F_1, …, F_n. Then their composition $G(F_1(\vec{x}), \ldots, F_n(\vec{x}))$ is also 1/2-definable. (PV suffices, if G and F_i's are uniformly witnessed.)*

Proof: For simplicity we will assume $n = 1$. By 3.2.4 and 3.2.5 there is a 1/3-definition of F given by functions $f(x, w)$ and $r(x)$, uniformly witnessed by $m(x, v)$. Similarly let $f'(y, w)$ and $r'(y)$ be a 1/3-definition of G, uniformly witnessed by $m'(y, v)$. Using the idea of the proof of lemma 3.2.2, we may assume that $r'(y) \mid r'(z)$ whenever $y \leq z$. Let $b(x)$ be a PV-function such that $f(x', w) \leq b(x)$ for all $x' \leq x$ and $w < r(x')$. Define

$$r''(x) := r'(b(x)) \cdot r(x),$$

$$f''(x, w) := \begin{cases} *, & \text{if } f(x, w_0) = *, \\ f'(f(x, w_0), w_1 \bmod r'(f(x, w_0))) & \text{otherwise,} \end{cases}$$

$$m''(x, v, 0) := \langle [w, v_1], i\rangle, \qquad \text{where } m(x, v_0) = \langle w, i\rangle,$$

$$m''(x, v, 1) := \begin{cases} \langle 0, 0\rangle, & \text{if } f(x, v_0) = *, \\ \langle [v_0, w + \lfloor \frac{v_1}{q} \rfloor \cdot q], i\rangle, & \text{if } q = r'(f(x, v_0)), \text{ and} \\ & \quad m'(f(x, v_0), v_1 \bmod q) = \langle w, i\rangle. \end{cases}$$

We claim that f'' and r'' is a 2/3-definition of $G \circ F$, witnessed by m''. Clearly, non-$*$ values of $f''(x, w)$ are just the values of $G \circ F$. Assume that $f''(x, u) \neq *$ for some $u < r''(x)$, and let $w < r''(x)$, $i < 3$, and $f''(x, w) = *$. This means that either $f(x, w_0) = *$, or $f'(y, w_1 \bmod r'(y)) = *$, where $y = f(x, w_0)$. In the former case, we put $j = 0$, and we find $v < r(x)$ such that $m(x, v) = \langle w_0, i\rangle$, then we have $m''(x, [v, w_1], j) = \langle w, i\rangle$. In the latter case, put $q = r'(y)$ and $j = 1$. Since $f(x, w_0) \in \operatorname{dom}(G)$ and G is composable with F, we have $y \in \operatorname{dom}(G)$, hence there is $v < q$ such that $m'(y, v) = \langle w_1 \bmod q, i\rangle$. Then $m''(x, [w_0, v + q\lfloor \frac{w_1}{q} \rfloor], j) = \langle [w_0, (w_1 \bmod q) + q\lfloor \frac{w_1}{q} \rfloor], i\rangle = \langle w, i\rangle$.

By 3.2.4, we may turn a 2/3-definition of $G \circ F$ into a 1/2-definition. □

3.2.9 Lemma ($PV + BB\Sigma^b_1 + dWPHP(PV) \vdash$:) *Let $F(x)$ be a 1/2-definable pmf. For every $n \in Log$ there exists a (polynomial size) circuit $C\colon 2^n \to 2^m \cup \{*\}$ such that*

$$F(x) \text{ is defined } \leftrightarrow C(x) \neq *,$$
$$y = C(x) \neq * \rightarrow F(x) = y,$$

for any x of length n.

Proof: Fix n, and a uniformly witnessed $1/2^{n+1}$-definition of F. We may assume that $r(x) = r$ is independent on x (for x of length n). The witnessing function

$$m(x, v)\colon 2^n \times r \to 2^{n+1} \times r$$

cannot be onto (by $dWPHP$), we may thus fix $\langle i,w\rangle \in (2^{n+1}\times r)\smallsetminus \mathrm{rng}(m)$, and define $C(x)=f(x,w)$. Clearly $F(x)=C(x)$ if $C(x)\neq *$. Moreover, if $C(x)=*$ then $F(x)$ is undefined, because otherwise there would be a $v<r$ such that $m(x,v)=\langle i,w\rangle$, a contradiction. □

3.2.10 Example (Rabin-Miller algorithm [36, 29].) There is a *coRP*-predicate $P(x)$, 1/2-definable in S^1_2, such that S^1_2 proves

$$P(x) \quad \text{iff} \quad x>1\,\&\,\forall y<x\ \ (y\neq 0\to y^{x-1}\equiv 1 \pmod{x}).$$

Any number satisfying this condition is provably prime, but the converse is equivalent to the Little Fermat's Theorem (hence unlikely to be provable in S^1_2, by [23]).

Proof: Define

$$r(x):=\begin{cases}1, & \text{if } x\leq 1\vee 2\mid x,\\ x-2 & \text{otherwise,}\end{cases}$$

$$f(x,w):=\begin{cases}*, & \text{if } x=2,\\ 0, & \text{if } x\leq 1\vee(2\mid x\,\&\,x>2),\\ *, & \text{if } x>2 \text{ odd, and}\\ & \quad\forall k\,(2^k\mid x-1\to (w+1)^{(x-1)/2^k}\equiv 1 \pmod{x}),\\ *, & \text{if } x>2 \text{ odd, } j\neq 0\text{, and}\\ & \quad (w+1)^{(x-1)/2^j}\equiv -1 \pmod{x},\\ 0 & \text{otherwise,}\end{cases}$$

where $j<|x|$ is the least number such that $2^j\mid x-1$ and

$$(w+1)^{(x-1)/2^j}\not\equiv 1 \pmod{x},$$

$$P(x):\leftrightarrow \forall w<r(x)\ \ f(x,w)=*,$$
$$Q(x):\leftrightarrow x>1\,\&\,\forall y<x\ \ (y\neq 0\to y^{x-1}\equiv 1 \pmod{x}).$$

(It would be more natural to consider random choices $w\in[1,x)$.) From now on, we will assume that $x>2$ and x is odd, other cases are trivial.

Let $x-1=y2^k$, where y is odd and $k>0$. Clearly $(-1)^y\equiv -1\not\equiv 1 \pmod{x}$, let $i\leq k$ be the least number such that $\exists a\in\mathbb{Z}^*_x\ a^{(x-1)/2^i}\not\equiv 1 \pmod{x}$, which exists by the Σ^b_1-*LENGTH-MIN* principle. (Here $a\in\mathbb{Z}^*_x$ means $a<x\&(a,x)=1$, which is in PV equivalent to $a<x\&\exists b<x\ \ ab\equiv 1 \pmod{x}$.)

Case 1: $i = 0$. Clearly, neither $P(x)$ nor $Q(x)$ holds. Let $b \in \mathbb{Z}_x^*$ be such that $b^{x-1} \not\equiv 1 \pmod{x}$. (Forgetting about S^1_2 for a moment, $\{w;\, f(x, w+1) = *\}$ is contained in a proper subgroup $\{w;\, w^{x-1} \equiv 1\} \lneqq \mathbb{Z}_x^*$, thus there are at most $|\mathbb{Z}_x^*|/2$ of them, and we may witness this using multiplication by a fixed element b of a nontrivial coset of this subgroup.) Define

$$C(x, v) := \begin{cases} \langle (b(v+1) \bmod x) - 1, 0\rangle, & \text{if } (b(v+1))^{x-1} \equiv 1 \pmod{x}, \\ \langle v, 1\rangle & \text{otherwise.} \end{cases}$$

Assume $w < r(x)$ is such that $f(x, w) = *$. Then we have $(w+1)^{x-1} \equiv 1 \pmod{x}$ and $(b(w+1))^{x-1} \not\equiv 1 \pmod{x}$, hence $C(x, w) = \langle w, 1\rangle$. Since $b, w+1 \in \mathbb{Z}_x^*$, there is $v < r(x)$ such that $b(v+1) \equiv w+1 \pmod{x}$. We have $(b(v+1))^{x-1} \equiv 1 \pmod{x}$, thus $C(x, v) = \langle w, 0\rangle$.

Case 2: $i > 0 \;\&\; \exists b \in \mathbb{Z}_x^* \; b^{(x-1)/2^i} \not\equiv \pm 1 \pmod{x}$. Fix any such b. Obviously $\neg P(x)$, moreover if we put $c \equiv b^{(x-1)/2^i} \pmod{x}$, we have $x \mid (c^2 - 1) = (c-1)(c+1)$ by minimality of i, but neither $x \mid c-1$ nor $x \mid c+1$. This means that x is not prime, and *a fortiori* $\neg Q(x)$ (as $Q(x)$ would imply $\mathbb{Z}_x^* = [1, x)$). Similarly to the Case 1, we define

$$C(x, v) := \begin{cases} \langle (b(v+1) \bmod x) - 1, 0\rangle, & \text{if } (b(v+1))^{(x-1)/2^i} \equiv \pm 1 \quad (x), \\ \langle v, 1\rangle & \text{otherwise.} \end{cases}$$

If $w < r(x)$ and $f(x, w) = *$, we have $(w+1)^{(x+1)/2^i} \equiv \pm 1 \pmod{x}$, hence $C(x, w) = \langle w, 1\rangle$ and $C(x, v) = \langle w, 0\rangle$ for some $v < r(x)$, by essentially the same argument as above.

Case 3: $i > 0 \;\&\; \forall b \in \mathbb{Z}_x^* \; b^{(x-1)/2^i} \equiv \pm 1 \pmod{x}$. We need some elementary number theory.

Claim 1 *PV proves the Chinese Remainder Theorem: if $a = \langle a_j \rangle_{j<\ell}$ is a sequence of pairwise coprime numbers and $b = \langle b_j \rangle_{j<\ell}$, then there is c such that $c \equiv b_j \pmod{a_j}$ for all $j < \ell$.*

Proof: For any $j < \ell$, $(a_j, \prod_{j' \neq j} a_{j'}) = 1$ (by Δ^b_1-*LIND*), hence there is $d_j < a_j$ such that $c_j := d_j \prod_{j' \neq j} a_{j'} \equiv 1 \pmod{a_j}$. Put $c = \sum_{j<\ell} b_j c_j$. We have $c_j \equiv 1 \pmod{a_j}$ and $c_j \equiv 0 \pmod{a_{j'}}$ for all $j' \neq j$, hence $c \equiv b_j \pmod{a_j}$. □ (claim 1)

Claim 2 *S^1_2 proves that*

(*i*) *$x > 0$ is a prime power iff there are no coprime proper divisors u and v of x such that $uv = x$.*

(*ii*) *Any $x > 0$ is uniquely representable as $x = \prod_{j<\ell} p_j^{e_j}$, where $\langle p_j \rangle_{j<\ell}$ is an increasing sequence of primes and each e_j is nonzero.*

Proof: Every number $x > 1$ is divisible by a prime. To see this, choose $p > 1$, $p \mid x$ with minimal length (using Δ^b_1-*LENGTH-MIN*). If $p = uv$, $u > 1$, then $v \mid x$ and $|v| < |p|$, hence $v = 1$, i.e., p is a prime.

If $x = p^e$ for a prime p, then any proper divisor of x is divisible by p (by Δ^b_1-*LIND* on e), hence $p \leq (u, v)$ for any $u, v > 1$ which divide x. On the other hand, assume that the right hand side of (i) holds, and w.l.o.g. $x > 1$. Let p be a prime divisor of x, and let $e < |x|$ be maximal such that $p^e \mid x$. If $p^e < x$, we have $(p^e, x/p^e) > 1$, thus $p \mid (x/p^e)$ and $p^{e+1} \mid x$, a contradiction. Hence $x = p^e$ is a prime power.

By Σ^b_1-*LENGTH-MAX*, there is the maximal $k < |x|$ such that there exists a sequence $a = \langle p_j \rangle_{j<k}$ of numbers greater than 1, such that $\prod_{j<k} p_j = x$. Every p_j in any maximal sequence is obviously prime. The sequence of p_j's may be arranged in non-decreasing order, and we may group together occurences of the same prime, yielding $x = \prod_{j<\ell} p_j^{e_j}$ as in the statement of the claim. If $x = \prod_{j<m} q_j^{f_j}$ is another such representation, Δ^b_1-*LIND* on $j < \min(\ell, m)$ shows that $p_j = q_j$ and $e_j = f_j$, hence also $\ell = m$. □ (claim 2)

Let us return to the analysis of the Case 3.

First assume that x is not a prime power. Choose coprime $a_1, a_2 > 1$ such that $a_1 a_2 = x$, and $b \in \mathbb{Z}^*_x$ such that $b^{(x-1)/2^i} \equiv -1 \pmod{x}$ (by the definition of i). The Chinese Remainder Theorem gives us c such that $c \equiv 1 \pmod{a_1}$ and $c \equiv b \pmod{a_2}$. We claim that $c \in \mathbb{Z}^*_x$: we have $(b, a_2) = 1$, hence we may find d such that $d \equiv 1 \pmod{a_1}$ and $bd \equiv 1 \pmod{a_2}$, then $cd \equiv 1 \pmod{x}$. By our assumption, $c^{(x-1)/2^i} \equiv 1 \pmod{x}$ or $c^{(x-1)/2^i} \equiv -1 \pmod{x}$. However, the former contradicts $c^{(x-1)/2^i} \equiv -1 \pmod{a_2}$, while the latter contradicts $c^{(x-1)/2^i} \equiv 1 \pmod{a_1}$, because a_1 and a_2 are odd.

We may thus write $x = p^e$, where p is an odd prime. Assume that $e > 1$. Notice that for any u and v, $(up^{e-1} + 1)(vp^{e-1} + 1) \equiv (u + v)p^{e-1} + 1 \pmod{x}$. This gives $(p^{e-1} + 1)^u \equiv up^{e-1} + 1 \pmod{x}$ by Δ^b_1-*PIND*, in particular $(p^{e-1} + 1)^{x-1} \equiv 1 - p^{e-1} \pmod{x}$. However $p^{e-1} + 1 \in \mathbb{Z}^*_x$, hence $(p^{e-1} + 1)^{x-1} \equiv 1 \pmod{x}$. This means $x \mid p^{e-1}$, a contradiction. Therefore $e = 1$ and $x = p$ is a prime.

We have $Q(x)$, because $\mathbb{Z}^*_p = [1, p)$ and $b^{x-1} \equiv 1 \pmod{x}$ for any $b \in \mathbb{Z}^*_x$ by our assumption. Also $P(x)$ holds: if $f(x, w) \neq *$, we would have $b^2 \equiv 1 \pmod{x}$ and $b \not\equiv \pm 1 \pmod{x}$ (where $b \equiv (w + 1)^{(x-1)/2^j} \pmod{x}$ for some $j > 0$), which we know is impossible for any prime x. □

3.3 Witnessing *dWPHP*

The following theorem is implicit in N. Thapen's alternative proof of Wilkie's witnessing theorem, presented in [48].

3.3.1 Theorem (Thapen) *Assume that*

$$S^1_2 + dWPHP(PV) \vdash \forall x\, \exists y\, \varphi(x, y),$$

where φ is Σ^b_1. Then for any ℓ there are $k \geq \ell$ and PV-function symbols G, g, and h such that

$$PV \vdash \forall x\, \forall w < 2^{2|x|^k}\, (g(x, w) < 2^{|x|^k}\ \&\ (G(g(x, w)) = w \vee \varphi(x, h(x, w)))).$$

More generally, there are PV-functions g, h, and a constant k such that

$$PV \vdash \forall x\, \forall b \geq 2^{|x|^k}\, \forall w < b^2\, (g(x, w, b) < b \\ \&\ (G(g(x, w, b)) = w \vee \varphi(x, h(x, w, b)))).$$

Proof (*sketch*)*:* If $f(x, y)$ is a PV-function, there is a parameter-free PV-function $G(z)$ such that G maps b^4 onto b^8, whenever there are $a, c < b$ such that $f(c, \cdot)$ maps a onto a^2. (This is lemma 3.8 of [48].) Take a "universal" function (e.g., a circuit evaluator) for f. Our assumption on φ gives

$$S^1_2 \vdash \forall x\, (\exists y\, \varphi(x, y) \vee \exists a, c\, \neg dWPHP(f(c))^a_{a^2}).$$

By Parikh's theorem, all existential quantifiers may be bounded by a term $t(x)$, and the properties of G imply

$$S^1_2 \vdash \forall x\, (\exists y \leq t(x)\, \varphi(x, y) \vee \forall b \geq t(x)^4\, \neg dWPHP(G)^b_{b^2}).$$

We may write this as

$$S^1_2 \vdash \forall x\, \forall w\, \forall b\, (\exists y \leq t(x)\, \varphi(x, y) \vee (b \geq t(x)^4\ \&\ w < b^2 \rightarrow \exists v < b\, G(v) = w)),$$

and an application of Buss' witnessing theorem gives us g and h as required. □

3.3.2 Corollary *The $\forall\Sigma^b_1$-consequences of $S^1_2 + dWPHP(PV)$ can be axiomatized over PV by $dWPHP'(PV)$, where $dWPHP'(f, g)$ denotes the formula*

$$a > 1 \rightarrow \exists x < a^2\, (g(x, a) \geq a \vee f(g(x, a), a) \neq x). \qquad □$$

By Wilkie's witnessing theorem, $\forall\Sigma^b_1$-consequences $S^1_2 + dWPHP(PV)$ are witnessed by randomized p-time functions (total *MFRP* in our notation). Our next theorem ensures that these witnessing functions can be chosen so that they are definable and provably total in $S^1_2 + dWPHP(PV)$. (Conversely, the statement that certain PV-functions define a uniformly witnessed total *MFRP* is $\forall\Sigma^b_1$.)

3.3.3 Theorem *Let $\varphi(\vec{x}, y)$ be a Σ^b_1-formula such that $\forall\vec{x}\,\exists y\,\varphi(\vec{x}, y)$ is provable in $S^1_2 + dWPHP(PV)$. Then for every PV-function s there is $F \in$ MFRP such that*

(i) F has a uniformly witnessed $1/s$-definition in PV,

(ii) F is weakly total in PV (in particular, F is provably total in $PV + dWPHP(PV)$),

(iii) $PV \vdash F(\vec{x}) = y \to \varphi(\vec{x}, y)$.

In particular, every formula which is Δ^b_1 in $S^1_2 + dWPHP(PV)$ is in $PV + dWPHP(PV)$ equivalent to a definable ZPP-predicate.

Proof: Fix ℓ such that $PV \vdash s(x) \le 2^{|x|^\ell}$, and find $k \ge \ell$, and PV-functions G, g, and h according to the theorem 3.3.1. Define

$$r(x) := 2^{|x|^k} \cdot 2^{|x|^k},$$

$$f(x, w) := \begin{cases} h(x, w), & \text{if } G(g(x, w)) \ne w, \\ * & \text{otherwise,} \end{cases}$$

$$m(x, v) := \langle G(v_0), v_1\rangle,$$

where $v = [v_0, v_1] = v_1 \cdot 2^{|x|^k} + v_0$ as in 3.2.2. We claim that f and r define in PV a weakly total $1/2^{|x|^k}$-*MFRP*, witnessed by m. To see this, let $w < 2^{2|x|^k}$ be such that $f(x, w) = *$, and $i < 2^{|x|^k}$. Put $v = [g(x, w), i]$. Then we have $m(x, v) = \langle w, i\rangle$, because $G(g(x, w)) = w$, and $v < r(x)$, because $g(x, w) < 2^{|x|^k}$.

If we put $F(x) = y$ iff $\exists w < r(x)\ f(x, w) = y \ne *$, then any value y of $F(x)$ satisfies $\varphi(x, y)$, because $y = h(x, w)$ for some $w < r(x)$ such that $G(g(x, w)) \ne w$. □

Chapter 4

Hard Boolean functions

In this chapter, we will investigate the concept of Boolean functions with large circuit complexity in the context of bounded arithmetic. In section 4.1, we establish the equivalence of $dWPHP(PV)$ and existence of functions with exponential circuit complexity. In section 4.2 we formalize the construction of the Nisan-Wigderson pseudorandom generator in a conservative extension of $S^1_2 + dWPHP(PV)$, to obtain a formalized derandomization result for definable probabilistic algorithms introduced in chapter 3. In section 4.3 we formalize the Impagliazzo-Wigderson strengthening of the Nisan-Wigderson derandomization theorem.

4.1 Hard functions and $dWPHP(PV)$

4.1.1 Definition Let $\varepsilon > 0$. A number x (viewed as an n-bit binary string, $n = |x|$) is *ε-hard,* if there is no Boolean circuit C on $|n|$ variables such that $|C| \le n^\varepsilon$, and $C(u) = bit(x,u)$ for all $u < n$. We write $Hard_\varepsilon(x)$ in such a case.

A Boolean function f on $k \in LogLog$ variables is identified with its truth table, i.e., a 2^k-bit number.

A function f is *ε-hard on average* (abbreviated $Hard^{\varnothing}_\varepsilon(f)$), if there does not exist a circuit C of size $|C| \le 2^{\varepsilon k}$ which approximates f, i.e.,

$$|\{u < 2^k;\, C(u) = f(u)\}| \ge (1/2 + 2^{-\varepsilon k})2^k.$$

Notice that $Hard_\varepsilon(x)$ and $Hard^{\varnothing}_\varepsilon(f)$ are Π^b_1.

4.1.2 Lemma ($PV + dWPHP(PV) \vdash$:) *For every $n \in Log$, there is an x of length n such that x cannot be computed by a circuit of size $n/(2|n|)$.*

Proof: Let $e\colon 2^{n-1} \to 2^n$ be a PV-function, which interprets its input as a circuit on $|n|$ variables, and outputs the truth table of the circuit. By $dWPHP(e)$ there is an $x \in 2^n \smallsetminus \mathrm{rng}(e)$. Since any circuit of size $m = n/(2|n|)$ may be described by a number of length at most $2m(|m|+1) \le n-1$, x is not computable by a circuit of size $\le m$. □

4.1.3 Corollary ($PV + dWPHP(PV) \vdash$:) *For every $k \in LogLog$, there is a Boolean function $f\colon 2^k \to 2$ such that $Hard_{1-o(1)}(f)$.* □

4.1.4 Lemma ($PV + dWPHP(PV) \vdash$:) *For any $k \in LogLog$, there are $(1/3 - o(1))$-hard on average functions $f\colon 2^k \to 2$.*

Proof: Put $n = 2^k$, and $m = (n/k)^{1/3}$. Consider the function

$$g\colon 2^{2m|m|} \times \sum_{i=0}^{n(1/2-1/m)} \binom{n}{i} \to 2^n,$$

whose first argument is a circuit $C\colon 2^k \to 2$ of size m, its second argument is a string $x \in 2^n$ containing at most $n(1/2 - 1/m)$ 1's, and its output is the truth-table of C XOR'ed by x. Clearly, a function $f\colon 2^k \to 2$ is $(1/3 - |k|/k)$-hard on average if $f \notin \mathrm{rng}(g)$. By Chernoff's inequality, provable in PV by theorem A.5, the domain of g is a number bounded by

$$d2^{\frac{2}{3}n^{1/3}k^{2/3}} 2^n 2^{-2n^{4/3}k^{2/3}n^{-1}} = d2^{n-\frac{4}{3}n^{1/3}k^{2/3}}$$

for some constant d. Since $d2^{n-\frac{4}{3}n^{1/3}k^{2/3}} < 2^{n-1}$ for $n \gg 0$, the function g cannot be onto, by $dWPHP(PV)$. □

4.1.5 Theorem ($S^1_2 \vdash$:) *Assume that $dWPHP(PV)$ fails. Then there is $s \in Log$ such that every string x is computable by a circuit of size at most s.*

Proof: Let $h\colon 2^m \twoheadrightarrow 2^{2m}$ be a surjection, computable by a circuit C. The main idea of the proof is: for any $i \in LogLog$, h may be amplified in i steps into a surjection $2^m \twoheadrightarrow (2^m)^{2^i}$, and this will allow us to express any $x \in 2^{2^i m}$ by a circuit of size $O(|C|i)$.

Let $D\colon 2 \times 2^m \to 2^m$ be the circuit defined by $D_j(b,y) = (\neg b \mathbin{\&} C_j(y)) \vee (b \mathbin{\&} C_{j+m}(y))$ for all $j < m$, where $\bullet_j$ is a shorthand for $bit(\bullet, j)$. Fix $i \in LogLog$, and define a sequence of circuits $E^k\colon 2^k \times 2^m \to 2^m$, $k \le i$ by

$$\begin{aligned} E^0(0,y) &:= y, \\ E^{k+1}(u,y) &:= E^k(u \restriction k, D(u_k, y)), \quad \text{where } u \restriction k = u \bmod 2^k, \end{aligned}$$

and put $E := E^i$. Notice that the size of E is bounded by $i|D|$. We claim that E represents an onto map $2^m \twoheadrightarrow 2^{2^i m}$ in the following sense: for any $x < 2^{2^i m}$, there is $y < 2^m$ such that $E_j(u, y) = x_{um+j}$ holds for every $u < 2^i$ and $j < m$. Indeed, we show by induction on $k \le i$ that there is a sequence w of numbers less than 2^m such that

$$\begin{aligned} &lh(w) = 2^{i-k} \& \\ &\qquad \forall v < 2^{i-k}\, \forall u < 2^k\, \forall j < m \;\; E^k_j(u, (w)_v) = bit(x, (v2^k + u)m + j). \end{aligned}$$

(This is Σ^b_1, because $i \in LogLog$, i.e., all universal quantifiers are sharply bounded.) The base step is trivial, we simply view x as a sequence of 2^i numbers less than 2^m. Assume that we have found a suitable w for $k < i$. Since C is onto, there is a sequence w' such that $C((w')_v) = [(w)_{2v}, (w)_{2v+1}]$ for any $v < 2^{i-k-1}$ (using $BB\Sigma^b_1$). We claim that w' works for $k+1$: given numbers $v < 2^{i-k-1}$, $u < 2^{k+1}$, and $j < m$, we have

$$\begin{aligned} E^{k+1}_j(u, (w')_v) &= E^k_j(u \restriction k, D(u_k, (w')_v)) = E^k_j(u \restriction k, (w)_{2v+u_k}) = \\ &= bit(x, ((2v + u_k)2^k + u \restriction k)m + j) \\ &= bit(x, (v2^{k+1} + u)m + j). \end{aligned}$$

Let $x < 2^{2^i m}$, and let $y < 2^m$ be its "inverse image" as described above. We may construct a small Boolean circuit $B\colon 2^{|n|} \to 2$ computing x as follows: $B(u) = E_{u \bmod m}(\lfloor \frac{u}{m} \rfloor, y)$. For simplicity, we may assume that m is a power of two, which means that the size of B is bounded by $2m|m| + i|D|$.

In other words, any x of length n is computable by a circuit of size $\le 2m|m| + |D| \cdot |\lceil n/m \rceil| \le c|n|$ for a suitable $c \in Log$. Take any $d \in Log \smallsetminus LogLog$ (this is possible, because $S^1_2 + Exp \vdash dWPHP(PV)$). Then $d > |n|$, hence x is computable by a circuit of size at most $s := c \cdot d \in Log$. □

4.1.6 Corollary *Let $0 < \varepsilon < 1$. There exists a standard constant c such that the following are equivalent over S^1_2:*

(i) $dWPHP(PV)$,

(ii) $\forall k \in LogLog\ (k \ge c \to \exists f\colon 2^k \to 2\ \ Hard_\varepsilon(f))$,

(iii) $\forall k_0 \in LogLog\ \exists k \in LogLog\ (k \ge k_0 \,\&\, \exists f\colon 2^k \to 2\ \ Hard_\varepsilon(f))$.

The same holds for hard on average functions, if $\varepsilon < 1/3$.

Proof: $(i) \to (ii)$ follows from 4.1.3 and 4.1.4, $(ii) \to (iii)$ is trivial. The implication $(iii) \to (i)$ follows from 4.1.5, because numbers $2^{\varepsilon k}$, $k \in \mathit{LogLog}$, are cofinal in Log for any fixed ε. □

4.1.7 Corollary *There is a PV-function $C(a, x)$ such that the negation of $dWPHP'(PV)$ implies*

$$\exists a\, \forall x \;\; C(a, x) \textit{ is a circuit of size } \leq |a| \textit{ computing } x$$

over PV. Actually, a can be itself computed by a PV-function from a counterexample to $dWPHP'(PV)$.

Proof: Let g and h be counterexamples to $dWPHP'(PV)$, i.e., $h\colon b \to b^2$, $g\colon b^2 \to b$, $g \circ h = id$. Given x, we proceed as in the proof of 4.1.5 to construct a small circuit for x, but instead of nondeterministically guessing preimages under h, we use g to find them explicitly (this way we also get rid of $BB\Sigma^b_1$, and Σ^b_1-$LIND$).

Alternatively, we may use Buss' witnessing theorem. Theorem 4.1.5 tells us

$$S^1_2 \vdash \exists b\, \forall v < 2b\, (h(v) < b \;\&\; g(h(v)) = v) \to \exists S\, \forall x\, \exists C \leq S\, (C \text{ computes } x),$$

and it is easy to see from its proof that S is actually bounded by a term $t(b)$, thus

$$S^1_2 \vdash \forall b \forall x\, (\exists v < 2b\, (h(v) \geq b \vee g(h(v)) \neq v) \vee \exists C \leq t(b)\, (C \text{ computes } x)).$$

The formula in parenthesis is Σ^b_1, hence there is a PV-function f such that

$$PV \vdash (f(b, x) < 2b \;\&\; (h(f(b, x)) \geq b \vee g(h(f(b, x))) \neq f(b, x)) \vee \\ \vee\, (f(b, x) \leq t(b) \;\&\; f(b, x) \text{ computes } x),$$

which means

$$PV + \neg dWPHP'(PV) \vdash \exists b\, \forall x\, (f(b, x) \leq t(b) \;\&\; f(b, x) \text{ computes } x).$$

It suffices to define $C(a, x) = \min\{f((a)_0, x), a\}$, as we can take $a = \langle b, t(b)\rangle$.

Notice that the converse to this corollary holds too, in a similar fashion to lemma 4.1.2. □

We will need the following refinement of lemmas 4.1.2 and 4.1.4 in section 4.2. Notice that we cannot use these lemmas directly to prove the improved version, as $BB\Pi^b_1$ is not available. The conclusion is $\forall\Sigma^b_2$, but it is not *a priori* clear that the Σ^b_2-conservativity of $BB\Sigma^b_2$ over S^1_2 extends to $S^1_2 + dWPHP(PV)$, although see 4.2.4.

4.1.8 Lemma *Let $\varepsilon < 1$. There is a constant c such that the following are provable in $PV + dWPHP(PV)$:*

(*i*) $\forall k \in \mathit{LogLog}\, \exists w\, \forall i < k\, (i \geq c \to (w)_i \colon 2^i \to 2 \;\&\; \mathit{Hard}_\varepsilon((w)_i))$,

(*ii*) $\forall k \in \mathit{LogLog}\, \exists w\, \forall i < k\, (i \geq c \to (w)_i \colon 2^i \to 2 \;\&\; \mathit{Hard}^{\varnothing}_{\varepsilon/3}((w)_i))$.

