LOGICAL STRENGTH OF COMPLEXITY THEORY AND A FORMALIZATION OF THE PCP THEOREM IN BOUNDED ARITHMETIC

JÁN PICH

Department of Algebra, Faculty of Mathematics and Physics, Charles University in Prague, Sokolovska 83, Prague, CZ-186 75, The Czech Republic

Abstract. We present several known formalizations of theorems from computational complexity in bounded arithmetic and formalize the PCP theorem in the theory PV_1 (no formalization of this theorem was known). This includes a formalization of the existence and of some properties of the (n, d, λ)-graphs in PV_1.

1. Introduction

The aim of this paper is to show that a lot of complexity theory can be formalized in low fragments of arithmetic like Cook's theory PV_1.

Our motivation is to demonstrate the power of bounded arithmetic as a counterpart to the unprovability results we already have or want to obtain, and generally to find out how complexity theory behaves in different worlds of bounded arithmetic.

Concerning the unprovability results, Pich [24] proves that under certain hardness assumptions the theory T_{NC^1}, the true universal first-order theory in the language containing names for all uniform NC^1 algorithms, cannot prove polynomial circuit lower bounds on SAT formalized naturally by a sentence $LB(SAT, n^k)$. In fact, that result generalizes basically to any theory weaker than PV_1 in terms of provably total functions. The question whether PV_1 proves $LB(SAT, n^k)$ remains open even if we allow standard complexity-theoretic hardness assumptions, see the discussion in Section 2.

Generally, it would be interesting to arrive at a complexity-theoretic statement, not necessarily circuit lower bounds, whose provability in PV_1 unexpectedly contradicts some other natural hypothesis. To understand better what are plausible candidates for such statements it might help us to investigate the theorems which are provable in low fragments of arithmetic.

In the present paper we will describe the formalization of just a few results; however, this should suffice to illustrate the power of the respective theories. Actually, many classical theorems from complexity theory have been already formalized in bounded arithmetic. In

1998 ACM Subject Classification: Complexity theory and logic, Proof complexity.
Key words and phrases: Bounded arithmetic, Complexity theory, Formalizations.

LOGICAL METHODS IN COMPUTER SCIENCE DOI:10.2168/LMCS-???

the table closing this section we list some representative examples. It should be understood that any of the formalized results is accompanied by a lot of other theorems that are formalizable in a similar fashion. In fact, some of the formalizations are so evident that they are used without a proof as a folklore. This is the case of Cook-Levin's theorem whose formalization we nevertheless describe for expository reasons in Section 4 as it gives us the opportunity to introduce some notions. For more details concerning the list see Section 3.

The main original contribution of this paper is a formalization of the exponential PCP theorem in the theory APC_1 and the PCP theorem in the theory PV_1. Perhaps the most challenging part here was to formalize properties of the (n, d, λ)-graphs needed to derive the PCP theorem. These are usually obtained using algebraic techniques involving norms over real vector spaces coming all the way down to the fundamental theorem of algebra etc. In order to avoid formalization of this machinery (and it is not clear whether this could be done) we employ certain approximations to derive slightly weaker properties of the (n, d, λ)-graphs in the theory PV_1 which, however, suffice to derive the PCP theorem in PV_1.

As the exponential PCP theorem follows trivially from the PCP theorem, the exponential version is actually also provable in PV_1. The PV_1 proof of the PCP theorem uses (among many other tools) the exponential PCP theorem but scaled down to constant size instances so that to prove the scaled down version we need to reason only about sets of constant size. On the other hand, in APC_1 we perform the standard proof of the exponential PCP theorem directly by formalizing a reasoning with p-time definable sets. Hence, the APC_1 proof shows different techniques to be available in low fragments of arithmetic.

The paper is organized as follows. In Section 2 we describe general properties of our formalizations and define theories of bounded arithmetic in which these formalizations take place. In Section 3 we discuss theorems that have been already formalized in bounded arithmetic as well as the new ones obtained in this paper. Section 4 illustrates a formalization of the Cook-Levin theorem in PV_1. In Section 5 we prove the exponential PCP theorem in APC_1. Section 6 formalizes pseudorandom constructions in PV_1 which are then used in Section 7 to formalize the PCP theorem in PV_1.

Theory	Theorem	Reference
PV_1	Cook-Levin's theorem	Section 4
	(n, d, λ)-graphs	Section 6
	the PCP theorem	Section 7
$PV_1 + WPHP(PV_1)$	PARITY $\notin AC^0$	[18]
APC_1	BPP, ZPP, AM,...	[15]
	Goldreich-Levin's theorem	[11]
	the exponential PCP theorem	Section 5
$HARD_\epsilon$	Impagliazzo-Wigderson's derandom.	[14]
$HARD^A$	Nisan-Wigderson's derandomization	[13]
$T^1_2 + rWPHP(PV_2)$	$S^P_2 \subseteq ZPP^{NP}$	[17]
APC_2	Graph isomorphism in coAM	[17]
$APC_2^{\oplus_p P}$	Toda's theorem	[5]

The theories are listed from the weakest to the strongest one.

2. Formalizations in bounded arithmetic: initial notes

The usual language of arithmetic contains well known symbols: $0, S, +, \cdot, =, \leq$. To encode reasoning about computations it is helpful to consider also symbols $\lfloor \frac{x}{2} \rfloor, |x|$ and $\#$ with the intended meaning "the whole part of $\frac{x}{2}$", "the length of the binary representation of x", and $x\#y = 2^{|x|\cdot|y|}$. The language L containing all these symbols was used by Buss [4] to define the theory S^1_2 (see below).

All theories we will work with, a subset of theories collectively known as bounded arithmetic, contain L as a part of their language.

The defining properties of symbols from L are captured by a set of basic axioms denoted as BASIC which we will not spell out, cf. Krajíček [18].

A quantifier is sharply bounded if it has the form $\exists x, x \leq |t|$ or $\forall x, x \leq |t|$ where t is a term not containing x. A quantifier is bounded if it is existential bounded: $\exists x, x \leq t$ for x not occuring in t, or universal bounded: $\forall x, x \leq t$ for x not occuring in t. By Σ^b_0 ($=\Pi^b_0 = \Delta^b_0$) we denote the set of all formulas in the language L with all quantifiers sharply bounded. For $i \geq 0$, the sets Σ^b_{i+1} and Π^b_{i+1} are the smallest sets satisfying

(a) $\Sigma^b_i \cup \Pi^b_i \subseteq \Sigma^b_{i+1} \cap \Pi^b_{i+1}$
(b) Σ^b_{i+1} and Π^b_{i+1} are closed under $\wedge, \vee$ and sharply bounded quantification
(c) Σ^b_{i+1} is closed under bounded existential quantification
(d) Π^b_{i+1} is closed under bounded universal quantification
(e) the negation of a Σ^b_{i+1}-formula is Π^b_{i+1}
(f) the negation of a Π^b_{i+1}-formula is Σ^b_{i+1}.

In words, the complexity of bounded formulas in language L (formulas with all quantifiers bounded) is defined by counting the number of alternations of bounded quantifiers, ignoring the sharply bounded ones. For $i > 0$, Δ^b_i denotes $\Sigma^b_i \cap \Pi^b_i$.

An example of a bounded arithmetic theory is the theory S^1_2 introduced by Buss [4]. The language of S^1_2 is L and its axioms consist of BASIC and Σ^b_1-PIND scheme which is the following kind of polynomial induction for Σ^b_1-formulas A:

$$A(0) \wedge \forall x, (A(\lfloor x/2 \rfloor) \rightarrow A(x)) \rightarrow \forall x A(x)$$

Buss [4] showed that whenever S^1_2 proves a formula of the form $\exists y, A(x, y)$ for Σ^b_1-formula A, then there is a p-time (i.e. polynomial time) function f such that $A(x, f(x))$ holds for all x.

Theories of bounded arithmetic generally cannot prove the totality of functions with superpolynomial growth of length. This follows from a theorem of Parikh [23]. In particular, $\forall k\ \exists x, |x| = k$ is unprovable. Consequently, if we want to prove in bounded arithmetic a statement of the form "for all k, n, there is an n^k-size circuit (encoded by a binary string of some number, i.e. $\exists x, |x| = n^k$) s.t. ..." we need to quantify the exponent k outside of the respective theory. That is, in such cases instead of proving

$$T \vdash \text{"for all } k, n, \text{ there is an } n^k\text{-size circuit s.t. ..."}$$

we prove

"for all k, $T \vdash$ for all m, n s.t. $|m| = n$, there is an n^k-size circuit s.t. ..."

Informally speaking, only the "feasible part" of the theorem is provable inside the theory.

In our formalizations numbers encode binary strings in a natural way. We then follow the convention that inputs of circuits, algorithms or functions are represented by binary strings. For example, when talking about n^k-size circuit lower bounds the number of inputs of n^k-size circuits is the length of some number, i.e $\exists x,\ n = |x|$. However, it does not necessarily follow that n is smaller, say, $\exists x,\ n = ||x||$. To indicate sizes of objects inside our theories we employ the shorthand notation $x \in Log \leftrightarrow \exists y, x = |y|$ and $x \in LogLog \leftrightarrow \exists y, x = ||y||$.

On the contrary, for example Razborov [25] considered (second-order) formalizations of circuit lower bounds (corresponding in first-order logic to the formalization) where p-size (i.e. polynomial size) circuits with n inputs were required to satisfy $n \in LogLog$. Thus, in his formalization, truth tables of functions computed by p-size circuits are encoded by binary strings. The respective theory is much stronger with respect to such formalization; it is as if it could manipulate with exponentially big objects. Formalizing known theorems is then easier and proving unprovability results is on the other hand formally much harder.

Similarly, in propositional proof complexity there are candidate hard tautologies for strong proof systems like Extended Frege which express circuit lower bounds on SAT (and other functions), see formulas $\neg Circuit_t(f)$ in Razborov [26] or $\tau(tt_{s,k})_f$ in Krajíček [19]. Using a standard translation into first-order logic they again correspond to the formalization where truth tables of SAT are encoded by binary strings. Therefore, by the known relation between propositional proof systems and bounded arithmetics, the hardness of such formulas for Extended Frege would imply a conditional unprovability of superpolynomial circuit lower bounds on SAT in PV_1 formalized in such a way that the theory PV_1 would be exponentially stronger than it is with respect to the formalization of circuit lower bounds $LB(SAT, n^k)$ considered in Pich [24]. The formalization $LB(SAT, n^k)$ follows the convention of our current paper.

However, the fact advocated here, that a lot of complexity theory is formalizable in theories like PV_1, suggests that it might be also hard to obtain the unprovability of $LB(SAT, n^k)$ in PV_1. Actually, the unprovability of $LB(SAT, n^k)$ in PV_1 would imply that there is no provable witnessing of errors of p-time algorithms claiming to solve SAT which is itself (interesting and) a reason to expect hardness of such unprovability result, see Pich [24].

2.1. **Theory PV_1: formalized p-time reasoning.** PV_1 introduced in Krajíček-Pudlák-Takeuti [20] is a conservative extension of an equational theory PV introduced by Cook [8].

The language of PV and PV_1 consists of symbols for all p-time algorithms given by Cobham's characterization of p-time functions, cf. [7]. In particular, it contains L. By a slight abuse of the notation we denote the language of PV_1 and PV also PV. A PV-formula is a first-order formula in the language PV. The hierarchy of $\Sigma^b_i(PV)$- and $\Pi^b_i(PV)$-formulas is defined similarly to Σ^b_i and Π^b_i (in first-order logic with equality) but in the language of PV.

In PV we can define p-time concepts and prove their basic properties. More precisely, every p-time function can be straightforwardly defined as a PV-function. Therefore, in the theory PV_1, which is a universal first-order theory, we can reason about p-time concepts. We can interpret provability in PV_1 as capturing the idea of what can be demonstrated when our reasoning is restricted to manipulation of p-time objects. However, strictly speaking, this description would also fit the theory S^1_2 which in addition uses NP-concepts in induction.

Anyway, it is a natural question which properties of p-time concepts are provable using only such p-time reasoning.

It can be shown that PV_1 proves $\Sigma^b_0(PV)$-induction, cf. Krajíček [18]. That is, for any $\Sigma^b_0(PV)$-formula A, PV_1 proves

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

In PV we can speak about formulas, circuits, Turing machines and other similar notions which can be encoded using finite sequences of numbers. These are encodable in PV in a well-behaved way so that basic operations on sequences like concatenation are definable by terms, i.e. by functions in the language. For more details see Krajíček [18] where the function $(w)_i$ which extracts the ith element from a sequence w is shown to be Δ^b_1-definable in S^1_2 but the definition is given by a p-time predicate so it can be written as an open PV-formula.

All PV-functions have well-behaved Δ^b_1-definitions in S^1_2. Hence, S^1_2 can be seen as an extension of PV_1, cf. Buss [4]. Moreover, Buss's witnessing theorem [4] implies that S^1_2 is $\forall\Sigma^b_1$-conservative over PV_1. This means that when proving a $\forall\Sigma^b_1$ statement in PV_1 we can actually use S^1_2. In particular, we will use an induction scheme denoted as Π^b_1-LLIND which is provable in S^1_2 and says that for any $\Pi^b_1(PV)$-formula A the following holds,

$$A(0) \wedge \forall x \leq ||a||\ (A(x) \to A(x+1)) \to A(||a||)$$

In Proposition 6.10, we will also use an induction scheme which we denote Π^b_1-LPIND. It is a weaker form of Π^b_1-PIND, cf. Krajíček [18], so it is derivable in S^1_2. Π^b_1-LPIND says that for any $\Pi^b_1(PV)$-formula A the following implication holds:

$$A(a) \wedge A(a^2) \wedge [\forall l \leq ||b||, (A(a^{\lfloor (l-1)/2 \rfloor}) \wedge A(a^{\lceil (l-1)/2 \rceil}) \to A(a^l))] \to A(a^{||b||})$$

2.2. **Theory APC_1: formalized probabilistic p-time reasoning.** To reason about probabilistic p-time concepts we will use an extension of PV_1 in which Jeřábek [15] developed a well-behaved notion of probability based on an approximate counting.