Proof: As in lemma 4.1.4, choose d such that

$$\sum_{j=0}^{2^i(1/2-2^{-\delta i})} \binom{2^i}{j} \leq 2^{d+2^i-2\cdot 2^{i(1-2\delta)}}$$

for all $i > 0$, where $\delta = \varepsilon/3$, and choose $c \geq 2$ such that $2 \cdot 2^{i(1-2\delta)} \geq \frac{2}{3} i 2^{\delta i} + d + i$ for all $i \geq c$. Put $k = ||b||$, and define a PV-function

$$g \colon \sum_{i=c}^{k-1} 2^{2^k-2^c-i} \to 2^{2^k-2^c}$$

as follows: given $i < k$ and $x < 2^{2^k-2^c-i}$, interpret the first $2^k - 2^c - 2^i$ bits of x as a sequence $\langle f_j;\, c \leq j < k, j \neq i\rangle$ of functions $f_j \colon 2^j \to 2$. The next $2\delta i 2^{\delta i}$ bits of x describe a circuit $C \colon 2^i \to 2$ of size $2^{\delta i}$, and the rest of x defines a binary string y of length 2^i with at most $2^{i-1} - 2^{i(1-\delta)}$ ones. (We need $d+2^i-2\cdot 2^{i(1-2\delta)}$ bits for y, and we have $2^i - 2\delta i 2^{\delta i} - i \geq 2^i - 2\cdot 2^{i(1-2\delta)} + d$ bits left.) We create a function $f_i \colon 2^i \to 2$ by taking the truth-table of C XOR'ed by y, and we let g output the sequence $\langle f_j;\, c \leq j < k\rangle$.

If $f = \langle f_j;\, c \leq j < k\rangle$ is a sequence of functions outside of the range of g, then all f_j are δ-hard on average. The domain of g is at most $2^{2^k-2^c-c+1} \leq 2^{2^k-2^c-1}$, hence g is not onto by $dWPHP(PV)$. A similar argument works for ε-hard functions. □

4.2 The Nisan-Wigderson generator

This section presents a derandomization result for definable probabilistic algorithms within bounded arithmetic. We will follow closely the Nisan-Wigderson construction [31]; however, we will present the derandomization in a relativized form: rather than postulating the existence of an explicit language in E with exponential average-case hardness, we will use an *oracle* for a family of hard Boolean functions, and our derandomized algorithms will have access to this oracle. We thus work in a theory with an extra unary function symbol α:

4.2.1 Definition Let $0 < \varepsilon < 1$ and c be standard constants. The theory $HARD^{\varnothing}_{\varepsilon,c}$ is an extension of $S^1_2(\alpha)$ by the following axioms:

$$\alpha(x)\colon 2^{||x||} \to 2,$$
$$x > c \to Hard^{\varnothing}_{\varepsilon}(\alpha(x)).$$

The theory $HARD_{\varepsilon,c}$ is defined similarly. We will usually ignore c in the sequel. (To avoid confusion: here $||x||$ means double iteration of the length function, it has nothing to do with the translation of Π^b_1-formulas into propositional logic. We will use this translation only in section 5.)

4.2.2 Observation $HARD^{\varnothing}_{\varepsilon} \vdash HARD_{\varepsilon} \vdash dWPHP(PV)$. □

4.2.3 Theorem *Let T denote the theory $HARD_{\varepsilon}$ or $HARD^{\varnothing}_{\varepsilon/3}$, with $0 < \varepsilon < 1$. Then T is fully conservative over $S^1_2 + dWPHP(PV)$. More generally, for any $i \geq 1$, $T + S^i_2(\alpha)$ and $T + T^i_2(\alpha)$ are conservative extensions of $S^i_2 + dWPHP(PV)$ and $T^i_2 + dWPHP(PV)$, respectively. Every countable model of $S^1_2 + dWPHP(PV)$ has an expansion into a model of T.*

Proof: Let $\mathcal{A}$ be a countable model of $S^1_2 + dWPHP(PV)$. Choose an increasing chain $p^0 \subseteq p^1 \subseteq p^2 \subseteq \dots$ of sequences $p^n \in A$ such that

$$\forall i < lh(p^n)\,(i \geq c \to (p^n)_i\colon 2^i \to 2 \,\&\, Hard_{\varepsilon}((p^n)_i)),$$

where c is the constant from lemma 4.1.8, and such that $\{lh(p^n);\, n \in \omega\}$ is cofinal in $LogLog(A)$. Define $\alpha^A = \bigcup_{n\in\omega} p^n$, i.e.

$$\alpha^A(a) := (p^n)_{||a||}, \qquad \text{for any } n \text{ s.t. } lh(p^n) > ||a||.$$

Clearly, $\langle \mathcal{A}, \alpha^A \rangle$ satisfies the hardness conditions from T.

Claim 1 *Let $\varphi(\vec{x})$ be a $\Sigma^b_\infty(\alpha)$-formula. Denote by $\tilde{\varphi}(p, \vec{x})$ the Σ^b_∞-formula which results from φ by substitution of $(p)_{||t||}$ for every subterm $\alpha(t)$. There is a constant c_φ such that*

$$\langle \mathcal{A}, \alpha^A \rangle \vDash \varphi(\vec{a}) \quad \textit{iff} \quad \mathcal{A} \vDash \tilde{\varphi}(p^n, \vec{a})$$

for any n such that $lh(p^n) > c_\varphi ||\vec{a}||$.

Proof: By straightforward induction on complexity of φ. If φ is atomic, it suffices to choose c_φ so that all $(p^n)_{||t||}$ are defined. If e.g. $\varphi(\vec{x}) = \exists y \leq s(\vec{x})\,\psi(y, \vec{x})$, take $c_\varphi = (d+1)c_\psi$, where d is such that $||s(\vec{x})|| < d||\vec{x}||$ for all $\vec{x}$. The assertion then follows from the induction hypothesis, because $y \leq s(\vec{x})$ and $lh(p^n) > c_\varphi||\vec{x}||$ imply $lh(p^n) > c_\psi(||y|| + ||\vec{x}||)$. □ (claim 1)

As a corollary of the claim we get that $\langle \mathcal{A}, \alpha^A \rangle \vDash \forall \vec{x}\, \varphi(\vec{x})$, whenever φ is a bounded $L(\alpha)$-formula, and $\mathcal{A} \vDash \forall \vec{x} \forall p\, \tilde{\varphi}(p, \vec{x})$. In particular, $\langle \mathcal{A}, \alpha^A \rangle \vDash S^1_2(\alpha)$, and additionally it is a model of $S^i_2(\alpha)$ or $T^i_2(\alpha)$, if S^i_2 or T^i_2 holds in $\mathcal{A}$. □

4.2.4 Corollary *$S^1_2 + dWPHP(PV) + BB\Sigma^b_2$ is $\forall\Sigma^b_2$-conservative over $S^1_2 + dWPHP(PV)$.*

Proof: This follows from 4.2.3, and $\Sigma^b_2(\alpha)$-conservation of $BB\Sigma^b_2(\alpha)$ over $S^1_2(\alpha)$ [39], because $HARD_{1/2}$ is a $\forall\Pi^b_1(\alpha)$-axiomatized extension of $S^1_2(\alpha)$. □

Now we turn attention to the actual derandomization. First, notice that we get a certain derandomization result for free, namely for definable $MFRP$ which are *provably total* in $S^1_2 + dWPHP(PV)$:

4.2.5 Lemma *Let $\varepsilon > 0$, and let F be a definable $MFRP$, provably total in $S^1_2 + dWPHP(PV)$. Then there is a $PV(\alpha)$ function f such that*

$$HARD_\varepsilon \vdash f(\vec{x}) = y \rightarrow F(\vec{x}) = y.$$

Proof: By our assumptions $HARD_\varepsilon$ proves $\forall \vec{x}\, \exists y\, F(\vec{x}) = y$, which is $\forall\Sigma^b_1$. Moreover, $HARD_\varepsilon$ is a $\forall\Pi^b_1(\alpha)$ extension of $S^1_2(\alpha)$, hence the result follows from the relativized Buss' witnessing theorem. □

However, we want to derandomize also functions which are not provably total (e.g., RP-predicates). Moreover, the Nisan-Wigderson construction will give a stronger result (see 4.2.11): f needs only *one* oracle query.

4.2.6 Definition ([31]) Let $k, \ell, t, m \in Log$, $k \leq \ell \leq t$. A $\langle k, \ell, t, m\rangle$-*design* is a sequence $\langle S_i \rangle_{i<m}$ of subsets $S_i \subseteq t$, such that $|S_i| = \ell$ and $|S_i \cap S_j| \leq k$ for all $i < j < m$.

4.2.7 Lemma *Let $0 < \gamma < 1$. There are constants $\delta > 0$, $c > 1$, and a PV-function d such that*

$$PV \vdash d(x) \text{ is a } \langle \gamma\ell, \ell, c\ell, 2^{\delta\ell}\rangle\text{-design, where } \ell = ||x||.$$

Proof: Put $c = 2/\gamma$, $\delta = c^{-2}$, and let $k = \gamma\ell$, $t = c\ell$, and $m = 2^{\delta\ell}$. The function d will iterate through all subsets $S \subseteq t$, putting S into the design if $|S| = \ell$ and its intersection with all elements of the design so-far constructed is at most k. We have to show that this algorithm will not stop with a design shorter than m. Clearly, it suffices to prove that for any design $\langle S_j \rangle_{j<i}$, $i < m$, there is an $S_i \subseteq t$ such that $\langle S_j \rangle_{j\leq i}$ is also a design.

We will do this by a counting argument (which works directly without any *PHP*, as $m \in Log$ and $k, \ell, t \in LogLog$). However, it turns out that instead of counting subsets $S \subseteq t$, it is easier to count functions $f\colon t \to t$ which represent $S = S(f) := f^{-1''}\ell$ (thus, choosing uniformly a random f means to choose S in such a way that $\Pr(a \in S) = \ell/t$ for all $a < t$).

The number of $f\colon t \to t$ such that $|S(f)| \geq \ell$ is

$$\sum_{i \geq \ell} \binom{t}{i} \ell^i (t-\ell)^{t-i} \geq \varepsilon t^t$$

for some constant $\varepsilon > 0$, by A.4. If S is a subset of t of size ℓ, the number of f such that $|S(f) \cap S| \geq k$ is

$$\begin{aligned}\sum_{i=k}^{\ell} \binom{\ell}{j} \ell^j (t-\ell)^{\ell-j} t^{t-\ell} &= t^t \ell^{-\ell} \sum_{i=k}^{\ell} \binom{\ell}{j} (k/2)^j (\ell - k/2)^{\ell-j} \\ &\leq t^t 4^{-(k/2)^2/\ell} = t^t 2^{-\gamma^2 \ell/2}\end{aligned}$$

by Chernoff's inequality (A.5). The number of f such that $|S(f) \cap S_j| \geq k$ for some $j < i$ is thus at most

$$t^t m 2^{-\gamma^2 \ell/2} = t^t 2^{(\delta - \gamma^2/2)\ell} \leq t^t 2^{-\gamma^2 \ell/4} < \varepsilon t^t,$$

hence there is f such that we may put $S(f)$ into the design. (If $|S(f)| > \ell$, we discard some of its elements.) □

4.2.8 Definition ([31]) Let $x < 2^t$, and $S \subseteq t$, $|S| = \ell$. Let $\{s_i\}_{i<\ell}$ be the increasing enumeration of the set S. Then we put $x \restriction S := y$, where $y < 2^\ell$ and $bit(y, i) = bit(x, s_i)$ for all $i < \ell$.

If $f\colon 2^\ell \to 2$, and $S = \langle S_i \rangle_{i<m}$ is a $\langle k, \ell, t, m\rangle$-design, the *Nisan-Wigderson generator* is a function $NW_{f,S}\colon 2^t \to 2^m$ defined by

$$bit(NW_{f,S}(x), i) = f(x \restriction S_i).$$

Let NW be a PV-function such that $NW(f, S, x) = NW_{f,S}(x)$.

4.2.9 Theorem *There is a PV-function $\pi(f, S, D, a, z)$, such that the following property is provable in S^1_2:*

Let $f\colon 2^\ell \to 2$ be a Boolean function such that $|\{x < 2^\ell;\, C(x) = f(x)\}| \leq 2^{\ell-1} + a$ for any circuit C of size $|C| \leq s$. Let S be a $\langle k, \ell, t, m\rangle$-design, and let $D\colon 2^m \to 2$ be a circuit of size $|D| < s - m2^k$. Put $e = am2^{m+t-\ell}$. Then

$$\begin{aligned}\pi(f, S, D, a, \cdot)\colon e \mathbin{\dot\cup} (2^m \times \{x < 2^t;\, D(NW_{f,S}(x)) = 1\}) &\twoheadrightarrow \\ \twoheadrightarrow\ 2^t \times \{r < 2^m;\, D(r) = 1\}.\end{aligned}$$

4.2.10 Remark The function π witnesses that $\Pr_x(D(NW_{f,S}(x)) = 1) \geq \Pr_r(D(r) = 1) - m\varepsilon$, where $\varepsilon = a2^{-\ell}$.

Proof: We will find (uniformly in $i < m$) surjections

$$G_i \colon a2^{m+t-\ell} \mathbin{\dot\cup} M_{i+1} \twoheadrightarrow M_i,$$

where $M_i = \{\langle \vec{r}, x\rangle;\ D(f(x \restriction S_0), \ldots, f(x \restriction S_{i-1}), r_i, \ldots, r_{m-1}) = 1\}$. Notice that $M_0 = \{\vec{r};\ D(\vec{r}) = 1\} \times 2^t$, and $M_m = 2^m \times \{x;\ D(NW_{f,S}(x)) = 1\}$.

Fix $i < m$, $y < 2^{t-\ell}$, and $r_{i+1}, \ldots, r_{m-1} < 2$. For any $u < 2^\ell$ and $j < m$ define $f^y_j(u) = f(x \restriction S_j)$, where $x \restriction S_i = u$ and $x \restriction (t \smallsetminus S_i) = y$. Finally put

$$\begin{aligned} A_0(u) &= D(f^y_0(u), \ldots, f^y_{i-1}(u), 0, r_{i+1}, \ldots, r_{m-1}),\\ A_1(u) &= \neg D(f^y_0(u), \ldots, f^y_{i-1}(u), 1, r_{i+1}, \ldots, r_{m-1}). \end{aligned}$$

Each $f^y_j(u)$, $j < i$, depends only on $|S_j \cap S_i| \leq k$ variables, hence it is computable by a circuit of size 2^k. This allows A_0 and A_1 to be represented as circuits of size at most $1 + |D| + i2^k \leq 1 + |D| + m2^k \leq s$, hence

$$|\{u;\ A_r(u) = f(u)\}| \leq 2^{\ell-1} + a, \qquad r = 0, 1.$$

By summing these two inequalities we get

$$\begin{aligned} 2a &\geq |\{u;\ f(u) = A_0(u)\}| + |\{u;\ f(u) = A_1(u)\}| - 2^\ell\\ &= |\{u;\ (A_0(u) \mathbin{\&} \neg(\neg f(u) \mathbin{\&} A_0(u))) \vee (\neg f(u) \mathbin{\&} \neg(\neg f(u) \mathbin{\&} A_0(u)))\}|\\ &\qquad + |\{u;\ (\neg A_1(u) \mathbin{\&} \neg(f(u) \mathbin{\&} \neg A_1(u))) \vee (f(u) \mathbin{\&} \neg(f(u) \mathbin{\&} \neg A_1(u)))\}|\\ &\qquad - 2^\ell\\ &= |\{u;\ A_0(u)\}| - |\{u;\ \neg f(u) \mathbin{\&} A_0(u)\}| + |\{u;\ \neg f(u)\}|\\ &\qquad - |\{u;\ \neg f(u) \mathbin{\&} A_0(u)\}| + |\{u;\ \neg A_1(u)\}| - |\{u;\ f(u) \mathbin{\&} \neg A_1(u)\}|\\ &\qquad + |\{u;\ f(u)\}| - |\{u;\ f(u) \mathbin{\&} \neg A_1(u)\}| - 2^\ell\\ &= |\{u;\ A_0(u)\}| + |\{u;\ \neg A_1(u)\}|\\ &\qquad - 2|\{u;\ (\neg f(u) \mathbin{\&} A_0(u)) \vee (f(u) \mathbin{\&} \neg A_1(u))\}|\\ &= |\{u;\ D(f^y_0(u), \ldots, f^y_{i-1}(u), 0, r_{i+1}, \ldots, r_{m-1})\}|\\ &\qquad + |\{u;\ D(f^y_0(u), \ldots, 1, r_{i+1}, \ldots)\}|\\ &\qquad - 2|\{u;\ D(f^y_0(u), \ldots, f(u), r_{i+1}, \ldots)\}|\\ &= |\{\langle r, u\rangle;\ D(f^y_0(u), \ldots, r, r_{i+1}, \ldots)\}|\\ &\qquad - |\{\langle r, u\rangle;\ D(f^y_0(u), \ldots, f(u), r_{i+1}, \ldots)\}|. \end{aligned}$$

Employing counting functions for the two sets in the last line, we get a surjection

$$\begin{aligned}g_{i,y,r_{i+1},\dots,r_{m-1}} \colon 2a \mathbin{\dot\cup} \{\langle r, u\rangle;\ D(f_0^y(u),\dots,f(u),r_{i+1},\dots,r_{m-1})\} &\twoheadrightarrow\\ \twoheadrightarrow \{\langle r,u\rangle;\ D(f_0^y(u),\dots,r,r_{i+1},\dots,r_{m-1})\}&.\end{aligned}$$

Define $G_i \colon M_{i+1} \mathbin{\dot\cup} a2^{m+t-\ell} \to M_i$ by

$$\begin{aligned}G_i(\vec r, x) = \langle r_0,\dots,r_{i-1},r'_i,r_{i+1},\dots,r_{m-1},x'\rangle,&\\ \text{if } g_{i,y,r_{i+1},\dots,r_{m-1}}(r_i, x\restriction S_i) = \langle r'_i, x'\restriction S_i\rangle,&\\ \text{and } x'\restriction (t\smallsetminus S_i) = y,&\\ G_i(2av+w) = \langle r_0,\dots,r_{i-1},r'_i,r_{i+1},\dots,r_{m-1},x'\rangle,&\\ \text{if } v = \langle y, r_0,\dots,r_{i-1},r_{i+1},\dots,r_{m-1}\rangle,&\\ g_{i,y,r_{i+1},\dots,r_{m-1}}(w) = \langle r'_i, x'\restriction S_i\rangle, \text{ and } x'\restriction(t\smallsetminus S_i) = y.&\end{aligned}$$

It is straightforward to check that the functions G_i are well defined and onto, using $f(x\restriction S_j) = f_j^{x\restriction(t\smallsetminus S_i)}(x\restriction S_i)$.

Now we define π as a composition of $G_0,\dots,G_{m-1}$. More precisely, we put

$$\pi(f,S,D,a,z) = G^m(z),$$

where $G^i \colon M_i \mathbin{\dot\cup} ai2^{m+t-\ell} \to M_0$ is defined inductively by

$$\begin{aligned}G^0(z) &= z,\\ G^{i+1}(z) &= \begin{cases} w - a2^{m+t-\ell}, & a2^{m+t-\ell} \le z < a(i+1)2^{m+t-\ell},\\ G^i(G_i(z)), & \text{otherwise.}\end{cases}\end{aligned}$$

Given $z \in M_0$, we prove by Σ^b_1-*LIND* on $i \le m$ that there is a $w \in M_i \mathbin{\dot\cup} ai2^{m+t-\ell}$ such that $G^i(w) = z$, in particular $\pi\colon M_m \mathbin{\dot\cup} am2^{m+t-\ell} \to M_0$ is onto, as required. □

4.2.11 Theorem *Let F be a MFRP definable in $S^1_2 + dWPHP(PV)$, and let $\varepsilon > 0$. Then there are PV-functions h and g such that $HARD^{\varnothing}_{\varepsilon}$ proves*

$$\begin{aligned}\exists y\ \ y = F(x) \ &\leftrightarrow\ h(x,\alpha(g(x))) \ne *,\\ \exists y\ \ y = F(x) &\to h(x,\alpha(g(x))) = F(x).\end{aligned}$$

Proof: Fix a 1/2-definition of $F(x)$ given by $f(x,w)$, $w < r(x)$. We may assume w.l.o.g. that $r(x) \ge x$. Choose a constant $b \ge 1$ such that for all $n \gg 0$, there is a circuit $C\colon 2^n \times 2^m \to 2$ of size at most m^b such that

$$C(x,w) = 1 \quad\text{iff}\quad f(x,w) \ne *,$$

where $m = |r(x)|$. Choose $\gamma < \varepsilon$, and let c, δ, and d be as in lemma 4.2.7. We may assume $\gamma + \delta < \varepsilon$ and $\delta < \varepsilon/b$, because we may shorten the design produced by d if necessary.

Define $g(x) = 2^{m^{1/\delta}}$, so that $\varphi = \alpha(g(x))$ is a Boolean function on $\ell = |m|/\delta$ variables. Put $t = c\ell$, $k = \gamma\ell$, and let $S = d(g(x))$ (hence S is a $\langle k, \ell, t, m\rangle$-design). Finally, define

$$h(x,\varphi) = \begin{cases} f(x, NW_{\varphi,S}(u)), & \text{if } u \text{ is the smallest } u < 2^t \text{ such that} \\ & \quad f(x, NW_{\varphi,S}(u)) \neq *, \\ *, & \text{if no such } u \text{ exists.} \end{cases}$$

Notice that $2^t = m^{c/\delta} = n^{O(1)}$, so the loop over all $u < 2^t$ may be done by a PV-function (i.e., it is p-time computable).

Clearly $h(x, \alpha(g(x))) = *$ if $F(x)$ does not have a value, and y is a value of $F(x)$ if $y = h(x, \alpha(g(x))) \neq *$. It remains to show that $h(x, \alpha(g(x))) \neq *$ if $F(x)$ is defined.

Put $s = 2^{\varepsilon\ell}$ and $a = 2^{(1-\varepsilon)\ell}$, so that φ satisfies the assumptions of theorem 4.2.9. The size bound on $D(w) := C(x,w)$ is also satisfied: $|D| + m2^k \leq m^b + m^{1+\gamma/\delta} < m^{\varepsilon/\delta} = s$, because $1 + \gamma/\delta < \varepsilon/\delta$ and $b < \varepsilon/\delta$.

Assume that we do not find a suitable $u < 2^t$. This means that

$$\forall u < 2^t \, D(NW_{\varphi,S}(u)) = 0,$$

hence by theorem 4.2.9 the function $\pi(\varphi, S, D, a, \cdot)$ is a surjection from $e = am2^{m+t-\ell}$ to $2^t \times \{w;\ D(w) = 1\}$. On the other hand, f is a $1/2$-definition of F and we assume that $F(x)$ is defined, hence we also have a surjection of 2^m onto $2 \times \{w;\ D(w) = 0\}$. We may modify this function to map 2^{m+t-1} onto $2^t \times \{w;\ D(w) = 0\}$, and combine it with π to get a surjection from $2^{m+t-1} + e$ onto 2^{m+t}.

However, $e = 2^{m+t+(\delta-\varepsilon)\ell} < 2^{m+t-2}$ because $\delta < \varepsilon$, hence we obtain a mapping of $3 \cdot 2^{m+t-2}$ onto $4 \cdot 2^{m+t-2}$. This contradicts $dWPHP(PV)$, which is available in $HARD^{\varnothing}_{\varepsilon}$. □

4.3 Finite fields in bounded arithmetic

Having succeeded in formalizing the Nisan-Wigderson derandomization result, the natural next step is to consider the Impagliazzo-Wigderson theorem [15], which draws the same conclusion assuming only worst-case hardness. The proof of the Impagliazzo-Wigderson result was later simplified by Sudan, Trevisan, and Vadhan [46], who realized the connection between hardness amplification, and list decoding of error-correcting codes; nevertheless, formalization of the theorem in bounded arithmetic turned out much harder than the Nisan-Wigderson construction. The main reason is that the Nisan-

Wigderson generator is based on simple combinatorics, whereas list decoding of error-correcting codes requires several algebraic tools concerning finite fields.

The proof is split in several subsections. The main results are presented in section 4.3.5. The key component, list decoding of Reed-Muller codes, is contained in section 4.3.4. It is built on list decoding of Reed-Solomon codes described in section 4.3.3, which in turn requires Gaussian elimination over function fields (section 4.3.2).

4.3.1 Basic properties of finite fields

This section is an overview of notation and a few elementary results on finite fields. An interested reader may find broader context in [25], or other algebra textbooks.

4.3.1 Definition (PV) A *finite field* is a sequence

$$F = \langle q, 0_F, 1_F, +_F, -_F, \cdot_F, {}^{-1}_F \rangle,$$

where $q \in Log$ represents the interval $[0, q)$, the rest are tables of operations on q of the correct arity, and the usual axioms of fields are satisfied. *Size* of the field F is q. A *univariate polynomial* over F is a sequence f of elements of F, such that $(f)_{lh(f)-1} \neq 0$ if $lh(f) \neq 0$. The number $lh(f) - 1$ is the *degree* of f.

4.3.2 Example If p is a prime, we can construct a field $\mathbb{F}_p$ of size p: $+$, $-$, and $\cdot$ are arithmetical operations modulo p, and ${}^{-1}$ is the modular inverse computed by the extended gcd algorithm.

4.3.3 Lemma $(PV \vdash:)$ *Addition, subtraction, multiplication, division with remainder, and extended* gcd *of polynomials over a finite field are well-defined, and computable by PV-functions.*

Proof: Straightforward. □

4.3.4 Example Let F be a finite field of size q, and f an irreducible polynomial of degree d over F, such that $q^d \in Log$. We can construct a field of size q^d by identifying elements of q^d with polynomials over F of degree less than d, and performing all operations modulo f. The last lemma ensures that the field operations are well-defined, and it is straightforward to check the field axioms.

4.3.5 Lemma $(PV \vdash:)$ *Let $f \in F[x]$ and $a \in F$. Then $f(a) = 0$ iff $x - a \mid f$.*

Proof: Write $f = (x - a)g + b$ using the division algorithm. Then $f(a) = b$. □

4.3.6 Lemma ($PV \vdash$:) *A nonzero polynomial $f \in F[x]$ has at most* $\deg(f)$ *roots.*

Proof: We show by induction on k, that if $\alpha_1, \dots, \alpha_k \in F$ are distinct roots of f, then $\prod_i (x - \alpha_i) \mid f$. The base case is trivial, and the induction step follows from lemma 4.3.5. Since $\prod_i (x - \alpha_i)$ is a polynomial of degree k, we must have $k \leq \deg(f)$ unless $f = 0$. □

4.3.7 Lemma ($PV \vdash$:) *Let F be a field of size q. There exists a prime $p \leq q$ such that F contains $\mathbb{F}_p$.*

Proof: For any integer n, define $\overline{n} \in F$ by $\overline{0} = 0_F$, $\overline{2n} = \overline{n} + \overline{n}$, $\overline{2n+1} = \overline{n} + \overline{n} + 1_F$, and $\overline{-n} = -\overline{n}$. Straightforward induction shows that $\overline{n+m} = \overline{n} + \overline{m}$, and $\overline{nm} = \overline{n}\,\overline{m}$. By $PHP_q^{q+1}(PV)$, there exist $n < m \leq q$ such that $\overline{n} = \overline{m}$, thus $\overline{n-m} = 0$. Let $p \leq q$ be the smallest positive integer such that $\overline{p} = 0$. Since F is an integral domain, p is prime, and it is easy to see that $\overline{\bullet}$ is an isomorphic embedding of $\mathbb{F}_p$ in F. □

4.3.8 Definition (PV) The prime p from the last lemma is called the *characteristic* of F, denoted by $\chi(F)$, and (the subfield of F isomorphic to) $\mathbb{F}_p$ is the *prime field* of F.

4.3.9 Lemma ($PV \vdash$:) *Let F be a field of size q. Then q is a power of $p = \chi(F)$.*

Proof: Fix D such that $p^D \leq q < p^{D+1}$. If s is a sequence of elements of F, and t a sequence of elements of $\mathbb{F}_p$ such that $d := lh(s) = lh(t) \leq D+1$, define $f(s,t) = \sum_{i<d} t_i s_i \in F$ (notice that f can be computed by a PV-function, resp. a circuit). A sequence s is *linearly independent*, if $f(s,t) \neq 0$ for every $t \neq \vec{0}$. Since $f(s, t+t') = f(s,t) + f(s,t')$, the function $f(s,\bullet)\colon \mathbb{F}_p^d \to F$ is injective for a linearly independent s, thus $p^d \leq q$, and $d \leq D$.

The universal quantifier in the definition of independence is sharply bounded, as $p^{D+1} \leq qp \in Log$. We can therefore find a maximal linearly independent s by greedy search. It follows that $f(s,\bullet)$ is a bijection of $\mathbb{F}_p^d$ and F: if $a \in F$, the sequence $s \cup \langle a \rangle$ is linearly dependent, i.e., $\sum_i t_i s_i + ta = 0$ for some nonzero $\langle \vec{t_i}, t \rangle$. Since s is independent, we must have $t \neq 0$, thus $a = -t^{-1} \sum_i t_i s_i = f(s,t')$, where $t'_i = -t^{-1} t_i$. □

4.3.10 Lemma ($PV \vdash$:) *Let F be a field of size q and characteristic p, and let $k \in Log$. Define a mapping $\Phi_{p^k} \colon F \to F$ by $\Phi_{p^k}(a) = a^{p^k}$. Then Φ_{p^k} is an automorphism of F.*

Proof: The property of being an automorphism is sharply bounded since $q \in Log$, thus we can use induction on k; moreover a composition of two automorphisms is an automorphism, therefore it is sufficient to consider the case $k = 1$. Φ_p obviously preserves all field operations except for addition. We have

$$(a+b)^n = \sum_{i \le n} \binom{n}{i} a^i b^{n-i}$$

by induction on $n \in Log$, moreover $p \mid \binom{p}{i}$ for every $0 < i < p$ by induction on i, thus $(a+b)^p = a^p + b^p$. Φ_p is injective because $a^p \neq 0$ if $a \neq 0$, and it is surjective by PHP^q_{q-1}. □

4.3.11 Lemma ($PV \vdash$:) *Let F be a field of size q. Then Φ_q is identity, and the polynomial $x^q - x$ is equal to $\prod_{a \in F}(x-a)$.*

Proof: Fix $a \in F^* = F \smallsetminus \{0\}$, we need to show $a^{q-1} = 1$. By PHP^q_{q-1}, there is an $r < q$ such that $a^r = 1$. Let r be the smallest such number, and put $G = \{a^i; i < r\}$. For each $b \in F^*$, let $f(b)$ be the least element of bG in the underlying ordering of q, and define $g(b) = \langle f(b), (f(b))^{-1}b \rangle$. It is easy to see that g is a bijection of F^* and $\mathrm{rng}(f) \times G$, in particular $r = |G|$ divides $q - 1 = |F^*|$, thus $a^{q-1} = 1$.

Since all elements of F are roots of $x^q - x$, we have $\prod_{a \in F}(x-a) \mid x^q - x$. Moreover, both polynomials are monic of degree q, thus they must be identical. □

4.3.12 Lemma ($PV \vdash$:) *Let F be a field of size q, and $q \le n \in Log$. There exists an extension H of F of size q^d such that $n \le q^d < n^2$. Moreover, we can construct such an H in time polynomial in n.*

Proof: There are q^2 monic polynomials of degree 2 over F, but only $\binom{q}{2} + q = q(q+1)/2$ of them are reducible. We can thus find an irreducible polynomial of degree 2 by an exhaustive search, and such a polynomial gives us an extension of F of size q^2. We iterate this process $\lceil \log_2(\log_q n) \rceil$ times. □

4.3.2 Some linear algebra

In the algorithm for finding roots of bivariate polynomials, we will need to compute gcd's of bivariate polynomials. The obvious approach, namely to use the Euclidean algorithm in the Euclidean domain $F(x)[y]$ does not work, since we do not know how to prove polynomial bounds on degrees of elements of $F(x)$ computed by the algorithm. An alternative approach is to reduce the gcd computation to solving linear systems over $F(x)$.

However, this is also problematic: in order to define Gaussian elimination, we need again to bound the degree of the polynomials constructed during the elimination process, and a simple induction on the length of the computation would only show exponential bounds. The degrees are actually polynomially bounded, because elements constructed during Gaussian elimination are given by determinants of certain minors of the original matrix; however, we cannot use this argument if we want to *define* determinants using Gaussian elimination. The way out of this vicious circle is to define determinants in some other way, to prove some of their properties, and only then proceed to Gaussian elimination. Fortunately, a lot of work in this direction was already done in M. Soltys' PhD thesis [42] (an extract of his thesis appeared in [43] and [44]; we will use [44] as the main reference).

Soltys defines three theories, $LA \subseteq LAP \subseteq \forall LAP$, for reasoning about matrix algebra. Originally, LA was defined as a quantifier-free sequent calculus, but it will be more convenient for us to treat it as a usual first-order theory, as in [49]. LA is a three sorted theory: the individual sorts correspond to elements of a field (or more generally, integral domain), to matrices, and to indices. The language of the theory contains field (or ring) operations, basic arithmetical operations for manipulation with indices, functions which extract the number of rows and columns and individual entries of a matrix, a form of definition by cases, λ-terms for construction of matrices by defining their elements, and a function which sums all entries of a matrix. It has over 30 open axioms (essentially, field (or integral domain) axioms, and defining equations for other function symbols), and the schema of open induction on indices. LAP extends LA by a function symbol for matrix powering. In $\forall LAP$, the induction schema is extended to Π^M_1-formulas: these are formulas of the form $\forall A \leq n\, \varphi$, where φ is open, n is an index term, and $A \leq n$ abbreviates $r(A) \leq n \wedge c(A) \leq n$ (i.e., the number of rows and columns of A is at most n).

LA proves basic ring identities of matrices, but otherwise it seems to be a rather weak theory. LAP is much more interesting: matrix powering enables to define the Berkowitz' algorithm [4] for computing the characteristic polynomial (and thus determinant) by a term of LAP. Some basic properties of the determinant are provable in LAP directly, and many other (e.g., the cofactor expansion formula) are (over LAP) implied by the Cayley-Hamilton theorem. Finally, $\forall LAP$ proves the Cayley-Hamilton theorem.

The language of LAP can be naturally interpreted in the language of arithmetic: we fix a definable field (integral domain) R to interpret the field sort, use logarithmically small number to represent the index sort, and sequences of field elements for the matrix sort. (If we are dealing with an infinite R, it is assumed that we pick its most natural realization: for example, elements of the field $\mathbb{Q}$ of rationals are all pairs of coprime integers, rather than, say, pairs of logarithmically small numbers.)

Soltys shows that S^1_2 proves the natural translation of $\forall LAP$ if we take a finite field as the field sort. Unfortunately for us, this interpretation does not work for infinite rings like $F(x)$ or $\mathbb{Q}$. It is straightforward to interpret LAP in these cases, but not the induction axioms, because Π^M_1-formulas translate into unbounded universal formulas (even quantification over 1×1 matrices requires an unbounded quantifier over the ring elements).

We provide more details here to get the records straight, because there seems to be a confusion on this question in the literature. Soltys and Cook [43, 44] claim without proof that the interpretation of $\forall LAP$ in S^1_2 works over $\mathbb{Q}$, but this is true only for open (or Π^M_1) consequences of $\forall LAP$, not in general (even if we restrict ourselves to sequents of Π^M_1-formulas, which is the original Soltys' formulation of $\forall LAP$). Let $(\forall LAP)^R$ denote S^1_2 extended by the interpretation of theorems of $\forall LAP$ over an (S^1_2-definable) integral domain R. To see that $S^1_2 \nvdash (\forall LAP)^R$ for most infinite rings R, we use the following result of [49]: if $t(u)$ is a ring term, then $\forall LAP$ proves (a Σ^M_1-formula equivalent to)

$$\forall a\, \forall n\, \exists X\, \forall i \leq n\, (X_{1,1} = a \wedge (i > 1 \to X_{1,i} = t(X_{1,i-1}))).$$

If we take $t(u) = u^2$, we get

$$(\forall LAP)^R \vdash \forall a \in R\, \forall n \in \mathit{Log}\, \exists b \in R\, b = a^{2^n}.$$

This means that $(\forall LAP)^R \vdash EXP$, if R is $\mathbb{Q}$, $\mathbb{Z}$ (take $a = 2$), $F[x]$ (take $a = x$), or just about any reasonable integral domain R which contains an element of infinite order.

If we consider $\forall LAP$ as a full first-order theory, we can do much better, at least for $R = \mathbb{Z}$: a block of c universal quantifiers over $\mathbb{Z}$ can be simulated by a quantifier over $1 \times c$ integer matrices, i.e., any universal property of the integers can be expressed by a Π^M_1-formula. Moreover, $I\Delta_0 + EXP \subseteq (\forall LAP)^{\mathbb{Z}}$ proves the MRDP theorem [11], thus any Π^0_1-formula is equivalent to a Π^M_1-formula, and $(\forall LAP)^{\mathbb{Z}} = I\Sigma_1$.

We can remedy the situation by restricting the induction schema.

4.3.13 Definition Let $\Pi^{M,b}_1$ be the class of LAP formulas of the form $\forall A \leq B\, \varphi$, where φ is an open formula, B is a matrix term not containing an occurrence of A, and $A \leq B$ is an abbreviation for

$$r(A) \leq r(B) \wedge c(A) \leq c(B) \wedge \\ \forall i \leq r(A), j \leq c(A)\, \exists k \leq r(B), \ell \leq c(B)\; A_{i,j} = B_{k,\ell}.$$

In words, the number of rows and columns of A is bounded by the respective parameters of B, and each entry of A appears as an entry of B. Let $\forall LAP^-$ be the extension of LAP by induction for $\Pi_1^{M,b}$-formulas. (Notice that $\forall LAP \vdash \forall LAP^-$: $\forall A \leq B\, \varphi$ is equivalent to

$$\forall A \leq \max(r(B), c(B))\, (A \leq B \rightarrow \varphi),$$

and $A \leq B$ is in $\forall LAP$ equivalent to a Σ_1^M-formula by [49].)

4.3.14 Theorem *S_2^1 proves the natural translation of $\forall LAP^-$, if we take $F(x)$ or $\mathbb{Q}$ as the base field, where F is a finite field.*

Proof: It is straightforward to adapt the interpretation of LAP from [42, 44] to this situation. Essentially, the open axioms pose no problems, once we manage to define LAP terms by PV-functions. This can be done by induction on the complexity of the term; for example, matrix powering is given by a PV-function, as the degree of coefficients of A^n is bounded by nd, where d is an upper bound to degrees of coefficients of A (if the matrix consists of polynomials; in general, we first compute the product of all denominators p which has degree at most n^2d, compute $(pA)^n$, and divide the result by p^n).

To see that $\Pi_1^{M,b}$-induction holds, notice that $\Pi_1^{M,b}$-formulas translate into Π_1^b-formulas: if $A \leq B$, then the size of A is bounded by the maximal size of its entries (which is bounded by the size of B), multiplied by the number of its rows and columns (which are also bounded by the size of B). Thus, Π_1^b-*LIND* suffices. □

The next theorem shows that we did not weaken the theory too much—$\forall LAP^-$ proves the most important Soltys' results, including some properties of the determinant which we will need later.

The *Cayley-Hamilton theorem* asserts that if $p_A(x)$ is the characteristic polynomial of a square matrix A, then $p_A(A) = 0$. *Cofactor expansion of determinant* along row i is the identity

$$|A| = \sum_j (-1)^{i+j} A_{i,j} |A[i;j]|,$$

where $|A|$ is the determinant of A, and $A[i_1, \dots, i_k; j_1, \dots, j_\ell]$ denotes the minor of A obtained by deleting rows $i_1, \dots, i_k$ and columns $j_1, \dots, j_\ell$. Cofactor expansion along columns is defined dually. *Multiplicativity of determinant* is the formula $|AB| = |A||B|$. *Axiomatic definition of determinant* consists of the following three conditions:

(i) determinant is multilinear in rows and columns,

(ii) determinant is alternating in rows and columns,

(iii) $|I| = 1$,

where I is the identity matrix.

4.3.15 Theorem *$\forall LAP^-$ proves the Cayley-Hamilton theorem, the cofactor expansion formula, axiomatic properties of determinant, and multiplicativity of determinant.*

Proof: The first three statements are equivalent over LAP by theorem 4.2 of [44]. An inspection of Soltys' proof of C-H in $\forall LAP$ (theorem 5.1 of [44]) reveals that it actually uses $\Pi_1^{M,b}$-induction rather than full Π_1^M-induction: the induction hypothesis is directly or indirectly (through corollary 4.1) applied only to permuted minors of the original matrix. The same argument applies to the proof of multiplicativity of determinant (theorem 5.2 of [44]). □

4.3.16 Lemma ($\forall LAP^- \vdash$:) *For any $n \times n$ matrix A such that $n \geq 2$,*

$$|A||A[n-1, n; n-1, n]| = \\ |A[n-1; n-1]||A[n; n]| - |A[n-1; n]||A[n; n-1]|.$$

Proof: Fix a matrix B, we will show by $\Pi_1^{M,b}$-induction on n that the identity is true for all square matrices $A \leq B$ with at most n rows. The statement holds for matrices of size 2 and 3 by direct computation. Let A be an $(n+1) \times (n+1)$ matrix, decomposed as

$$A = \begin{pmatrix} M & S & Q \\ R & a & b \\ P & c & d \end{pmatrix},$$

and assume the statement is true for all permutations of M. By expansion on the last row, we have

$$\begin{vmatrix} M & Q \\ R & b \end{vmatrix} \begin{vmatrix} M & S \\ P & c \end{vmatrix} = \Big(\sum_j (-1)^j r_j |M[;j]\, Q| + (-1)^n b|M|\Big) \times$$

$$\begin{aligned}
&\Big(\sum_j (-1)^j p_j |M[;j]\,S| + (-1)^n c|M|\Big)\\
&= |M|\Big(bc|M| + \sum_j (-1)^{n+j} cr_j |M[;j]\,Q|\\
&\quad + \sum_j (-1)^{n+j} bp_j |M[;j]\,S|\Big)\\
&\quad + \sum_{j,k} (-1)^{j+k} r_j p_k |M[;j]\,Q||M[;k]\,S|.
\end{aligned}$$

Similarly,

$$\begin{aligned}
\begin{vmatrix} M & S \\ R & a \end{vmatrix}\begin{vmatrix} M & Q \\ P & d \end{vmatrix} &= |M|\Big(ad|M| + \sum_j (-1)^{n+j} dr_j |M[;j]\,S|\\
&\quad + \sum_j (-1)^{n+j} ap_j |M[;j]\,Q|\Big)\\
&\quad + \sum_{j,k} (-1)^{j+k} r_j p_k |M[;j]\,S||M[;k]\,Q|,
\end{aligned}$$

thus

$$\begin{aligned}
&\begin{vmatrix} M & Q \\ R & b \end{vmatrix}\begin{vmatrix} M & S \\ P & c \end{vmatrix} - \begin{vmatrix} M & S \\ R & a \end{vmatrix}\begin{vmatrix} M & Q \\ P & d \end{vmatrix}\\
&= |M|\Big((bc - ad)|M| + \sum_j (-1)^{n+j} (cr_j - ap_j)|M[;j]\,Q|\\
&\qquad + \sum_j (-1)^{n+j} (bp_j - dr_j)|M[;j]\,S|\Big)\\
&\quad + \sum_{j,k} (-1)^{j+k} (r_j p_k - r_k p_j)|M[;j]\,Q||M[;k]\,S|\\
&= |M|\Big((bc - ad)|M| + \sum_j (-1)^{n+j} (cr_j - ap_j)|M[;j]\,Q|\\
&\qquad + \sum_j (-1)^{n+j} (bp_j - dr_j)|M[;j]\,S|\Big)\\
&\quad + \sum_{j<k} (-1)^{j+k} (r_j p_k - r_k p_j)\Big(|M[;j]\,Q||M[;k]\,S| - |M[;j]\,S||M[;k]\,Q|\Big).
\end{aligned}$$

Similarly, expanding $|A|$ along the last two rows yields

$$\begin{vmatrix} M & S & Q \\ R & a & b \\ P & c & d \end{vmatrix} = (ad - bc)|M| + \sum_j (-1)^{n+j} (ap_j - cr_j)|M[;j]\,Q|$$

$$\begin{aligned}&+\sum_{j}(-1)^{n+j}(dr_j-bp_j)|M[;j]\,S|\\&+\sum_{j<k}(-1)^{j+k}(r_kp_j-r_jp_k)|M[;j,k]\,S\,Q|,\end{aligned}$$

thus

$$\begin{aligned}&|M|\begin{vmatrix}M&S&Q\\R&a&b\\P&c&d\end{vmatrix}-\begin{vmatrix}M&S\\R&a\end{vmatrix}\begin{vmatrix}M&Q\\P&d\end{vmatrix}+\begin{vmatrix}M&Q\\R&b\end{vmatrix}\begin{vmatrix}M&S\\P&c\end{vmatrix}=\\&\quad\sum_{j<k}(-1)^{j+k}(r_kp_j-r_jp_k)\Big(|M||M[;j,k]\,S\,Q|-|M[;j]\,Q||M[;k]\,S|\\&\qquad\qquad+|M[;j]\,S||M[;k]\,Q|\Big).\end{aligned}$$

It suffices to show that

$$|M||M[;j,k]\,S\,Q|-|M[;j]\,Q||M[;k]\,S|+|M[;j]\,S||M[;k]\,Q|=0$$

for any $j<k$. Using expansion on columns, a similar computation as above shows

$$\begin{aligned}&|M[;j]\,S||M[;k]\,Q|-|M[;j]\,Q||M[;k]\,S|\\&\qquad=\sum_{i<\ell}(-1)^{i+\ell}(s_iq_\ell-s_\ell q_i)\Big(|M[i;j]||M[\ell;k]|-|M[i;k]||M[\ell;j]|\Big)\end{aligned}$$

and

$$|M[;j,k]\,S\,Q|=\sum_{i<\ell}(-1)^{i+\ell}(s_\ell q_i-s_iq_\ell)|M[i,\ell;j,k]|,$$

thus

$$\begin{aligned}&|M||M[;j,k]\,S\,Q|-|M[;j]\,Q||M[;k]\,S|+|M[;j]\,S||M[;k]\,Q|=\\&\quad\sum_{i<\ell}(-1)^{i+\ell}(s_\ell q_i-s_iq_\ell)\Big(|M||M[i,\ell;j,k]|-|M[i;j]||M[\ell;k]|\\&\qquad\qquad+|M[i;k]||M[\ell;j]|\Big).\end{aligned}$$

However,

$$|M||M[i,\ell;j,k]|-|M[i;j]||M[\ell;k]|+|M[i;k]||M[\ell;j]|=0$$

for every $i<\ell$: we move the ith and ℓth row to the bottom of M, do similarly for the columns (the net change to the expression is a multiplicative factor of $(-1)^{i+j+k+\ell}$, by alternation in rows and columns), and use the induction hypothesis. □

Now we have everything we need for Gaussian elimination. Armed with lemma 4.3.16, we could proceed to define the textbook Gaussian elimination algorithm over $F(x)$ in S^1_2, and to prove that it is a p-time algorithm (more precisely, it would go the other way around: polynomial bounds have to be already built-in to get a PV-function, and we would use lemma 4.3.16 to prove the *soundness* of the algorithm).

We will take a different approach, and formalize Gaussian elimination in $\forall LAP^-$. The benefit is that it automatically extends to other S^1_2-definable fields like $\mathbb{Q}$; in any case, the result is more elegant, as it stresses the field-agnostic nature of Gaussian elimination. The drawback is that we cannot directly define the elimination *algorithm* in $\forall LAP^-$, we have to present the result as an existential statement. This is inessential, as we will obtain a PV-function for free from Buss' witnessing theorem.

4.3.17 Definition ($\forall LAP^-$) We let *elementary matrices* be square matrices which result from the identity matrix by putting a nonzero element c on position $\langle i, j\rangle$ in the matrix, or by swapping the ith and jth row for some i, j. We denote the first type of elementary matrices by $E_{i,j}(c)$, and the second type by $E'_{i,j}$. A *reduced row-echelon* matrix is a matrix A with the following properties:

(i) The left-most nonzero entry of every nonzero row is 1. This entry is called the *pivot*.

(ii) If $i < j$ and row j is nonzero, then the pivot of row i is to the left of the pivot of row j (and, in particular, row i is nonzero).

(iii) Entries above any pivot are 0.

A is a *row-echelon* matrix if it satisfies (i) and (ii).

4.3.18 Theorem ($\forall LAP^- \vdash$:) *For every $n \times m$ matrix A there exists a sequence of elementary $n \times n$ matrices $E_1, \dots, E_s$ such that $E_1 \cdots E_s A$ is a reduced row-echelon matrix.*

Proof: By a standard argument, $\Pi^{M,b}_1$-induction is equivalent to induction for $\Sigma^{M,b}_1$-formulas. We observe several closure properties of $\Sigma^{M,b}_1$-formulas: a quantifier over a sequence $\{M(\ell); \ell < k\}$ of $n \times m$ matrices can be simulated by a single quantifier over an $n \times km$ matrix M by putting $M(\ell)_{i,j} := M_{i,\ell m+j}$. In particular, we can merge together a block of bounded quantifiers (if their bounds are independent). Using methods of [49], open formulas in LAP are closed under bounded index quantifiers (the characteristic function of any open formula is definable by a term of LAP, and a bounded universal index quantifier can then be simulated by a product).

For any matrix B, let $B^{j_1,\dots,j_\ell}_{i_1,\dots,i_k}$ be the $k \times \ell$ minor of B consisting of rows $i_1,\dots,i_k$ and columns $j_1,\dots,j_\ell$.

Claim 1 *There exists a sequence of elementary $n \times n$ matrices $E_1,\dots,E_s$ such that the product $E_1 \cdots E_s A$ is a row-echelon matrix.*

Proof: We will show the following statement by induction on $k \le n$, the claim follows if we take $k = n$:

> There exists a sequence $C(1),\dots,C(k) \le A$ of $n \times m$ matrices, a sequence $F(1),\dots,F(k) \le I$ of $n \times n$ matrices, and an $n \times m$ matrix $B \le A$ such that
>
> - $\forall i \le k\, \exists j \le n\; F(i) = E'_{i,j}$,
> - the first k rows of M are in row-echelon form,
> - if c is the pivot column of row k, $i > k$, and $j \le m$, then
>
> $$M_{i,j} = \frac{|B^{1\dots k,j}_{1\dots k,i}|}{|B^{1\dots k}_{1\dots k}|}$$
>
> for $j > c$, and $M_{i,j} = 0$ for $j \le c$,
>
> where M is the $n \times m$ matrix defined as
>
> $$\prod_{\ell=k}^{1}\left(\prod_{i=\ell+1}^{n}\left(E_{i,\ell}\Big(-\frac{|C(\ell)^{1\dots\ell}_{1\dots\ell-1,i}|}{|C(\ell)^{1\dots\ell-1}_{1\dots\ell-1}|}\Big)\,E_{\ell,\ell}\Big(\frac{|C(\ell)^{1\dots\ell-1}_{1\dots\ell-1}|}{|C(\ell)^{1\dots\ell}_{1\dots\ell}|}\Big)\right) \times F(\ell)\right) \times A.$$

Notice that the bulleted conditions contain only bounded index quantifiers, thus the whole thing is a $\Sigma^{M,b}_1$-formula.

The base case $k = 0$ is trivial, we just take $B = A$. Assume the statement is true for $k - 1$, we will demonstrate it for k. Let c be the pivot column of row $k - 1$. Find the least $d > c$ such that $M_{r,d} \neq 0$ for some $r \ge k$, and fix one such r. (If there is no suitable d, then M is already in row-echelon form, and the claim is true.) Put $F(k) = E'_{k,r}$, $N = F(k)M$, and $D = F(k)B$ (i.e., N and D result from M and B by swapping rows k and r).

Notice that $N_{k,d} \neq 0$, we may thus construct a matrix P from N as follows: first divide row k by $N_{k,d}$, then subtract $N_{i,d}$ times row k from row i for every $i = k + 1,\dots,n$. Clearly the first k rows of P are in row-echelon form. Also $P_{i,j} = 0$ for $i > k$ and $j \le d$, and for $j > d$ we have

$$P_{i,j} = N_{i,j} - \frac{N_{i,d}N_{k,j}}{N_{k,d}} = \frac{|D^{1\dots k-1,j}_{1\dots k-1,i}|}{|D^{1\dots k-1}_{1\dots k-1}|} - \frac{|D^{1\dots k-1,d}_{1\dots k-1,i}||D^{1\dots k-1,j}_{1\dots k}|}{|D^{1\dots k-1}_{1\dots k-1}||D^{1\dots k-1,d}_{1\dots k}|}.$$

By lemma 4.3.16, we have

$$|D^{1\dots k-1,d}_{1\dots k}||D^{1\dots k-1,j}_{1\dots k-1,i}| - |D^{1\dots k-1,d}_{1\dots k-1,i}||D^{1\dots k-1,j}_{1\dots k}| = |D^{1\dots k-1}_{1\dots k-1}||D^{1\dots k-1,d,j}_{1\dots k,i}|,$$

thus

$$P_{i,j} = \frac{|D^{1\dots k-1}_{1\dots k-1}||D^{1\dots k-1,d,j}_{1\dots k,i}|}{|D^{1\dots k-1}_{1\dots k-1}||D^{1\dots k-1,d}_{1\dots k}|} = \frac{|D^{1\dots k-1,d,j}_{1\dots k,i}|}{|D^{1\dots k-1,d}_{1\dots k}|},$$

which has the required form, if we take D with column d moved to column k as the new B. Also

$$\begin{aligned} P &= \prod_{i=k+1}^{n} \left(E_{i,k}(-N_{i,d})\, E_{k,k}(N^{-1}_{k,d})\right) \times N \\ &= \prod_{i=k+1}^{n} \left(E_{i,k}(-N_{i,d})\, E_{k,k}(N^{-1}_{k,d})\right) \times F(k)\, M, \end{aligned}$$

and we have

$$N^{-1}_{k,d} = \frac{|D^{1\dots k-1}_{1\dots k-1}|}{|D^{1\dots k-1,d}_{1\dots k}|},$$

and

$$-N_{i,d} = \frac{|D^{1\dots k-1,d}_{1\dots k-1,i}|}{|D^{1\dots k-1}_{1\dots k-1}|}.$$

It suffices to put $C(k) = D^{1\dots k-1,d}_{1\dots n}$. □ (claim 1)

Claim 2 *For any row-echelon matrix M, there exists a sequence of elementary $n \times n$ matrices $E_1, \dots, E_s$ such that the product $E_1 \cdots E_s M$ is a reduced row-echelon matrix.*

Proof: Let k be the last nonzero row of M, and let c_ℓ be the pivot column in row ℓ, for $\ell = 1, \dots, k$. We cannot define c_ℓ directly as a term in ℓ, but the following will be sufficient: the formula $M_{\ell,c} = 1 \wedge \forall j < c\, M_{\ell,j} = 0$ expressing that c is the pivot column of row ℓ contains only a bounded index quantifier, therefore it has a characteristic function $t(c, \ell)$. If we let T be the matrix such that $T_{j,\ell} = t(j, \ell)$, then $(MT)_{i,\ell} = M_{i,c_\ell}$. We claim that

$$\prod_{\ell=k}^{1} \prod_{i=1}^{\ell-1} E_{i,\ell}(-M_{i,c_\ell}) \times M$$

is a reduced row-echelon matrix. To see this, we will prove by open induction on $h = k, \dots, 1$ that

$$N(h) = \prod_{\ell=k}^{h} \prod_{i=1}^{\ell-1} E_{i,\ell}(-M_{i,c_\ell}) \times M$$

is a row-echelon matrix with pivots in same columns as M, $N(h)$ agrees with M in all columns $j < c_h$, and has zeros above pivots of rows $h, \dots, k$.

Assume that the statement is true for $h+1$. The effect of

$$N(h) = \prod_{i=1}^{h-1} E_{i,h}(-M_{i,c_h}) \times N(h+1)$$

is to subtract M_{i,c_h} times row h from row i, for every $i < h$. By assumption, $M_{i,c_h} = N(h+1)_{i,c_h}$, thus the result is that entries above the pivot c_h are cleared. Columns $j < c_h$ are unchanged, and entries above pivots c_ℓ for $\ell > h$ remain zero, because $N(h+1)_{h,c_\ell} = 0$. Thus the statement is true for $N(h)$, and the claim holds. □ (claim 2)

The theorem follows by applying claim 1 and claim 2. □

4.3.19 Theorem ($PV \vdash$:) *There is a PV-function which computes a solution to a system of linear equations over $F(x)$ if one exists, and a PV-function which computes a basis for the space of all solutions of a homogeneous linear system over $F(x)$, where F is a finite field.*

Proof: By theorems 4.3.18 and 4.3.14, and Buss' witnessing theorem, there is a PV-function $GE(F, A)$, which computes a sequence of elementary matrices $E_1, \dots, E_s$ such that $E_1 \cdots E_s A$ is a reduced row-echelon matrix. Let $Ax = c$ be a linear system. We use GE to find $E_1, \dots, E_s$, and put $B = E_1 \cdots E_s A$. We compute $d = E_1 \cdots E_s c$, and solve the system $Bx = d$. (Let X be the set of columns of B which do not contain a pivot. It is easy to see that $Bx = d$ is solvable iff $d_i = 0$ for every zero row i of B, and the solutions are given by choosing arbitrary x_j for $j \in X$, and computing $x_\ell = d_\ell - \sum_{j \in X} B_{k,j} x_j$ for $\ell \notin X$, where k is the row with pivot in column ℓ.) Clearly any solution to $Ax = c$ is a solution to $Bx = d$. Conversely, elementary matrices are trivial to invert, and thus if $Bx = d$, we have $Ax = E_s^{-1} \cdots E_1^{-1} Bx = E_s^{-1} \cdots E_1^{-1} d = c$. □

As mentioned in the introduction to this section, the reason for our interest in linear algebra over function fields was to obtain a gcd algorithm for bivariate polynomials. The exact statement we want is theorem 4.3.23, and we finish the present section by its proof.

4.3.20 Lemma ($PV \vdash$:) *There is a PV-function which, given a finite field F and polynomials $f, g \in F(x)[y]$, computes $h, a, b \in F(x)[y]$ such that $h = af + bg$, and $h \mid f, g$.*

Proof: Taking care of trivial cases, we may assume $\deg(f), \deg(g) > 0$. Put $d_f = \deg(f)$ and $d_g = \deg(g)$, and consider the system of equations

$$\deg(a) < d_g, \quad \deg(b) < d_f,$$
$$\deg(af + bg) = d,$$
$$af + bg \text{ is monic}$$

for a fixed $d \le \min(d_f, d_g)$. This is a linear system with $d_f + d_g$ unknowns for coefficients of a and b, and $d_f + d_g - d$ equations stating that the dth coefficient of $h := af + bg$ is 1, and the ith coefficients are 0 for $i = d + 1, \dots, d_f + d_g - 1$. The system is clearly solvable for $d = \min(d_f, d_g)$, and using theorem 4.3.19, we can find the minimal d such that the system is solvable, and compute the solutions a, b, and h.

We claim that $h = \gcd(f, g)$, i.e., $h \mid f$ and $h \mid g$. Using the division algorithm, write $f = uh + v$, $\deg(v) < \deg(h)$. We have $v = (1 - ua)f - ubg$. Moreover, we can write $v = a'f + b'g$ with $\deg(a') < \deg(g)$, $\deg(b') < \deg(f)$: if we put $1 - ua = wg + a'$, $\deg(a') < \deg(g)$, we have $v = a'f + (wf - ub)g$. Putting $b' = wf - ub$, we have $\deg(b'g) = \deg(v - a'f) < \deg(f) + \deg(g)$, thus $\deg(b') < \deg(f)$. We can make v monic by dividing it by its leading coefficient, if $v \neq 0$. Then v, a', b' form another solution to the linear system with $\deg(v) < \deg(h)$, thus $v = 0$ by minimality of h, which means that $h \mid f$. We can show $h \mid g$ in the same way. □

4.3.21 Definition (PV) Let $f \in F[x, y]$, and write

$$f(x, y) = \sum_{i \le d} f_i(x) y^i.$$

The *content* of f is $\operatorname{cont}(f) := \gcd(f_0, \dots, f_d)$.

4.3.22 Lemma (Gauss' lemma) $(PV \vdash{:})$ *Let $f, g \in F[x, y]$, $h \in F(x)[y]$ be such that $f = gh$, and $\operatorname{cont}(g) = 1$. Then $h \in F[x, y]$.*

Proof: Write $h = h'/c$, where $h' \in F[x, y]$, $c \in F[x]$ is nonzero, and $\gcd(c, \operatorname{cont}(h')) = 1$. Then $cf = gh'$. Assume that $\deg_x(c) > 0$ for contradiction. We may assume $S^1_2(PV)$ w.l.o.g. because we are proving a Σ^b_1-statement, thus we can find a nonconstant divisor $p \mid c$ of the smallest possible degree; it follows that $p \in F[x]$ is irreducible, and thus prime (by extended gcd in $F[x]$). Since c is coprime with $\operatorname{cont}(h')$, and $\operatorname{cont}(g) = 1$, we can find the smallest i and j such that p does not divide g_i and h'_j. The $(i + j)$th coefficient of $cf = gh'$ is

$$\sum_{k \le i+j} g_k h'_{i+j-k},$$

and every term except for $g_i h'_j$ in this sum is divisible by p, but $p \nmid g_i h'_j$, thus $(gh')_{i+j}$ is not divisible by p. However, $(cf)_{i+j} = cf_{i+j}$ is divisible by p, a contradiction. □

4.3.23 Theorem $(PV \vdash:)$ *There is a PV-function which, given a finite field F and polynomials $f, g \in F[x,y]$, computes $h, a, b \in F[x,y]$ and $c \in F[x]$ such that $ch = af + bg$, $h \mid f, g$, and $c \neq 0$.*

Proof: By lemma 4.3.20 and clearing denominators, compute polynomials $h, a, b \in F[x,y]$ and nonzero $c \in F[x]$ such that

$$ch = af + bg,$$
$$h \mid f, g \text{ in } F(x)[y].$$

Compute $d = \operatorname{cont}(h)$, divide h by d, and multiply c by d. Since d is invertible in $F(x)$, we still have $h \mid f, g$ in $F(x)[y]$. Moreover now $\operatorname{cont}(h) = 1$, thus $h \mid f, g$ in $F[x,y]$ by Gauss' lemma. □

4.3.3 List decoding of Reed-Solomon codes

Our next step is a list decoding algorithm for Reed-Solomon codes, which will be used in the next section as a subprocedure of a list decoder for Reed-Muller codes. We will not define the actual codes in bounded arithmetic, to avoid the machinery of coding theory which we would have to introduce. We just remind the reader briefly here, to get an idea of what is going on.