In this section, we recall a part of his work which we will use to formalize the exponential PCP theorem.

The dual (or surjective) pigeonhole principle for f, written as $dWPHP(f)$, is the universal closure of the formula

$$x > 0 \to \exists v < x(|y|+1)\forall u < x|y| f(u) \neq v$$

For a set of functions Γ, $dWPHP(\Gamma) := \{dWPHP(f) | f \in \Gamma\}$.

The theory APC_1 is defined as $PV_1 + dWPHP(PV)$ where PV stands for the set of PV-functions.

When a number a is used in a context which asks for a set it is assumed to represent the integer interval $[0, a)$, e.g. $X \subseteq a$ means that all elements of X are less than a. If $X \subseteq a$, $Y \subseteq b$, then $X \times Y := \{bx + y | x \in X, y \in Y\} \subseteq ab$ and $X \dot\cup Y := X \cup \{y + a | y \in Y\} \subseteq a + b$.

We will often work with rational numbers which are assumed to be represented by pairs of integers in the natural way. By a definable set we mean a collection of numbers satisfying some formula, possibly with parameters.

Let $n, m \in Log$, $C : 2^n \to 2^m$ be a circuit and $X \subseteq 2^n, Y \subseteq 2^m$ definable sets.We write $C : X \twoheadrightarrow Y$ if $Y \subseteq C[X]$, i.e. $\forall y \in Y \exists x \in X,\ C(x) = y$. The following definitions are taken from Jeřábek [15].

Definition 2.1 (in APC_1)**.** *Let $X, Y \subseteq 2^n$ be definable sets, and $\epsilon \leq 1$. We say that the size of X is approximately less than the size of Y with error ϵ, written as $X \preceq_\epsilon Y$, if there exists a circuit G, and $v \neq 0$ such that*

$$G : v \times (Y \dot{\cup} \epsilon 2^n) \twoheadrightarrow v \times X$$

The sets X and Y have approximately the same size with error ϵ, written as $X \approx_\epsilon Y$, if $X \preceq_\epsilon Y$ and $Y \preceq_\epsilon X$.

A number s identified with the interval $[0, s)$, so $X \preceq_\epsilon s$ means that the size of X is at most s with error ϵ.

Definition 2.2 (in APC_1)**.** *Let $X \subseteq 2^{|t|}$ be a definable set and $0 \leq \epsilon, p \leq 1$. We define*

$$Pr_{x<t}[x \in X] \preceq_\epsilon p \quad iff \quad X \cap t \preceq_\epsilon pt$$

and similarly for $\approx$.

The definition of $\preceq_\epsilon$ is an unbounded $\exists\Pi^b_2$-formula so it cannot be used freely in bounded induction. This problem was solved by Jeřábek [15] by working in a suitable conservative extension of APC_1.

Definition 2.3 (in PV_1)**.** *Let $f : 2^k \mapsto 2$ be a truth-table of a Boolean function with k inputs (f is encoded as a string of 2^k bits, hence $k \in LogLog$). We say that f is (worst-case) ϵ-hard, written as $Hard_\epsilon(f)$ if no circuit C of size $2^{\epsilon k}$ computes f. The function f is average-case ϵ-hard, written as $Hard^A_\epsilon(f)$, if for no circuit C of size $\leq 2^{\epsilon k}$:*

$$|\{u < 2^k | C(u) = f(u)\}| \geq (1/2 + 2^{-\epsilon k})2^k$$

Proposition 2.1 (Jeřábek [13])**.** *For every constant $\epsilon < 1/3$ there exists a constant c such that APC_1 proves: for every $k \in LogLog$ such that $k \geq c$, there exist average-case ϵ-hard functions $f : 2^k \mapsto 2$.*

PV_1 can be relativized to $PV_1(\alpha)$. The new function symbol α is then allowed in the inductive clauses for introduction of new function symbols. This means that the language of $PV_1(\alpha)$, denoted also $PV(\alpha)$, contains symbols for all p-time oracle algorithms.

Definition 2.4 (Jeřábek [13])**.** *The theory $HARD^A$ is an extension of the theory $PV_1(\alpha) + dWPHP(PV(\alpha))$ by the axioms*

1. *$\alpha(x)$ is a truth-table of a Boolean function in $||x||$ variables*
2. *$x \geq c \to Hard^A_{1/4}(\alpha(x))$*
3. *$||x|| = ||y|| \to \alpha(x) = \alpha(y)$*

where c is the constant from the previous lemma.

Theorem 2.1 (Jeřábek [13, 15])**.** *$HARD^A$ is a conservative extension of APC_1. Moreover, there is a $PV(\alpha)$-function $Size$ such that $HARD^A$ proves: if $X \subseteq 2^n$ is definable by a circuit C, then*

$$X \approx_\epsilon Size(C, 2^n, e)$$

where $\epsilon = |e|^{-1}$

We will abuse the notation and write $Size(X, \epsilon)$ instead of $Size(C, 2^n, e)$.

Definition 2.5 (in APC_1)**.** *If $X \subseteq 2^{|t|}$ is defined by a circuit and $\epsilon^{-1} \in Log$, we put*

$$Pr_{x<t}[x \in X]_\epsilon := \frac{1}{t} Size(X \cap t, \epsilon)$$

Jeřábek [15] showed that these definitions are well-behaved:

Proposition 2.2. *(in PV_1) Let $X, X', Y, Y', Z \subseteq 2^n$ be definable sets and $\epsilon, \delta < 1$. Then*

i) $X \subseteq Y \Rightarrow X \preceq_0 Y$

ii) $X \preceq_\epsilon Y \wedge Y \preceq_\delta Z \Rightarrow X \preceq_{\epsilon+\delta} Z$

iii) $X \preceq_\epsilon X' \wedge Y \preceq_\delta Y' \Rightarrow X \times Y \preceq_{\epsilon+\delta+\epsilon\delta} X' \times Y'$

Proposition 2.3. *(in APC_1)*

1. *Let $X, Y \subseteq 2^n$ be definable by circuits, $s, t, u \leq 2^n$, $\epsilon, \delta, \theta, \gamma \leq 1, \gamma^{-1} \in Log$. Then*

   i) $X \preceq_\epsilon Y \Rightarrow 2^n - Y \preceq_{\epsilon+\delta} 2^n - X$

   ii) $X \approx_\epsilon s \wedge Y \approx_\delta t \wedge X \cap Y \approx_\theta u \Rightarrow X \cup Y \approx_{\epsilon+\delta+\theta+\gamma} s + t - u$

2. *Let $X \subseteq 2^n \times 2^m$ and $Y \subseteq 2^m$ be definable by circuits, $t \preceq_\epsilon Y$ and $s \preceq_\delta X_y$ for every $y \in Y$, where $X_y := \{x | \langle x, y\rangle \in X\}$. Then for any $\gamma^{-1} \in Log$*

$$st \preceq_{\epsilon+\delta+\epsilon\delta+\gamma} X \cap (2^n \times Y)$$

3. *(Chernoff's bound) Let $X \subseteq 2^n, m \in Log, 0 \leq \epsilon, \delta, p \leq 1$ and $X \succeq_\epsilon p2^n$. Then*

$$\{w \in (2^n)^m | \ |\{i < m | w_i \in X\}| \leq m(p - \delta)\} \preceq_0 c4^{m(c\epsilon - \delta^2)} 2^{nm}$$

for some constant c, where w is treated as a sequence of m numbers less than 2^n and w_i is its i-th member.

3. Previous formalizations of complexity theory and our contribution

Many classical theorems from complexity theory have been already formalized in bounded arithmetic. In the following sections we present some representative examples from different areas of complexity theory. The last section describes the formalizations that are obtained in this paper.

3.1. NP-completeness.

Actually, formalization of some theorems is a folklore used without a proof. For example, Cook-Krajíček [9] mention that NP-completeness of SAT can be formalized in PV_1.

Theorem 3.1 (Cook-Levin's theorem in PV_1)**.** *(a) For every Σ^b_1-formula $\phi(x)$, there is a PV-function $f(x)$ such that*

$$PV_1 \vdash \phi(x) \leftrightarrow \exists y SAT(f(x), y)$$

where $SAT(z, y)$ is an open PV-formula which holds iff truth assignment y satisfies propositional formula z.

(b) For each k we have a PV-function f such that PV_1 proves: for any M, x,

$$\exists w, z; |z|, |w| \leq |x|^k, M(x, z, w) = 1 \leftrightarrow \exists y, |y| \leq 3|M||x|^{2k}, SAT(f(M, x), y)$$

where $M(x, z, w) = 1$ is an open PV-formula which holds iff w is an accepting computation of Turing machine M on input x, z (so we are slightly abusing the notation as M is actually a free variable in the formula $M(x, z, w) = 1$) and $|M|$ is the length of M's code.

Note that formulations (a) and (b) are essentially equivalent because the formula $\exists w, z; |z|, |w| \leq |x|^k, M(x, z, w) = 1$ is Σ^b_1 and any Σ^b_1-formula $\phi(x)$ is equivalent in PV_1 to a formula $\exists w, z; |z|, |w| \leq |x|^k, M(x, z, w) = 1$ for some k and M. In (b) we have in addition also an explicit bound on y.

For expository reasons we present a proof of (b) in Section 4.

3.2. **Randomized computation.** The main application of approximate counting in APC_1 is in the formalization of probabilistic algorithms in APC_1 and complexity classes like BPP and AM. Jeřábek's formalizations involve many other results we will not state explicitly like "promise BPP $\subseteq$ P/poly" (Lemma 3.10 in Jeřábek [15]), Rabin-Miller algorithm (Example 3.2.10 in Jeřábek [14]) but also principles like Stirling's bound on binomial coefficients.

Definition 3.1 (Jeřábek [15])**.** *(in APC_1) A PV-function r and a PV-predicate A define a BPP language if for each x either $Pr_{w<r(x)}[\neg A(x, w)] \preceq_0 1/4$ or $Pr_{w<r(x)}[A(x, w)] \preceq_0 1/4$.*

Theorem 3.2 (Jeřábek [15])**.** *Let A be a PV-predicate and r a PV-function. There are Σ^b_2-formulas $\sigma^+(x), \sigma^-(x)$ and Π^b_2-formulas $\pi^+(x), \pi^-(x)$ such that APC_1 proves*

$$Pr_{w<r(x)}[\neg A(x, w)] \preceq_0 1/4 \Rightarrow \pi^+(x) \Rightarrow \sigma^+(x) \Rightarrow Pr_{w<r(x)}[\neg A(x, w)] \preceq_0 1/3$$

$$Pr_{w<r(x)}[A(x, w)] \preceq_0 1/4 \Rightarrow \pi^-(x) \Rightarrow \sigma^-(x) \Rightarrow Pr_{w<r(x)}[A(x, w)] \preceq_0 1/3$$

In particular, any definable BPP language is in $\Sigma^b_2 \cap \Pi^b_2$.

In [17] Jeřábek formalized Cai's [6] result stating that $S^P_2 \subseteq ZPP^{NP}$ in the theory $T^1_2 + rWPHP(PV_2)$. The complexity class S^P_2 consists of languages for which there exists a p-time predicate R such that

$$x \in L \Rightarrow \exists y \forall z R(x, y, z)$$

$$x \notin L \Rightarrow \exists z \forall y \neg R(x, y, z)$$

where $|y|, |z|$ are implicitly bounded by a polynomial in $|x|$.

The theory T^1_2 is defined as S^1_2 but with induction for Σ^b_1-formulas, PV_2 denotes functions computable in polynomial time relative to NP, and $rWPHP(PV_2)$ is a set of axioms

$$x > 0 \rightarrow \exists y < x(|y| + 1)(g(y) \geq x|y| \vee f(g(y)) \neq y)$$

for PV_2-functions f, g.

Note that $rWPHP(f, g)$ follows from $dWPHP(f)$.

Theorem 3.3 (Jeřábek [17])**.** *(in $T^1_2 + rWPHP(PV_2)$) The complexity class S^P_2 is contained in ZPP^{NP}. That is, for each p-time relation R defining a language $L \in S^P_2$, there exists ZPP^{NP}-predicate P definable in $T^1_2 + rWPHP(PV_2)$ such that the same theory proves $x \in L \Leftrightarrow P(x)$.*

3.3. **Circuit lower bounds.** In [18, Section 15.2] Krajíček proves PARITY $\notin AC^0$ in the theory $PV_1 + WPHP(PV_1)$. By $WPHP(PV_1)$ he denotes the set of axioms

$$a > 0 \to \exists y \leq 2a \forall x \leq a, f(x) \neq y$$

for every PV_1-function symbol $f(x)$ where f may have other arguments besides x and they are treated as parameters in the axioms.

It is known that $WPHP(PV_1)$ and $dWPHP(PV)$ are equivalent over S^1_2. Further, the theory $PV_1 + dWPHP(PV)$ is $\forall\Sigma^b_1$-conservative over

$$PV_1 + \{\exists y < a \# a \ \forall x < a, \ f(x) \neq y | \text{ for PV-functions } f\}$$

(noted in Jeřábek [16] as a corollary of earlier results).

Theorem 3.4 (Krajíček [18], Section 15.2)**.** *Let d, k be arbitrary constants. Then the theory $PV_1 + WPHP(PV_1)$ proves that for any sufficiently large $n \in Log$ there are no depth d circuits of size $\leq kn^k$ computing $PARITY(x_1, ..., x_n)$.*

In [25] Razborov developes a logical formalism supporting his feeling that S^1_2 is the right theory to capture that part of reasoning in Boolean complexity which led to actual lower bounds for explicitly given Boolean functions. He formalizes lower bounds for constant-depth circuits over the standard basis, lower bounds for monotone circuits, lower bounds for constant-depth circuits with MOD-q gates, and lower bounds for monotone formulas based on communication complexity.

Importantly, his formalizations presented in second-order logic correspond in first-order logic to the formalization where the number of inputs of circuits in the respective theorems is in *LogLog*. This makes it more suitable for encoding into the propositional setting but it also makes the formalization results formally weaker.