Reed-Solomon code [38] with alphabet size q, message length k, and codeword length n works as follows: we identify the alphabet with a finite field $F = GF(q)$, and messages $p \in F^k$ with polynomials of degree less than k. The codeword $C(p)$ is then the evaluation of p at n fixed points $\alpha_1, \dots, \alpha_n \in F$. List decoding is the task of finding all messages p, such that $C(p)$ has a good agreement with a given $f \in F^n$. In the case of Reed-Solomon codes, this can be rephrased as follows: given a function $f : X \to F$, where $X \subseteq F$, find the list of polynomials $p \in F[x]$ of a given degree, which agree with f on at least t elements of X. This is formalized in theorem 4.3.34.

The list decoding algorithm we use was described by Sudan [45]. Its most involved component is a factoring algorithm for bivariate polynomials. We do not know whether it is possible to formalize general bivariate factoring[1] in S^1_2, however it will be sufficient to construct a root finding algorithm,

[1] Univariate factoring is the real problem; the reduction from bivariate to univariate factoring using Hensel's lifting is relatively straightforward.

which is much easier (theorem 4.3.32). Another simplification is that we can allow the algorithm to be polynomial in the field size, rather than its logarithm; this makes univariate root finding particularly trivial.

The idea of finding roots of a bivariate polynomial $f(x,y)$ is to find roots of the univariate polynomial $f(0,y)$, or equivalently, to find roots of f modulo x, and then to lift them to roots modulo x^{2^k} for sufficiently large k. This process, called Hensel's lifting, can be viewed as approximating a root of f in $F((x))$ (the field of formal Laurent series over F) by Newton's iteration.

4.3.24 Definition (PV) If $f \in F[x,y]$, we let $f'(x,y)$ denote the partial derivative of f wrt y, i.e., if $f = \sum_i f_i y^i$, where $f_i \in F[x]$, then $f'(x,y) := \sum_{i>0} i f_i y^{i-1}$.

4.3.25 Lemma ($PV \vdash$:) *Let $f \in F[x,y]$ and $p \in F[x]$. Then $h(x,y) = \frac{f(x,y)-f(x,p(x))}{y-p(x)}$ is a polynomial, and $h(x,p(x)) = f'(x,p(x))$.*

Proof: Since $(f(x,y) - f(x,p(x)))(p(x)) = 0$, we have $y - p(x) \mid f(x,y) - f(x,p(x))$. Deriving the equation $f - f(x,p) = (y-p)h$ yields

$$f' = h + (y-p)h',$$

thus $f'(x,p) = h(x,p)$. □

4.3.26 Lemma (Hensel's lifting) ($PV \vdash$:) *Let $0 < k \in Log$, $f \in F[x,y]$, and $p, b \in F[x]$ satisfy*

(*i*) $f(x,p(x)) \equiv 0 \pmod{x^k}$,

(*ii*) $b(x)f'(x,p(x)) \equiv 1 \pmod{x^k}$.

Let

$$\begin{aligned} p^*(x) &\equiv p(x) - b(x)f(x,p(x)) \pmod{x^{2k}}, \\ b^*(x) &\equiv (2 - b(x)f'(x,p^*(x)))b(x) \pmod{x^{2k}}. \end{aligned}$$

Then

(*i*) $p(x) \equiv p^*(x) \pmod{x^k}$,

(*ii*) $f(x,p^*(x)) \equiv 0 \pmod{x^{2k}}$,

(*iii*) $b^*(x)f'(x,p^*(x)) \equiv 1 \pmod{x^{2k}}$.

Moreover, conditions (i) and (ii) determine p^* uniquely modulo x^{2k}.

Proof: Clearly $f(x,p(x))\equiv 0 \pmod{x^k}$ implies $p\equiv p^* \pmod{x^k}$. Put

$$q = bf'(x,p^*)-1.$$

Then $b^*\equiv (1-q)b \pmod{x^{2k}}$, and $q\equiv bf'(x,p)-1\equiv 0 \pmod{x^k}$, thus

$$\begin{aligned}b^*f'(x,p^*)\equiv (1-q)bf'(x,p^*) &=\\ &=(1-q)(1+q)=1-q^2\equiv 1 \pmod{x^{2k}}.\end{aligned}$$

Let $h=(f-f(x,p))/(y-p)$. We have

$$\begin{aligned}f(x,p^*)&=f(x,p)+(p^*-p)h(x,p^*)\\ &\equiv f(x,p)(1-bh(x,p^*)) \pmod{x^{2k}}.\end{aligned}$$

Also $f(x,p)\equiv 0 \pmod{x^k}$ and

$$1-bh(x,p^*)\equiv 1-bh(x,p)=1-bf'(x,p)\equiv 0 \pmod{x^k},$$

thus $f(x,p^*)\equiv 0 \pmod{x^{2k}}$.

Assume that $\tilde p\equiv p \pmod{x^k}$ and $f(x,\tilde p)\equiv 0 \pmod{x^{2k}}$. Then $h(x,\tilde p)\equiv h(x,p)=f'(x,p) \pmod{x^k}$, thus $bh(x,\tilde p)\equiv 1 \pmod{x^k}$, and $\tilde p - p\equiv 0 \pmod{x^k}$ implies

$$\tilde p - p\equiv(\tilde p-p)h(x,\tilde p)b=(f(x,\tilde p)-f(x,p))b\equiv -bf(x,p) \pmod{x^{2k}},$$

i.e., $\tilde p\equiv p-bf(x,p)\equiv p^* \pmod{x^{2k}}$. □

4.3.27 Definition (PV) Let $Lift(f,F,S)$ be the formalization of the following algorithm.

```
**input:** $F$ finite field, $f\in F[x,y]$, $S$ set of roots of $f(0,y)$
**output:** set of roots of $f$
**algorithm:**
  $R\leftarrow\emptyset$
  $k\leftarrow|\deg_x(f)|$
  for each $\alpha\in S$ do:
    $p(x)\leftarrow\alpha$
    $b(x)\leftarrow(f'(0,\alpha))^{-1}$, or reject if $f'(0,\alpha)=0$
    for $j\leftarrow 1,\dots,k$ do:
      $p(x)\leftarrow(p(x)-b(x)f'(x,p(x)))\bmod x^{2^j}$
      $b(x)\leftarrow(b(x)(2-b(x)f'(x,p(x))))\bmod x^{2^j}$
    if $f(x,p(x))=0$ then $R\leftarrow R\cup\{p\}$
  output $R$
```

4.3.28 Lemma ($PV \vdash$:) *Let F be a finite field, $f \in F[x,y]$, and S is the set of all roots of $f(0,y)$ in F. If all of these roots are simple, then $Lift(f,F,S)$ computes the set of all roots of f.*

Proof: Clearly all polynomials output by $Lift$ are roots of f. Let $q(x)$ be any root of f. Then $\alpha := q(0)$ is a root of $f(0,y)$, and $f'(0,\alpha) \neq 0$ because α is simple, thus the lifting stage of the algorithm computes a polynomial p of degree less than 2^k such that $p(0) = \alpha$ and $f(x,p(x)) \equiv 0 \pmod{x^{2^k}}$. By uniqueness of Hensel's lifting, we have $p \equiv q \pmod{x^{2^k}}$. Since $y - q(x) \mid f(x,y)$, the degree of q is at most $\deg_x(f) < 2^k$, thus $p = q$, and the output of the algorithm includes q. □

We have to deal with polynomials f such that $f(0,y)$ has multiple roots. The idea is to split f into square-free factors using gcd with its derivative. Then, for square-free f, we have a good chance of finding $\alpha \in F$ such that $f(\alpha,y)$ is also square-free.

4.3.29 Definition (PV) Let *Split*(f,F) be a formalization of the following recursive algorithm.

```
input: $F$ finite field of size $q$, $f \in F[x,y]$
output: sequence of pairs $\langle f_i, d_i \rangle$, $f_i \in F[x,y]$, $d_i \leq |\deg_y(f)|$
algorithm:
  $h \leftarrow \gcd(f,f')$ as in theorem 4.3.23
  if $\deg_y(h) = 0$ then output $\langle\langle f, 0 \rangle\rangle$
  if $\deg_y(h) < \deg_y(f)$ then output $Split(h,F) \cup Split(f/h,F)$
  write $f = \sum_{i \leq d} a_i y^i$, $a_i \in F[x]$
  let $p$ be the smallest number such that $q$ is a power of $p$
  $g \leftarrow \sum_{i \leq d/p} a_{pi} y^i$
  $S \leftarrow Split(g,F)$
  output $\langle\langle f_i, d_i + 1 \rangle; i < lh(S), (S)_i = \langle f_i, d_i \rangle\rangle$
```

In order to make sure that the algorithm is p-time and PV-definable, we should avoid the recursion, and use a loop instead: the algorithm would maintain a list of partial factors which need to be split, and iteratively reduce them as above. The number of iterations would be explicitly bounded by $\deg_y(f)$; it will be clear from the next lemma that this bound does not change the semantics of the algorithm.

4.3.30 Lemma ($PV \vdash$:) *Let F be a finite field of characteristic p, and $f \in F[x,y]$ a nonzero polynomial. Then Split(f,F) computes a sequence of* *pairs* $\langle f_i, d_i\rangle$, $i < k$, *such that*

$$f(x,y) = \prod_{i<k} f_i(x, y^{p^{d_i}}),$$

and $\deg_y(\gcd(f_i, f_i')) = 0$ *for each* $i < k$.

Proof: By induction on the length of the computation. If $\deg_y(h) = 0$, the algorithm is obviously correct. If $\deg_y(h) < \deg_y(f)$, the correctness follows from $f = (f/h)h$, and the induction hypothesis. (Notice that $\deg_y(f/h) < \deg_y(h)$, since $\deg_y(h) > 0$.)

Assume that $\deg_y(h) = \deg_y(f)$. Since $h \mid f'$ and $\deg_y(f') < \deg_y(f)$, we must have $f' = 0$. Notice that p computed by the algorithm is equal to $\chi(F)$, since $\chi(F)$ is prime, and q is a power of $\chi(F)$, by lemma 4.3.7. We have

$$f'(x,y) = \sum_{i\le d} i a_i y^{i-1},$$

thus $f' = 0$ implies that $a_i = 0$ for every i not divisible by p. This means that $f(x,y) = g(x, y^p)$, and correctness of the algorithm follows by the induction hypothesis. □

4.3.31 Definition (PV) Let $BRoots(f, F)$ be the following algorithm.

```
input: $F$ finite field of size $q$, $f \in F[x,y]$
output: set of all roots of $f$
algorithm:
  $R \leftarrow \emptyset$
  $\langle\langle f_i, d_i\rangle;\ i < m\rangle \leftarrow Split(f, F)$
  for each $i < m$ do:
    compute $a, b \in F[x,y]$, $0 \ne c \in F[x]$ such that
            $c = af_i + bf_i'$ by theorem 4.3.23
    $d \leftarrow \deg(c)$
    if $d \ge q$ then:
      let $K$ be an extension of $F$ of size $q' > d$, $q' \le d^2$,
              by lemma 4.3.12
    else $K \leftarrow F$
    search the first $d+1$ elements of $K$ to find $\alpha \in K$ s.t. $c(\alpha) \ne 0$
    compute the set $S$ of all roots of $f_i(\alpha, y)$ in $K$ by brute force search
    $P \leftarrow Lift(f_i(x+\alpha, y), K, S)$
    $P \leftarrow \{r(x-\alpha);\ r \in P\}$
    if $F \ne K$, exclude from $P$ polynomials with coefficients
            outside of $F$
    $p \leftarrow \chi(F)$, $s \leftarrow p^{d_i}$, write $q = p^t$
    $k \leftarrow -d_i \bmod t$
    for $r \in P$ do:
        if $r$ has a nonzero coefficient $r_j$ s.t. $s \nmid j$, skip to the next $r$
        $R \leftarrow R \cup \{\sum_j r_{js}^{p^k} x^j\}$
output $R$
```

4.3.32 Theorem ($PV \vdash$:) *Let F be a finite field, and $f \in F[x,y]$ a nonzero polynomial. Then $BRoots(f,F)$ computes the set of all roots of f.*

Proof: By soundness of *Split*, the first step in the main loop makes sense. The search for an α which is not a root of c succeeds, because c cannot have more roots than its degree, as in lemma 4.3.6. Then $c(\alpha) = a(\alpha,y)f_i(\alpha,y) + b(\alpha,y)f_i'(\alpha,y) = a(\alpha,y)f_i(\alpha,y) + b(\alpha,y)(f_i(\alpha,y))'$ implies that $\gcd(f_i(\alpha,y),(f_i(\alpha,y))') = 1$, hence all roots of $f_i(\alpha,y)$ are simple (i.e., $f_i(\alpha,y)$ and its derivation have no common roots). Then it follows from the soundness of *Lift* that P is the set of all roots of f_i.

Observe that if $k \equiv -d \pmod t$, and $a \in F$, then $(a^{p^k})^{p^d} = a^{p^{k+d}} = a$, since $a^q = a$ by lemma 4.3.11. Moreover, $(\sum_i g_i)^p = \sum_i g_i^p$ for any polynomials g_i, as in lemma 4.3.10; thus, if r is a polynomial whose only nonzero coefficients are at positions divisible by $s = p^d$, then $r' = \sum_j r_{js}^{p^k} x^j$ satisfies $(r')^s = r$.

Thus, the output of the algorithm contains only polynomials r' such that $(r')^{p_i^d}$ is a root of f_i. But $f_i(x, y^{p^{d_i}}) \mid f$, hence $f(x,r') = 0$.

Conversely, let r' be a root of f. Then r' is a root of some $f_i(x, y^{p^{d_i}})$, thus $(r')^{p^{d_i}}$ is a root of f_i, and it is included in the set P computed in the ith iteration of the algorithm. This means that R includes a polynomial r'' such that $(r'')^{p^{d_i}} = (r')^{p^{d_i}}$, and it is easy to see from lemma 4.3.10 that such a polynomial is unique, i.e., $r' = r''$. □

4.3.33 Definition (**PV**) The $\langle u,v\rangle$-*weighted degree* of a monomial x^iy^j is $ui + vj$. The $\langle u,v\rangle$-weighted degree of a bivariate polynomial f is the maximum of the $\langle u,v\rangle$-weighted degrees of monomials appearing in f with nonzero coefficients.

4.3.34 Theorem ($PV \vdash$:) *There is a PV-functions which, given a finite field F, a table of a partial function $\{\langle x_i,y_i\rangle;\ i<n\} \subseteq F \times F$, and parameters d, and $t \geq 1 + \sqrt{2dn}$, computes the list of all polynomials $p \in F[x]$ of degree at most d satisfying $|\{i;\ p(x_i) = y_i\}| \geq t$.*

Proof: The function will formalize the following algorithm: find a nonzero $f \in F[x,y]$ of $\langle 1,d\rangle$-weighted degree at most $t-1$ such that $\forall i < n\, f(x_i,y_i) =$

0 by solving a homogeneous linear system over F, compute the roots p of f using $BRoots$, and output those p of degree at most d which satisfy $|\{i;\, p(x_i) = y_i\}| \geq t$.

If p is a polynomial of degree at most d such that $|\{i;\, p(x_i) = y_i\}| \geq t$, then $f(x, p(x))$ is a univariate polynomial of degree at most $t-1$ which has at least t zeros, thus $f(x, p(x)) = 0$.

It thus suffices to show that the linear system has a nonzero solution. The system has n equations, and the number of variables is equal to the number of monomials of $\langle 1, d\rangle$-weighted degree at most $t-1$, which is

$$\sum_{j \leq \left\lfloor \frac{t-1}{d} \right\rfloor} (t-1-jd) = \frac{t-1+((t-1) \bmod d)}{2} \left(\left\lfloor \tfrac{t-1}{d} \right\rfloor + 1\right)$$
$$\geq \frac{t-1}{2} \left(\left\lfloor \tfrac{t-1}{d} \right\rfloor + 1\right) > \frac{(t-1)^2}{2d} \geq n,$$

since $t - 1 \geq \sqrt{2dn}$. Thus the system has more variables than equations, which implies it has a nontrivial solution. □

4.3.35 Corollary ($PV \vdash$:) *The output list in theorem 4.3.34 has at most $\sqrt{2n/d}$ elements.*

Proof: By the proof of the theorem, each solution $p(x)$ satisfies $y - p(x) \mid f(x, y)$, thus the number ℓ of solutions is at most $\deg_y(f) \leq (t-1)/d$. Clearly, ℓ decreases if t increases, hence the worst case is $t = 1 + \sqrt{2dn}$, which gives $\ell \leq \sqrt{2n/d}$. □

4.3.4 List decoding of Reed-Muller codes

The heart of hardness amplification is an efficient list decoding procedure for Reed-Muller codes, due to Sudan, Trevisan, and Vadhan [46]. Again, we do not formally introduce the codes as such; Reed-Muller code [30, 37] is a generalization of Reed-Solomon codes to multivariate polynomials: we interpret the message as a table of a function $z \colon H^m \to F$ for some $H \subseteq F$, we extend it to a low-degree polynomial $p \in F[x_1, \dots, x_m]$, and output the table of p on the whole space F^m. Thus, list decoding of a Reed-Muller code amounts to finding low-degree polynomials p which have a prescribed agreement with a given function $f \colon F^m \to F$.

An extra complication is that we need the algorithm to run in time smaller than q^m. Therefore, both input and output is represented implicitly: the algorithm has oracle access to the function f, and outputs oracle circuits $C^f(x)$ which can evaluate p on a given point $x \in F^m$.

In this setting, the only parameter of the decoding algorithm which is significant for hardness amplification is the size of the circuits C. We thus will not formalize the decoding algorithm itself, but only prove that a small C exists for a given f and p. The exact statement is theorem 4.3.43.

We start with a few observations about logarithmically small probability spaces.

4.3.36 Definition (PV) A *probability space* P on $n \in Log$ is a sequence of nonnegative rationals $\{p_i;\ i < n\}$ such that $\sum_i p_i = 1$. A *random variable* X on P is a sequence $\{x_i;\ i < n\}$ of rationals. We define $\operatorname{E} X = \sum_i p_i x_i$, and $\operatorname{var} X = \operatorname{E}(X - \operatorname{E} X)^2$. A *random event* is a subset $X \subseteq P$. We identify events with random variables taking values 0 and 1, and define accordingly $\Pr(X) = \operatorname{E} X$. Two variables X, Y on P are *independent* if $\Pr(X = a \wedge Y = b) = \Pr(X = a) \Pr(Y = b)$ for every rational a, b.

Observe that we can restrict the universal quantifiers in the definition of independence to $a \in \operatorname{rng}(X)$ and $b \in \operatorname{rng}(Y)$. The range of a sequence is an encoded set computable by a PV-function, thus independence is definable by an open PV-formula. Also notice that

$$\operatorname{E} X = \sum_i x_i p_i = \sum_a \sum_{x_i = a} a p_i = \sum_a a \Pr(X = a),$$

where the sum here and below is taken over $a \in \operatorname{rng}(X)$.

4.3.37 Lemma (Markov's inequality) ($PV \vdash$:) *If X is a nonnegative random variable, and $a > 0$, then*

$$\Pr(X \geq a) \leq \frac{\operatorname{E} X}{a}.$$

Proof: $a \Pr(X \geq a) = a \sum_{x_i \geq a} p_i \leq \sum_{x_i \geq a} x_i p_i \leq \sum_i x_i p_i = \operatorname{E} X$. □

4.3.38 Lemma (Chebyshev's inequality) ($PV \vdash$:) *If X is a random variable, and $a > 0$, then*

$$\Pr(|X - \operatorname{E} X| \geq a) \leq \frac{\operatorname{var} X}{a^2}.$$

Proof: $\Pr(|X - \operatorname{E} X| \geq a) = \Pr((X - \operatorname{E} X)^2 \geq a^2)$. □

4.3.39 Lemma ($PV \vdash$:) *If X and Y are independent random variables, then $\operatorname{E} XY = \operatorname{E} X \operatorname{E} Y$.*

Proof:

$$\begin{aligned}\mathrm{E}\,X\,\mathrm{E}\,Y &= \Big(\sum_a a\Pr(X=a)\Big)\Big(\sum_b b\Pr(Y=b)\Big)\\ &= \sum_{a,b} ab\Pr(X=a)\Pr(Y=b)\\ &= \sum_{a,b} ab\Pr(X=a\wedge Y=b)\\ &= \sum_c c\sum_{ab=c}\Pr(X=a\wedge Y=b)\\ &= \sum_c c\Pr(XY=c)=\mathrm{E}\,XY.\end{aligned}$$

□

4.3.40 Lemma $(PV\vdash:)$ *If $X_1,\dots,X_m$ is a sequence of pairwise independent random variables, then* $\operatorname{var}\sum_i X_i=\sum_i\operatorname{var}X_i$.

Proof: Let $X=\sum_i X_i$, $Y_i=X_i-\mathrm{E}\,X_i$, and $Y=\sum_i Y_i$. Then $\mathrm{E}\,Y_i=\mathrm{E}\,Y=0$, $\operatorname{var}Y_i=\operatorname{var}X_i$, and $\operatorname{var}Y=\operatorname{var}X$. If $i\neq j$, we have

$$\mathrm{E}\,Y_iY_j=\mathrm{E}(X_i-\mathrm{E}\,X_i)(X_j-\mathrm{E}\,X_j)=\mathrm{E}\,X_iX_j-\mathrm{E}\,X_i\,\mathrm{E}\,X_j=0,$$

thus $\operatorname{var}Y=\mathrm{E}(\sum_i Y_i)^2=\mathrm{E}\sum_{i,j}Y_iY_j=\sum_i\mathrm{E}\,Y_i^2+\sum_{i\neq j}\mathrm{E}\,Y_iY_j=\sum_i\operatorname{var}Y_i$. □

The list decoding procedure splits in two steps. The first step is, given a function f and a polynomial p with nonnegligible agreement with f, to find a circuit which approximates p on most of the inputs. The second step is to construct a (randomized) circuit which computes p everywhere, given f and p with agreement close to 1. These two steps correspond to lemmas 4.3.41 and 4.3.42 below.

The idea is to restrict f and p to a random line in F^m, and solve the resulting univariate polynomial reconstruction problem, which is just Reed-Solomon decoding.

4.3.41 Lemma $(PV\vdash:)$ *Let F be a field of size q, $q^m\in Log$, $f\colon F^m\to F$, and $p\in F[x_1,\dots,x_m]$ of total degree d. If*

$$\Pr_{z\in F^m}(p(z)=f(z))>\sqrt{8d/q},$$

then there is an oracle circuit $C\colon F^m\to F$ of size $poly(q,m)$ such that $\Pr_{x\in F^m}(p(x)=C^f(x))\geq 1-\sqrt{8d/q}$.

Proof: Put $\varepsilon = \sqrt{8d/q}$. For any $z, x \in F^m$, define $\ell_{z,x}(t) = (1-t)z + tx$, and $L_{z,x} = \{\ell_{z,x}(t);\, t \in F\}$. Notice that for fixed z, x and uniformly random $t \in F$, $\ell_{z,x}(t)$ is a uniformly random element of $L_{z,x}$, and for uniformly random $x, z \in F^m$ the random variables $X_t = \ell_{z,x}(t)$ are pairwise independent uniformly distributed elements of F^m (for any $t \neq t'$, and $a, a' \in F^m$, there exists a unique pair $\langle z, x\rangle$ such that $(1-t)z+xt = a$, and $(1-t')z+xt' = a'$). Define

$$E(g,z,x) :\equiv g \in F[x] \wedge \deg(g) \leq d \wedge \Pr_t(g(t) = f(\ell_{z,x}(t))) \geq \varepsilon/2.$$

For any $z \in F^m$ and $a \in F$, let $C^f_{z,a}\colon F^m \to F$ be the circuit performing the following computation on a given $x \in F^m$:

- if $x = z$, output a
- ask the oracle for values of f on $L_{z,x}$
- compute the list $g_1, \dots, g_\ell$ of all g_i s.t. $E(g_i, z, x)$ by Reed-Solomon list decoding
- if $\exists! i\, g_i(0) = a$, then output $g_i(1)$

Claim 1 $\Pr_{z,x}(\Pr_t(p(\ell_{z,x}(t)) = f(\ell_{z,x}(t))) \leq \varepsilon/2) < \sqrt{2/(dq)}$.

Proof: Assume for simplicity that $\Pr_{z\in F^m}(p(z) = f(z)) = \varepsilon$. Let X_t be the indicator random variable for the predicate $p(\ell_{z,x}(t)) = f(\ell_{z,x}(t))$, and $X = \sum_i X_i = |\{t;\, p(\ell_{z,x}(t)) = f(\ell_{z,x}(t))\}|$. We have $\operatorname{E} X = \sum_t \operatorname{E} X_t = q\varepsilon$, and $\operatorname{var} X = \sum_t \operatorname{var} X_t = q\varepsilon(1-\varepsilon)$ by pairwise independence, thus

$$\Pr_{z,x}(X \leq q\varepsilon/2) \leq \frac{q\varepsilon(1-\varepsilon)}{(q\varepsilon/2)^2} < \frac{4}{q\varepsilon} = \sqrt{\frac{2}{dq}}$$

by Chebyshev's inequality. □ (claim 1)

Claim 2 $\Pr_{z,x}(z \neq x \wedge \exists g\,(E(g,z,x) \wedge g(0) = p(z) \wedge g \neq p \circ \ell_{z,x})) \leq \sqrt{2d/q}$.

Proof: The relation $\langle z, x\rangle \sim \langle z_0, x_0\rangle$ defined by $L_{z,x} = L_{z_0,x_0}$ is an equivalence, it thus suffices to show that for any fixed $z_0 \neq x_0$,

$$\alpha := \Pr_{z,x}(\exists g\,(E(g,z,x) \wedge g(0) = p(z) \wedge g \neq p \circ \ell_{z,x}) \mid \langle z, x\rangle \sim \langle z_0, x_0\rangle) \\ \leq \sqrt{2d/q}.$$

Notice that the equivalence class of $\langle z_0, x_0\rangle$ is uniquely parametrized by

$$\{\langle \ell_{z_0,x_0}(t), \ell_{z_0,x_0}(u)\rangle;\, t, u \in F, t \neq u\},$$

thus

$$\begin{aligned}\alpha &= \Pr_{t\neq u}(\exists g\, E(g, \ell_{z_0,x_0}(t), \ell_{z_0,x_0}(u)) \wedge g(0) = p(\ell_{z_0,x_0}(t)) \\ &\qquad \wedge g \neq p \circ \ell_{\ell_{z_0,x_0}(t),\ell_{z_0,x_0}(u)}) \\ &= \Pr_{t\neq u}(\exists g\, E(g, z_0, x_0) \wedge g(t) = p(\ell_{z_0,x_0}(t)) \wedge g \neq p \circ \ell_{z_0,x_0}) \\ &= \Pr_{t}(\exists g\, E(g, z_0, x_0) \wedge g(t) = p(\ell_{z_0,x_0}(t)) \wedge g \neq p \circ \ell_{z_0,x_0}),\end{aligned}$$

since a linear substitution preserves the degree of a polynomial. For any given $g \neq p \circ \ell_{z_0,x_0}$, $\Pr_t(g(t) = p(\ell_{z_0,x_0}(t))) \leq d/q$, since both g and $p \circ \ell_{z_0,x_0}$ are polynomials of degree at most d. The number ℓ of all g such that $E(g, z_0, x_0)$ is at most $\sqrt{2q/d}$ by corollary 4.3.35, thus $\alpha \leq \ell d/q = \sqrt{2d/q}$.

□ (claim 2)

Claim 3 $\Pr_{z,x}(C^f_{z,p(z)}(x) = p(x)) \geq 1 - \sqrt{8d/q}$.

Proof: Since $p \circ \ell_{z,x}$ is a polynomial of degree at most d, claim 1 implies that $g_i = p \circ \ell_{z,x}$ for some i whp. Such a g_i satisfies $g_i(0) = p(z)$, and whp is unique with this property by claim 2, thus $C^f_{z,p(z)}(x) = g_i(1) = p(x)$ with probability at least $1 - (\sqrt{2/(dq)} + \sqrt{2d/q}) \geq 1 - 2\sqrt{2d/q} = 1 - \sqrt{8d/q}$.

□ (claim 3)

To finish the proof of the lemma, notice that $\Pr_{z,x}(C^f_{z,p(z)}(x) = p(x)) = \mathrm{E}_z \Pr_x(C^f_{z,p(z)}(x) = p(x)) \leq \max_z \Pr_x(C^f_{z,p(z)}(x) = p(x))$, thus there exists z such that $C^f_{z,p(z)}$ has the required properties. □

4.3.42 Lemma ($PV \vdash$:) *Let F be a field of size q, $q^m \in Log$, $f\colon F^m \to F$, $p \in F[x_1, \dots, x_m]$ of total degree $d < q/8$. If*

$$\Pr_{z\in F^m}(p(z) \neq f(z)) \leq \delta,$$

then there is an oracle circuit $C\colon F^m \times F^m \to F$ of size $poly(q, m)$ such that for every $x \in F^m$, $\Pr_a(p(x) \neq C^f(x, a)) < 3\delta$.

Proof: Put $q' = q - 1$, and define $C(x, a)$ as follows:

- ask the oracle for values of $f(\ell_{x,a}(t))$, $t \in F^*$
- compute (if it exists) g of degree $\leq d$ such that

  $$|\{t \in F^*;\, g(t) = f(\ell_{x,a}(t))\}| > (q' + d)/2$$

  by Reed-Solomon decoding
- output $g(0)$

Notice that g is unique if it exists, because distinct degree d polynomials can agree on at most d points. Also notice that we can use the algorithm from theorem 4.3.34, since $\sqrt{2dq} < q/2 < (q+d)/2$. Thus C is correct if $p \circ \ell_{x,a}$ has sufficient agreement with $f \circ \ell_{x,a}$. The expected number of $t \in F^*$ such that $p(\ell_{x,a}(t)) \neq f(\ell_{x,a}(t))$ is $q'\delta$, since for a fixed t, elements $\ell_{x,a}(t)$ are uniformly distributed in F^m. By Markov's inequality, we have

$$\Pr_a(|\{t;\, p(\ell_{x,a}(t)) \neq f(\ell_{x,a}(t))\}| \geq (q'-d)/2) \leq \frac{2q'\delta}{q'-d} = \frac{2\delta}{1-d/q'} \leq \frac{16\delta}{7}.$$

□

4.3.43 Theorem ($PV \vdash$:) *Let F be a field of size q, $q^{m^2} \in Log$, $f\colon F^m \to F$, and $p \in F[x_1, \dots, x_m]$ of total degree $d \leq \sqrt{q}$. If*

$$\Pr_{z \in F^m}(p(z) = f(z)) > \sqrt{8d/q},$$

then there is an oracle circuit $C\colon F^m \to F$ of size $poly(q,m)$ such that $C^f(x) = p(x)$ for every $x \in F^m$.

Proof: Let $D^f(x,a)$ be the circuit given by lemma 4.3.42, into which we substitute the circuit from lemma 4.3.41 as its oracle. For any $x \in F^m$, we have

$$\eta := \Pr_a(D^f(x,a) \neq p(x)) < 3\sqrt{8d/q} < 9q^{-1/4}.$$

Put $k := 10m$, and let $C(x,\vec{a})\colon F^m \times F^{mk} \to F$ be the circuit which computes $D(x,a_i)$ for $i < k$, and outputs the majority answer.

Claim 1 *For any $x \in F^m$, $\Pr_{\vec{a}}(C^f(x,\vec{a}) \neq p(x)) \leq (2\sqrt{\eta})^k$.*

Proof: Notice that $q^{mk} \in Log$, we may thus count $\vec{a} \in F^{mk}$ directly. Clearly $C^f(x,\vec{a}) = p(x)$ if $D^f(x,a_i) = p(x)$ holds for more than $k/2$ of i's. The number of a such that $D^f(x,a) \neq p(x)$ is ηq^m, thus the number of $\vec{a}$ such that $C^f(x,\vec{a}) \neq p(x)$ is at most

$$\sum_{i \leq k/2} \binom{k}{i} (q^m - \eta q^m)^i (\eta q^m)^{k-i} = q^{mk} \sum_{i \leq k/2} \binom{k}{i} (1-\eta)^i \eta^{k-i} = q^{mk} \sum_{i \leq k/2} c_i,$$

where $c_i := \binom{k}{i}(1-\eta)^i \eta^{k-i}$. We have $c_{i-1} = c_i \eta i/((k-i+1)(1-\eta)) \leq c_i \eta/(1-\eta)$, thus $c_i \leq c_{k/2}(\eta/(1-\eta))^{k/2-i}$ by induction on i, and

$$\sum_{i \leq k/2} c_i \leq c_{k/2} \sum_{i \leq k/2} \left(\frac{\eta}{1-\eta}\right)^i < c_{k/2} \frac{1}{1-\eta/(1-\eta)} = c_{k/2} \frac{1-\eta}{1-2\eta},$$

thus

$$\sum_{i\le k/2} c_i \le \binom{k}{k/2}(1-\eta)^{k/2}\eta^{k/2}\frac{1-\eta}{1-2\eta} < 2^k\eta^{k/2}. \qquad \square \text{ (claim 1)}$$

It follows that

$$\Pr_{\vec{a}}(C^f(x,\vec{a}) \ne p(x)) \le (6q^{-1/8})^{10m} \le (q^{-1/9})^{10m} = (q^m)^{-10/9},$$

thus

$$\Pr_{\vec{a}}(\exists x\, C^f(x,\vec{a}) \ne p(x)) \le (q^m)^{-1/9} < 1,$$

and we can fix $\vec{a} \in F^m$ such that $C'(x) := C(x,\vec{a})$ computes $p(x)$ correctly on all inputs. □

4.3.5 Hardness amplification

The idea of hardness amplification from [46] is as follows: if f is a hard Boolean function, we encode its truth table by an error-correcting code to get a table of a new function $g := C(f)$. If C admits efficient implicit list decoding, then g will be hard to approximate: a small circuit approximating g can be error-corrected to get a small circuit computing f, contradicting its assumed hardness. Also C is computable in time polynomial in its input size, i.e., exponential in the input size of f, thus a uniform exponential-time family of hard Boolean functions is transformed into a uniform family of functions hard on average.

Reed-Muller codes from the previous section have almost all the desired parameters, except that we have to concatenate them with Hadamard codes to make the alphabet binary. Properties of the concatenated codes are summarized in theorem 4.3.46, and the actual hardness amplification is formalized in theorem 4.3.47.

We do not need a special list decoding procedure for Hadamard codes, because a simple brute-force search suffices due to the size of the parameters involved. However, we do need a Johnson-like bound on the number of codewords, which is stated in the next lemma.

4.3.44 Definition (PV) Let $q = 2^k \in Log$. *Hadamard function* $H_k\colon 2^k \to 2^q$ is given by $H_k(x) = \lambda y\, \langle x, y\rangle$, where $\langle,\rangle$ is the inner product modulo 2. *Hamming distance* of $u, v \in 2^q$ is $\Delta(u,v) := |\{x < q;\ u_x \ne v_x\}|$, and *relative Hamming distance* is $\delta(u,v) := \frac{1}{q}\Delta(u,v)$.

4.3.45 Lemma ($PV \vdash$:) *Let $q = 2^k \in Log$, $w \in 2^q$, and $\varepsilon > 0$. Then*

$$\sum_{x \in 2^k} (1 - 2\delta(w, H_k(x)))^2 = 1,$$

therefore $|\{x;\ \delta(w, H_k(x)) \le \frac{1}{2}(1 - \varepsilon)\}| \le 1/\varepsilon^2$.

Proof: We embed 2^q into $\mathbb{Z}^q$ by replacing $i \in \{0, 1\}$ with $(-1)^i \in \{1, -1\}$. In the sequel, vectors of length k are considered elements of $(\mathbb{Z}_2)^k$, i.e., additions and inner products are evaluated modulo 2, whereas vectors of length q are understood as elements of $\mathbb{Z}^q$. For any $x \in 2^k$, we let $H_x = H_k(x) \in \{-1, 1\}^q$.

Clearly, $\langle w, w \rangle = q$ for any $w \in \{-1, 1\}^q$, in particular $\langle H_x, H_x \rangle = q$. If $x \neq y$, we have

$$\langle H_x, H_y \rangle = \sum_z (-1)^{\langle x,z \rangle} (-1)^{\langle y,z \rangle} = \sum_z (-1)^{\langle x,z \rangle + \langle y,z \rangle} = \sum_z (-1)^{\langle x+y,z \rangle}.$$

Choose $i < k$ such that $x_i \neq y_i$, and let α be the ith elementary vector in 2^k. Since $z \mapsto z + \alpha$ is a permutation of 2^k, we have

$$\begin{aligned} 2\langle H_x, H_y \rangle &= \sum_z (-1)^{\langle x+y,z \rangle} + \sum_z (-1)^{\langle x+y,z+\alpha \rangle} \\ &= \sum_z (-1)^{\langle x+y,z \rangle} (1 + (-1)^{\langle x+y,\alpha \rangle}) = 0, \end{aligned}$$

thus $\langle H_x, H_y \rangle = 0$.

Claim 1 *$\sum_x H_x \langle w, H_x \rangle = qw$ for any $w \in \mathbb{Z}^q$.*

Proof: Let e_y be the yth elementary vector in $\mathbb{Z}^q$. We have

$$\sum_x H_x \langle w, H_x \rangle = \sum_{x,y} H_x w_y \langle e_y, H_x \rangle = \sum_{x,y} H_x w_y (-1)^{\langle x,y \rangle},$$

thus

$$\begin{aligned} \sum_x H_x \langle w, H_x \rangle &= \sum_{x,y,z} e_z w_y (-1)^{\langle x,z \rangle} (-1)^{\langle x,y \rangle} \\ &= \sum_{y,z} e_z w_y \langle H_z, H_y \rangle = \sum_z e_z w_z q = qw. \end{aligned}$$

□ (claim 1)

For any $w \in \{-1,1\}^q$ we have

$$\begin{aligned}\langle w, H_x\rangle &= \sum_y w_y(-1)^{\langle x,y\rangle}\\ &= |\{y;\, w_y = (-1)^{\langle x,y\rangle}\}| - |\{y;\, w_y \neq (-1)^{\langle x,y\rangle}\}|\\ &= q - 2\Delta(w, H_x),\end{aligned}$$

therefore

$$q^2 = \langle w, qw\rangle = \langle w, \sum_x \langle w, H_x\rangle\rangle = \sum_x \langle w, H_x\rangle^2 = \sum_x (q - 2\Delta(w,H_x))^2.$$

This means

$$\begin{aligned}1 &= \sum_x (1 - 2\delta(w,H_x))^2\\ &\geq \sum_{\delta(w,H_x)\leq\frac12(1-\varepsilon)} (1-2\delta(w,H_x))^2\\ &\geq \varepsilon^2|\{x;\, \delta(w,H_x) \leq \tfrac12(1-\varepsilon)\}|,\end{aligned}$$

thus $|\{x;\, \delta(w,H_x) \leq \frac12(1-\varepsilon)\}| \leq 1/\varepsilon^2$. □

4.3.46 Theorem *There is a PV-function $C(k,e,z)$, a constant c, and a polynomial $n(k,e)$, such that PV proves the following statement.*

Assume $k, e \in Log$, $|k| \leq e$, and $k^{|k|/|e|} \in Log$. Put $C_{k,e}(z) := C(k,e,z)$ for $z \in 2^k$. Then $C_{k,e}\colon 2^k \to 2^{n(k,e)}$, and for every $y \in 2^n$ and $x \in 2^k$ such that $\delta(y, C_{k,e}(z)) \leq 1/2 - 1/e$, there exists an oracle circuit $D^y\colon k \to 2$ of size e^c such that $D^y(i) = z_i$ for every $i < k$.

Proof: The function C will work as follows. Given k and e, put $d := e^3$, and $m := |k|/|d|$. Construct a field F of size $q \geq d^3|k|^2$ by lemma 4.3.12, fix a subset $H \subseteq F$ of size d, and an injection $b\colon k \to H^m$. Put $t := |q|$, $n' := q^m$, $n := 2^t n'$, and identify F with a subset of 2^t. Given a $z \in 2^k$, construct (by interpolation) a polynomial $p \in F[x_1, \dots, x_m]$ of degree less than d in each variable, such that $p(b(i)) = z_i$ for every $i < k$, and output the sequence $\lambda u \in F^m\, H_t(p(u))$.

Notice that $|n'| = m|q| = O(m|d|) = O(|k|)$, thus $n = O(qn') = poly(e,k)$. Using this bound, it is straightforward to see that C is p-time computable.

Assume that $\delta(y, C_{k,e}(z)) \leq 1/2 - \varepsilon$, where $\varepsilon = 1/e$. Split y into blocks y_u of length 2^t, indexed by $u \in F^m$. We have

$$\tfrac12 - \varepsilon \leq \Pr_{u\in F^m, a<2^t}(y_{u,a} \neq (C_{k,e}(z))_{u,a}) = \mathrm{E}_u \Pr_a(y_{u,a} \neq (H_t(p(u)))_a),$$

thus

$$\Pr_u(\Pr_a(y_{u,a} \neq (H_t(p(u)))_a) \geq \tfrac{1}{2}(1-\varepsilon)) \leq \frac{1-2\varepsilon}{1-\varepsilon} \leq 1-\varepsilon$$

by Markov's inequality, i.e., $\Pr_u(\delta(y_u, H_t(p(u))) < \frac{1}{2}(1-\varepsilon)) \geq \varepsilon$. For any $u \in F^m$, let $r_{u,0}, \dots, r_{u,\ell}$ be the sequence of all $r \in F$ such that $\delta(y_u, H_t(r)) < \frac{1}{2}(1-\varepsilon)$, ordered lexicographically. By lemma 4.3.45, $\ell \leq e^2$. We have

$$\varepsilon \leq \Pr_u(\exists j \leq \ell\, r_{u,j} = p(u)) \leq \sum_{j \leq \ell} \Pr_u(r_{u,j} = p(u)),$$

thus we can fix $j \leq \ell$ such that $\Pr_u(r_{u,j} = p(u)) \geq \varepsilon/\ell \geq \varepsilon^3 = 1/d$. Define a function $f\colon F^m \to F$ as $f(u) := r_{u,j}$. Notice that f can be implemented as a circuit of size $O(q\ell) = \mathit{poly}(e)$ with oracle access to y, using a simple brute-force search. If we plug this circuit in the circuit given by theorem 4.3.43, we obtain an oracle circuit $C^y\colon F^m \to F$ of size $\mathit{poly}(e,q,m) = \mathit{poly}(e)$ such that $C^y(u) = p(u)$ for every $u \in F^m$, then it suffices to put $D^y(i) := C^y(b(i))$.

However, we have to check that assumptions of theorem 4.3.43 are satisfied. We already know that $q^m = \mathit{poly}(k)$, and $m = |k|/|d| = O(|k|/|e|)$, thus $q^{m^2} = (q^m)^m = k^{O(m)} = k^{O(|k|/|e|)} \in \mathit{Log}$. Also $md \leq d|k| \leq \sqrt{q/d} \leq \sqrt{q}$, and finally

$$\Pr_u(f(u) = p(u)) \geq 1/d > \sqrt{8md/q}$$

follows from $8md^3 \leq 8d^3|k| < d^3|k|^2 \leq q$. □

Recall the definition of the theories HARD and $\mathit{HARD}^{\varnothing}$ from section 4.2.

4.3.47 Theorem *For every $\varepsilon > 0$, there exist constants $\delta > 0$ and $d > 0$, and a PV-function A, such that*

$$PV \vdash \mathit{Hard}_\varepsilon(f) \to ||A(f)|| = d||f|| \,\&\, \mathit{Hard}^{\varnothing}_\delta(A(f)).$$

Proof: Let c and $n(k,e)$ be as in theorem 4.3.47, let c' be a constant such that $n(k,e) \leq (ke)^{c'}$, and put $\eta = \varepsilon/(c+2)$. If $f\colon 2^\ell \to 2$, define $A(f) = C_{k,e}(f)\colon 2^{c'(1+\eta)\ell} \to 2$, where $k = 2^\ell$ and $e = 2^{\eta\ell}$. Assume that S is a circuit of size e which computes $A(f)$ correctly for at least a fraction $1/2 + 1/e$ of the inputs. We have $e \leq k \in \mathit{Log}$, $|k| = \ell \leq e$, and $k^{|k|/|e|} = k^{1/\eta} \in \mathit{Log}$, thus we can apply theorem 4.3.46 to get an oracle circuit D of size e^c which computes f. We can substitute S for the oracle, and obtain a Boolean circuit of size $e^{c+1} = 2^{(c+1)\eta\ell} < 2^{\varepsilon\ell}$ which computes f, a contradiction. Thus $A(f)$ is δ-hard on average, where $\delta = \eta/d$, and $d = c'(1+\eta)$. □

4.3.48 Corollary *For every $\varepsilon > 0$ there exists $\delta > 0$, and an interpretation of $HARD_{\delta}^{\varnothing}$ in $HARD_{\varepsilon}$, which leaves L_{PV} absolute.* □

The following corollary is a strengthening of theorem 4.2.11.

4.3.49 Corollary *Let F be a MFRP definable in $S_2^1 + dWPHP(PV)$, and let $\varepsilon > 0$. Then there are PV-functions h and g such that $HARD_{\varepsilon}$ proves*

$$\exists y \;\; y = F(x) \;\; \leftrightarrow \;\; h(x, \alpha(g(x))) \neq *,$$
$$\exists y \;\; y = F(x) \rightarrow h(x, \alpha(g(x))) = F(x).$$

□

Chapter 5

A propositional proof system associated with $dWPHP(PV)$

In this section, we will present a propositional proof system *WF* which corresponds to the theory $S^1_2 + dWPHP(PV)$, i.e., *WF* is the strongest proof system whose consistency is provable in $S^1_2 + dWPHP(PV)$, and tautologies resulting from translation of $\forall\Pi^b_1$-consequences of $S^1_2 + dWPHP(PV)$ have polynomial-size proofs in *WF*. Obviously, such a system has to contain Extended Frege; we could indeed formulate *WF* as an extension of *EF*, but it will be more convenient to use a variant of *EF* which manipulates Boolean circuits instead of formulas, to get rid of *EF*'s extension axioms. We will describe this variant in the first section[1].

5.1 Circuit Frege

5.1.1 Definition Any Boolean circuit C can be "unfolded" into a unique (possibly huge) formula φ_C. Circuits C and D are *similar*, written as $C \simeq D$, if φ_C and φ_D are the same formulas.

5.1.2 Lemma *Similarity of circuits is polynomial-time decidable.*

Proof: As $NLOG \subseteq P$, it suffices to show $\simeq\ \in coNLOG$, which is clearly accomplished by the following algorithm:

```
$c \leftarrow$ output node of $C$,   $d \leftarrow$ output node of $D$
loop
  $\ell_c \leftarrow$ label of $c$,   $\ell_d \leftarrow$ label of $d$   $\{$connective or variable$\}$
```

[1] Although it is folklore that *EF* is essentially "a Frege system operating with circuits", we were unable to find a reference making this explicit.

```
        if $\ell_c \neq \ell_d$ then REJECT
        if $\ell_c$ is a variable or a constant then ACCEPT
        non-deterministically choose $i$ smaller than the arity of $\ell_c$
        $c \leftarrow i^{\text{th}}$ input of $c$,    $d \leftarrow i^{\text{th}}$ input of $d$
    end loop
```

□

5.1.3 Definition A CF (*circuit Frege*) proof system is defined as follows: choose a finite basis $\mathcal{B}$ of Boolean connectives, and a finite, sound, and implicationally complete set $\mathcal{R}$ of Frege rules over $\mathcal{B}$. A CF-proof of a circuit A is a sequence of $\mathcal{B}$-circuits $A_0, \dots, A_k = A$, such that for every $i \leq k$, either there are $j_1, \dots, j_\ell < i$ such that

$$\frac{A_{j_1} \cdots A_{j_\ell}}{A_i}$$

is an instance of a rule $R \in \mathcal{R}$, or there is $j < i$ such that $A_j \simeq A_i$. (Lemma 5.1.2 ensures that CF indeed fulfills the definition of a propositional proof system. Also, when we work with CF in bounded arithmetic, we cannot use definition 5.1.1 directly as it involves exponentially large objects, we thus use the algorithm from lemma 5.1.2 instead.)

5.1.4 Lemma *Any CF system p-simulates any EF system.*

Proof: All EF systems simulate each other, hence we may assume w.l.o.g. that both proof systems use the same set of connectives and Frege rules. Let $\pi\colon \varphi_0, \dots, \varphi_k$ be an EF-proof, and let

$$\begin{gathered} q_1 \equiv \psi_1 \\ q_2 \equiv \psi_2(q_1) \\ \dots \\ q_\ell \equiv \psi_\ell(q_1, \dots, q_{\ell-1}) \end{gathered}$$

be all extension axioms used in π. We define circuits $Q_{i,j}(q_1, \dots, q_j)$, $0 \leq j < i \leq \ell$, as follows:

$$\begin{aligned} Q_{i,i-1}(q_1, \dots, q_{i-1}) &:= \psi_i(q_1, \dots, q_{i-1}), \\ Q_{i,j-1}(q_1, \dots, q_{j-1}) &:= Q'_{i,j}(q_1, \dots, q_{j-1}, \psi_j(q_1, \dots, q_{j-1})), \end{aligned}$$

where $Q'_{i,j}$ differs from $Q_{i,j}$ by joining all occurrences of q_j together. We put $Q_i := Q_{i,0}$. It is easy to see that $Q_i \simeq \psi_i(Q_1, \dots, Q_{i-1})$.

We modify the proof π by putting a (constant size) Frege proof of $q_i \equiv q_i$ before every extension axiom $q_i \equiv \psi_i$, and then we substitute circuits $Q_1, \dots, Q_\ell$ for variables $q_1, \dots, q_\ell$ in the whole proof. This makes up a correct CF proof π': substitution does not break Frege rules, and extension axioms translate to circuits $Q_i \equiv \psi_i(Q_1, \ldots, Q_{i-1})$, each preceded by a similar circuit $Q_i \equiv Q_i$.

The size of $Q_{i,j}$ is bounded by $|\psi_{j+1}| + \cdots + |\psi_i|$, in particular the size of Q_i is bounded by $|\pi|$, hence the size of π' is $O(|\pi|^2)$. □

5.1.5 Lemma *Any EF system p-simulates proofs of formulas in any CF system.*

Proof: Let $\pi\colon A_0, \ldots, A_k = \varphi$ be a CF proof, where φ is a formula. We assign an extension variable $q_i =: q[C]$ to each subcircuit C of each A_j in such a way that similar circuits get the same variable, and every circuit gets a variable with higher index than all its subcircuits. The EF proof π' will start with extension axioms for q_i's, which describe the relation of the corresponding circuits to their subcircuits. For example, if $C = p_1 \vee \neg(p_2 \to p_1)$, we could have

$$\begin{aligned} q_1 &\equiv p_1 \\ q_2 &\equiv p_2 \\ q_3 &\equiv q_2 \to q_1 \\ q_4 &\equiv \neg q_3 \\ q_5 &\equiv q_1 \vee q_4 \end{aligned}$$

Then we extend the proof to contain the sequence $q[A_0], \ldots, q[A_k]$. If $A_i \simeq A_j$, $j < i$, we have nothing to do, because $q[A_i] = q[A_j]$. Assume that $A_i = \chi(B_1, \ldots, B_m)$ was inferred by a Frege rule R from $A_{j_1} = \psi_1(B_1, \ldots, B_m)$, $\ldots$, $A_{j_\ell} = \psi_\ell(B_1, \ldots, B_m)$, where $j_1, \ldots, j_\ell < i$. There is a constant size Frege proof of

$$\begin{gathered} q[A_{j_1}] \equiv \psi_1(q[B_1], \ldots, q[B_m]) \\ \ldots \\ q[A_{j_\ell}] \equiv \psi_\ell(q[B_1], \ldots, q[B_m]) \\ q[A_i] \equiv \chi(q[B_1], \ldots, q[B_m]) \end{gathered}$$

from the extension axioms. By the induction hypothesis our proof already contains the formulas $q[A_{j_1}], \ldots, q[A_{j_\ell}]$, hence we get a proof of

$$\begin{gathered} \psi_1(q[B_1], \ldots, q[B_m]) \\ \ldots \\ \psi_\ell(q[B_1], \ldots, q[B_m]) \end{gathered}$$

$$\frac{\chi(q[B_1],\dots,q[B_m])}{q[A_i]}$$

by a constant-size simulation of R and Modus Ponens (or rather its variant for $\equiv$).

We thus have an $O(|\pi|)$ proof of $q[\varphi]$, and we finish it by an $O(|\varphi|^2)$ proof of $q[\varphi]\equiv\varphi$ and Modus Ponens. □

5.2 *WPHP* Frege

5.2.1 Definition The *WF* (*WPHP Frege*) proof system is defined as follows: a *WF*-proof of a circuit A is a sequence of circuits $A_0,\dots,A_k$ such that $A_k = A$, and every A_i is inferred from some $A_{j_1},\dots,A_{j_\ell}$, $j_1,\dots,j_\ell < i$ by a Frege rule, or it is similar to some A_j, $j < i$, or it is a special axiom

$$\bigvee_{\ell=1}^{m}(r_\ell \not\equiv C_{i,\ell}(D_{i,1},\dots,D_{i,n})),$$

where $n < m$, and r_ℓ are pairwise distinct variables which do not occur in A, $C_{i,\ell'}$, or A_j for $j < i$, but may occur in $D_{i,1},\dots,D_{i,n}$.

5.2.2 Remark In principle, different choices of connectives and Frege rules give different variants of *WF*. We ignore this ambiguity, as all such systems are polynomially equivalent.

We will see in 5.2.8 that we could restrict *WF*-proofs to contain only *one* special axiom, and still get an equivalent system. On the other hand, we could allow special axioms with the same C's to share the same sequence of special variables: the proof of 5.2.3 can be easily modified to show the consistency of such a system in $S^1_2 + dWPHP(PV)$, hence it is polynomially equivalent to the original *WF* by 5.2.8.

5.2.3 Theorem *$S^1_2 + dWPHP(PV)$ proves 0-RFN(WF).*

Proof: Let $\pi = \langle A_0,\dots,A_k\rangle$ be a *WF*-proof of a circuit $A = A_k$, and let e be a truth assignment to the variables occurring in A. W.l.o.g. we may assume that every variable in π either occurs in A, or it is a special variable of a *WF*-axiom from π. We will show by induction on $i \le k < |\pi|$ that there is an assignment $e' \supseteq e$, which makes A_j true for every $j \le i$ (this is Σ^b_1-*LIND*).

If A_i is inferred by a Frege rule from $A_{j_1},\dots,A_{j_\ell}$, $j_1,\dots,j_\ell < i$, the induction step from $i-1$ to i is easy because the rule is sound: its verification consists of checking only finitely many cases involving the inductive definition of satisfaction for some top-level subcircuits of the A_j's, hence it goes through in S^1_2.

If $A_i \simeq A_j$, $j < i$, we have $e'(A_i) = e'(A_j)$ by induction on the depth of the circuit.

Assume that A_i is the special axiom $\bigvee_{j=1}^{m}(r_j \not\equiv C_j(D_1, \dots, D_n))$. Notice that the truth value of all variables occurring in $C_j(s_1, \dots, s_n)$ is fixed by e', except for the placeholders $s_1, \dots, s_n$ (the definition of *WF* implies that special variables from $A_{i'}$, $i' \geq i$, cannot occur in C_j). Hence the sequence of circuits $C = \langle C_1, \dots, C_m \rangle$ computes a function $g\colon 2^n \to 2^m$. More precisely, there is a PV-function symbol $f(u, v, x)$ with the following property: if u is a sequence of circuits, and v a partial truth assignment, then the j-th bit of $f(u, v, x)$ is $u_j(a)$, where a extends v and the j'-th variable not assigned by v is given the value $bit(x, j')$ by a. Then we put $g(x) = f(\langle C_1, \dots, C_m \rangle, e', x)$. By definition $n < m$, i.e. $2 \cdot 2^n \leq 2^m$, hence $dWPHP(PV)$ implies that there is $y < 2^m$ such that $y \neq g(x)$ for any $x < 2^n$. We extend e' by putting $e'(r_j) = bit(y, j-1)$, and we claim that $e'(A_i) = 1$: if $x < 2^n$ is such that $bit(x, j') = e'(D_{j'+1})$, then the value of $C_j(\vec{D}_{j'})$ under e' is $bit(g(x), j-1)$, which is distinct from $e'(r_j)$ for some $j \leq m$. □

Recall that G is a propositional proof system operating with quantified Boolean formulas, defined (in [22]) as an extension of the usual Gentzen sequent calculus by rules for introducing existential and universal quantifiers. G_2 is a fragment of G, which allows only sequents consisting of Σ^q_2-formulas (these are, roughly, formulas of the form $\exists x_1 \cdots \exists x_k \forall y_1 \cdots \forall y_\ell\, \varphi$, with φ quantifier-free).

5.2.4 Corollary *G_2 polynomially simulates WF.*

Proof: By [35] (see also [28], and chapter 11.2 of [20]), $dWPHP(PV)$ is provable in T^2_2, hence also $T^2_2 \vdash 0\text{-}RFN(WF)$. By [22], this implies $S^1_2 \vdash WF \leq_p G_2$. See also [20], chapters 9.2, 9.3. □

Recall the definition of the $\|\varphi\|$ translation of $\Pi^b_1(PV)$-formulas into propositional logic from section 2.3. In this translation, it is necessary to encode the computation of the circuit $\{\!\{f\}\!\}^{\vec{n}}$ by a formula introducing extra auxiliary variables, as it is unlikely that $PV \vdash P \subseteq NC^1$. This seems to obfuscate things a bit, and we will use a proof system handling Boolean circuits directly, we thus avoid this inconvenience by introducing a more natural modified translation, which produces circuits instead of formulas. It is defined as follows:

5.2.5 Definition Let $\varphi(\vec{x})$ be a $\Pi^b_1(PV)$-formula, and $b(\vec{x})$ its bounding polynomial. We define a Boolean circuit $\{\!\!\{\varphi\}\!\!\}^{\vec{n}}(\vec{p};\vec{q})$ by induction on complexity of φ:

$$\begin{aligned}
\{\!\!\{f(\vec{x}) = g(\vec{x})\}\!\!\}^{\vec{n}}(\vec{p}) &:= \bigwedge_{i<b(\vec{n})} (\{\!\!\{f\}\!\!\}^{\vec{n}}_i(\vec{p}) \equiv \{\!\!\{g\}\!\!\}^{\vec{n}}_i(\vec{p})),\\
\{\!\!\{f(\vec{x}) \le g(\vec{x})\}\!\!\}^{\vec{n}}(\vec{p}) &:= \bigwedge_{i<b(\vec{n})} \big(\{\!\!\{f\}\!\!\}^{\vec{n}}_i \mathbin{\&} \bigwedge_{j>i}(\{\!\!\{f\}\!\!\}^{\vec{n}}_j \equiv \{\!\!\{g\}\!\!\}^{\vec{n}}_j) \to \{\!\!\{g\}\!\!\}^{\vec{n}}_i\big),\\
\{\!\!\{\neg\varphi\}\!\!\}^{\vec{n}}(\vec{p}) &:= \neg\{\!\!\{\varphi\}\!\!\}^{\vec{n}}(\vec{p}), \quad \varphi \in \Sigma^b_0(PV),\\
\{\!\!\{\varphi \circ \psi\}\!\!\}^{\vec{n}}(\vec{p};\vec{q}) &:= \{\!\!\{\varphi\}\!\!\}^{\vec{n}}(\vec{p};\vec{q}) \circ \{\!\!\{\psi\}\!\!\}^{\vec{n}}(\vec{p};\vec{q}), \quad \circ \in \{\&, \vee\},\\
\{\!\!\{\forall y \le |t(\vec{x})|\, \varphi(\vec{x},y)\}\!\!\}^{\vec{n}}(\vec{p};\vec{q^0},\dots,\vec{q^m}) &:= \\
&\bigwedge_{j\le m} \{\!\!\{y \le |t(\vec{x})| \to \varphi(\vec{x},y)\}\!\!\}^{\vec{n},|m|}(\vec{p},\vec{\varepsilon};\vec{q^j}),\\
\{\!\!\{\exists y \le |t(\vec{x})|\, \varphi(\vec{x},y)\}\!\!\}^{\vec{n}}(\vec{p};\vec{q^0},\dots,\vec{q^m}) &:= \\
&\bigvee_{j\le m} \{\!\!\{y \le |t(\vec{x})| \mathbin{\&} \varphi(\vec{x},y)\}\!\!\}^{\vec{n},|m|}(\vec{p},\vec{\varepsilon};\vec{q^j}),\\
\{\!\!\{\forall y \le t(\vec{x})\, \varphi(\vec{x},y)\}\!\!\}^{\vec{n}}(\vec{p};\vec{q},\vec{p'}) &:= \{\!\!\{y \le t(\vec{x}) \to \varphi(\vec{x},y)\}\!\!\}^{\vec{n},m}(\vec{p},\vec{p'};\vec{q}),
\end{aligned}$$

where $m = m(\vec{n})$ is a bounding polynomial to $t(\vec{x})$, and $\vec{\varepsilon}$ is the representation of j as a sequence of $|m|$ binary digits (= truth constants). Notice that auxiliary variables $\vec{q}$ are introduced only for (non-sharply) bounded universal quantifiers.