3.4. **Interactive proofs.** Jeřábek [17] formalized the equivalence of public-coin and private-coin interactive protocols in the theory $APC_2 := T^1_2 + dWPHP(PV_2)$. This is illustrated on the example of the isomorphism problem: given two structures G_0 and G_1 (as tables) of the same signature, determine whether $G_0 \simeq G_1$.

Definition 3.2 (Jeřábek [15])**.** *(in APC_2) A pair $\langle \phi, r \rangle$ where $\phi(x, w)$ is a Σ^b_1-formula, and r is a PV-function, defines an AM language if for each x either $Pr_{w<r(x)}[\neg\phi(x, w)] \preceq^1_0 1/4$ or $Pr_{w<r(x)}[\phi(x, w)] \preceq^1_0 1/4$ where $\preceq^1_0$ denotes $\preceq_0$ relativized with a Σ^b_1-complete oracle.*

Theorem 3.5 (Jeřábek [17])**.** *(in APC_2) Graph nonisomorphism is in AM.*

3.5. **Cryptography.** Recently, Dai Tri Man Le [11] formalized Goldreich-Levin's theorem in APC_1.

Theorem 3.6 (Dai Tri Man Le [11])**.** *(in APC_1) Let $f : \{0,1\}^n \to \{0,1\}^n$ be a function computed by a circuit of size t, and suppose that there exists a circuit C of size s such that*

$$Pr_{(x,r)\in\{0,1\}^{2n}}[C(f(x), r) = \bigoplus_{i=1}^{n} x_i r_i]_\epsilon \geq \frac{1}{2} + \frac{1}{p(n)}$$

If $\epsilon = \frac{1}{poly(n)}$ is sufficiently small, then there is a circuit C' of size at most $(s+t)poly(n, 1/\epsilon)$ and $q = poly(n)$ such that

$$Pr_{(x,r')\in\{0,1\}^n\times\{0,1\}^q}[f(C'(f(x), r')) = f(x)]_\epsilon \geq \frac{1}{4p(n)} - \frac{15\epsilon}{2}$$

3.6. **Complexity of counting.** In [5], Buss, Kołodziejczyk and Zdanowski derived Toda's theorem in an extension of the theory APC_2.

For a fixed prime $p \geq 2$, they denote by C_p^k for $k \in [p]$ quantifiers counting mod p. The intended meaning of $C_p^k x \leq tA(x)$ is that the number of values $x \leq t$ for which A is true is congruent to k mod p. See [5] for the explicit list of axioms defining C_p^k.

A $\oplus_p P$ formula is a formula which is either atomic, or of the form $C_p^k x \leq tA(x)$ where A is sharply bounded. $\Sigma_0^{b,\oplus_p P} = \Pi_0^{b,\oplus_p P}$ is the set of formulas obtained as the closure of $\oplus_p P$ formulas under Boolean connectives $\vee, \wedge, \neg$ and under sharply bounded quantifiers. For $i \geq 1$, the strict formula sets $\hat{\Sigma}^{b,\oplus_p P_i}$ are defined in the usual way by counting the number of alternations of bounded quantifiers.

$T_2^{1,\oplus_p P}$ is the theory axiomatized by the axioms for PV_1 symbols, the C_p^k axioms for sharply bounded formulas $A(x)$, and $\hat{\Sigma}_1^{b,\oplus_p P}$-IND.

$APC_2^{\oplus_p P} := T_2^{1,\oplus_p P} + dWPHP(PV_2^{\oplus_p P})$ where $PV_2^{\oplus_p P}$ means functions that can be computed in polynomial time relative to $NP^{\oplus_p P}$.

$\Sigma_\infty^b(\oplus_p)$ denotes formulas formed from bounded existential, universal, and C_p quantifiers.

In $APC_2^{\oplus_p P}$, we say that a language is in $BP \cdot \oplus_p P$ if there exists PV_1 functions f and u such that for all x,

$$x \in L \Leftrightarrow Pr_{r<u(x)}[f(x, r) \notin \oplus_p^1 SAT] \preceq_0 1/4$$
$$x \notin L \Leftrightarrow Pr_{r<u(x)}[f(x, r) \notin \oplus_p^0 SAT] \preceq_0 1/4$$

where $\oplus_p^i SAT$ is the set of propositional formulas ϕ such that the number of satisfying assignments of ϕ is congruent to i mod p for some prime p.

Theorem 3.7 (Buss, Kołodziejczyk, Zdanowski [5])**.** *$APC_2^{\oplus_p P}$ proves that any $\Sigma_\infty^b(\oplus_p)$ formula defines a property in $BP\cdot\oplus_p P$.*

3.7. **Derandomization.** The approximate counting developed in APC_1 relies on a formalization of the derandomization result by Nisan and Wigderson [22].

Definition 3.3 (Jeřábek [15])**.** *(in APC_1) A definable randomized algorithm is given by a pair of PV-functions f, r such that*

$$\exists w < r(x)\ f(x, w) \neq * \rightarrow Pr_{w<r(x)}[f(x, w) = *] \preceq_0 1/2$$

where $*$ is a special symbol signaling a rejecting computation.

The special symbol $*$ could be avoided but it is useful for denoting a "failure-state" of probabilistic algorithms. It can be used when the input random string does not encode the expected structure, say a graph or a formula.

Theorem 3.8 (Jeřábek [13]). *Let F be a randomized algorithm definable in S^1_2+$dWPHP(PV)$. Then there are PV-functions h and g such that $HARD^A$ proves*

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

Jeřábek [14] formalized also Impagliazzo-Wigderson's [12] derandomization which draws the same conclusion assuming only worst-case hardness. This turned out to be much harder than the Nisan-Wigderson construction mainly because list decoding of error-correcting codes used in the construction requires several algebraic tools concerning finite fields.

Theorem 3.9 (Jeřábek [14]). *Let F be a randomized algorithm definable in S^1_2+$dWPHP(PV)$, and let $\epsilon > 0$. Then there are PV-functions h and g such that $HARD_\epsilon$ proves*

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

Here, $HARD_\epsilon$ is defined as an extension of $S^1_2(\alpha)$, i.e. relativized S^1_2, by the following axioms:

1. $\alpha(x) : 2^{||x||} \rightarrow 2$
2. $x \geq c \rightarrow Hard_\epsilon(\alpha(x))$

for a standard constant c.

3.8. **Contribution of our paper: the PCP theorem and the (n, d, λ)-graphs.** We add to the list of formalized results mentioned in previous sections formalizations of the exponential PCP theorem, the PCP theorem, and certain pseudorandom constructions involving the so called (n, d, λ)-graphs which are needed in the proof of the PCP theorem. The exponential PCP theorem was proved in Arora-Safra [2], and the PCP theorem is originally from Arora-Safra [2] and Arora et.al. [3]. In [10] Dinur gave a simpler proof of the PCP theorem which we will formalize.

Definition 3.4. *(in APC_1) Let k, k', d be constants, $x \in \{0,1\}^n$ for $n \in Log$. Further, let $w \in \{0,1\}^{kn^k}$ (represent random bits), π be a $k'n^{k'}$-size circuit with m inputs where m might differ from n, and D be a kn^k-time algorithm.*

Denote by $D^{\pi,w}(x)$ the output of D on input x and with access to π specified by (random bits) w as follows. D computes π on at most d different inputs: first, it produces strings $\hat{w}_1, ..., \hat{w}_d$ where each $\hat{w}_i \in \{0,1\}^m$, then it computes $\pi(\hat{w}_1), ..., \pi(\hat{w}_d)$ and finally computes its output which is either 1 or 0.

We formulate the exponential PCP theorem in APC_1 as follows. For an explanation and a discussion concerning the choice of the formulation see Section 5.

Theorem 3.8 (The exponential PCP theorem in APC_1). *There are constants d, k, k' and a kn^k-time algorithm D (given as a PV-function) computing as in Definition 3.4 such that APC_1 proves that for any $x \in \{0,1\}^n$, $n \in Log$:*

$$\exists y SAT(x, y) \rightarrow \exists k'n^{k'} size\ circuit\ \pi\ \forall w < 2^{kn^k}, D^{\pi,w}(x) = 1$$
$$\forall y \neg SAT(x, y) \rightarrow \forall k'n^{k'} size\ circuit\ \pi, Pr_{w<2^{kn^k}}[D^{\pi,w}(x) = 1] \preceq_0 1/2$$

We also formalize pseudorandom constructions involving the (n, d, λ)-graphs in PV_1 but leave the presentation of these results to Section 6 as it would require introducing too many definitions now.

In order to formalize the PCP theorem we use the notion of probability Pr on spaces of polynomial size $poly(n)$ for $n \in Log$ which is assumed to be defined in a natural way using an exact counting of sets of polynomial size which is also assumed to be defined in PV_1 in a standard way. The notion of probability Pr should not be confused with the definition of Pr in APC_1. We formulate (the more important implication of) the PCP theorem in PV_1 as follows.

Definition 3.5. *(in PV_1) Let k, c, d be constants, $x \in \{0,1\}^n, n \in Log, w \in \{0,1\}^{c \log n}$, $\pi \in \{0,1\}^{dn^c}$, and be D be a kn^k-time algorithm.*

Denote by $D^{\pi,w}(x)$ the output of D on input x and with access to π specified by w as follows. D uses at most $c \log n$ random bits w and makes at most d nonadaptive queries to locations of π, i.e. D can read bits $\pi_{i_1}, ..., \pi_{i_d}$ for $i_1, ..., i_d$ produced by D. Then it computes its outputs, 1 or 0.

In Definition 3.5 we abuse the notation and use the shortcut $D^{\pi,w}(x)$ in different meaning than in Definition 3.4. This should not lead into confusion.

Theorem 7 (The PCP theorem in PV_1)**.** *There are constants d, k, c and a kn^k-time algorithm D (given as a PV-function) computing as in Definition 3.5 such that PV_1 proves that for any $x \in \{0,1\}^n, n \in Log$:*

$$\exists y SAT(x, y) \rightarrow \exists \pi \in \{0,1\}^{dn^c}\ \forall w < n^c, D^{\pi,w}(x) = 1$$

$$\forall y \neg SAT(x, y) \rightarrow \forall \pi \in \{0,1\}^{dn^c}, Pr_{w<n^c}[D^{\pi,w}(x) = 1] \leq 1/2$$

Note that the exponential PCP theorem follows from the PCP theorem. Hence, the exponential version is also provable in PV_1. The PV_1 proof of the PCP theorem uses (among many other tools) the exponential PCP theorem but scaled down to constant size instances so that to prove the scaled down version we need to reason only about sets of constant size. On the other hand, in APC_1 we perform a reasoning with p-time definable sets. Hence, the APC_1 proof shows different tools to be available in low fragments of arithmetic.

4. The Cook-Levin theorem in PV_1

This section serves mainly as an illustration of some techniques available in PV_1 which we later use freely in our arguments.

Theorem 4.1. *(The Cook-Levin theorem in PV_1) For each k, we have a PV-function f such that PV_1 proves: for any M, x,*

$$\exists w, z; |z|, |w| \leq |x|^k, M(x, z, w) = 1 \leftrightarrow \exists y, |y| \leq 3|M||x|^{2k}, SAT(f(M, x), y)$$

where $M(x, z, w) = 1$ is an open PV-formula which holds iff w is an accepting computation of Turing machine M on input x, z, and $|M|$ is the length of M's code.

Proof. First, we show that for some PV-function f, PV_1 proves:

$$\forall M, x, z, w; |z|, |w| \leq |x|^k \exists y; |y| \leq 3|M||x|^{2k}\ (M(x, z, w) = 1 \rightarrow SAT(f(M, x), y)) \quad (*)$$

The Turing machine M is represented as a binary string encoding a tuple (Q, Σ, b, F, ρ) where Q is the set of states, Σ is the set of tape symbols, $b \in Q$ is the initial state, $F \subseteq Q$

is the set of accepting states, and $\rho \subseteq ((Q-F)\times\Sigma)\times(Q\times\Sigma\times\{-1,1\})$ is the transition function.

We assume that the open PV-formulas $M(x,z,w)=1$ and $SAT(x,y)$ are already constructed in a well-behaved way.

The propositional formula encoded by $f(M,x)$ will be built from atoms $T_{i,j,s}$ with intended interpretation "tape cell i of M contains symbol j at step s", atoms $H_{i,s}$ for "M's head is at tape cell i at step s", and atoms $Q_{q,s}$ for "M is in state q at step s". These atoms are assumed to be encoded in a standard way.

Given M,x we define $f(M,x)$ gradually by introducing more and more complex functions. This is supposed to illustrate the way in which PV_1 introduces new functions.

Let us start with a definition of function $f_{input}(x,y)$ mapping x,y to a conjunction of $|y|$ atoms representing first $|y|$ bits of binary string x:

$f_{input}(x,0) := 0$
$f_{input}(x,s_i(y)) :=' f_{input}(x,y)\wedge T_{|y|,i,0}\ '$ if $|y|\le|x|\wedge x_{|y|}=i$, $i=0,1$

where $'A\wedge B'$ is a code of the conjunction of propositional formulas encoded in A and B.

Next, put $f_{ins}(M,x) :=' f_{input}(x,x)\wedge Q_{b,0}\ '$.

Then, define $f_{symb}(M,x,[t,l,m]) =' f_{ins}(M,x)\wedge G'$ where G is a conjunction of formulas $(T_{t',l',m'}\to\neg T_{t',l'',m'})$ for all $l'\neq l''$ and t',m' such that $[t',l',m'],[t',l'',m']\le[t,l,m]$. This guarantees that cell $t'\le t$ contains only one symbol at step $m'\le m$.

$f_{symb}(M,x,0) := f_{ins}(M,x)$
$f_{symb}(M,x,s_i([t,l,m])) :=' f_{symb}(M,x,[t,l,m])\wedge(T_{t',l',m'}\to\neg T_{t',l'',m'})\ '$
if $l'\neq l''\wedge[t',l',m'],[t',l'',m']\le[t,l,m]$, $i\in\{0,1\}$

Similarly, define $f_{state}(M,x,[t,l,m])$ by extending $f_{symb}(M,x,[t,l,m])$ with
1. $Q_{t',m'}\to\neg Q_{t'',m'}$ for $t'\neq t''$ (M cannot be in two different states at step m')
2. $H_{t',m'}\to\neg H_{t'',m'}$ for $t'\neq t''$ (Head cannot be in two different positions at step m')
3. $T_{t',l'',m'}\wedge T_{t',l',m'+1}\to H_{t',m'}$ for $l'\neq l''$ and $t',t''\le t; l',l''\le l; m'\le m$

Further, in this way introduce function f_{trans} capturing M's transition function ρ.