5.2.6 Lemma *Let $\varphi(\vec{x}) \in \Pi^b_1(PV)$. There are circuits $\vec{C}^{\vec{n}}_\varphi$, and a p-time constructible sequence of CF-proofs of*

$$\begin{gathered}
\{\!\!\{\varphi\}\!\!\}^{\vec{n}}(\vec{p};\vec{q}) \to \|\varphi\|^{\vec{n}}(\vec{p};\vec{q},\vec{q'}),\\
\|\varphi\|^{\vec{n}}(\vec{p};\vec{q},\vec{C}^{\vec{n}}_\varphi(\vec{p},\vec{q})) \to \{\!\!\{\varphi\}\!\!\}^{\vec{n}}(\vec{p};\vec{q}).
\end{gathered}$$

Proof: This follows by straightforward induction on complexity of φ. We need the following property for the base case: for any PV-function f, there are circuits $\vec{C}^{\vec{n}}_f$, and p-time constructible CF-proofs of

$$\begin{gathered}
\|f\|^{\vec{n}}(\vec{p};\{\!\!\{f\}\!\!\}^{\vec{n}}(\vec{p});\vec{C}^{\vec{n}}_f(\vec{p})),\\
\|f\|^{\vec{n}}(\vec{p};\vec{r};\vec{q}) \to \bigwedge_i (r_i \equiv \{\!\!\{f\}\!\!\}^{\vec{n}}_i(\vec{p})).
\end{gathered}$$

We may take subcircuits of $\{\!\!\{f\}\!\!\}$ for $\vec{C}_f$. The second part essentially states that the computation of $\{\!\!\{f\}\!\!\}$ is unique, and its proof in CF may be constructed by induction on the size of $\{\!\!\{f\}\!\!\}$. □

5.2.7 Theorem *If $S^1_2 + dWPHP(PV) \vdash \forall x\, \varphi(x)$, where $\varphi \in \Pi^b_1$, then tautologies $\|\varphi\|^n$ have polynomial size WF-proofs. Actually, these proofs are constructible by a p-time function, and PV proves this fact.*

Proof: Assume that $S^1_2 + dWPHP(PV) \vdash \forall x\, \varphi(x)$, where $\varphi \in \Sigma^b_0(PV)$ for simplicity. By theorem 3.3.1, there is a constant k, and PV-functions G and g such that

$$PV \vdash 2^{|x|^k} \leq b \,\&\, w < b^2 \to (G(g(x,w,b)) = w \vee \varphi(x)),$$
$$PV \vdash g(x,w,b) < b.$$

Given n (bounding x), and $m = 2n^k$ (bounding w and b), there are $poly(n)$-size CF-proofs (constructible in PV) of the circuits

$$\{\!\{2^{|x|^k} \leq b\}\!\}^{n,m}(\vec{p},\vec{q}) \,\&\, \{\!\{w < b^2\}\!\}^{m,m}(\vec{r},\vec{q}) \to$$
$$\to \{\!\{G(g(x,w,b)) = w\}\!\}^{n,m,m}(\vec{p},\vec{r},\vec{q}) \vee \{\!\{\varphi(x)\}\!\}^n(\vec{p}),$$

$$\{\!\{g(x,w,b) < b\}\!\}^{n,m,m}(\vec{p},\vec{r},\vec{q}),$$

using the simulation of PV by EF [9], and lemmas 5.1.4, 5.2.6. We substitute the binary representation of $b := 2^{n^k}$ for the variables $\vec{q}$, i.e., $q_{n^k} = 1$, $q_j = 0$ for $j \neq n^k$. Then there are poly-size CF-proofs of $\{\!\{2^{|x|^k} \leq b\}\!\}^{n,m}$ and $\{\!\{w < b^2\}\!\}^{m,m}$, hence by modus ponens

$$\{\!\{G(g(x,w,b)) = w\}\!\}^{n,m,m} \vee \{\!\{\varphi(x)\}\!\}^n,$$

which is the circuit

$$\bigwedge_{i<m} (r_i \equiv \{\!\{G\}\!\}^{q(n)}_i(\{\!\{g\}\!\}_0, \ldots, \{\!\{g\}\!\}_{q(n)-1})) \vee \{\!\{\varphi(x)\}\!\}^n,$$

where $q(n)$ is the bounding polynomial for g. However,

$$\{\!\{g(x,w,b) < b\}\!\}^{n,m,m}$$

implies

$$\bigwedge_{i=n^k}^{q(n)-1} \neg\{\!\{g\}\!\}^{n,m,m}_i,$$

thus we get a proof π of

$$\bigwedge_{i<m} (r_i \equiv \{\!\{G\}\!\}^{q(n)}_i(\{\!\{g\}\!\}_0, \ldots, \{\!\{g\}\!\}_{n^k-1}, 0, \ldots, 0)) \vee \{\!\{\varphi(x)\}\!\}^n.$$

If we define

$$C_j(s_0, \ldots, s_{n^k-1}) = \{\!\{G\}\!\}^{q(n)}_j(\vec{s}, 0, \ldots, 0),$$

$$D_i(\vec{p},\vec{r}) = \{\!\{g\}\!\}_i^{n,m,m}(\vec{p},\vec{r},\vec{q}),$$

we may rewrite this as

$$\bigwedge_{i<m}(r_i \equiv C_i(D_0,\dots,D_{n^k-1})) \vee \{\!\{\varphi(x)\}\!\}^n.$$

Since $m = 2n^k > n^k$ for every $n > 0$, and C_j does not contain any of the $r_{j'}$, we may put a special axiom

$$\bigvee_{i<m}(r_i \not\equiv C_i(D_0,\dots,D_{n^k-1}))$$

before the first line of π, and we finish the proof by De Morgan rules and modus ponens to get a *WF*-proof of

$$\{\!\{\varphi(x)\}\!\}^n.$$

Lemma 5.2.6, and another modus ponens give

$$\|\varphi(x)\|^n.$$

□

5.2.8 Corollary

(i) *For any Π_1^b-formula $\varphi(x)$, $S_2^1 + dWPHP(PV) \vdash (WF \vdash \|\varphi\|^{|x|}) \to \varphi(x)$.*

(ii) *$PV + Con(WF)$ axiomatizes strict $\forall\Pi_1^b$-consequences of the theory $S_2^1 + dWPHP(PV)$.*

(iii) *If $S_2^1 + dWPHP(PV) \vdash 0\text{-}RFN(P)$, where P is a propositional proof system, then $PV \vdash (P \leq_p WF)$.*

(iv) *WF is polynomially simulated by a modified WF proof system, in which we allow only the first formula of the proof to be a special axiom.*

Proof: (i) follows from 5.2.3 together with $S_2^1 \vdash Taut(\|\varphi\|^{|x|}) \to \varphi(x)$.

(ii): if $\varphi \in strict\Pi_1^b$, the formula $Taut(\|\varphi\|^{|x|}) \to \varphi(x)$ just mentioned is provable already in PV, and $Con(WF)$ implies $0\text{-}RFN(WF)$ as WF is provably closed under substitution and modus ponens. This, together with 5.2.7, shows the harder inclusion of (ii), the other one follows from 5.2.3.

(iii): we have $PV \vdash (WF \vdash \{\!\{P(p) = f \to Taut(f)\}\!\})$ by 5.2.7, and it is easy to see that $PV \vdash (P(\pi) = \varphi \to CF \vdash \{\!\{P(p) = f\}\!\}(\pi,\varphi))$ and $PV \vdash (WF \vdash \{\!\{Taut\}\!\}(\varphi) \to WF \vdash \varphi)$, hence $PV \vdash (P(\pi) = \varphi \to WF \vdash \varphi)$.

(iv): the proof of (iii) works for the modified *WF*-system from (iv) as well, because the proof constructed in 5.2.7 used only one special axiom; then (iv) follows from 5.2.3. □

5.2.9 Definition Let p be a prime. *Unstructured Extended Nullstellensatz* of [7] is a proof system for multivariate polynomials over $\mathbb{Z}_p$: a $UENS_p$-refutation of a set of polynomials $f_0, \dots, f_{n-1} \in \mathbb{Z}_p[x_0, \dots, x_{m-1}]$ shows that the f_i's do not have a 0–1 solution (i.e., a common zero at a point from $\{0,1\}^m$). A $UENS_p$-refutation is given by two sequences of polynomials $g_0, \dots, g_{\ell-1}$ and $g'_0, \dots, g'_{\ell+n+m-1}$, such that

$$\sum_{i<\ell} g_i g'_i + \sum_{i<n} f_i g'_{i+\ell} + \sum_{i<m} (x_i^2 - x_i) g'_{i+\ell+n} = 1,$$

and each g_i has the form

$$\prod_{j<k} (h_{i,j} - r_{i,j}),$$

where $r_{i,j}$ are pairwise distinct variables not occurring among $x_0, \dots, x_{m-1}$, $h_{i,j}$ does not contain any of $r_{i,0}, \dots, r_{i,k-1}$, and $\ell < e^{k/p}$ (where e is the base of natural logarithm).

The $UENS$ proof system simulates Extended Frege, but the converse is an open problem. In fact, it was not clear whether *any* "traditional" proof system simulates $UENS$. We show that it is possible to simulate $UENS$ in WF (hence also in G_2).

5.2.10 Theorem *For any prime p, the WF proof system polynomially simulates $UENS_p$.*

Proof: By corollary 5.2.8, it suffices to prove the soundness of $UENS_p$ in $S^1_2 + dWPHP(PV)$. It is not clear how to express base e exponentiation in bounded arithmetic, however we may simply relax the last condition of 5.2.9 to $\ell < \beta^{k/p}$, where β is any fixed rational such that $e < \beta < (1 - 1/p)^{-p}$.

Consider any $UENS_p$-refutation as in 5.2.9, and assume for contradiction that $f_i(\vec{a}) = 0$ for all $i < n$, with $a_j \in \{0,1\}$. Put $t = k\ell$. W.l.o.g. we assume that every variable in g_i and g'_i is one of x_j or $r_{i',j}$. We will find an assignment $b_{0,0}, \dots, b_{\ell-1,k-1} \in \mathbb{Z}_p$ to $\{r_{i,j}\}_{i<\ell, j<k}$ such that $g_i(\vec{a}, \vec{b}) = 0$ for all i, then

$$\sum_{i<\ell} g_i(\vec{a}, \vec{b}) g'_i(\vec{a}, \vec{b}) + \sum_{i<n} f_i(\vec{a}) g'_{i+\ell}(\vec{a}, \vec{b}) + \sum_{i<m} (a_i^2 - a_i) g'_{i+\ell+n}(\vec{a}, \vec{b}) = 0 \neq 1,$$

contradicting the definition of a $UENS_p$-proof.

We define a function

$$F \colon \ell \times (p-1)^k \times p^{t-k} \to p^t$$

by $F(i, u_0, \dots, u_{k-1}, v_0, \dots, v_{t-k-1}) = \langle b_{0,0}, \dots, b_{\ell-1,k-1} \rangle$, where $b_{i',j}$ are assigned according to $\vec{v}$ if $i' \neq i$, and

$$b_{i,j} = \begin{cases} u_j, & \text{if } u_j < h_{i,j}(\vec{a}, \vec{b}), \\ u_j + 1, & \text{otherwise.} \end{cases}$$

Notice that the value of $h_{i,j}(\vec{a}, \vec{b})$ depends only on $\vec{v}$, as $r_{i,0}, \dots, r_{i,k-1}$ do not occur in $h_{i,j}$.

It is clear from the definition that the values of $F(i, \bullet)$ are exactly the assignments $\vec{b}$ such that $g_i(\vec{a}, \vec{b}) \neq 0$, hence it suffices to show that $\mathrm{rng}(F) \neq p^t$. Choose a rational constant $\alpha > 1$ such that $\beta\alpha^p < (p/(p-1))^p$. Then $\alpha\ell(p-1)^k p^{t-k} < \beta^{k/p}\alpha^k(p-1)^k p^{t-k} < p^t$, hence F is not onto by $dWPHP(PV)^x_{\alpha x}$. □

5.2.11 Remark By an easy modification of the proof of 5.2.10, we could simulate a slightly stronger system than *UENS*: the extension variables $r_{i,j}$ could be reused in $g_{i'}$, $i' \neq i$, and we could allow $r_{i,0}, \dots, r_{i,j-1}$ to occur in $h_{i,j}$. (However, it is quite possible that this modification is polynomially equivalent to the original *UENS*.)

Chapter 6

Theories with explicit counting

This chapter deals with two unrelated bounded arithmetical theories, which have counting terms or quantifiers in their language. In section 6.1, we will look at a theory invented by R. Impagliazzo and B. Kapron, aimed at formalizing cryptographic reasoning. We will propose a modification of the theory, and generalize the soundness theorem from [14] to $\forall\exists$-consequences. In section 6.2, we will define and develop a bounded theory of approximate counting, formulated in a variant of Kleene's 3-valued logic.

6.1 Impagliazzo-Kapron logic

6.1.1 Description of the theory

The Impagliazzo-Kapron logic, as described in [14], is a multi-sorted theory in a second-order language. First-order objects are intended to be natural numbers, or equivalently, binary strings. Second-order objects of sort $k > 0$ are k-ary functions (intended to be polynomial-time computable). The language of the theory contains usual arithmetical operations 0, S, $+$, $\cdot$, $|\cdot|$, $\#$[1], and relation $<$. The function symbol $\circ$ denotes string concatenation, and $x_{\{i\dots j\}}$ is the substring of x consisting of bit positions i through j. The constants $\mathbf{s}$ and $\mathbf{n}$ denote a non-standard string and its length, respectively. We have also application function symbols, which apply a k-ary second-order object f to first-order objects $x_1, \dots, x_k$; we will however leave them

[1] [14] use $\otimes$ instead of $\#$, to avoid clash with counting terms. We stick to the standard notation, as the smash function can be easily distinguished from counting terms by syntactic context.

out in formulas, writing just $f(x_1, \ldots, x_k)$ instead. We have only a first-order equality relation symbol.

The functions just introduced are called *basic functions*, and terms using only basic functions are *basic terms*. Apart from these, we have also *counting terms*: for any open formula φ, there is a function symbol $\#(|x| = |t|)\varphi$, denoting the number of strings of length $|t|$ which satisfy φ. These counting terms are introduced recursively, i.e., the formula φ may also contain counting terms.

The original Impagliazzo-Kapron theory, which we denote IK^-, has the following axioms.

- *BASIC*, modified appropriately to the selection of arithmetical function symbols in the language.
- Security parameter axioms: $\mathbf{n} = |\mathbf{s}|$, and $\mathbf{n} > \overline{k}$ for every $k \in \omega$.
- Function axioms: functions (i.e., second-order objects, rather than function symbols of the language) are closed under composition and bounded recursion on notation. There are function objects for all projections, non-constant basic function symbols, and constant 0.
- The *LIND* schema for open formulas.
- Counting axioms:

$$\begin{gathered}
\#(|x| = |y|)\top = y \# 1, \\
\#(|x| = |y|)\varphi = \#(|x| = |y|)(\varphi \wedge \psi) + \#(|x| = |y|)(\varphi \wedge \neg\psi), \\
\forall x\, (|x| = |y| \to (\varphi \to \psi)) \to \#(|x| = |y|)\varphi \leq \#(|x| = |y|)\psi, \\
\#(|x| = 0)C(x) = C(0), \\
\#(|x| = |y| + 1)C(x) = \\
\#(|x| = |y|)C(2x+1) + \#(|x| = |y|)C(2x+2),
\end{gathered}$$

  where φ is an open formula, and C is a counting term—this is to be understood as follows: we define

$$\#(|x_1| = |y_1|)\#(|x_2| = |y_2|)\varphi(x_1, x_2)$$

  as an abbreviation for

$$\#(|x| = |y_1 \circ y_2|)\varphi(x_{\{0\ldots|x_1|-1\}}, x_{\{|x_1|\ldots|x|-1\}}).$$

One problem with this theory is that it does not prove the existence of constant functions $c_x(y) = x$ for arbitrary x. In particular, we cannot construct new functions by substitution of a constant value for a variable. This makes the theory a bit unnatural to work with, leading to subtle errors; indeed, substitution of constants is implicitly used in several applications in [14], most importantly in the proof of soundness of a system for computational indistinguishability (which we will not define here).

For this reason, we define a new theory, *IK*, to be IK^- plus the axiom

$$\forall x\, \exists f\, f(0) = x.$$

This is not just a cosmetic change; the soundness theorem for IK^- from [14] does not hold for *IK* as stated. In essence, IK^- is sound for cryptographic reasoning in uniform adversary model, whereas *IK* works only in non-uniform setting (i.e., we consider second-order objects to be functions from *FP*/*poly*). This difference is essential for applications to indistinguishability: soundness of preservation of indistingushability by substitution into a p-time context seems to require a non-uniform (or at least, randomized) adversary model.

For technical reasons, we also define IK_0 to be *IK* without the axioms $\mathbf{n} > \overline{k}$.

6.1.2 The soundness theorem

6.1.1 Definition Let $q(n)$ be a polynomial. A function f is *q-bounded* if $|f(x)| \leq q(|x|)$ for every x. Notice that every function from $FP^A/poly$ is q-bounded for some q, where A is an arbitrary oracle.

6.1.2 Theorem *Assume that IK proves*

$$\forall \vec{f}\, \forall \vec{z}\, \exists \vec{g}\, \exists \vec{y}\, \varphi(\vec{f}, \vec{g}, \vec{z}, \vec{y}, \mathbf{s}),$$

where φ is a bounded formula. Then there exists a constant k such that for every polynomial q, there is a polynomial p with the following property.

For every q-bounded functions $\vec{f}$, every $\vec{z}$, and every $x \geq k$, there exist numbers $\vec{y}$, and oracle circuits $\vec{C}$ such that

$$\mathbb{N} \vDash \varphi(\vec{f}, \vec{C}^{\vec{f}}, \vec{z}, \vec{y}, x),$$

and the size of $\vec{y}$ and $\vec{C}$ is at most $p(|\vec{z}|, |x|)$.

Proof: We start with some simplifications. An *IK*-proof may contain only a finite number of the axioms $\mathbf{n} > k$, thus there is a constant k such that

$$IK_0 \vdash \forall \vec{f}\, \forall \vec{z}\, \exists \vec{g}\, \exists \vec{y}\, (\mathbf{s} \geq \overline{k} \to \varphi(\vec{f}, \vec{g}, \vec{z}, \vec{y}, \mathbf{s})).$$

Since $\mathbf{s} \geq \overline{k} \to \varphi$ is a bounded formula, we can without loss of generality assume $k = 0$, i.e., it suffices to show the theorem for IK_0 in place of IK. We can get rid of the remaining security parameter axiom by substituting $|\mathbf{s}|$ for $\mathbf{n}$ in the whole proof; then we may replace $\mathbf{s}$ by a variable, and treat it as one of the variables $\vec{z}$. In other words: without loss of generality, φ does not contain $\mathbf{s}$. To simplify the notation, we will also ignore all the vectors, and consider only the case

$$IK_0 \vdash \forall f \, \forall z \, \exists g \, \exists y \, \varphi(f, g, z, y).$$

Fix a polynomial $q(n)$, and assume for contradiction that the conclusion of the theorem is false for every polynomial $p(n)$. This means that each finite subset of

$$\begin{aligned} S := Th(\mathbb{N}^2) \cup \{\forall u \, |\mathbf{f}(u)| \leq q(|u|)\} \\ \cup \{\forall |y|, |C| \leq p(|\mathbf{z}|) \, \neg\varphi(\mathbf{f}, C^{\mathbf{f}}, \mathbf{z}, y);\ p \text{ a polynomial}\} \end{aligned}$$

is satisfiable, where $\mathbb{N}^2$ is the standard model of second-order arithmetic, $\mathbf{z}$ is a new first-order constant, and $\mathbf{f}$ is a new second-order constant. By compactness, there exists a model M of S. Let

$$I = \{y \in M;\ \exists \text{ a standard polynomial } p \text{ s.t. } |y| \leq p(|\mathbf{z}|)\},$$

and let F be the set of all $PV(\mathbf{f})$-functions with parameters from I, restricted to I. Notice that I is closed under F: it is closed under $\mathbf{f}$ because $|\mathbf{f}(u)| \leq q(|u|)$, and closure under more complicated $PV(\mathbf{f})$-functions follows by induction on build-up of the function, using the fact that I is a cut closed under smash. In particular, I is closed under basic function symbols of IK_0; it is also closed under the counting functions, because $\#(|x| = |y|)\varphi \leq 2^{|y|} \leq 2y$. (Counting functions are well-defined in M, as it is a model of true arithmetic.)

It follows that $\langle I, F\rangle$ is a submodel of M (expanded to the language of IK_0). As such, it is elementary for open formulas; moreover I is a cut, thus it is elementary with respect to all bounded formulas. In particular, $\langle I, F\rangle \vDash IK_0$: the function axioms are satisfied as $PV(\mathbf{f})$ is closed under limited recursion on notation; the other axioms are bounded, and valid in M (as $M \vDash Th(\mathbb{N})$). By the definition of S and elementarity for bounded formulas, we have

$$\langle I, F\rangle \vDash \forall C \, \forall y \, \neg\varphi(\mathbf{f}, C^{\mathbf{f}}, \mathbf{z}, y).$$

Let $y \in I$, $g \in F$, and let $v \in I$ be the sequence of parameters used to define g. As in the case of usual arithmetical theories, the formula φ has a bounding polynomial r: i.e., in order to evaluate φ, we only need to know the value of functions and relation symbols for numbers of length at most $r(|y|,|v|,|\mathbf{z}|)$ (this follows by induction on complexity of φ; the only property needed is a (standard) polynomial bound to all functions involved in φ). Since g is given by a $PV(\mathbf{f})$-function with parameter v, the restriction of g to $2^{r(|y|,|v|,|\mathbf{z}|)}$ is computable by an oracle circuit C^f of size polynomial in $|y|$, $|v|$, and $|\mathbf{z}|$; it follows that $C \in I$, and

$$\langle I, F\rangle \vDash \neg\varphi(\mathbf{f}, g, \mathbf{z}, y).$$

Since y and g were arbitrary, we get

$$\langle I, F\rangle \vDash IK_0 + \neg\exists g\,\exists y\,\varphi(\mathbf{f}, g, \mathbf{z}, y),$$

which is a contradiction. □

As a corollary to theorem 6.1.2, we derive a soundness theorem for $(\forall \to \forall)$-consequences of IK, formulated more in line with the original Impagliazzo-Kapron soundness theorem from [14].

6.1.3 Corollary *Let φ and ψ be bounded formulas, and assume*

$$IK \vdash \forall g\,\forall\vec{z}\,\varphi(\vec{f}, g, \vec{z}, \mathbf{s}) \to \forall g\,\forall\vec{z}\,\psi(\vec{f}, g, \vec{z}, \mathbf{s}).$$

Let $\vec{f} \in FP/poly$. If for every polynomial $q(n)$, the formula $\varphi(\vec{f}, C, \vec{z}, x)$ holds for all sufficiently large x and all $|\vec{z}|, |C| \le q(|x|)$, then the same is true of ψ.

Proof: We rewrite the assumption as

$$IK \vdash \forall\vec{f}\,\forall h\,\forall\vec{z}\,\exists g\,\exists\vec{w}\,(\varphi(\vec{f}, g, \vec{w}, \mathbf{s}) \to \psi(\vec{f}, h, \vec{z}, \mathbf{s})).$$

Since IK proves that any circuit defines a function, we get

$$IK \vdash \forall\vec{f}\,\forall C\,\forall\vec{z}\,\exists g\,\exists\vec{w}\,(\varphi(\vec{f}, g, \vec{w}, \mathbf{s}) \to \psi(\vec{f}, C, \vec{z}, \mathbf{s})).$$

By theorem 6.1.2, there is a polynomial p and a constant k such that the standard model satisfies

$$\forall x > k\,\forall C, \vec{z}\,\exists |D|, |\vec{w}| \le p(|x|, |\vec{z}|, |C|)\,(\varphi(\vec{f}, D^{\vec{f}}, \vec{w}, x) \to \psi(\vec{f}, C, \vec{z}, x)).$$

Since $\vec{f}$ are computable by polynomial-size circuits, we can get rid of the oracles in D if we switch to a larger $p(n)$. If we substitute q into p, we get a polynomial r such that

$$\forall x > k\,\forall |C|, |\vec{z}| \le q(|x|)\,\exists |D|, |\vec{w}| \le r(|x|)\,(\varphi(\vec{f}, D, \vec{w}, x) \to \psi(\vec{f}, C, \vec{z}, x)),$$

which means

$$\forall x > k\,\big(\forall |C|, |\vec{z}| \le r(|x|)\,\varphi(\vec{f}, C, \vec{z}, x) \to \forall |C|, |\vec{z}| \le q(|x|)\,\psi(\vec{f}, C, \vec{z}, x)\big).$$

By assumption, $\forall |C|, |\vec{z}| \le r(|x|)\,\varphi(\vec{f}, C, \vec{z}, x)$ holds for all sufficiently large x, thus

$$\forall x \gg 0\,\forall |C|, |\vec{z}| \le q(|x|)\,\psi(\vec{f}, C, \vec{z}, x). \qquad \square$$

6.2 A theory of approximate counting

This section is devoted to a bounded arithmetical theory with approximate counting quantifiers. Unlike exact counting ($\#P$), approximate counting has a low complexity: it can be realized in randomized polynomial time, and thus in the second level of the polynomial-time hierarchy. One of our goals was thus to create a theory which can formalize basic approximate counting arguments, but still remain "feasible": e.g., existential theorems of the theory are witnessable by probabilistic polynomial-time algorithms.

Another motivation stems from definability of randomized computation: using approximate counting quantifiers, we can easily express algorithms from probabilistic complexity classes like BPP, MA, AM.

Approximate counting quantifiers can be naturally presented as promise problems: positive instances of the problem are the cases where the formula in question is satisfied by many elements, whereas in negative instances it is satisfied by only few elements. However, predicates and formulas in classical first-order logic can only express usual languages, not promise problems. The natural solution is to use a non-classical, 3-valued logic; the truth values 1 (true) and 0 (false) correspond respectively to positive and negative instances of the promise problem, and the third truth value, $*$, corresponds to the grey area of the remaining instances, where the promise is broken. The reader can think of $*$ as meaning "undefined", or better yet, "uncertain".

In section 6.2.1 we define precisely the logic we want to use, and state its basic properties (such as the completeness theorem). In section 6.2.2 we introduce the language and theory C^1_2 of approximate counting. Section 6.2.3 contains a witnessing theorem for C^1_2.

6.2.1 Kleene's logic

6.2.1 Definition *Kleene's 3-valued logic* [19] is a propositional logic using truth values $\{0, *, 1\}$ with the only designated value 1. Its connectives are given by the following tables.

$\wedge$	0	$*$	1
0	0	0	0
$*$	0	$*$	$*$
1	0	$*$	1

$\vee$	0	$*$	1
0	0	$*$	1
$*$	$*$	$*$	1
1	1	1	1

	$\neg$
0	1
$*$	$*$
1	0

We may also include nullary connectives $\bot$ and $\top$, with $v(\bot) = 0$ and $v(\top) = 1$.

3-valued first-order models are defined as usual, except that predicate symbols (and formulas in general) are realized by functions with range in $\{0, *, 1\}$. We denote $v^M(\varphi[e])$ the value of a formula φ in a model M under an assignment e. *First-order Kleene's logic* has quantifiers $\forall$ and $\exists$ with the following semantics.

$$v^M(\forall x\, \varphi[e]) = \begin{cases} 1, & \text{if } \forall a \in M\; v^M(\varphi[e(x/a)]) = 1, \\ 0, & \text{if } \exists a \in M\; v^M(\varphi[e(x/a)]) = 0, \\ * & \text{otherwise,} \end{cases}$$

$$v^M(\exists x\, \varphi[e]) = \begin{cases} 1, & \text{if } \exists a \in M\; v^M(\varphi[e(x/a)]) = 1, \\ 0, & \text{if } \forall a \in M\; v^M(\varphi[e(x/a)]) = 0, \\ * & \text{otherwise.} \end{cases}$$

Kleene's logic has no reasonable implication connective, which makes it impossible to formulate a nontrivial theory (e.g., induction axioms). We thus borrow an implication from the LPF system [17] (see also [2]), given by the following table.

$\to$	0	$*$	1
0	1	1	1
$*$	1	1	1
1	0	$*$	1

We also define an abbreviation $\varphi \leftrightarrow \psi = (\varphi \to \psi) \wedge (\psi \to \varphi)$, which has the following table.

$\leftrightarrow$	0	$*$	1
0	1	1	0
$*$	1	1	$*$
1	0	$*$	1

A formula φ is *satisfied* by an assignment e in a model M, if $v^M(\varphi[e]) = 1$. A formula is *valid* in a model M, if it is in M satisfied by all assignments. A formula is a *tautology*, if it is valid in all models.

A theory T *entails* a formula φ iff for any model M, if all formulas $\psi \in T$ are valid in M, then φ is also valid in M.

A sequent $\Gamma \Longrightarrow \Delta$ is *tautological* if any assignment in any model which satisfies all formulas in Γ also satisfies some formula in Δ. (Equivalently: the characteristic formula $\bigwedge\Gamma \to \bigvee\Delta$ is a tautology.)

We will denote the resulting system as $K^\forall$.

6.2.2 Remark The intuition behind our choice of $\to$ is that it behaves like an internalization of the sequent arrow—it makes the deduction theorem hold for $K^\forall$, and more generally,

$$\Gamma, \varphi \Longrightarrow \psi, \Delta \text{ is tautological} \quad \text{iff} \quad \Gamma \Longrightarrow (\varphi \to \psi), \Delta \text{ is tautological.}$$

(This condition determines the table of $\to$ almost completely, it only leaves open whether the values $1 \to 0$ and $1 \to *$ are 0 or $*$. It is possible to give another condition involving $\neg(\varphi \to \psi)$, which makes the table unique.) One nice feature of this $\to$ is that the negation-free fragment of $K^\forall$ (i.e., $\{\wedge, \vee, \to, \bot, \top, \forall, \exists\}$) is equivalent to the classical predicate calculus, because the function f defined as $f(0) = f(*) = 0$, $f(1) = 1$ is a homomorphism with respect to the connectives and quantifiers, and preserves the set of designated values.