$f_{trans}(M,x,c) :=' f_{state}(M,x,[|x|^k,|x|^k,|x|^k])\wedge$
$(H_{j,c}\wedge Q_{q,c}\wedge T_{j,\sigma,c}\to\bigvee_{(q,\sigma,q',\sigma',d)\in\rho}(H_{j+d,c+1}\wedge Q_{q',c+1}\wedge T_{j,\sigma',c+1})'$

Finally, $f(M,x) :=' f_{trans}(M,x,|x|^k)\wedge\bigvee_{r\in F,t\le|x|^k}Q_{r,t}\ '$.

This defines a PV-function f. To see that $(*)$ holds, given M,x,w, we define y assigning 0 or 1 to atoms of the formula $f(M,x)$ as follows:

1. $y(T_{j,i,0})=1$ iff $x_j=i$ for $i=0,1$ and $j<|x|$.
   $y(T_{j,i,t})=1$ iff w says that tape cell j of M at step t contains i
2. $y(H_{j,c})=1$ iff w says that at step c head is in position j
3. $y(Q_{r,t})=1$ iff w contains M in state r at step t

Informally, if w indeed encodes an accepting computation of Turing machine M on input x,z, then the previous definition produces y which satisfies all conjuncts in formula $f(M,x)$ because these are copying the conditions from the definition of $M(x,z,w)=1$. Therefore, we can conclude that $M(x,z,w)=1\to SAT(f(M,x),y)$ in the theory PV_1.

Analogously, $PV_1 \vdash \forall M, x, y, \exists w, z(SAT(f(M,x),y) \rightarrow M(x,z,w) = 1)$.

□

5. The exponential PCP theorem in APC_1

The exponential PCP theorem was proved in Arora-Safra [2]. We formalize it in the theory APC_1 basically following the presentation in Arora-Barak [1]. However, there is a crucial change: we cannot use the Fourier transformation to derive the linearity test because it would require manipulations with exponentially big objects and it is not clear whether this could be done (for example, using a representation by circuits). Instead, we formalize the so called majority correction argument as it is presented in Moshkovitz [21]. Other parts of the proof work without much change. It is essential that all sets used to express probabilities are definable by p-size circuits so that APC_1 can work with them and the proof itself does not use more than basic operations on these sets which are available in APC_1.

Recall Definition 3.4 introducing the predicate $D^{\pi,w}(x)$. The algorithm D will represent the so called verifier of probabilistically checkable proofs π. The verifier is usually defined so that π is allowed to be any string of arbitrary length and D has an oracular access to π, it can ask for any bit of π. Then, for a language L, $L \in PCP(poly(n), 1)$ standardly means that there is a p-time algorithm D such that:

1. If $x \in L$, then there is a string π (proof) such that D with input x of length n and $poly(n)$ random bits asks for at most $O(1)$ bits of π and accepts (with probability 1);
2. If $x \notin L$, then for any π, D with input x of length n and $poly(n)$ random bits asks for at most $O(1)$ bits of π and accepts with probability $\leq 1/2$.

The exponential PCP theorem says that $NP \subseteq PCP(poly(n), 1)$. As the verifier uses $poly(n)$ random bits, the proof π can be seen as a string of size $2^{poly(n)}$. In our formalization, $n \in Log$ so bounded arithmetic cannot encode the exponentially big proofs by binary strings. In order to be able to speak about them we represent such proofs by p-size circuits. More precisely, for a $k'n^{k'}$-size circuit π with m inputs and $x \in \{0,1\}^m$, $\pi(x)$ is the x-th bit of the proof represented by π. Hence, the condition 1.) in our formulation of the exponential PCP theorem will look formally stronger but it follows trivially from the standard proof. In condition 2.) our D will recognize errors only in proofs that are represented by $k'n^{k'}$-size circuits. We can interpret it as if the proofs that are not represented by such circuits were automatically rejected. Alternatively, we could also represent proofs by oracles which would maybe better reflect the nature of the exponential PCP theorem. However, then we would need to perform the formalization in the theory APC_1 extended by such oracles.

As the NP-completeness of SAT is provable in PV_1 it is sufficient to show in APC_1 that SAT $\in PCP(poly(n), 1)$. This should justify Theorem 3.8 as the right formulation of the exponential PCP theorem in APC_1.

Proof. (of Theorem 3.8) For any $x \in \{0,1\}^n$, the algorithm D firstly reduces SAT instance x to a set of quadratic equations: It obtains 3SAT formula equivalent to x by introducing new variable for each gate of the formula encoded in x and clauses representing the gate. For each clause of the form $x_1 \vee x_2 \vee x_3$ it produces two equations $(1 - x_1)y = 0$ and $y - (1 - x_2)(1 - x_3) = 0$ where y is a new variable. Analogously for other possible clauses, if some x_i occurs in the clause negatively, $1 - x_i$ in the resulting equations is replaced by

x_i. In this way D produces a set of quadratic equations which is solvable in F_2 if and only if x is satisfiable. More precisely, there is k_0 such that if x encodes a propositional formula with n_0 variables it can be efficiently mapped to a set of $m \leq |x|^{k_0}$ quadratic equations on $n_1 \leq |x|^{k_0}$ variables $u_1, ..., u_{n_1}$ (w.l.o.g. $u_1 = 1$). The set of equations can be represented by an $m \times n_1^2$ matrix A and a string $b \in \{0,1\}^m$ satisfying:

$$\exists y\ SAT(x,y) \to \exists u\ Au \otimes u = b$$

$$\forall y\ \neg SAT(x,y) \to \forall u\ Au \otimes u \neq b$$

where $u \in \{0,1\}^{n_1}$ and $u \otimes u$ is a vector of bits $u_i u_j, i, j \in [n_1]$ ordered lexicographically.

The algorithm D will interpret $k'n^{k'}$-size circuits π with $n_1^2 + n_1 + 1$ inputs b, z, z', where $b \in \{0,1\}, z \in \{0,1\}^{n_1}, z' \in \{0,1\}^{n_1^2}$, as circuits allowing us to access functions $f_\pi = WH(u)$ and $g_\pi = WH(u \otimes u)$ for some $u \in \{0,1\}^{n_1}$ in the following way, $\pi(0, z, z') = WH(u)(z)$ and $\pi(1, z, z') = WH(u \otimes u)(z')$. Here, $WH(u)(z) := \Sigma_{i=1}^{n_1} u_i z_i \ mod\ 2$. Similarly for $WH(u \otimes u)(z')$. WH stands for "Walsh-Hadamard".

For any $x \in \{0,1\}^n$, the algorithm D with $\leq kn^k$ random bits $w = r_1^l, ..., r_7^l$ for $l = 1, ..., m_0$, where m_0 is a constant, $r_1^l, r_2^l, r_3^l \in \{0,1\}^{n_1}$, $r_4^l, r_5^l, r_6^l \in \{0,1\}^{n_1^2}, r_7^l \in \{0,1\}^m$ and with access to an $k'n^{k'}$-size circuit π accepts if and only if for each $l = 1, ..., m_0$, π passes the following tests

- "linearity": $f(r_1^l + r_2^l) = f(r_1^l) + f(r_2^l)$ and $g(r_4^l + r_5^l) = g(r_4^l) + g(r_5^l)$
- "g_π encodes $u \otimes u$": $g'(r_1^l \otimes r_2^l) = f'(r_1^l) f'(r_2^l)$
- "g_π encodes a satisfying assignment": $g'(z) = \Sigma_{i=1}^m (r_7^l)_i b_i$ for z representing the sum $\Sigma_{i=1}^m (r_7^l)_i (A_i u \otimes u)$ where $A_i u \otimes u$ is the lefthand-side of the i-th equation in $Au \otimes u = b$

Here, $f = f_\pi, g = g_\pi$, $f'(r_1^l) = f(r_1^l + r_3^l) + f(r_3^l), f'(r_2^l) = f(r_2^l + r_3^l) + f(r_3^l)$ and similarly $g'(r_1^l \otimes r_2^l) = g(r_1^l \otimes r_2^l + r_6^l) + g(r_6^l), g'(z) = g(z + r_6^l) + g(r_6^l)$.

For any $x \in \{0,1\}^n$, if $\exists y SAT(x,y)$ then there is $u \in \{0,1\}^{n_1}$ solving the corresponding equations $Au \otimes u = b$. Thus there is a $k'n^{k'}$-size circuit π with $n_1^2 + n_1 + 1$ inputs given by $\pi(0, z, z') := WH(u)(z)$ and $\pi(1, z, z') := WH(u \otimes u)(z')$ which passes all the tests: for any w, the linearity is clearly satisfied by the definition. Further:

$$g'(r_1^l \otimes r_2^l) = g(r_1^l \otimes r_2^l + r_6^l) + g(r_6^l) = g(r_1^l \otimes r_2^l) = \Sigma_{i,j=1}^{n_1} u_i u_j (r_1^l)_i (r_2^l)_j$$
$$= \Sigma_{i=1}^{n_1} u_i (r_1^l)_i \Sigma_{j=1}^{n_1} u_j (r_2^l)_j = f(r) f(r') = f'(r) f'(r')$$

and as $Au \otimes u = b$ also $g'(z) = \Sigma_{i=1}^m (r_7^l)_i b_i$.

Now we will show that the algorithm D recognizes incorrect proofs with high probability. The argument relies on the Test of linearity which we prove in Section 5.1.

Proposition 5.1 (Test of linearity in APC_1)**.** *Let ϵ be sufficiently small, $\epsilon^{-1} \in Log$ and let f be a function on $n_1 \in Log$ inputs represented by a circuit such that for each linear function g with n_1 inputs,*

$$Pr_{x \in \{0,1\}^{n_1}}[f(x) = g(x)]_\epsilon < p$$

Then $Pr_{x,y}[f(x+y) = f(x) + f(y)]_\epsilon \preceq_{11\epsilon + 13\epsilon^2 + 2\epsilon^3} max\{29/32, 1/2 + p/2\}$.

We abuse the notation and use f also in place of circuits representing f. Note that g is represented by n_1 coefficients.

Claim 1 (Local decoding in APC_1)**.** *Let $s < 1/4, \epsilon \leq 1$ and f be a function on $n_1 \in Log$ inputs represented by a circuit such that there is a linear function f_l which satisfies $Pr_{x<2^{n_1}}[f(x) = f_l(x)]_\epsilon \geq 1 - s$. Then for each $x < 2^{n_1}$,*

$Pr_{r<2^{n_1}}[f_l(x) = f(x+r) + f(r)]_\epsilon \succeq_{6\epsilon} 1 - 2s$.

Proof of the claim: By the assumption and Proposition 2.3 1.i), for $x < 2^{n_1}$, $\{r|f(r) \neq f_l(r)\} \cap 2^{n_1} \preceq_{2\epsilon} s2^{n_1}$ and $\{r|f(x+r) \neq f_l(x+r)\} \cap 2^{n_1} \preceq_{2\epsilon} s2^{n_1}$ which implies $\{r|f(r) \neq f_l(r) \vee f(x+r) \neq f_l(x+r)\} \cap 2^{n_1} \preceq_{4\epsilon} 2s2^{n_1}$. By linearity of f_l, for any $x < 2^{n_1}$, $\{r|f_l(x) \neq f(x+r) + f(r)\} \subseteq \{r|f_l(r) \neq f(r) \vee f_l(x+r) \neq f(x+r)\}$.

Thus, $Pr_r[f_l(x) = f(x+r) + f(r)]_\epsilon \succeq_{6\epsilon} 1 - 2s$, which proves the claim.

Assume that $\forall y \neg SAT(x, y)$, so $\forall u, Au \otimes u \neq b$ and let π be arbitrary circuit of size $k'n^{k'}$. Further, let ϵ be sufficiently small, $\epsilon^{-1} \in Log$ and denote by $D_1^{\pi,w}(x)$, $D^{\pi,w}(x)$ with $m_0 = 1$, i.e. D performing only one round of testing.

If for each linear function g_l, $Pr_{x\in\{0,1\}^{n_1^2}}[g(x) = g_l(x)]_\epsilon < 31/32$ or for each linear function f_l, $Pr_{x\in\{0,1\}^{n_1}}[f(x) = f_l(x)]_\epsilon < 31/32$, then by the test of linearity, we have $Pr_w[D_1^{\pi,w}(x) = 1]_\epsilon \preceq_{13\epsilon+13\epsilon^2+2\epsilon^3} 63/64$. Otherwise, there are linear functions g_l, f_l such that by local decoding, for each $x \in \{0,1\}^{n_1^2}$, it holds $Pr_r[g_l(x) = g'(x)]_\epsilon \succeq_{6\epsilon} 15/16$ where $g'(x) = g(x+r) + g(r)$ and for each $x \in \{0,1\}^{n_1}$, $Pr_r[f_l(x) = f'(x)]_\epsilon \succeq_{6\epsilon} 15/16$ where $f'(x) = f(x+r) + f(r)$.

We need to show that even in the latter situation verifier D accepts with small probabilty. For this, we distinguish two cases: 1. $g_l \neq WH(u\otimes u)$, i.e. $\exists x, y$, $g_l(x\otimes y) \neq f_l(x)f_l(y)$; 2. $g_l = WH(u \otimes u)$. Here, by the linearity of f_l, we have $f_l = WH(u)$ for some u and $f_l f_l = WH(u \otimes u)$.

Claim 2. *If $g_l \neq WH(u \otimes u)$, then $Pr_{r_1,r_2}[g_l(r_1 \otimes r_2) \neq f_l(r_1)f_l(r_2)] \succeq_{2\epsilon} 1/4$*

Proof: Let U, W be matrices such that $g_l(x \otimes y) = xUy$ and $f_l(x)f_l(y) = xWy$.