6.2.3 Remark To put our variant of Kleene's logic in context, we note that $K^\forall$ is closely connected with 3-valued fuzzy logics. Apart from having a different "official" implication, the basic Kleene's logic is a fragment of 3-valued Gödel logic with involutive negation, and $K^\forall$ has the same expressive power as 3-valued Łukasiewicz or Gödel logic with Baaz delta [26, 3]: the Łukasiewicz connectives and delta are definable as

$$\begin{aligned}
\triangle\varphi &= \neg(\varphi \to \neg\varphi),\\
\varphi \supset \psi &= (\varphi \to \psi) \wedge (\neg\psi \to \neg\varphi),\\
\varphi \mathbin{\&} \psi &= \neg(\varphi \supset \neg\psi),
\end{aligned}$$

on the other hand we have

$$\varphi \to \psi = \triangle\varphi \supset \psi.$$

From an algebraic point of view, the structure $\langle\{0, *, 1\}, \vee, \wedge, \neg, \bot, \top\}\rangle$ is a bounded distributive lattice with an involutive dual automorphism, satisfying $x \wedge \neg x \leq y \vee \neg y$ (and, in fact, it generates the variety of such algebras). It follows that the usual associative, commutative, distributive, and de Morgan rules hold in Kleene's logic.

6.2.4 Definition The Hilbert-style calculus $\mathit{HK}^{\forall}$ consists of the following axioms and rules.

$$((\varphi \to \psi) \to \chi) \to ((\chi \to \varphi) \to (\omega \to \varphi))$$
$$\varphi \wedge \psi \to \varphi$$
$$\varphi \wedge \psi \to \psi$$
$$\varphi \to (\psi \to \varphi \wedge \psi)$$
$$\varphi \to \varphi \vee \psi$$
$$\psi \to \varphi \vee \psi$$
$$(\varphi \to \chi) \to ((\psi \to \chi) \to (\varphi \vee \psi \to \chi))$$
$$\forall x\, \varphi \to \varphi(x/t)$$
$$\varphi(x/t) \to \exists x\, \varphi$$
$$\neg\varphi \to (\varphi \to \psi)$$
$$\neg\neg\varphi \leftrightarrow \varphi$$
$$\neg(\varphi \wedge \psi) \leftrightarrow \neg\varphi \vee \neg\psi$$
$$\neg(\varphi \vee \psi) \leftrightarrow \neg\varphi \wedge \neg\psi$$
$$\neg(\varphi \to \psi) \leftrightarrow \varphi \wedge \neg\psi$$
$$\neg\forall x\, \varphi \leftrightarrow \exists x\, \neg\varphi$$
$$\neg\exists x\, \varphi \leftrightarrow \forall x\, \neg\varphi$$
$$\varphi,\ \ \varphi \to \psi \ \ \vdash \ \ \psi$$
$$\psi \to \varphi \ \ \vdash \ \ \psi \to \forall v\, \varphi$$
$$\varphi \to \psi \ \ \vdash \ \ \exists v\, \varphi \to \psi$$

In the generalization rules, the variable v cannot be free in ψ. (If the reader is puzzled by the first axiom [27], she may replace it by her favourite complete axiomatization of the classical implicational fragment.)

6.2.5 Theorem (completeness of $\mathit{HK}^{\forall}$) *A theory T entails a formula φ iff φ is $\mathit{HK}^{\forall}$-provable from T.*

Proof (*sketch*)*:* The propositional case was proved by Avron [2]. Soundness of $\mathit{HK}^{\forall}$ is straightforward. As for completeness, first notice that the $\neg$-free part of $\mathit{HK}^{\forall}$ is an axiomatization of the positive fragment of the classical logic. The deduction theorem for $\mathit{HK}^{\forall}$, and conservativity of the Henkin completion follow easily from this observation, as their standard proof only relies on provability of certain $\neg$-free schemata. (E.g., the usual theorem on conservative introduction of constants follows from the deduction theorem and the second generalization rule; then we can justify adding Henkin constants by provability of $\exists y\,(\exists x\,\varphi(x) \to \varphi(y))$ and $\exists y\,(\varphi(y) \to \forall x\,\varphi(x))$.)

Assume that $T \nvdash \varphi$ is a Henkin theory. By Zorn's lemma, there exists a maximal theory $S \supseteq T$ in the same language such that $S \nvdash \varphi$. Notice that S is a deductively closed Henkin theory, and moreover it is prime (i.e., it has the disjunction property), because of the deduction theorem and the axiom

$$(\psi \to \varphi) \to ((\chi \to \varphi) \to (\psi \vee \chi \to \varphi)).$$

We take the term model for the language of S, with satisfaction defined by

$$v(\psi) = \begin{cases} 1, & \text{if } \psi \in S, \\ 0, & \text{if } \neg\psi \in S, \\ * & \text{otherwise} \end{cases}$$

for any sentence ψ. We can check that this is indeed a correct definition of a $K^\forall$-model, thus $T \nvDash \varphi$. For example, consider a formula $\psi \wedge \chi$. We have $v(\psi \wedge \chi) = 1$ iff $v(\psi) = v(\chi) = 1$, due to the axioms

$$\begin{gathered} \psi \wedge \chi \to \psi \\ \psi \wedge \chi \to \chi \\ \psi \to (\chi \to \psi \wedge \chi). \end{gathered}$$

Similarly, $v(\psi \wedge \chi) = 0$ iff $v(\psi) = 0$ or $v(\chi) = 0$, because S is prime, and contains the axiom

$$\neg(\psi \wedge \chi) \leftrightarrow \neg\psi \vee \neg\chi. \qquad \square$$

6.2.6 Definition Sequent calculus $GK^\forall$ consists of the usual structural rules (exchange, contraction, weakening, cut), and the following rules.

$$\Gamma,\, \varphi \Longrightarrow \varphi,\, \Delta \qquad\qquad \Gamma,\, \varphi,\, \neg\varphi \Longrightarrow \Delta$$

$$\frac{\Gamma \Longrightarrow \varphi,\, \Delta \qquad \Gamma \Longrightarrow \psi,\, \Delta}{\Gamma \Longrightarrow \varphi \wedge \psi,\, \Delta} \qquad\qquad \frac{\Gamma \Longrightarrow \neg\varphi,\, \neg\psi,\, \Delta}{\Gamma \Longrightarrow \neg(\varphi \wedge \psi),\, \Delta}$$

$$\frac{\Gamma,\, \varphi,\, \psi \Longrightarrow \Delta}{\Gamma,\, \varphi \wedge \psi \Longrightarrow \Delta} \qquad\qquad \frac{\Gamma,\, \neg\varphi \Longrightarrow \Delta \qquad \Gamma,\, \neg\psi \Longrightarrow \Delta}{\Gamma,\, \neg(\varphi \wedge \psi) \Longrightarrow \Delta}$$

$$\frac{\Gamma \Longrightarrow \varphi,\, \psi,\, \Delta}{\Gamma \Longrightarrow \varphi \vee \psi,\, \Delta} \qquad\qquad \frac{\Gamma \Longrightarrow \neg\varphi,\, \Delta \qquad \Gamma \Longrightarrow \neg\psi,\, \Delta}{\Gamma \Longrightarrow \neg(\varphi \vee \psi),\, \Delta}$$

$$\frac{\Gamma,\, \varphi \Longrightarrow \Delta \qquad \Gamma,\, \psi \Longrightarrow \Delta}{\Gamma,\, \varphi \vee \psi \Longrightarrow \Delta} \qquad\qquad \frac{\Gamma,\, \neg\varphi,\, \neg\psi \Longrightarrow \Delta}{\Gamma,\, \neg(\varphi \vee \psi) \Longrightarrow \Delta}$$

$$\frac{\Gamma,\ \varphi \Longrightarrow \psi,\ \Delta}{\Gamma \Longrightarrow (\varphi \to \psi),\ \Delta} \qquad \frac{\Gamma \Longrightarrow \varphi,\ \Delta \qquad \Gamma \Longrightarrow \neg\psi,\ \Delta}{\Gamma \Longrightarrow \neg(\varphi \to \psi),\ \Delta}$$

$$\frac{\Gamma \Longrightarrow \varphi,\ \Delta \qquad \Gamma,\ \psi \Longrightarrow \Delta}{\Gamma,\ (\varphi \to \psi) \Longrightarrow \Delta} \qquad \frac{\Gamma,\ \varphi,\ \neg\psi \Longrightarrow \Delta}{\Gamma,\ \neg(\varphi \to \psi) \Longrightarrow \Delta}$$

$$\frac{\Gamma \Longrightarrow \varphi,\ \Delta}{\Gamma \Longrightarrow \neg\neg\varphi,\ \Delta} \qquad \frac{\Gamma,\ \varphi \Longrightarrow \Delta}{\Gamma,\ \neg\neg\varphi \Longrightarrow \Delta}$$

$$\frac{\Gamma \Longrightarrow \varphi(v),\ \Delta}{\Gamma \Longrightarrow \forall x\, \varphi,\ \Delta} \qquad \frac{\Gamma \Longrightarrow \neg\varphi(t),\ \Delta}{\Gamma \Longrightarrow \neg\forall x\, \varphi,\ \Delta}$$

$$\frac{\Gamma,\ \varphi(t) \Longrightarrow \Delta}{\Gamma,\ \forall x\, \varphi \Longrightarrow \Delta} \qquad \frac{\Gamma,\ \neg\varphi(v) \Longrightarrow \Delta}{\Gamma,\ \neg\forall x\, \varphi \Longrightarrow \Delta}$$

$$\frac{\Gamma \Longrightarrow \varphi(t),\ \Delta}{\Gamma \Longrightarrow \exists x\, \varphi,\ \Delta} \qquad \frac{\Gamma \Longrightarrow \neg\varphi(v),\ \Delta}{\Gamma \Longrightarrow \neg\exists x\, \varphi,\ \Delta}$$

$$\frac{\Gamma,\ \varphi(v) \Longrightarrow \Delta}{\Gamma,\ \exists x\, \varphi \Longrightarrow \Delta} \qquad \frac{\Gamma,\ \neg\varphi(t) \Longrightarrow \Delta}{\Gamma,\ \neg\exists x\, \varphi \Longrightarrow \Delta}$$

Rules $\forall$-*right*, $\neg\forall$-*left*, $\exists$-*left*, and $\neg\exists$-*right* are subject to the eigenvariable condition: variable v cannot be free in the conclusion of the rule.

Essentially, the positive fragment of the calculus coincides with the calculus for classical logic, but rules for introduction of $\neg$ are replaced by a set of rules introducing negations of the individual positive connectives (plus introduction of double negation, and the *ex falso quodlibet* axiom).

6.2.7 Definition A sequent is *regular* if it does not contain free and bound occurrences of the same variable.

6.2.8 Theorem (completeness of $GK^\forall$) *A sequent is tautological iff it is provable in $GK^\forall$. Moreover, every regular tautological sequent has a $GK^\forall$-proof which does not use any cut or weakening inferences.*

Proof (sketch): The propositional case is again due to Avron [2]. Soundness of $GK^\forall$ is obvious. Completeness of $GK^\forall$ follows from straightforward simulation of $HK^\forall$, but this way we would not obtain the cut-elimination theorem.

Assume that a closed sequent $\Gamma_0 \Longrightarrow \Delta_0$ in a language L has no cut-free proof. Let $\{\langle\varphi_n, t_n\rangle;\ n \in \omega\}$ be a sequence with infinitely many appearances of every pair of a sentence and a close term in language $L' = L \cup \{c_n;\ n \in \omega\}$. We construct a sequence of cut-free unprovable sequents $\Gamma_n \Longrightarrow \Delta_n$ by induction on n. If the formula φ_n appears in $\Gamma_n \cup \Delta_n$, we put a "witness" for it in $\Gamma_{n+1} \Longrightarrow \Delta_{n+1}$. For example, let $\varphi_n = \neg(\alpha \vee \beta)$: if $\varphi_n \in \Gamma_n$, we define $\Gamma_{n+1} = \Gamma_n \cup \{\neg\alpha, \neg\beta\}$, $\Delta_{n+1} = \Delta_n$; if $\varphi_n \in \Delta_n$, we put $\Gamma_{n+1} = \Gamma_n$, and

$$\Delta_{n+1} = \begin{cases} \Delta_n \cup \{\neg\alpha\}, & \text{if } \Gamma_n \Longrightarrow \neg\alpha, \Delta_n \text{ has no cut-free proof,} \\ \Delta_n \cup \{\neg\beta\} & \text{otherwise.} \end{cases}$$

There are similar conditions for all connectives and quantifiers, and their negations. (Terms t_n and constants c_n are used for witnessing the quantifiers.)

We put $\Gamma = \bigcup_n \Gamma_n$, $\Delta = \bigcup_n \Delta_n$, and we take the term model for L' with satisfaction of *atomic* formulas defined by

$$v(\varphi) = \begin{cases} 1, & \text{if } \varphi \in \Gamma, \\ 0, & \text{if } \neg\varphi \in \Gamma, \\ * & \text{otherwise.} \end{cases}$$

We check by induction on complexity of a sentence φ that

$$\begin{aligned} \varphi \in \Gamma &\Rightarrow v(\varphi) = 1, \\ \varphi \in \Delta &\Rightarrow v(\varphi) \neq 1, \\ \neg\varphi \in \Gamma &\Rightarrow v(\varphi) = 0, \\ \neg\varphi \in \Delta &\Rightarrow v(\varphi) \neq 0, \end{aligned}$$

in particular we have a counterexample showing that $\Gamma_0 \Longrightarrow \Delta_0$ is not tautological. □

6.2.2 The theory C_2^1

6.2.9 Definition The first-order language L_{Cnt} is defined inductively:

(*i*) L_{Cnt} contains the language of PV,

(*ii*) if $\varphi(y, \vec{x})$ is an L_{Cnt}-formula, there is a predicate $Cnt_\varphi(b, c, d, \vec{x})$ in L_{Cnt}.

We will usually write $\mathbf{C}^b y < c\, \varphi(y, \vec{x})$ instead of $Cnt_\varphi(b, c, d, \vec{x})$ (d is implicit in this notation).

The intended meaning of $\mathbf{C}^b y < c\, \varphi(y)$ is an approximate counting quantifier. If the number of $y < c$ which satisfy φ is at least b, the quantifier is true; if it is at most $a := b - c/|d|$, the quantifier is false; otherwise it is $*$ ("undefined"). For convenience, we allow b or a to be negative integers (which are encoded in arithmetic in a straightforward way).

As we want to stay within the p-time hierarchy, we must not allow exact counting as a special case of approximate counting, the gap between a and b has to be at least a polynomial fraction of c. This motivates the curious indirect definition of a. (More precisely, some cases of exact counting are allowed. However if $b = a + 1$, we have $c \leq |d|$, i.e. c is sharply bounded.)

We make this intuition precise in the next definition.

6.2.10 Definition The *standard model for L_{Cnt}*, $\mathbb{N}$, is an expansion of the standard model of arithmetic such that

$$v^{\mathbb{N}}(Cnt_\varphi(b, c, d, \vec{x})) = \begin{cases} 1, & \text{if } |\{y < c;\ v^{\mathbb{N}}(\varphi(y, \vec{x})) = 1\}| \geq b, \\ 0, & \text{if } |\{y < c;\ v^{\mathbb{N}}(\varphi(y, \vec{x})) \neq 0\}| \leq b - c/|d|, \\ * & \text{otherwise.} \end{cases}$$

6.2.11 Definition The class of Σ^c_0*-formulas* contains all $\Sigma^b_0(PV)$-formulas, and it is closed under conjuction, disjunction, negation, sharply bounded quantifiers, and counting quantifiers.

Σ^c_1*-formulas* are built from Σ^c_0-formulas using disjuction, conjunction, bounded existential quantifiers, sharply bounded universal quantifiers, and counting quantifiers.

MA-formulas are built from Σ^c_0-formulas using disjuction, conjunction, bounded existential quantifiers, and sharply bounded universal quantifiers.

6.2.12 Definition *Promise problem* is a pair $L = \langle L^+, L^- \rangle$ of disjoint languages. Elements of L^+ and L^- are respectively called *positive* and *negative* instances (or inputs). A promise problem L is in *promise-BPP* (*prBPP*), if there is a probabilistic p-time Turing machine which accepts all positive inputs with probability at least 2/3, and rejects all negative inputs with probability at least 2/3. L is in *promise-MA* (*prMA*), if there is a predicate $P(x, y, z)$ decidable in deterministic time $|x|^{O(1)}$ such that

$$x \in L^+ \Rightarrow \exists y\ \mathrm{Pr}_z(P(x, y, z)) \geq 2/3,$$
$$x \in L^- \Rightarrow \forall y\ \mathrm{Pr}_z(P(x, y, z)) \leq 1/3.$$

L is in *promise-AM* (*prAM*), if there is a predicate $P(x, y, z)$ decidable in deterministic time $|x|^{O(1)}$ such that

$$x \in L^+ \Rightarrow \mathrm{Pr}_y(\exists z\, P(x, y, z)) \geq 2/3,$$
$$x \in L^- \Rightarrow \mathrm{Pr}_y(\exists z\, P(x, y, z)) \leq 1/3.$$

An ordinary language L is identified with the promise problem $\langle L, \{0,1\}^* \smallsetminus L\rangle$.

A promise problem L is *definable* by an L_{Cnt}-formula φ if

$$\begin{aligned} x \in L^+ &\Rightarrow v^{\mathbb{N}}(\varphi(x)) = 1, \\ x \in L^- &\Rightarrow v^{\mathbb{N}}(\varphi(x)) = 0. \end{aligned}$$

Note that the input of L must contain all free variables of φ, including those "hidden" in the implicit parameters d.

6.2.13 Theorem *Let L be a promise problem.*

(i) $L \in prBPP$ iff L is Σ^c_0-definable,

(ii) $L \in prAM$ iff L is Σ^c_1-definable.

(iii) $L \in prMA$ iff L is definable by an MA-formula.

Proof: We will show the right-to-left direction of (i) by induction on complexity. The base case is trivial, as Σ^c_0-formulas without counting quantifiers are $\Sigma^b_0(PV)$-formulas, thus decidable in deterministic polynomial time. Connectives correspond to

$$\begin{aligned} L_1 \cap L_2 &= \langle L_1^+ \cap L_2^+, L_1^- \cup L_2^- \rangle, \\ L_1 \cup L_2 &= \langle L_1^+ \cup L_2^+, L_1^- \cap L_2^- \rangle, \\ \overline{L} &= \langle L^-, L^+ \rangle, \end{aligned}$$

and it is easy to see that *prBPP* is closed under these operations. Sharply bounded quantifiers are similar. Consider a counting predicate $Cnt_\varphi(b, c, d)$. Notice that $prBPP^{prBPP} = prBPP$ by usual amplification methods, it thus suffices to express Cnt_φ as a *prBPP* problem with oracle access to φ. The algorithm simply picks a random $y \leq c$, and outputs the oracle answer to $\varphi(y, \vec{x})$. For positive inputs, the answer is 1 with probability at least b/c, and for negative inputs it is at most a/c. The gap is at least $1/|d|$, we may thus amplify the success probability to $2/3$ by using polynomially many samples.

For the left-to-right direction, let $P(x, z)$ be a PV-predicate such that

$$\begin{aligned} x \in L^+ &\Rightarrow \Pr_{|z| \leq p(|x|)}(P(x, z)) \geq 2/3, \\ x \in L^- &\Rightarrow \Pr_{|z| \leq p(|x|)}(P(x, z)) \leq 1/3, \end{aligned}$$

for a polynomial $p(n)$. Then L is definable by the formula

$$Cnt_{P(x,z)}(2t(x)/3, t(x), 4, x),$$

where $t(x) = 2^{p(|x|)}$.

The proofs of (ii) and (iii) are similar. □

6.2.14 Definition The theory C^1_2 in the language L_{Cnt} has the following axioms.

(i) equality axioms,

(ii) axioms of PV, and the law of excluded middle for atomic formulas,

(iii) Σ^c_1-*PIND*: for any Σ^c_1-formula φ,

$$\varphi(0) \wedge \forall u \leq x\,(\varphi(\lfloor \tfrac{u}{2} \rfloor) \rightarrow \varphi(u)) \rightarrow \varphi(x),$$

(iv) axioms about counting quantifiers ($\varphi, \psi \in \Sigma^c_1$, $\vartheta \in \Sigma^c_0$, f is a PV-function):

$$(1) \qquad \neg\, Cnt_{\neg\vartheta}(b, c, d, \vec{u}) \leftrightarrow \mathbf{C}^{c-b+\lceil c/|d| \rceil} x < c\, \vartheta(x, \vec{u}),$$

$$(2) \quad \forall v < c\,(\varphi(v) \vee \exists y < c-b\, f(y) = v) \wedge b \leq c \rightarrow \mathbf{C}^{b-\lfloor c/|e| \rfloor} x < c\, \varphi(x),$$

(3)
$$\forall x < c\, \exists y < z\, f(y) = x \wedge \mathbf{C}^b x < c\, \varphi(x) \rightarrow \mathbf{C}^{b-\lfloor z/|e| \rfloor} y < z\, \varphi(f(y)),$$

$$(4) \quad \forall x < c\, \exists i < |z|\, \varphi(x, i)\ \&\ \sum_{i<|z|} (b)_i + \lceil c/|e| \rceil \leq c + |z|$$
$$\rightarrow \exists i < |z|\, \mathbf{C}^{(b)_i} x < c\, \varphi(x, i),$$

$$(5) \quad \mathbf{C}^b x < c\, \varphi(x) \wedge \mathbf{C}^{b'} x < c\, \psi(x) \rightarrow \mathbf{C}^{b+b'-c} x < c\, (\varphi(x) \wedge \psi(x)),$$

$$(6) \quad \mathbf{C}^b x < c\, \varphi(x) \wedge \mathbf{C}^{b'} x < c'\, \psi(x) \rightarrow$$
$$\mathbf{C}^{b+b'} x < c + c'\, ((x < c \wedge \varphi(x)) \vee (x \geq c \wedge \psi(x - c))),$$

$$(7) \quad \mathbf{C}^b x < c\, \mathbf{C}^{b'} y < c'\, \varphi(x + yc) \wedge b \geq 0 \wedge b' \geq 0 \rightarrow \mathbf{C}^{bb'} v < cc'\, \varphi(v),$$

$$(8) \quad \mathbf{C}^b x < cz\, \varphi(x \bmod c) \vee \mathbf{C}^b x < cz\, \varphi(\lfloor \tfrac{x}{z} \rfloor) \rightarrow \mathbf{C}^{\lceil b/z \rceil} y < c\, \varphi(y),$$

$$\text{(9)}\qquad \mathbf{C}^{bc'+b'c-bb'}v < cc'\,\varphi(v) \to \mathbf{C}^{b}x < c\,\mathbf{C}^{b'}y < c'\,\varphi(x+yc),$$

and the following derivation rule, for $\chi \in \Sigma_1^c$:

$$\text{(10)}\qquad \frac{\chi \to \forall x < c\,(\varphi(x) \to \psi(x))}{\chi \to (\mathbf{C}^b x < c\,\varphi(x) \to \mathbf{C}^b x < c\,\psi(x))}.$$

6.2.15 Remark Axiom (ii) actually implies the law of excluded middle for all formulas which do not contain counting quantifiers, thus C^1_2 extends the classical theory $S^1_2(PV)$. In fact, it contains $S^1_2 + dWPHP(PV)$, which is easily seen from axiom (2).

The "error terms" in axioms (2) and (3), and the formulation of (10) as a rule rather than axiom, are needed for witnessing.

6.2.16 Definition The theory C^1_2 can be reformulated as a sequent calculus GC^1_2, extending $GK^\forall$. We add new initial sequents for the open axioms (i) and (ii). Induction (iii) corresponds to the rule

$$\frac{\Gamma,\ \varphi(\lfloor \tfrac{v}{2} \rfloor) \Longrightarrow \varphi(v),\ \Delta}{\Gamma,\ \varphi(0) \Longrightarrow \varphi(t),\ \Delta}\,,$$

subject to eigenvariable condition for v. Axioms (1) and (5)–(9), which have the form $\varphi_1 \to \varphi_2$, are given as initial sequents $\Gamma, \varphi_1 \Longrightarrow \varphi_2, \Delta$. (Axioms (1) and (8) are rewritten as two sequents.) The remaining axioms are given as

$$\text{(2)}\,\frac{\Gamma,\ v < c \Longrightarrow \varphi(v),\ \exists y < c - b\, f(y) = v,\ \Delta}{\Gamma,\ b \le c \Longrightarrow \mathbf{C}^{b-\lfloor c/|e| \rfloor} x < c\,\varphi(x),\ \Delta}\,,$$

$$\text{(3)}\,\frac{\Gamma,\ v < c \Longrightarrow \exists y < z\, f(y) = v,\ \Delta}{\Gamma,\ \mathbf{C}^b x < c\,\varphi(x) \Longrightarrow \mathbf{C}^{b-\lfloor z/|e| \rfloor} y < z\,\varphi(f(y)),\ \Delta}\,,$$

$$\text{(4)}\,\frac{\Gamma,\ v < c \Longrightarrow \exists i < |z|\,\varphi(x,\ i),\ \Delta}{\Gamma,\ \sum_{i<|z|}(b)_i + \lceil c/|e| \rceil \le c + |z| \Longrightarrow \exists i < |z|\,\mathbf{C}^{(b)_i} x < c\,\varphi(x,\ i),\ \Delta}\,,$$

$$\text{(10)}\,\frac{\Lambda,\ v < c,\ \varphi(v) \Longrightarrow \psi(v)}{\Lambda,\ \mathbf{C}^b x < c\,\varphi(x) \Longrightarrow \mathbf{C}^b x < c\,\psi(x)}\,,$$

with EVC on v, and φ, ψ, Λ restricted to Σ^c_1-formulas.

We also add rules for bounded quantifiers, which are no longer seen as abbreviations. The rules from definition 7.1.1 of [20] will do, except that we have to include also rules for negated quantifiers, as in $GK^{\forall}$.

The notion of a principal formula of a rule is defined as usual, except that the formulas from Λ in rule (10) are also considered principal.

6.2.17 Lemma *$C^1_2 \vdash \varphi$ iff the sequent $\Longrightarrow \varphi$ is provable in GC^1_2.*

Proof: Straightforward. □

6.2.18 Theorem *A regular sequent provable in GC^1_2 has a GC^1_2-proof, in which every cut-formula has an identical ancestor which is a principal formula of one of the special rules from 6.2.16.*

Proof (sketch): We modify the proof of 6.2.8 so that the resulting model satisfies C^1_2. For example, consider an instance of axiom (2). At certain stage in the construction, we put

$$\begin{aligned} \Gamma_{n+1} &= \Gamma_n \cup \{\mathbf{C}^{b-\lfloor c/|e|\rfloor} x < c\,\varphi(x)\}, \\ \Delta_{n+1} &= \Delta_n, \end{aligned}$$

if the sequent $\Gamma_n, \mathbf{C}^{b-\lfloor c/|e|\rfloor} x < c\,\varphi(x) \Longrightarrow \Delta_n$ is unprovable,

$$\begin{aligned} \Gamma_{n+1} &= \Gamma_n, \\ \Delta_{n+1} &= \Delta_n \cup \{b \leq c\}, \end{aligned}$$

if the sequent $\Gamma_n \Longrightarrow b \leq c, \Delta_n$ is unprovable, and

$$\begin{aligned} \Gamma_{n+1} &= \Gamma_n \cup \{c_k < c\}, \\ \Delta_{n+1} &= \Delta_n \cup \{\varphi(c_k), \exists y < c - b\, f(y) = c_k\} \end{aligned}$$

otherwise, where c_k is the first unused constant. In either case, the sequent $\Gamma_{n+1} \Longrightarrow \Delta_{n+1}$ is not provable: if it were, we could derive (putting $b' = b - \lfloor c/|e|\rfloor$)

$$\frac{\Gamma_n,\ v < c \Longrightarrow \varphi(v),\ \exists y < c - b\, f(y) = v,\ \Delta_n}{\Gamma_n,\ b \leq c \Longrightarrow \mathbf{C}^{b'} x < c\,\varphi(x),\ \Delta_n}$$

$$\frac{\Gamma_n,\ \mathbf{C}^{b'} x < c\,\varphi(x) \Longrightarrow \Delta_n \qquad \Gamma_n,\ b \leq c \Longrightarrow \mathbf{C}^{b'} x < c\,\varphi(x),\ \Delta_n}{\Gamma_n,\ b \leq c \Longrightarrow \Delta_n}$$

$$\frac{\Gamma_n \Longrightarrow b \leq c,\ \Delta_n \qquad \Gamma_n,\ b \leq c \Longrightarrow \Delta_n}{\Gamma_n \Longrightarrow \Delta_n}$$

by the rule for (2) and two cuts, contradicting the unprovability of $\Gamma_n \Longrightarrow \Delta_n$.

The final model thus either satisfies $\mathbf{C}^{b-\lfloor c/|e|\rfloor}x < c\,\varphi(x)$, or does not satisfy $b \leq c$, or contains an element (namely c_k) which does not satisfy $v < c \rightarrow (\varphi(v) \vee \exists y < c - b\, f(y) = v)$. In any case, the implication

$$\forall x < c\,(\varphi(x) \vee \exists y < c - b\, f(y) = x) \wedge b \leq c \rightarrow \mathbf{C}^{b-\lfloor c/|e|\rfloor}x < c\,\varphi(x)$$

is satisfied.

Other rules are treated in a similar way, with the exception of rule (10), which requires an extra complication: we must proceed by induction on the depth of nesting of rules (10) in the proof. □

6.2.3 A witnessing theorem for C^1_2

6.2.19 Definition For every Σ^c_1-formula $\varphi(\vec{x})$, we define a new Σ^c_0-formula $Wit_\varphi(e,\vec{x})$ by induction on complexity (we assume that negations are pushed down by de Morgan rules):

(*i*) $Wit_\varphi = \varphi$, if $\varphi \in \Sigma^b_0$, or the topmost symbol of φ is a counting quantifier or its negation,

(*ii*) $Wit_{\varphi\wedge\psi}(e,\vec{x}) = Wit_\varphi((e)_0,\vec{x}) \wedge Wit_\psi((e)_1,\vec{x})$.

(*iii*) $Wit_{\varphi\vee\psi}(e,\vec{x}) = ((e)_0 = 0 \wedge Wit_\varphi((e)_1,\vec{x})) \vee ((e)_0 = 1 \wedge Wit_\psi((e)_1,\vec{x}))$.

(*iv*) $Wit_{\forall u\leq|t|\,\varphi}(e,\vec{x}) = \forall u \leq |t|\ Wit_\varphi((e)_u,u,\vec{x})$.

(*v*) $Wit_{\exists u\leq t\,\varphi} = (e)_0 \leq t \wedge Wit_\varphi((e)_1,(e)_0,\vec{x})$.

Any e such that $Wit_\varphi(e,\vec{x})$ holds is called a *witness* for $\varphi(\vec{x})$.

A *witnessing function* for a sequent $\varphi_1,\dots,\varphi_n \Longrightarrow \psi_1,\dots,\psi_m$ is a probabilistic algorithm $f(r,e_1,\dots,e_n,\vec{x})$ such that if e_i, $i = 1,\dots,n$ are witnesses for $\varphi_i(\vec{x})$, f outputs with probability at least $1-1/r$ a pair $\langle j,e\rangle$ such that e is a witness for $\psi_j(\vec{x})$ (in the standard model).

6.2.20 Theorem *Let $\Gamma \Longrightarrow \Delta$ be a sequent consisting of Σ^c_1-formulas. If $\Gamma \Longrightarrow \Delta$ is provable in C^1_2, it has a polynomial-time probabilistic witnessing function.*

Proof: By theorem 6.2.18 and (a suitable version of) the subformula property, $\Gamma \Longrightarrow \Delta$ has a GC^1_2-proof which contains only Σ^c_1-formulas. We will construct the witnessing function by induction on the length of the proof.