If $U \neq W$, then $\{r_2 \in 2^{n_1}|Ur_2 \neq Wr_2\} \succeq_0 2^{n_1}/2$ as witnessed by the following circuit: Let (i, j) be a position where U and W differ. Consider the circuit mapping r_2 from $\{r_2 \in 2^{n_1}|Ur_2 \neq Wr_2\}$ to $\hat{r_2}$ where $\hat{r_2} < 2^{n_1}/2$ is obtained from r_2 by erasing its jth bit $(r_2)_j$. For each $r_2 < 2^{n_1}/2$, let $r_2^0 < 2^n$ be such that $r_2 = \hat{r_2^0}$ and $(r_2^0)_j = 0$ and let $r_2^1 < 2^{n_1}$ be such that $r_2 = \hat{r_2^1}$ and $(r_2^1)_j = 1$. Then, for each $r_2 < 2^n/2$, r_2^0 or r_2^1 is in $\{r_2 \in 2^{n_1}|Ur_2 \neq Wr_2\}$.

Furthermore, if $U \neq W$, we similarly observe that $\{r_1 \in 2^{n_1}|r_1Ur_2 \neq r_1Wr_2\} \succeq_0 2^n/2$ for each $r_2 < 2^{n_1}$. Hence, by Proposition 2.3 2., $\{\langle r_1, r_2\rangle |g_l(r_1\otimes r_2) \neq f_l(r_1)f_l(r_2)\} \succeq_\epsilon 2^{2n}/4$. This proves the claim.

Suppose now that $g_l \neq WH(u \otimes u)$. As $\{\langle r_1, r_2\rangle |g'(r_1 \otimes r_2) = f'(r_1)f'(r_2)\}$ is a subset of

$\{\langle r_1, r_2\rangle |g'(r_1\otimes r_2) = g_l(r_1\otimes r_2)\wedge g_l(r_1\otimes r_2) = f_l(r_1)f_l(r_2)\wedge f'(r_1) = f_l(r_1)\wedge f'(r_2) = f_l(r_2)\}\cup$
$\{\langle r_1, r_2\rangle |g'(r_1 \otimes r_2) \neq g_l(r_1 \otimes r_2) \vee f'(r_1) \neq f_l(r_1) \vee f'(r_2) \neq f_l(r_2)\}$

which is $\preceq_{28\epsilon} 15/16(2^{2n_1})$ by Claim 2, we can conclude that

$Pr_w[D_1^{\pi,w}(x) = 1]_\epsilon \preceq_{2\epsilon} Pr_{r_1,r_2}[g'(r_1 \otimes r_2) = f'(r_1)f'(r_2)]_\epsilon \preceq_{28\epsilon} 15/16$.

It remains to consider the case that $g_l = WH(u \otimes u)$.

For each $u < 2^{2n_1}$, $R = \{r|\Sigma_i r_i(A_i u \otimes u) \neq \Sigma_i r_i b_i\} \cap 2^m \succeq_0 2^m/2$ as it is witnessed by the following circuit. Let j be the first such that $A_j u \otimes u \neq b_j$. The circuit maps each $r \in R$ to $\hat{r}$ where $\hat{r} < 2^m/2$ is obtained from r by erasing its jth bit r_j. For each $r < 2^m/2$, let $r^0 < 2^m$ be such that $r = \hat{r^0}$ and $r^0_j = 0$ and let $r^1 < 2^m$ be such that $r = \hat{r^1}$ and $r^1_j = 1$. Then, for each $r < 2^m/2$, $r^0 \in R$ or otherwise $\Sigma_i r^0_i(A_i u \otimes u) = \Sigma_i r^0_i b_i$ and hence $r^1 \in R$.

Furthermore, assuming $g_l = WH(u \otimes u)$, $\{r|g'(z) = \Sigma_i r_i b_i\}$ is a subset of

$$\{r|\Sigma r_i(A_i u \otimes u) = \Sigma_i r_i b_i \wedge g_l(z) = g'(z)\} \cup \{r|g_l(z) \neq g'(z)\}$$

Thus, $Pr_w[D_1^{\pi,w}(x) = 1]_\epsilon \preceq_{2\epsilon} Pr_r[g'(z) = \Sigma_i r_i b_i]_\epsilon \preceq_{10\epsilon} 9/16$.

In all cases, $Pr_w[D_1^{\pi,w}(x) = 1]_\epsilon \preceq_{28\epsilon} 63/64$ so

$$\{w \in 2^{3n_1+n_1^2+m} | D_1^{\pi,w}(x) = 0\} \succeq_{30\epsilon} 1/64(2^{3n_1+n_1^2+m})$$

Therofore, for sufficiently big constant m_0, Chernoff's bound from Proposition 2.3 with $\delta^2 := c30\epsilon + 1/100^2$ and sufficiently small ϵ implies that $Pr_{w<2^{kn^k}}[D^{\pi,w}(x) = 1] \preceq_0 1/2$.

To conclude the proof of the exponential PCP theorem in APC_1 it thus remains to derive the Test of linearity.

5.1. **Test of linearity in APC_1.** In this section we prove Proposition 5.1 in the theory APC_1.

We cannot use the Fourier transformation argument directly as in Arora-Barak [1] which would require to prove the existence of exponentially long Fourier expansions (and it is not clear if this could be managed, for example, using a representation by p-size circuits). Instead we formalize the so called majority correction argument. Our presentation is a minor modification of Moshkovitz [21].

In the argument we first define a function $g_\epsilon(x)$ as the majority value of the expression $f(y) + f(x + y)$ for possible y's. Assuming that $f(x) = f(x + y) + f(y)$ holds with high probability for random x, y it is shown then that such $g_\epsilon(x)$ is linear and close to f.

Let $\epsilon > 0$ be sufficiently small and $\epsilon^{-1} \in Log$. Define $g_\epsilon : 2^n \mapsto 2$ by

$$g_\epsilon(x) = 1 \qquad \equiv_{def} \qquad Pr_{y<2^n}[f(y) + f(x + y) = 1]_\epsilon \geq 1/2$$

Therefore, for any $x < 2^n$, $P_x := Pr_{y<2^x}[g_\epsilon(x) = f(y) + f(x + y)]_\epsilon \geq 1/2$.

We will now derive three claims that can be combined into a proof of Proposition 5.1.

Claim 1.: $Pr_{\langle x,y\rangle}[f(x + y) \neq f(x) + f(y)]_\epsilon \succeq_{8\epsilon+13\epsilon^2+2\epsilon^3} \frac{1}{2} Pr_x[f(x) \neq g_\epsilon(x)]_\epsilon$

This holds trivially if $Size(\{x|g_\epsilon(x) \neq f(x)\} \cap 2^n, \epsilon) = 0$. Otherwise, define sets $T := \{\langle x, y\rangle \,|f(x + y) \neq f(x) + f(y)\}$ and $G := \{x|g_\epsilon(x) \neq f(x)\}$. Then,

$Pr_{x<2^n,y<2^n}[f(x + y) \neq f(x) + f(y)]_\epsilon \geq Size(T \cap (G \times 2^n) \cap 2^{2n}, \epsilon)/2^{2n} =$

$$\frac{Size((G \cap 2^n) \times 2^n, \epsilon)}{2^{2n}} \frac{Size(T \cap (G \times 2^n) \cap 2^{2n}, \epsilon)}{Size((G \cap 2^n) \times 2^n, \epsilon)}$$

By Proposition 2.2*iii*), $(G \cap 2^n) \times 2^n \approx_\epsilon Size(G \cap 2^n, \epsilon)2^n$, so the first fraction in the expression above is $\approx_{2\epsilon} Pr_{x<2^n}[g_\epsilon(x) \neq f(x)]_\epsilon$.

Further, for each $x \in G \cap 2^n$, $P_x \geq 1/2$ and in particular, $2^n/2 \preceq_\epsilon T_x = \{y| \langle x, y\rangle \in T\}$.

Hence, by Proposition 2.3 2., $Size(G,\epsilon)2^n/2 \preceq_{3\epsilon+\epsilon^2} T \cap (G \times 2^n)$, and

$$\frac{Size(T \cap (G \times 2^n) \cap 2^{2n}, \epsilon)}{Size((G \cap 2^n) \times 2^n, \epsilon)} \succeq_{4\epsilon+\epsilon^2} \frac{Size(G,\epsilon)2^n}{2Size((G \cap 2^n) \times 2^n, \epsilon)} \succeq_{2\epsilon} 1/2$$

Applying now Proposition 2.2 *iii*) we obtain Claim 1.

Claim 2.: If $Pr_{\langle x,y\rangle}[f(x+y) \neq f(x)+f(y)]_\epsilon < \frac{3}{32}$, then $\forall x < 2^n$, $P_x > \frac{3}{4}$.

Fix $x < 2^n$ and define
$A := \{\langle y,z\rangle \,|g_\epsilon(x) = f(y)+f(x+y) \wedge g_\epsilon(x) = f(x+z)+f(z)\}$
$B := \{\langle y,z\rangle \,|g_\epsilon(x) \neq f(y)+f(x+y) \wedge g_\epsilon(x) \neq f(x+z)+f(z)\}$

Then, $Pr_{y,z}[f(y)+f(x+y) = f(z)+f(x+z)]_\epsilon = Pr_{y,z}[\langle y,z\rangle \in A \cup B]_\epsilon$.

By 2.3 1.*ii*), $(A \cup B) \cap 2^{2n} = (A \cap 2^{2n}) \cup (B \cap 2^{2n}) \approx_{3\epsilon} Size(A \cap 2^{2n}, \epsilon) + Size(B \cap 2^{2n}, \epsilon)$.
Thus, $Pr_{y,z}[\langle y,z\rangle \in A \cup B]_\epsilon \approx_{4\epsilon} Pr_{y,z}[\langle y,z\rangle \in A] + Pr_{y,z}[\langle y,z\rangle \in B]$.

Next, let $A' := \{y | g_\epsilon(x) = f(x+y)+f(x)\}$. Using Proposition 2.2 *iii*) twice, $A \cap 2^{2n}$ is $(A' \cap 2^n) \times (A' \cap 2^n) \approx_{2\epsilon} Size(A' \cap 2^n, \epsilon) Size(A' \cap 2^n, \epsilon)$. Therefore, $Pr_{y,z}[\langle y,z\rangle \in A] \approx_{3\epsilon} P_x^2$.
As by Proposition 2.3 1.*i*), $\{y | g_\epsilon(x) \neq f(x+y)+f(y)\} \cap 2^n = 2^n - A' \cap 2^n$ is $\approx_{2\epsilon} 2^n - Size(A' \cap 2^n, \epsilon)$, we analogously obtain $Pr_{y,z}[\langle y,z\rangle \in B] \approx_{9\epsilon} (1-P_x)^2$.

Therefore, $Pr_{y,z}[f(y)+f(y+x) = f(z)+f(x+z)] \approx_{17\epsilon} P_x^2 + (1-P_x)^2$.

Define now,

$C := \{\langle y,z\rangle \,|f(y+z) \neq f(y)+f(z)\}$
$D := \{\langle y,z\rangle \,|f(y+z) \neq f(x+y)+f(x+z)\}$

Then, $2^{2n} - (C \cap 2^{2n}) \cup (D \cap 2^{2n}) \subseteq (A \cup B) \cap 2^{2n}$ and by Proposition 2.2 *i*) we have $2^{2n} - (C \cap 2^{2n}) \cup (D \cap 2^{2n}) \preceq_0 (A \cup B) \cap 2^{2n}$.
By Proposition 2.3 1.*ii*), $(C \cap 2^{2n}) \cup (D \cap 2^{2n}) \preceq_{3\epsilon} Size(C \cap 2^{2n}, \epsilon) + Size(D \cap 2^{2n}, \epsilon)$, so $2^{2n} - Size(C \cap 2^{2n}, \epsilon) - Size(D \cap 2^{2n}, \epsilon) \preceq_{4\epsilon} 2^{2n} - (C \cap 2^{2n}) \cup (D \cap 2^{2n})$.
Moreover, by the assumption, $Pr_{y,z}[f(y)+f(z) \neq f(y+z)]_\epsilon < 3/32$ and similarly, $Pr_{y,z}[f(y+z) \neq f(x+y)+f(x+z)]_\epsilon < 3/32$. Therefore,

$$Pr_{y,z}[f(y)+f(x+y) = f(z)+f(x+z)]_\epsilon \succeq_{5\epsilon} 13/16$$

This shows that $P_x^2+(1-P_x)^2 \succeq_{22\epsilon} \frac{13}{16}$ and $2(P_x-\frac{1}{4})(P_x-\frac{3}{4})+\frac{10}{16} \succeq_{22\epsilon} \frac{13}{16}$. As $P_x \geq 1/2$, $P_x < 3/4$ would imply $\frac{10}{16}2^n \succeq_{22\epsilon} \frac{13}{16}2^n$ contradicting dual weak pigeonhole principle. Hence, Claim 2 follows.

Claim 3.: If $Pr_{x,y}[f(x+y) \neq f(x)+f(y)]_\epsilon < 3/32$, then g_ϵ is linear.

By Claim 2, $\forall x,y < 2^n$,

$Pr_z[g_\epsilon(x) \neq f(x+z)+f(z)]_\epsilon \preceq_{3\epsilon} 1/4$
$Pr_z[g_\epsilon(y) \neq f(y+z)+f(z)]_\epsilon \preceq_{3\epsilon} 1/4$
$Pr_z[g_\epsilon(x+y) \neq f(y+z)+f(z+x)]_\epsilon \preceq_{3\epsilon} 1/4$

Therefore,

$$Pr_z[g_\epsilon(x) = f(x+z)+f(z) \wedge g_\epsilon(y) = f(y+z)+f(z) \wedge g_\epsilon(x+y) = f(y+z)+f(z+x)]_\epsilon \succeq_{16\epsilon} 1/4$$

The last estimation implies that if ϵ is sufficiently small, there exists z_0 (and we can efficiently find it) such that

$g_\epsilon(x) = f(x + z_0) + f(z_0)$,
$g_\epsilon(y) = f(y + z_0) + f(z_0)$,
$g_\epsilon(x + y) = f(y + z_0) + f(z_0 + x)$

which shows that $g_\epsilon(x) + g_\epsilon(y) = g_\epsilon(x + y)$ and proves Claim 3.

We can now derive Proposition 5.1. Assume that for each linear function g we have $Pr_x[g(x) = f(x)]_\epsilon < p$. By Claim 3, $Pr_{x,y}[f(x + y) \neq f(x) + f(y)]_\epsilon \geq 3/32$ or g_ϵ is linear. This means that either $Pr_{x,y}[f(x + y) = f(x) + f(y)]_\epsilon \preceq_{3\epsilon} 29/32$ or $Pr_x[g_\epsilon(x) = f(x)] < p$. In the latter case, $Pr_x[g_\epsilon(x) \neq f(x)] \succeq_{3\epsilon} 1 - p$ and by Claim 1,
$Pr_{x,y}[f(x + y) = f(x) + f(y)]_\epsilon \preceq_{11\epsilon+13\epsilon^2+2\epsilon^3} 1/2 + p/2$.

6. Pseudorandom constructions in PV_1

In order to derive the PCP theorem in PV_1 we will need to prove in the theory PV_1 the existence and some properties of the (n, d, λ)-graphs (see their definition below). While the construction itself is very combinatorial, its analysis uses algebraic techniques, e.g. properties of eigenvectors, which we do not know how to formalizable in PV_1.

Using an equivalent combinatorial definition of the (n, d, λ)-graphs it is possible to derive their existence and main properties by only combinatorial tools. However, we need it for the algebraic equivalent and the implication producing the algebraic (n, d, λ)-graphs from the combinatorial (n, d, λ)-graphs is one of those which seem to require the algebraic techniques we are trying to avoid.

Therefore, we will employ an approximation of some algebraic tools which will allows us to derive slightly weaker results about the algebraic (n, d, λ)-graphs that are, however, sufficient to derive the PCP theorem.

For the history of the field leading to the results presented in this section see Arora-Barak [1, Chapter 21].

6.1. Definition and some properties of the (n, d, λ)-graphs.

In PV_1 we say that a graph G is d-regular if each vertex appears in exactly d edges. We allow G to have multiple edges and self-loops. The random-walk $n \times n$ matrix A of a d-regular graph G with n vertices consists of elements $A_{i,j}$ being the number of edges between the i-th and the j-th vertex in G divided by d. All our graphs will be undirected, hence, their random-walk matrices will be symmetric. For any k and a graph G with n vertices, we denote by G^k the graph with n vertices which has an edge between the ith and the jth vertex for each k step path between the ith and the jth vertex in G.

We would like to define now the second largest eigenvalue of G denoted as $\lambda(G)$. The parameter $\lambda(G)$ corresponds to a certain expansion property of G (see Proposition 6.3) and normally it is defined as the maximum value of $||Ax||$ over all vectors x in n-dimensional real vector space such that $||x|| = 1$ and $\Sigma_i x_i = 0$. Here, $||y|| = (\Sigma_i y_i^2)^{1/2}$ and A is the random-walk matrix of graph G with n vertices. In PV_1 we will approximate this definition using a sufficiently dense net of rational numbers.

The theory PV_1 proves that each x is the value of an expression of the form $\Sigma_{i=0}^{|x|} 2^i y_i$ for $y_i \in \{0,1\}$ which is encoded in a natural way. In PV_1 we write that $x \in Q^n/m$ if $x = (x_1, ..., x_n)$ and each x_i is $\frac{a}{b}$ or $-\frac{a}{b}$ for $a \in [m] \cup \{0\}, b \in [m] = \{1, ..., m\}$ where a, b are represented by products of such expressions $\Sigma_i 2^i y_i, y_i \in \{0,1\}$. These products are also encoded in a natural way. In such cases we might write $a = c \cdot d$ to specify that a is represented by a product of c and d where c, d might be products of other expressions of the form $\Sigma_i 2^i y_i$.

Let L be a sufficiently big constant, then $SQRT$ is a function which given nonnegative $r \in Q/m$, $m > 1$, produces $SQRT(r) \in Q/(Lm)^7$ such that

$$0 \leq (SQRT(r))^2 - r \leq \frac{1}{L}$$

where we ignore the difference between $SQRT(r)$ and the value of the expression it encodes. Moreover, $SQRT$ satisfies the following: If input r is a fraction of the form $\frac{c \cdot c \cdot e}{d \cdot d \cdot f} \in Q/m$ where c, d are sums $\Sigma_i 2^i y_i$ with $y_i \in \{0,1\}$ (and e, f might be products of such sums), then

$$SQRT(\frac{c \cdot c \cdot e}{d \cdot d \cdot f}) = \frac{c}{d} \cdot SQRT(\frac{e}{f}) \qquad (*)$$

which is illustrating the representation of the number encoded in $SQRT(r)$. The representation of $\frac{c^2 e}{d^2 f}$ guarantees that $SQRT$ does not need to perform factorization.

The function $SQRT$ is essentially the usual algorithm approximating square root by a digit-by-digit search. We will assume that $SQRT$ works as follows: given $r \in Q/m$, it first finds out maximal $e, f \in [m]$ such that the current representation of r is $\frac{e \cdot e}{f \cdot f} \frac{p}{q}$ for some $p, q \in [m]$, and then by a digit-by-digit search it finds the first $c \in [L^7 m^6]$ such that $SQRT(r)$ which is $\frac{ec}{2fLqm^4} \in Q/(Lm)^7$ satisfies $0 \leq (\frac{ec}{2fLqm^4})^2 - r \leq \frac{1}{L}$. To get such c we want to satisfy $c^2 - 4pqL^2 m^8 \leq 4Lm^6$. Thus $c \leq 2\sqrt{pq}Lm^4 + 2\sqrt{L}m^3 \leq 7m^6$. The value c is then produced by a p-time algorithm approximating $2\sqrt{pq}Lm^4$ so it is unique and its existence is provable in PV_1.

For $x \in Q^n/m$, put $||x|| := SQRT(\Sigma_i x_i^2)$ where the input $\Sigma_i x_i^2 \in Q/(nm^{2n})$ is computed so that if each $x_i = \pm\frac{a_i c}{b_i d}$ for some common c, d, then $\Sigma_i x_i^2$ is represented as $\frac{e \cdot c \cdot c}{f \cdot d \cdot d}$ for some e, f.

By the definition, if $x \in Q^n/m$, $x \neq 0$, then $\frac{x}{||x||} \in Q^n/((Lnm^{2n})^7 m)$ and using $(*)$, $||\frac{x}{||x||}|| = 1$. Note that $||x||$ might be a fraction so we assume that $\frac{x}{||x||}$ is rearranged appropriately.

However, by $||x||^2$ we always mean $\langle x, x \rangle$ where $\langle x, y \rangle := \Sigma_i x_i y_i$ for $x, y \in Q/m$. The n-dimensional unite vector is defined as $\mathbf{1} := (1/n, ..., 1/n)$.

The parameter $\lambda(G)$ is defined as the maximum value of $||Ax||$ over all possible vectors $x \in Q^n/(Ln)^{(Ln)^L}$ such that $||x|| = 1$ and $\langle x, \mathbf{1} \rangle = 0$. Here again, the vector $Ax \in Q^n/(n(d(Ln)^{(Ln)^L})^n)$ (with elements of length $poly(n)$) is computed so that if each $x_i = \pm\frac{a_i c}{b_i d}$ for some common c, d, then $(Ax)_j = \pm\frac{c \cdot e_j}{d \cdot f_j}$ for some e_j, f_j.

We will not need to prove $\exists y,\ y = \lambda(G)$ in PV_1 but we will work with formulas of the form $\lambda(G) \leq y$ which are Π_1^b. To see this note that in $\lambda(G) \leq y$ we universally quantify over all x's in $Q^n/(Ln)^{(Ln)^L}$. For each j, there are $\leq m^j$ ways how to represent $b \in [m]$ as a product of j numbers so this is a universal quantification over $\leq 2^{n^{O(1)}}$ x's. For each such x, predicates $||x|| = 1$ and $||Ax|| \leq y$ are computable in time $n^{O(1)}$.

Definition 6.1. *A d-regular graph G with n vertices is (n,d,λ)-graph if $\lambda(G) \leq \lambda < 1$.*

We will often use Cauchy-Schwarz inequality in PV_1 which can be obtained in the standard way.

Proposition 6.1. *(Cauchy-Schwarz inequality in PV_1) For every n, m and $x, y \in Q^n/m$, $\langle x,y\rangle^2 \leq ||x||^2 \cdot ||y||^2$ and therefore, if $n \in Log$ (and thus $||x||$ exists), also $\langle x,y\rangle \leq ||x|| \cdot ||y||$.*

Proof. If $y = 0$, the inequality holds. Otherwise, let $z := x - \frac{\langle x,y\rangle}{\langle y,y\rangle} y$. Then, $\langle z,y\rangle = \langle x,y\rangle - \frac{\langle x,y\rangle}{\langle y,y\rangle}\langle y,y\rangle = 0$. Therefore, $||x||^2 = (\frac{\langle x,y\rangle}{\langle y,y\rangle})^2||y||^2 + ||z||^2 = \frac{\langle x,y\rangle^2}{||y||^2} + ||z||^2 \geq \frac{\langle x,y\rangle^2}{||y||^2}$. □

In Peano Arithmetic, regular graphs G satisfy $\lambda(G) \leq 1$ but in PV_1 we will have just $\lambda(G) \leq 1 + \epsilon + 1/L$ for any rational $\epsilon > 0$. Fortunately, this is enough to derive the PCP theorem in PV_1.

Proposition 6.2. *For any d and any rational $\epsilon > 0$, PV_1 proves that for any d-regular graph G with $n \in Log$ vertices, $\lambda(G) < 1 + \epsilon + 1/L$.*

Proof. As the statement we want to prove is $\forall\Sigma^b_1$, by $\forall\Sigma^b_1$-conservativity of S^1_2 over PV_1, we can work in the theory S^1_2.

Let A be the random-walk matrix of G. We want to show that $\lambda(G) < 1 + \epsilon + 1/L$. Using Cauchy-Schwarz inequality, for every $x \in Q^n/(Ln)^{(Ln)^L}$ such that $||x|| = 1$,

$$||Ax||^2 = \Sigma_i(\Sigma_j A_{i,j}x_j)^2 \leq \Sigma_i(\Sigma_j A^2_{i,j}\Sigma_j x^2_j) \leq \Sigma_i\Sigma_j A^2_{i,j} \leq \Sigma_i\Sigma_j A_{i,j} = \Sigma_i 1 = n$$

As $A_{i,j} = A_{j,i}$, we have $\langle x, Ay\rangle = \Sigma_i(x_i\Sigma_j A_{i,j}y_j) = \Sigma_j(y_j\Sigma_i(x_i A_{j,i})) = \langle Ax, y\rangle$ and $||Ax||^4 = \langle Ax, Ax\rangle^2 = \langle A^2x, x\rangle^2 \leq ||A^2x||^2$ where A^2 is the random-walk matrix of G^2, so also $||A^2x||^2 \leq n$ and $||Ax||^4 \leq n$. This shows that

$$\forall k \leq K \log\log n\ (\forall A, ||Ax||^2 \leq n^{1/(2^k)} \rightarrow \forall A, ||Ax||^2 \leq n^{1/(2^{k+1})})$$

where K is a sufficiently big constant depending only on ϵ and the universal quantifier before A goes only over random-walk matrices of d-regular graphs with n vertices. Note also that $n^{1/(2^k)}$ might be irrational but we can assume that it is approximated with a sufficiently small constant error so that the predicate $||Ax||^2 \leq n^{1/(2^k)}$ is Π^b_1.

Then, by Π^b_1-LLIND (available in S^1_2), we have $\forall A, ||Ax||^2 \leq n^{1/(\log n)^K}$ which is $< (1+\epsilon)^2$ by the choice of K and therefore $||Ax|| \leq 1 + \epsilon + 1/L$.

□

We can now prove that the (n,d,λ)-graphs satisfy a useful expansion property. The term $\frac{\lambda d}{Ln^2}$ occuring in its formulation is an error resulting from our approximations in PV_1.

Proposition 6.3. *(in PV_1) If G is (n,d,λ)-graph with $n \in Log$ vertices V and edges E, then for every $S \subseteq V, |S| \leq n/2$,*

$$|E(S, V-S)| \geq \frac{d|S|(1-\lambda)}{2} - \frac{\lambda d}{Ln^2}$$

where $E(S,T)$ denotes the set of edges $(i,j) \in E$ with $i \in S, j \in T$.

Proof. It suffices to show:

$$|E(S, V-S)| \geq (1-\lambda)\frac{d|S||V-S|}{n} - \frac{\lambda}{Ln^2}$$

Let $x \in Q^n/n$ be the following vector: $x_i = |V - S|$ if $i \in S$ and $x_i = -|S|$ if $i \in V - S$. Put $Z := \Sigma_{i,j} A_{i,j}(x_i - x_j)^2$ for the random-walk matrix A of G. Then,
$Z = \frac{2}{d}|E(S, V - S)|(|S| + |V - S|)^2$. As A's rows and columns sum up to one, we have also

$$Z = \Sigma_{i,j} A_{i,j} x_i^2 - 2\Sigma_{i,j} A_{i,j} x_i x_j + \Sigma_{i,j} A_{i,j} x_j^2 = 2||x||^2 - 2\langle x, Ax\rangle$$

Further, $\Sigma x_i = 0$ and $\frac{x}{||x||} \in Q^n/((Lnn^{2n})^7 n)$ so $||Ax|| = ||A\frac{x}{||x||}||||x|| \leq \lambda||x||$. By Cauchy-Schwarz inequality, $\langle x, Ax\rangle \leq ||x|| \cdot ||Ax||$. Therefore,

$$\frac{1}{d}|E(S, V - S)|(|S| + |V - S|)^2 \geq (1 - \lambda)||x||^2 - \lambda/L$$

It remains to observe that $||x||^2 = |S||V - S|(|S| + |V - S|)$ □

In the following proposition we use the notion of probability Pr on sets of polynomial size $poly(n)$ for $n \in Log$. We assume that this is defined in PV_1 in a natural way using an exact counting of sets of polynomial size $poly(n), n \in Log$ which is also definable in PV_1 in a usual way. This should not be confused with the definition of Pr in APC_1.