Initial sequents are witnessed trivially, as their principal formulas are Σ^b_0 or start with a counting quantifier. A similar argument works for rule (10), as its conclusion has only one formula in the succedent.

Consider a $\wedge$-*right* inference

$$\frac{\Gamma \Longrightarrow \varphi,\ \Delta \qquad \Gamma \Longrightarrow \psi,\ \Delta}{\Gamma \Longrightarrow \varphi \wedge \psi,\ \Delta}\ ,$$

and assume that f_1 and f_2 witness the assumptions of the rule. We construct a witnessing function for the conclusion as follows: apply f_1 and f_2. If one of them outputs a witness to some formula in Δ, return it; otherwise we have witnesses e and e' to the formulas φ and ψ, and we return $\langle e, e' \rangle$.

Assume that f witnesses the assumption of a $\wedge$-*left* inference

$$\frac{\Gamma,\ \varphi,\ \psi \Longrightarrow \Delta}{\Gamma,\ \varphi \wedge \psi \Longrightarrow \Delta}\ ,$$

To witness the conclusion, we simply take the input corresponding to $\varphi \wedge \psi$, decompose it into witnesses for φ and ψ, and feed them to f.

The other logical rules are treated similarly.

Let f witness the assumption of an induction rule

$$\frac{\Gamma,\ \varphi(\lfloor \frac{v}{2} \rfloor) \Longrightarrow \varphi(v),\ \Delta}{\Gamma,\ \varphi(0) \Longrightarrow \varphi(t),\ \Delta}\ .$$

To witness the conclusion, we iterate f $|t|$-times to construct witnesses for $\varphi(\lfloor \frac{t}{2^i} \rfloor)$, $i = |t|, \dots, 0$, starting with a given witness to $\varphi(0)$. We either obtain a witness to $\varphi(t)$, or get a witness to some formula from Δ in the process.

Consider an inference

$$(2)\ \frac{\Gamma,\ v < c \Longrightarrow \varphi(v),\ \exists y < c - b\, g(y) = v,\ \Delta}{\Gamma,\ b \le c \Longrightarrow \mathbf{C}^{b - \lfloor c/|z| \rfloor} x < c\, \varphi(x),\ \Delta}\ ,$$

with f witnessing its assumption. Given witnesses to Γ, we choose independently $O(|z|)$ numbers $v_i < c$, and apply f to them. If we obtain a witness to Δ, we are done; otherwise all v_i happened to satisfy φ or to be in the range of g. The probability of v *not* being in $\mathrm{rng}(g)$ is at least b/c, which means that $\mathbf{C}^{b - \lfloor c/|z| \rfloor} x < c\, \varphi(x)$ holds with high probability, and we may output anything as a witness to the last formula.

A similar argument works for rule (3).

Finally, assume that f witnesses the assumption of

$$(4)\ \frac{\Gamma,\ v < c \Longrightarrow \exists i < |z|\, \varphi(x,\ i),\ \Delta}{\Gamma,\ \sum_{i<|z|}(b)_i + \lceil c/|e|\rceil \leq c + |z| \Longrightarrow \exists i < |z|\, \mathbf{C}^{(b)_i} x < c\, \varphi(x,\ i),\ \Delta}\ .$$

We witness the conclusion as follows: choose $k = O((|z||e|)^2)$ numbers $v_j < c$, and apply f. If we get a witness to Δ, we are done. Otherwise we find $i < |z|$ such that we got at least $k(((b)_i - 1)/c + 1/(|z||e|))$ witnesses to $\varphi(u_j, i)$, and we output i as a witness to $\exists i < |z|\, \mathbf{C}^{(b)_i} x < c\, \varphi(x, i)$. Chernoff's inequality guarantees that this choice is correct with high probability. □

Bibliography

[1] Miklós Ajtai, *The complexity of the pigeonhole principle*, in: Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science, 1988, pp. 346–355.

[2] Arnon Avron, *Natural* 3*-valued logics—characterization and proof theory*, Journal of Symbolic Logic 56 (1991), no. 1, pp. 276–294.

[3] Matthias Baaz, *Infinite-valued Gödel logics with 0-1-projections and relativizations*, in: Proceedings of Gödel '96, Logic Foundations of Mathematics, Computer Science and Physics—Kurt Gödel's Legacy (P. Hájek, ed.), Lecture Notes in Logic vol. 6, Springer, 1996, pp. 23–33.

[4] Stuart J. Berkowitz, *On computing the determinant in small parallel time using a small number of processors*, Information Processing Letters 18 (1984), no. 3, pp. 147–150.

[5] Samuel R. Buss, *Bounded arithmetic*, Bibliopolis, Naples, 1986.

[6] ———, *Axiomatizations and conservation results for fragments of bounded arithmetic*, in: Logic and Computation, Proceedings of a Workshop held at Carnegie Mellon University (W. Sieg, ed.), Contemporary Mathematics vol. 106, American Mathematical Society, 1990, pp. 57–84.

[7] Samuel R. Buss, Russell Impagliazzo, Jan Krajíček, Pavel Pudlák, Alexander A. Razborov, and Jiří Sgall, *Proof complexity in algebraic systems and bounded depth Frege systems with modular counting*, Computational Complexity 6 (1997), no. 3, pp. 256–298.

[8] Alan Cobham, *The intrinsic computational difficulty of functions*, in: Proceedings of the 2nd International Congress of Logic, Methodology and Philosophy of Science (Y. Bar-Hillel, ed.), North Holland, 1965, pp. 24–30.

[9] Stephen A. Cook, *Feasibly constructive proofs and the propositional calculus*, in: Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, 1975, pp. 83–97.

[10] Stephen A. Cook and Robert A. Reckhow, *The relative efficiency of propositional proof systems*, Journal of Symbolic Logic 44 (1979), no. 1, pp. 36–50.

[11] Haim Gaifman and Constantinos Dimitracopoulos, *Fragments of Peano's arithmetic and the MRDP theorem*, in: Logic and algorithmic, Monographie de L'Enseignement Mathématique no. 30, Université de Genève, 1982, pp. 187–206.

[12] Petr Hájek and Pavel Pudlák, *Metamathematics of first-order arithmetic*, Perspectives in Mathematical Logic, Springer, 1993, second edition 1998.

[13] Armin Haken, *The intractability of resolution*, Theoretical Computer Science 39 (1985), pp. 297–308.

[14] Russell Impagliazzo and Bruce M. Kapron, *Logics for reasoning about cryptographic constructions*, in: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, 2003, pp. 372–383.

[15] Russell Impagliazzo and Avi Wigderson, *P=BPP unless E has subexponential circuits: Derandomizing the XOR Lemma*, in: Proceedings of the 29th Annual ACM Symposium on Theory of Computing, 1997, pp. 220–229.

[16] Emil Jeřábek, *Dual weak pigeonhole principle, Boolean complexity, and derandomization*, Annals of Pure and Applied Logic 129 (2004), pp. 1–37.

[17] Cliff B. Jones, *Systematic software development using VDM*, Prentice-Hall International, UK, 1990.

[18] Valentine Kabanets and Russell Impagliazzo, *Derandomizing polynomial identity tests means proving circuit lower bounds*, Electronic Colloquium on Computational Complexity 9 (2002), no. 55.

[19] Stephen C. Kleene, *Introduction to metamathematics*, D. Van Nostrand, Princeton, New Jersey, 1950.

[20] Jan Krajíček, *Bounded arithmetic, propositional logic, and complexity theory*, Encyclopedia of Mathematics and Its Applications vol. 60, Cambridge University Press, 1995.

[21] ———, *On the weak pigeonhole principle*, Fundamenta Mathematicae 170 (2001), pp. 123–140.

[22] Jan Krajíček and Pavel Pudlák, *Quantified propositional calculi and fragments of bounded arithmetic*, Zeitschrift für mathematische Logik und Grundlagen der Mathematik 36 (1990), no. 1, pp. 29–46.

[23] ———, *Some consequences of cryptographical conjectures for S^1_2 and EF*, Information and Computation 140 (1998), no. 1, pp. 82–94.

[24] Jan Krajíček, Pavel Pudlák, and Gaisi Takeuti, *Bounded arithmetic and the polynomial hierarchy*, Annals of Pure and Applied Logic 52 (1991), no. 1–2, pp. 143–153.

[25] Rudolf Lidl and Harald Niederreiter, *Introduction to finite fields and their applications*, Cambridge University Press, 1986, second edition 1996.

[26] Jan Łukasiewicz, *O logice trójwartościowej*, Ruch Filozoficzny 5 (1920), pp. 170–171.

[27] ———, *The shortest axiom of the implicational calculus of propositions*, Proceedings of the Royal Irish Academy 52 (1948), no. A 3, pp. 25–33.

[28] Alexis Maciel, Toniann Pitassi, and Alan R. Woods, *A new proof of the weak pigeonhole principle*, Journal of Computer and System Sciences 64 (2002), no. 4, pp. 843–872.

[29] Gary L. Miller, *Riemann's hypothesis and tests for primality*, Journal of Computer and System Sciences 13 (1976), no. 3, pp. 300–317.

[30] David E. Muller, *Application of Boolean algebra to switching circuit design and to error detection*, IEEE Transactions on Computers 3 (1954), pp. 6–12.

[31] Noam Nisan and Avi Wigderson, *Hardness vs. randomness*, Journal of Computer and System Sciences 49 (1994), no. 2, pp. 149–167.

[32] Kerry E. Ojakian, *Combinatorics in bounded arithmetic*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, 2004.

[33] Christos H. Papadimitriou, *Computational complexity*, Addison-Wesley, 1994.

[34] Rohit Parikh, *Existence and feasibility in arithmetic*, Journal of Symbolic Logic 36 (1971), no. 3, pp. 494–508.

[35] Jeff B. Paris, Alex J. Wilkie, and Alan R. Woods, *Provability of the pigeonhole principle and the existence of infinitely many primes*, Journal of Symbolic Logic 53 (1988), no. 4, pp. 1235–1244.

[36] Michael O. Rabin, *Probabilistic algorithm for testing primality*, Journal of Number Theory 12 (1980), no. 1, pp. 128–138.

[37] Irving S. Reed, *A class of multiple-error-correcting codes and the decoding scheme*, IEEE Transactions on Information Theory 4 (1954), pp. 38–49.

[38] Irving S. Reed and Gustave Solomon, *Polynomial codes over certain finite fields*, SIAM Journal of Applied Mathematics 8 (1960), pp. 300–304.

[39] Jean-Pierre Ressayre, *A conservation result for system of bounded arithmetic*, unpublished manuscript, 1986.

[40] Claude E. Shannon, *The synthesis of two-terminal switching circuits*, Bell System Technical Journal 28 (1949), no. 1, pp. 59–98.

[41] Michael Sipser, *A complexity theoretic approach to randomness*, in: Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 1983, pp. 330–335.

[42] Michael Soltys-Kulinicz, *The complexity of derivations of matrix identities*, Ph.D. thesis, University of Toronto, Department of Mathematics, 2001.

[43] Michael Soltys and Stephen A. Cook, *The proof complexity of linear algebra*, in: Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science, 2002, pp. 335–344.

[44] __________, *The proof complexity of linear algebra*, Annals of Pure and Applied Logic 130 (2004), pp. 277–323.

[45] Madhu Sudan, *Decoding Reed-Solomon codes beyond the error-correction diameter*, in: Proceedings of the 35th Annual Allerton Conference on Communication, Control and Computing, 1997, pp. 215–224.

[46] Madhu Sudan, Luca Trevisan, and Salil Vadhan, *Pseudorandom generators without the XOR lemma*, Journal of Computer and System Sciences 62 (2001), no. 2, pp. 236–266.

[47] Neil Thapen, *A model-theoretic characterization of the weak pigeonhole principle*, Annals of Pure and Applied Logic 118 (2002), no. 1–2, pp. 175–195.

[48] ———, *The weak pigeonhole principle in models of bounded arithmetic*, Ph.D. thesis, Oxford University, 2002.

[49] Neil Thapen and Michael Soltys, *Weak theories of linear algebra*, Archive for Mathematical Logic (2004), to appear.

[50] Seinosuke Toda, *On the computational power of PP and $\oplus P$*, in: Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science, 1989, pp. 514–519.

Appendix A

Some bounds on binomial coefficients

Here we show that several well-known inequalities, useful for counting, are provable in PV when their parameters are restricted to logarithmically small numbers. One cannot avoid using these technical results, and although it is not unexpected that the formalization is provable in PV, the only way how to verify this seems to be to actually prove them in PV.

A.1 Definition Let $n \in Log$, $k, i \le n$. Define

$$\begin{bmatrix} n \\ i \end{bmatrix}_k := \binom{n}{i} k^i (n-k)^{n-i},$$

$\binom{n}{<i} := \sum_{j<i} \binom{n}{j}$, $\begin{bmatrix} n \\ <i \end{bmatrix}_k := \sum_{j<i} \begin{bmatrix} n \\ j \end{bmatrix}_k$.

A.2 Theorem (Stirling's bound) *There is a $c > 1$ such that PV proves*

$$0 < k < n \in Log \;\; \rightarrow \;\; \frac{1}{c}\binom{n}{k} \le \frac{n^n}{k^k (n-k)^{n-k}} \sqrt{\frac{n}{k(n-k)}} \le c\binom{n}{k}.$$

(*We will abbreviate this as "*$\binom{n}{k} = \Theta(\cdots)$*."*)

Proof: Define $f(i) := \begin{bmatrix} n \\ i \end{bmatrix}_k$, and

$$\gamma(i) := f(i+1)/f(i) = k(n-i)/((n-k)(i+1)).$$

We have $j < i \rightarrow \gamma(j) > \gamma(i)$, because $(j+1)(n-i) < (i+1)(n-j)$. Also

$$\gamma(k-1) = (n-k+1)/(n-k) > 1 > k/(k+1) = \gamma(k),$$

hence $f(i+1) > f(i)$ for $i < k$, and $f(i+1) < f(i)$ for $i \ge k$.

Let $i < k$. We have $k(n-i+1)f(j-1) \leq (n-k)i\,f(j)$ for any $0 < j \leq i$, hence

$$(k(n-i+1))^\ell f(i-\ell) \leq ((n-k)i)^\ell f(i)$$

for any $0 \leq \ell \leq i$ (by Δ^b_1-induction on ℓ). Using induction once again, we find that

$$\begin{aligned}(k(n-i+1))^\ell(k(n-i+1)-(n-k)i)\sum_{j=i-\ell}^{i-1} f(j) \leq&\\ \leq (n-k)i((k(n-i+1))^\ell - ((n-k)i)^\ell)f(i),&\end{aligned}$$

in particular,

$$\begin{aligned}(k(n-i+1))^i(kn+k-in)\sum_{j<i} f(j)&\\ &\leq (n-k)i((k(n-i+1))^i - ((n-k)i)^i)f(i)\\ &\leq (n-k)i(k(n-i+1))^i f(i),\end{aligned}$$

hence

$$\sum_{j<i} f(j) \leq \frac{(n-k)i}{kn+k-in} f(i) \leq \frac{(n-k)i}{n(k-i)} f(i).$$

Similarly,

$$\sum_{j>i} f(j) \leq \frac{k(n-i)}{n(i-k)} f(i) \qquad \text{for any } i > k.$$

Put $s := \left\lfloor \sqrt{\frac{k(n-k)}{n}} \right\rfloor$. Then

$$\begin{aligned}\frac{1}{f(k-s-1)}\sum_{j<k-s-1} f(j) &\leq \frac{n-k}{n}\left(\frac{k}{s+1}-1\right)\\ &\leq \frac{n-k}{n}\left(\sqrt{\frac{kn}{n-k}}-1\right)\\ &= \sqrt{\frac{k(n-k)}{n}} - \frac{n-k}{n} \leq s + \frac{k}{n},\end{aligned}$$

hence

$$\begin{aligned}\sum_{j\leq k} f(j) \leq \sum_{j<k-s-1} f(j) + \sum_{j=k-s-1}^{k} f(j)&\\ \leq \left(s+\frac{k}{n}\right) f(k-s-1) + (s+2)f(k) \leq \left(2s+2+\frac{k}{n}\right) f(k).&\end{aligned}$$

Similarly we may show

$$\sum_{j>k} f(j) \leq \left(2s+2-\frac{k}{n}\right) f(k),$$

hence

$$n^n = \sum_{j=0}^{n} f(j) \leq 4(s+1)f(k) \leq 8sf(k),$$

in other words

$$\binom{n}{k} \geq \frac{n^n}{8sk^k(n-k)^{n-k}} \geq \frac{n^n}{8k^k(n-k)^{n-k}}\sqrt{\frac{n}{k(n-k)}}.$$

Claim 1 *PV proves*

$$b \in \mathit{Log},\ \ b>0 \quad \to \quad (b+1)^{b+1} \leq 4b^{b+1}.$$

Proof: By induction on b. The claim holds if $b=1$. Assume $b>1$ and $b^b \leq 4(b-1)^b$. Straightforward induction on d shows that

$$c^d \leq (c+1)^d - d(c+1)^{d-1} + \binom{d}{2}(c+1)^{d-2}, \qquad d \geq 2,$$

hence

$$\begin{aligned}(b-1)^{b+1}(b+1)^{b+1} &= (b^2-1)^{b+1} \\ &\leq b^{2b+2} - (b+1)b^{2b} + \frac{b^2+b}{2}b^{2b-2} \\ &= b^{2b+2} - b^{2b-1} - \frac{1}{2}b^{2b} + \frac{1}{2}b^{2b-1} \\ &\leq b^{2b+2} - b^{2b+1} = (b-1)b^{b+1}b^b \\ &\leq 4(b-1)^{b+1}b^{b+1},\end{aligned}$$

thus $(b+1)^{b+1} \leq 4b^{b+1}$. □ (claim 1)

Claim 2 *PV proves*

$$a, b \in \mathit{Log},\ \ b>0 \quad \to \quad (b+a)^b \leq 4^a b^b.$$

Proof: The case $a=0$ is trivial. If $a=1$, we have $(b+1)^{b+1} \leq 4b^{b+1} \leq 4(b+1)b^b$ by previous claim, hence $(b+1)^b \leq 4b^b$. We proceed by induction on a. Using the induction hypothesis for a and 1, we have

$$(b+a+1)^{b+a} \leq 4(b+a)^{b+a} \leq 4^{a+1}b^b(b+a)^a \leq 4^{a+1}b^b(b+a+1)^a,$$

hence $(b+a+1)^b \leq 4^{a+1}b^b$. □ (claim 2)

Let $i \le s$. Then $in(i-1) \le k(n-k)$, hence

$$\gamma(k-i) = \frac{k(n-k+i)}{(n-k)(k-i+1)} = 1 + \frac{k+n(i-1)}{(n-k)(k-i+1)} \le 1 + \frac{in}{k(n-k)}.$$

Since (assuming i even) $f(k-i/2) \le f(k-i)\gamma^{i/2}(k-i)$, this implies

$$(k(n-k))^{i/2} f(k-i/2) \le (k(n-k)+in)^{i/2} f(k-i),$$

and, using claim 2,

$$\begin{aligned}(k(n-k))^{\frac{i}{2}k(n-k)}&(f(k-i/2))^{k(n-k)}\\ &\le (k(n-k)+in)^{\frac{i}{2}k(n-k)}(f(k-i))^{k(n-k)}\\ &\le (4^{in}(k(n-k))^{k(n-k)})^{i/2}(f(k-i))^{k(n-k)}\\ &= (k(n-k))^{\frac{i}{2}k(n-k)}2^{i^2n}(f(k-i))^{k(n-k)},\end{aligned}$$

hence

$$(f(k-i/2))^{k(n-k)} \le 2^{i^2n}(f(k-i))^{k(n-k)}.$$

Choose ℓ such that $2^\ell \le s < 2^{\ell+1}$. Then $4^\ell \le k(n-k)/n$, and an induction shows that

$$\begin{aligned}(f(k-1))^{k(n-k)} &\le (f(k-2^\ell))^{k(n-k)}2^{\frac{4}{3}(4^\ell-1)n}\\ &\le (f(k-2^\ell))^{k(n-k)}2^{\frac{4}{3}k(n-k)} \le (3f(k-2^\ell))^{k(n-k)},\end{aligned}$$

hence $f(k-2^\ell) \ge f(k-1)/3 \ge f(k)/6$. This implies

$$n^n \ge \sum_{j=1}^{2^\ell} f(k-j) \ge 2^\ell f(k-2^\ell) \ge \frac{2^\ell}{6}f(k) \ge \frac{s+1}{12}f(k),$$

which means

$$\binom{n}{k} \le \frac{12n^n}{(s+1)k^k(n-k)^{n-k}} \le \frac{12n^n}{k^k(n-k)^{n-k}}\sqrt{\frac{n}{k(n-k)}}. \qquad \square$$

A.3 Corollary *PV proves: for any $0 < k < n \in Log$,*

$$|k-i| \le \sqrt{\frac{k(n-k)}{n}} \quad \to \quad \begin{bmatrix} n \\ i \end{bmatrix}_k = \Theta\left(\begin{bmatrix} n \\ k \end{bmatrix}_k\right).$$

(*Here $|\cdot|$ denotes absolute value, not the length function.*) □

A.4 Theorem *The following is provable in PV. Let $k, n \in Log$ be such that $n > k > 0$, and denote $s = \sqrt{\frac{k(n-k)}{n}}$.*

(*i*) *Assume* $i \le s$. *Then*

$$\left[{n \atop < k-i}\right]_k = \Theta\left(s\left[{n \atop k-i}\right]_k\right) = \Theta(n^n),$$

$$\left[{n \atop > k+i}\right]_k = \Theta\left(s\left[{n \atop k+i}\right]_k\right) = \Theta(n^n).$$

(*ii*) *Assume* $i \ge s$.

$$\left[{n \atop < k-i}\right]_k = \Theta\left(\left(1-\frac{k}{n}\right)\left(\frac{k}{i}-1\right)\left[{n \atop k-i}\right]_k\right),$$

$$\left[{n \atop > k+i}\right]_k = \Theta\left(\frac{k}{n}\left(\frac{n-k}{i}-1\right)\left[{n \atop k+i}\right]_k\right).$$

Proof: It suffices to show the $\left[{n \atop <\cdots}\right]$-part, as $\left[{n \atop j}\right]_k = \left[{n \atop n-j}\right]_{n-k}$.

First assume $i \le s$. We already know from the proof of A.2 that

$$\left[{n \atop < k-i}\right]_k \le \left[{n \atop < k}\right]_k = O\left(s\left[{n \atop k}\right]_k\right) = O\left(s\left[{n \atop k-i}\right]_k\right) = O(n^n).$$

If $i \le s/2$, we also have

$$\left[{n \atop < k-i}\right]_k \ge \left[{n \atop < k-s/2}\right]_k \ge \frac{s}{2}\left[{n \atop k-s}\right]_k = \Omega\left(s\left[{n \atop k-i}\right]_k\right) = \Omega(n^n).$$

The case of $s/2 < i \le s$ is treated similarly: the proof of $\left[{n \atop k-s}\right]_k = \Omega\left(\left[{n \atop k}\right]_k\right)$ can be easily adapted to $\left[{n \atop k-2s}\right]_k$.

Now assume $k \ge i > s$. We have already proved that

$$\left[{n \atop < k-i}\right]_k \le \frac{(n-k)(k-i)}{ni}\left[{n \atop k-i}\right]_k.$$

If $i = k$, clearly $\left[{n \atop < k-i}\right]_k = 0 = \frac{k}{i} - 1$. If $k > i \ge k/4$, we have

$$\begin{aligned}\frac{\left[{n \atop < k-i}\right]_k}{\left[{n \atop k-i}\right]_k} &\ge \frac{1}{\gamma(k-i-1)} = \frac{(n-k)(k-i)}{k(n-k+i+1)}\\ &\ge \frac{(n-k)(k-i)}{kn} \ge \frac{(n-k)(k-i)}{4ni}.\end{aligned}$$

Let $k/4 > i > s$, and define f and γ as in the proof of A.2. By the monotonicity of γ and simple induction, we have

$$f(k-i-j) \ge f(k-i)(\gamma(k-2i))^{-j},$$

hence (putting $\gamma = \gamma(k-2i)$)

$$\begin{aligned}\begin{bmatrix} n \\ < k-i \end{bmatrix}_k &\geq \sum_{j=1}^{i} f(k-i-j) \geq f(k-i)\sum_{j=1}^{i}\gamma^{-j} \\ &= f(k-i)\frac{1}{\gamma-1}(1-\gamma^{-i}) \\ &= f(k-i)\frac{(n-k)(k-2i+1)}{n(2i-1)+k}\left(1-\left(\frac{(n-k)(k-2i+1)}{k(n-k+2i)}\right)^i\right).\end{aligned}$$

Notice that

$$\begin{aligned}\frac{(n-k)(k-2i+1)}{n(2i-1)+k} &\geq \frac{(n-k)(k-2i+1)}{2ni} \\ &\geq \frac{(n-k)k}{4ni} \geq \frac{(n-k)(k-i)}{4ni}.\end{aligned}$$

Claim 1 *$b^{a+b}2^a \leq (a+b)^{a+b}$ for any $a, b \in Log$.*

Proof: Case $a = 0$ is trivial. If $a = 1$, we have

$$(b+1)^{b+1} \geq b^{b+1} + (b+1)b^b \geq 2b^{b+1}.$$

Proceed by induction on a. Assuming the hypothesis for a, we have

$$b^{a+b+1}2^{a+1} \leq 2b(a+b)^{a+b} \leq 2(a+b)^{a+b+1} \leq (a+b+1)^{a+b+1}.$$

□ (claim 1)

Put $a = n(2i-1)+k$, $b = (n-k)(k-2i+1)$ (hence $a+b = k(n-k+2i)$). We have

$$2^{ai}b^{i(a+b)} \leq (a+b)^{i(a+b)}.$$

On the other hand, $i^2n \geq k(n-k)$ implies

$$ia - (a+b) = 2i^2n - in - kn + k^2 - ik \geq i^2n - i(n+k) \geq in(i-2) \geq 0,$$

hence

$$2^{a+b}b^{i(a+b)} \leq (a+b)^{i(a+b)}.$$

This means that

$$1 - \left(\frac{b}{a+b}\right)^i \geq 1 - \frac{1}{2} = \frac{1}{2},$$

thus

$$\begin{bmatrix} n \\ < k-i \end{bmatrix}_k \geq \frac{(n-k)(k-i)}{8ni}\begin{bmatrix} n \\ k-i \end{bmatrix}_k.$$

□

A.5 Theorem (Chernoff's bound) *PV proves: for any $n, k, i \in Log$ such that $k \leq n$ and $n > 0$,*

$$\frac{1}{n^n}\left(\begin{bmatrix} n \\ \leq k-i \end{bmatrix}_k + \begin{bmatrix} n \\ \geq k+i \end{bmatrix}_k\right) = O(4^{-i^2/n}).$$

Proof: The interesting case is to bound $\left[{n \atop <k-i}\right]_k$ when $0 < i < k < n$. If $i \leq s = \sqrt{\frac{k(n-k)}{n}}$, there is also nothing to prove, because $i^2/n \leq (1-k/n)k/n \leq 1/4$, and the left-hand side is bounded by 1. Assume $i > s$. We know from A.2 and A.4 that

$$\frac{1}{n^n}\begin{bmatrix} n \\ < k-i \end{bmatrix}_k \leq c\left(1-\frac{k}{n}\right)\left(\frac{k}{i}-1\right)\sqrt{\frac{n}{(k-i)(n-k+i)}}\times$$
$$\times\frac{k^{k-i}(n-k)^{n-k+i}}{(k-i)^{k-i}(n-k+i)^{n-k+i}}$$

for some c. Since $i > s$, we have

$$\left(1-\frac{k}{n}\right)\left(\frac{k}{i}-1\right)\sqrt{\frac{n}{(k-i)(n-k+i)}} = \frac{n-k}{i}\sqrt{\frac{k-i}{n(n-k+i)}}$$
$$\leq \sqrt{\frac{(n-k)(k-i)}{k(n-k+i)}} = \sqrt{1-\frac{ni}{k(n-k+i)}} \leq 1.$$

As with the proof of A.4, it is not hard to show that $(1+1/a)^a \leq (1+1/b)^b$ and $(1+1/b)^{b+1} \leq (1+1/a)^{a+1}$ whenever $0 < a \leq b \in Log$, hence also $(1+1/a)^a \leq (1+1/b)^{b+1}$ for any a, b, in other words

$$(1+1/b)^{b+1}(1-1/(a+1))^a \geq 1.$$

Let $a, b, j \in Log$, $0 < j < b$. Then

$$\begin{aligned}
&\left(1+\frac{1}{b-j}\right)^b\left(1-\frac{1}{a+j}\right)^a \\
&\quad= \left(1+\frac{1}{b-j}\right)^{b-j+1}\left(1-\frac{1}{a+j}\right)^{a+j-1} \\
&\quad\quad\times\left(1+\frac{1}{b-j}\right)^{j-1}\left(1-\frac{1}{a+j}\right)^{-(j-1)} \\
&\quad\geq \left(1+\frac{1}{b-j}\right)^{j-1}\left(1+\frac{1}{a+j-1}\right)^{j-1},
\end{aligned}$$

thus

$$\left[\left(1+\frac{1}{b-j}\right)^b\left(1-\frac{1}{a+j}\right)^a\right]^{(b-j)(a+j-1)(a+b)}$$

$$\geq \left[\left(1+\frac{1}{b-j}\right)^{(b-j)(a+j-1)}\left(1+\frac{1}{a+j-1}\right)^{(b-j)(a+j-1)}\right]^{(j-1)(a+b)}$$
$$\geq 2^{(j-1)(a+b-1)^2} \geq 2^{4(j-1)(b-j)(a+j-1)},$$

because $(x+y)^2 \geq 4xy$. Therefore

$$4^{2(j-1)}(a+j)^{a(a+b)}(b-j)^{b(a+b)} \leq (a+j-1)^{a(a+b)}(b-j+1)^{a(a+b)},$$

and by induction on i we have

$$4^{i^2-i}(a+i)^{(a+b)}(b-i)^{b(a+b)} \leq a^{a(a+b)}b^{a(a+b)}$$

for any $0 \leq i < b$. Put $a = k-i$ and $b = n-k+i$. Then

$$\begin{aligned}4^{i^2-n}\left(k^{k-i}(n-k)^{n-k+i}\right)^n &\leq 4^{i^2-i}\left(k^{k-i}(n-k)^{n-k+i}\right)^n\\ &\leq \left((k-i)^{k-i}(n-k+i)^{n-k+i}\right)^n,\end{aligned}$$

hence

$$\frac{k^{k-i}(n-k)^{n-k+i}}{(k-i)^{k-i}(n-k+i)^{n-k+i}} \leq 4^{(-i^2+n)/n} = 4 \cdot 4^{-i^2/n},$$

and finally

$$\frac{1}{n^n}\begin{bmatrix} n \\ < k-i \end{bmatrix}_k \leq 4c \cdot 4^{-i^2/n}. \qquad \square$$