Proposition 6.4. *For any $d, l < L$, PV_1 proves that for each (n, d, λ)-graph G with $n \in Log$ vertices V, for any $S \subseteq V, |S| \leq |V|/2$,*

$$Pr_{(i,j)\in E(G^l)}[i \in S \wedge j \in S] \leq \frac{|S|}{|V|}(\frac{|S|}{|V|} + 2\lambda^l)$$

where $E(G^l)$ denotes the set of all edges in G^l.

Proof. For empty S the statement holds. Otherwise put $S := \{i_1, ..., i_{|S|}\}$. If $\langle x, \mathbf{1}\rangle = 0$, then $\langle Ax, \mathbf{1}\rangle = 0$ for the random-walk matrix A of G. As A^l is the random-walk matrix of d^l-regular graph G^l, $A^{l-1} \in Q^{n\times n}/d^{l-1}$ and $\frac{A^{l-1}x}{||A^{l-1}x||} \in Q^n/(Ln((d^{l-1}n)^n n)^{2n})^7(d^{l-1}n)^n n$ for $x \in Q^n/n$. By the choice of d, l, this does not exceed the range $(Ln)^{Ln^L}$ and we can apply $\lambda(G) \leq \lambda$ to obtain $||A^l x|| \leq \lambda^l||x||$ for any $x \in Q^n/n$ with $\langle x, \mathbf{1}\rangle = 0$. Now, use the inequality from the proof of Proposition 6.3:

$$\frac{|E(S, V - S)|}{d^l} \geq \frac{|S||V - S|(1 - \lambda^l)}{|V|} - \frac{\lambda^l}{Ln^2}$$

Then, $Pr_{(i,j)\in E(G^l)}[i \in S \wedge j \in S] = \frac{1}{|V|}\Sigma_{m=1}^{|S|}(1 - Pr[j \notin S | i = i_m])$ is

$$\frac{|S|}{|V|}(1 - \Sigma_{m=1}^{|S|}\frac{|E(i_m, V - S)|}{|S|d^l}) = \frac{|S|}{|V|}(1 - \frac{|E(S, V - S)|}{|S|d^l}) \leq \frac{|S|}{|V|}(\frac{|S|}{|V|} + 2\lambda^l)$$

□

6.2. **A technical tool.** Sometimes we will need to use an assumption which has the form "$||Ax|| \leq \lambda$ for $x \in Q^n/(Ln)^{(Ln)^L}$" even for x's exceeding the range fixed by $(Ln)^{(Ln)^L}$. We will now prove a simple approximation lemma which allows this in some cases. It illustrates a type of approximation which we use more often. The matrix A in its formulation will not need to represent a random-walk matrix. In our applications A will be a result of certain operations on random-walk matrices.

Proposition 6.5. *(in PV_1) Let A be an $n \times n$ matrix of elements from $Q/(2L^2n^5d)$, for $n \in Log$. Further, let $s \in Log$. If $||Ax||^2 \leq y(||x||^2 + 1/L)$ for any $x \in Q^n/(Ln)^{(Ln)^L}$, then for any $x \in Q^n/m$,*

$$||Ax||^2 \leq (y(1 + \frac{1}{L}) + \frac{1}{L})(||x||^2 + \frac{1}{Ls})$$

Proof. For $x \in Q^n/m$ and $s \in Log$, define $||x||'$ in the same way as $||x||$ but with $SQRT$ redefined so that $0 \leq (SQRT(||x||^2))^2 - ||x||^2 \leq 1/(Ls)$.

It suffices now to approximate $\frac{x}{||x||'}, x \neq 0$ by $c \in Q^n/(Ln)^{(Ln)^L}$ with $||c||^2 \leq 1$ such that $|||A\frac{x}{||x||'}||^2 - ||Ac||^2| \leq \frac{1}{L}$. Then,

$||Ax||^2 \leq ||A\frac{x}{||x||'}||^2(||x||^2 + \frac{1}{Ls}) \leq (y(||c||^2 + 1/L) + \frac{1}{L})(||x||^2 + \frac{1}{Ls}) \leq$
$\leq (y(1 + 1/L) + \frac{1}{L})(||x||^2 + \frac{1}{Ls})$

The approximation: for each i, $|\frac{x_i}{||x||'}| \leq 1$ so we can find c_i (i.e. PV_1 can prove its existence) such that $0 \leq \frac{x_i}{||x||'} - c_i \leq 1/(18L^5n^{13}d^2)$. Then $||c||^2 \leq ||\frac{x}{||x||'}||^2 \leq 1$ and for each l, $|A_{l,i}\frac{x_i}{||x||'} - A_{l,i}c_i| \leq 1/(9L^3n^8d)$. Hence, $|(A\frac{x}{||x||'})_l - (Ac)_l| \leq 1/(9L^3n^7d)$. As $(A\frac{x}{||x||'})_l, (Ac)_l \leq 3L^2n^6d$, we can conclude $|||A\frac{x}{||x||'}||^2 - ||Ac||^2| \leq 1/L$ □

Using a similar approximation, we will derive one more useful lemma.

For any $n \times n$ matrix A with elements from Q/m, we say that $||A|| \leq 1$ iff for every $x \in Q^n/(Ln)^{(Ln)^L}$, $||Ax||^2 \leq (1 + 2/L)(||x||^2 + 1/L)$.

Proposition 6.6. *For any λ and $d < L$, PV_1 proves the following. Let A be a random-walk matrix of a d-regular graph G with $n \in Log$ vertices such that $\lambda(G) \leq \lambda \in Q/(Ln^2)$. Let J be $n \times n$ matrix such that $J_{i,j} = 1/n$ for every i, j. Then,*

$$A = (1 - \lambda)J + \lambda C$$

for some C with $||C|| \leq 1$

Proof. Define $C := \frac{1}{\lambda}(A - (1 - \lambda)J) \in Q^{n\times n}/(2L^2n^5d)$. We want to prove that for any $x \in Q^n/(Ln)^{(Ln)^L}$, $||Cx||^2 \leq (||x||^2 + 1/L)(1 + 2/L)$. Decompose x as $x = \alpha\mathbf{1} + y$ for some $\alpha \in Q/((Ln)^{(Ln)^L})^{n+1}$ where $\langle \mathbf{1}, y\rangle = 0$.

Similarly as in Proposition 6.5, approximate $\frac{y}{||y||}$ by vector c with $||c||^2 \leq 1$ so that $||A\frac{y}{||y||}||^2 \leq ||Ac||^2 + \lambda^2/L$ and $\frac{c}{||c||} \in Q^n/(Ln)^{(Ln)^L}$. This time we can do it without the absolute value because all elements of A are positive. Note also that for $d < L$ the range of $\frac{c}{||c||}$ does not exceed $(Ln)^{(Ln)^L}$.

Since $A\mathbf{1} = \mathbf{1}$ and $J\mathbf{1} = \mathbf{1}$, we have $C\alpha\mathbf{1} = \alpha\mathbf{1}$. As $\langle y, \mathbf{1}\rangle = 0$, $Jy = 0$ and $Cy = \frac{1}{\lambda}Ay$. Using $\langle Ay, \alpha\mathbf{1}\rangle = 0$ and $||Ac|| \leq \lambda||c||$, we obtain,

$||Cx||^2 = ||\alpha 1 + \frac{1}{\lambda}Ay||^2 = ||\alpha\mathbf{1}||^2 + ||\frac{1}{\lambda}Ay||^2 \leq ||\alpha\mathbf{1}||^2 + \frac{1}{\lambda^2}(||Ac||^2 + \frac{\lambda^2}{L})(||y||^2 + \frac{1}{L}) \leq$
$||\alpha\mathbf{1}||^2 + (1 + 2/L)(||y||^2 + 1/L) \leq (1 + 2/L)(||x||^2 + 1/L)$ □

6.3. **The tensor product.** The explicit construction of the (n, d, λ)-graphs needs two graph products, the tensor product and the replacement product, which we describe in this and the next section. More details about the tensor product and the replacement product can be found in [1, Section 21.3.3] resp. [1, Section 21.3.4] .

Definition 6.2. *(in PV_1) If $A = \{a_{i,j}\}_{i,j=1,...,n}$ is the $n \times n$ random-walk matrix of d-degree graph G and $A' = \{a'_{i',j'}\}$ is the $n' \times n'$ random-walk matrix of d'-degree graph G', then the random-walk matrix of $G \otimes G'$, denoted as $A \otimes A'$ is the $nn' \times nn'$ matrix that in the $\langle i, i' \rangle$th row and the $\langle j, j' \rangle$th column has the value $a_{i,j}a'_{i',j'}$.*

This means that $G \otimes G'$ has a cluster of n' vertices for every vertex in G. If (i, j) is an edge in G and (i', j') is an edge in G', then there is an edge between the i'-th vertex in the cluster corresponding to i and the j'-th vertex in the cluster corresponding to j. Therefore, $G \otimes G'$ has degree $d'd$ and nn' vertices. We can see matrix $A \otimes A'$ as consisting of blocks of the form $a_{i,j}A'$, that is, intuitively, $A \otimes A'$ is matrix A with elements multiplied by copies of A'.

In Peano Arithmetic, $\lambda(G \otimes G') \leq max\{\lambda(G), \lambda(G')\}$ for regular graphs G, G'. The standard derivation of this bound uses the existence of an orthogonal basis of eigenvectors for symmetric matrices which uses the fundamental theorem of algebra (applied to determinant of matrix $A - xI$ consisting of exponentially many terms). We do not know how to formalize this in PV_1. Instead, we will derive a weaker bound which is sufficient for our purposes.

Note first a simple consequence of Cauchy-Schwarz inequality.

Proposition 6.7. *(in PV_1) For every two $n \times n$ matrices A, B and $x \in Q^n/m$ where $n \in Log$, we have $||(A + B)x|| \leq ||Ax|| + ||Bx|| + 1/L^{1/2}$.*

Proof. $$||(A+B)x||^2 = \langle (A+B)x, (A+B)x \rangle = ||Ax||^2 + 2\langle Ax, Bx \rangle + ||Bx||^2 \leq$$
$$\leq ||Ax||^2 + 2||Ax||||Bx|| + ||Bx||^2 \leq (||Ax|| + ||Bx||)^2$$
and so $||(A + B)x|| \leq ||Ax|| + ||Bx|| + 1/L^{1/2}$. □

Proposition 6.8. *PV_1 proves that if G is a d-regular graph with $n \in Log$ vertices and G' is a d'-regular graph with $n' \in Log$ vertices such that $d, d' < L$, $\lambda(G) \leq \lambda \in Q/(Ln^2)$ and $\lambda(G') \leq \lambda' \in Q/(L(n')^2)$, then*

$$\lambda(G \otimes G') \leq ((1 + 6/L)^2 + 1/L)(max\{\lambda + \lambda' - \lambda\lambda', \lambda\lambda', \lambda', \lambda\}) + 3/L^{1/2}$$

(Note that PV_1 does not need to know that $\lambda(G) \leq 1$ or $\lambda(G') \leq 1$.)

Proof. Let A be the random-walk matrix of G of the form $n \times n$ and A' be the random-walk matrix of G' of the form $n' \times n'$. By Proposition 6.6 $A = (1 - \lambda)J_n + \lambda C$ for some C with $||C|| \leq 1$ and $n \times n$ all $1/n$ matrix J_n. Similarly, $A' = (1 - \lambda')J_{n'} + \lambda' C'$ for some C' with $||C'|| \leq 1$ and $n' \times n'$ all $1/n'$ matrix $J_{n'}$.

As tensor product satisfies $(A+B)\otimes C = A\otimes C+B\otimes C$ and $A\otimes(B+C) = A\otimes B+A\otimes C$, for any $x \in Q^{nn'}/(Lnn')^{(Lnn')^L}$ we have $(*)$:

$$||A \otimes A'x|| \leq (1 - \lambda)||(J_n \otimes J_{n'})x|| + (1 - \lambda)\lambda'||(J_n \otimes C')x||$$
$$+\lambda(1 - \lambda')||(C \otimes J_{n'})x|| + \lambda'\lambda||(C \otimes C')x|| + 3/L^{1/2}$$

If $\Sigma_i x_i = 0$, then $J_n \otimes J_{n'}x = 0$. If $x \in Q^n/(Ln)^{(Ln)^L}$, $||J_n x||^2 = \frac{1}{n}(\Sigma_i x_i)^2 \leq ||x||^2$ where we used $\langle x, (1, ..., 1) \rangle^2 \leq n||x||^2$ which follows from Cauchy-Schwarz inequality. Therefore, $||J_n|| \leq 1$ and similarly $||J_{n'}|| \leq 1$.

If $\lambda > 1$ or $\lambda' > 1$, we can trivially upper bound the term corresponding to $1-\lambda$ resp. $1-\lambda'$ in $(*)$ by 0. In all cases, to finish the proof it suffices to show that for any $n \times n$ matrix $A \in Q^{n\times n}/(2L^2n^5d)$, $n' \times n'$ matrix $B \in Q^{n'\times n'}/(2L^2(n')^5d)$ such that $||A|| \leq 1, ||B|| \leq 1$, for any $x \in Q^{nn'}/(Lnn')^{(Lnn')^L}$ with $||x|| = 1$, $||(A \otimes B)x|| \leq (1 + 6/L)^2 + 1/L$ holds.

For any $x \in Q^{nn'}/m'$ and $i \in [n']$ define $x^i \in Q^n/m$ so that for each $j \in [n]$,

$$x^i_j = \Sigma_{k\in\{n'(j-1)+1,\ldots,n'j\}} B_{i,(k-n'(j-1))}x_k$$

Then, $||(A \otimes B)x||^2 = \Sigma_{i\in[n']}||Ax^i||^2$ and as by Proposition 6.5 for each i, $||Ax^i||^2 \leq (\Sigma_{j\in[n]}(x^i_j)^2 + 1/(Ln'))((1 + 1/L)(1 + 2/L) + 1/L)$, we have,

$$||(A \otimes B)x||^2 \leq (1/L + \Sigma_{i\in[n']}\Sigma_{j\in[n]}(x^i_j)^2)(1 + 6/L)$$

Since also $||Bx||^2 \leq (||x||^2 + 1/(Ln))((1 + 1/L)(1 + 2/L) + 1/L)$, for each $j \in [n]$,

$$\Sigma_{i\in[n']}(\Sigma_{k\in\{n'(j-1)+1,\ldots,n'j\}} B_{i,(k-n'(j-1))}x_k)^2 \leq (\frac{1}{Ln} + \Sigma_{k\in\{n'(j-1)+1,\ldots,n'j\}}(x_k)^2)(1 + \frac{6}{L})$$

Therefore, if $||x|| = 1$, then $||(A \otimes B)||^2 \leq (1/L + (1 + 6/L)(1 + 1/L))(1 + 6/L)$, and $||(A \otimes B)x|| \leq (1 + 6/L)^2 + 1/L$.

□

6.4. **The replacement product.** If G is an n-vertex d-degree graph, we can give a number from 1 to d to each neighbor of each vertex and then the rotation map $\hat{G} : [n]\times[d] \mapsto [n]\times[d]$ maps a pair $\langle v, i\rangle$ to $\langle u, j\rangle$ where u is the i-th neighbor of v and v is the j-th neighbor of u. Using this rotation map, we define the replacement product.

Let G, G' be graphs such that G has n vertices and degree D, and G' has D vertices and degree d. Further, let A, A' denote the random-walk matrices of G and G' respectively, and $\hat{A}$ be the permutation matrix corresponding to the rotation map of G which means that $\hat{A}$ is an $nD \times nD$ matrix whose (i, j)th column is all zeroes except a single 1 in the (i', j') position where $(i', j') = \hat{G}(i, j)$. Then the replacement product of G and G', denoted $G \oslash G'$, is the graph with the random-walk matrix

$$A \oslash A' := 1/2\hat{A} + 1/2(I_n \otimes A')$$

where I_n is $n \times n$ 0-1 matrix with 1's only on the diagonal.

This means that $G \oslash G'$ has a copy of G' for every vertex in G and if (i, j) is an edge in G then there are d parallel edges between the i'-th vertex in the copy of G' corresponding to i and the j' vertex in the copy of G' corresponding to j where i' is the index of j as neighbor of i and j' is the index of i as neighbor of j in G. Therefore, $G \oslash G'$ has degree $2d$ and nD vertices.

Proposition 6.9. *(in PV_1) Let $d, D < L$. Suppose G is a D-degree graph with $n \in Log$ vertices and G' is a d-degree graph with D vertices. If $\lambda(G) \leq 1 - \epsilon \in Q/(Ln^2)$ and $\lambda(H) \leq 1 - \delta \in Q/(LD^2)$ for $n \in Log$, rational ϵ and rational $\delta \in [0, 1]$, then*

$$\lambda((G \oslash H)^3) \leq (1 - \epsilon\delta^2/8)(1 + 8/L^{1/2})^9 + \delta^2/(2L^{1/2}) + 2/L^{1/2}$$

In Proposition 6.9, Peano Arithmetic could prove $\lambda(G \oslash H) \leq 1 - \frac{\epsilon\delta^2}{24}$ following the argument in Arora-Barak [1]. In [1] this is derived using the equation $\lambda(G^l) = \lambda(G)^l$ which uses the existence of an orthogonal basis of eigenvectors for symmetric matrices. Again, in PV_1 we prove just a weaker bound for $(G \oslash H)^3$ (i.e. not for the product $G \oslash H$ but its power) which is sufficient for our purposes.

Proof. Let A resp. B be the random-walk matrix of graph G with n vertices resp. graph H with D vertices and $\hat{A}$ be the permutation matrix corresponding to the rotation map of G. By definition, $A \oslash B = \frac{1}{2}(\hat{A} + I_n \otimes B)$ and

$$(A \oslash B)^3 = \tfrac{1}{8}(\hat{A}^3 + \hat{A}(I \otimes B)\hat{A} + (I \otimes B)\hat{A}^2 + (I \otimes B)^2\hat{A} + \hat{A}^2(I \otimes B) + \\ + \hat{A}(I \otimes B)^2 + (I \otimes B)\hat{A}(I \otimes B) + (I \otimes B)^3)$$

By Proposition 6.6, $B = \delta J + (1-\delta)C$ for some C with $||C|| \leq 1$ and $D \times D$ all $1/D$ matrix J. Therefore,

$$(I \otimes B)\hat{A}(I \otimes B) = \delta^2(I \otimes J)\hat{A}(I \otimes J) + \delta(1-\delta)(I \otimes J)\hat{A}(I \otimes C) + \\ + \delta(1-\delta)(I \otimes C)\hat{A}(I \otimes J) + (1-\delta)^2(I \otimes C)\hat{A}(I \otimes C)$$

Since $||C|| \leq 1$ and $||I|| \leq 1$, for any x with $||x|| \leq 1$, we have $||(I \otimes C)x||^2 \leq (1+6/L)^4$ as in the proof of Proposition 6.8. Similarly, $||(I \otimes J)x||^2 \leq (1+6/L)^4$.

If a matrix A satisfies $||Ax||^2 \leq (1+6/L)^4$ for $||x|| \leq 1$, then for any B and x, $||(AB)x||^2 = ||A\frac{Bx}{||Bx||}||^2(SQRT(||Bx||^2))^2 \leq (1+\frac{6}{L})^4(SQRT(||Bx||^2))^2$. Consequently, $||(AB)x|| \leq (1+6/L)^2||Bx|| + 1/L^{1/2}$.

As $||\hat{A}|| \leq 1$, this shows that for any $x, ||x|| \leq 1$ and $\delta \in [0,1]$,

$$||((I \otimes B)\hat{A}(I \otimes B))x|| \leq \delta^2||((I \otimes J)\hat{A}(I \otimes J))x|| + (1-\delta^2)((1+\tfrac{6}{L})^8 + \\ + (1+\tfrac{6}{L})^4/L^{1/2} + (1+\tfrac{6}{L})^2/L^{1/2} + \tfrac{1}{L^{1/2}}) + \tfrac{3}{L^{1/2}}$$

Further, for any $x, ||x|| = 1$ and $\delta \in [0,1]$,

$$||(I \otimes B)x|| \leq \delta||(I \otimes J)x|| + (1-\delta)||(I \otimes C)x|| + 1/L^{1/2} \leq (1+6/L)^2 + 2/L^{1/2}$$

Hence, $||(I \otimes B)x||^2 \leq (1+8/L^{1/2})^4$, and using an analogous argument as above we can bound $||(A \oslash B)^3 x||$. For any $x, ||x|| = 1$,

$$||(A \oslash B)^3 x|| \leq (1-\tfrac{\delta^2}{8})(1+8/L^{1/2})^9 + \tfrac{\delta^2}{8}||((I \otimes J)\hat{A}(I \otimes J))x|| + 2/L^{1/2}$$

Observe that $(I \otimes J)\hat{A}(I \otimes J) = A \otimes J$ because $(I \otimes J)\hat{A}(I \otimes J)$ is the random-walk matrix of a graph with the number of edges between its nodes (i,j) and (i',j') being the number of k's in $[D]$ for which there is k' such that $\hat{G}(i,k) = (i',k')$. That is,

$((I \otimes J)\hat{A}(I \otimes J))_{(i,j),(i',j')} = \frac{1}{D}a_{i,i'} = (A \otimes J)_{(i,j),(i,j')}$

Then, by Proposition 6.8, for any $x, ||x|| = 1$ such that Σ_i, x_i (and so $Jx = 0$) we have:

$$||(I \otimes J)\hat{A}(I \otimes J)x|| = ||(A \otimes J)x|| \leq (1-\epsilon)((1+6/L)^2 + 1/L) + 3/L^{1/2}$$

which completes the proof. □

6.5. **The construction of the (n, d, λ)-graphs.** Finally, we are ready to construct the (n, d, λ)-graphs in the theory PV_1, see Arora-Barak [1, Chapter 21] for the history of the result. However, we will do it just for n's of the form c^k where c is a constant and $k \in LogLog$. It is possible to extend the construction to any n (cf. [1]) but at least a straightforward application of the extension requires algebraic techniques which we are avoiding. More specifically, it uses a converse of Proposition 6.3 which in turn uses facts about eigenvectors derived from the fundamental theorem of algebra. Nevertheless, the weaker construction is sufficient to derive the PCP theorem in PV_1.

Proposition 6.10. *For any rational $c \in (0, 1)$ there are d, b and L (the constant from the definition of $\lambda(G)$) such that PV_1 proves that for each $k \in LogLog$ and $n = (2d)^{100k}$ there is a $(2d)^b$-regular graph G_n with n vertices and $\lambda(G_n) < c$.*

Proof. For $c \in (0, 1)$, let e be such that $1/2^e < c$ and $b > e$ be a sufficiently big constant. Then, define $((2d)^{100k}, (2d)^b, 1/2^e)$-graphs in PV_1 as follows.

1. Let H be a $((2d)^{100}, d, 0.01)$-graph where d is a sufficiently big constant so that such a graph exists. Let G_1 be a $((2d)^{100}, (2d)^b, \frac{1}{2^b})$-graph and G_2 be a $((2d)^{200}, (2d)^b, \frac{1}{2^b})$-graph. These graphs can be found by brute force, cf. [1]. More precisely, as our H take the graph H from the proof of Theorem 21.19 in [1] and as our G_1, G_2 take G_1^b, G_2^b for G_1, G_2 from the same proof in [1].
2. For $(2d)^{100k}$ with $k > 2$, define $G_k := ((G_{\lfloor (k-1)/2 \rfloor} \otimes G_{\lceil (k-1)/2 \rceil}) \oslash H)^b$

Note that for given $(2d)^{100k}$, G_k is produced by a specific p-time computation which exists provably in PV_1.

Claim.: For every $(2d)^{100k}$, G_k is a $((2d)^{100k}, (2d)^b, 1/2^e)$-graph.

The claim is proved by $\Pi^b_1(PV)$-LPIND induction. As graphs G_k are constructed by a p-time function, the statement we want to obtain is $\forall\Sigma^b_1$. Hence, by $\forall\Sigma^b_1$-conservativity of S^1_2 over PV_1, we can work in the theory S^1_2 (which proves $\Pi^b_1(PV)$-LPIND).

For $k = 1, 2$, PV_1 can verify the claim directly. For $(2d)^{100k}$ with $k > 2$, let n_k be the number of vertices of G_k. If $n_{\lfloor (k-1)/2 \rfloor} = (2d)^{100\lfloor (k-1)/2 \rfloor}$ and $n_{\lceil (k-1)/2 \rceil} = (2d)^{100\lceil (k-1)/2 \rceil}$, then $n_k = n_{\lfloor (k-1)/2 \rfloor} n_{\lceil (k-1)/2 \rceil} (2d)^{100} = (2d)^{100k}$.

Considering the degree, if $G = G_{\lfloor (k-1)/2 \rfloor}$ has degree $(2d)^b$, then $(G \otimes G)$ has degree $(2d)^{2b}$, $(G \otimes G) \oslash H$ has degree $2d$ and G_k has degree $(2d)^b$.

The eigenvalue analysis: if $\lambda(G) \leq 1/2^e$ (which is a $\Pi^b_1(PV)$-formula), then assuming L is sufficiently big, $1/2^e \in Q/(Ln^2)$ and by Proposition 6.8 $\lambda(G \otimes G) \leq 2/2^e$. Hence, by Proposition 6.9,

$$\lambda(((G \otimes G) \oslash H)^3) \leq (1 - (1 - 2/2^e)\frac{(0.99)^2}{8})(1 + 8/L^{1/2})^9 + \frac{(0.99)^2}{2L^{1/2}} + 2/L^{1/2}$$

The conclusion $\lambda(((G \otimes G) \oslash H)^b) \leq 1/2^e$ is a consequence of the fact that the assumption $\lambda(G) \leq \lambda$ implies $\lambda(G^b) \leq \lambda^b(1 + 4/L) + 3d^b/L^{1/2}$ (where L is quantified after d, b so the term $3d^b/L^{1/2}$ can be made arbitrarily small). To see that the implication holds, note that similarly as in the proof of Proposition 6.4, $\lambda(G) \leq \lambda$ implies that for any $x \in Q^n/((Ln^3)^n n)$ with $\langle x, \mathbf{1} \rangle = 0$, we have $||A^b x|| \leq \lambda^b ||x||$ where $A^b \in Q^{n \times n}/d^b$ is the random-walk matrix of G^b. We need a similar bound even for $x \notin Q^n/((Ln^3)^n n)$. Fortunately, if $x \notin Q^n/((Ln^3)^n n), ||x|| = 1$, $\langle x, 1 \rangle = 0$, we can again approximate x by vector

$c \in Q^n/((Ln^3)^n n)$: for each i, $|x_i| \leq 1$ (otherwise $||x|| > 1$) so we can find $c_i \in Q/((Ln^3)^n n)$ such that $|x_i - c_i| \leq 1/(Ln^2)$ and $\langle c, 1\rangle = 0$. The values c_i are produced provably in PV_1 by a p-time algorithm which choses i_0 satisfying $x_{i_0} \geq 1/(Ln^2)$, then finds the smallest $c_i > x_i$ such that $c_i - x_i < 1/(Ln^3), c_i \in Q/(Ln^3), i \neq i_0$ and puts $c_{i_0} = \Sigma_{i \neq i_0} c_i \in Q/((Ln^3)^n n)$. The chosen c satisfies $||c||^2 \leq 1 + 3/(Ln)$ and $|(A^b x)_j - (A^b c)_j| \leq d^b/(Ln)$. Since also $(A^b x)_j, (A^b c)_j \leq 2d^b n$, we have $|||A^b x||^2 - ||A^b c||^2| \leq 5d^{2b}/L$ and

$$||A^b x||^2 \leq \lambda^{2b}(||c||^2 + 1/L) + 5d^{2b}/L \leq \lambda^{2b}(1 + 4/L) + 5d^{2b}/L$$

Thus, $||A^b x|| \leq \lambda^b(1 + 4/L) + 3d^b/L^{1/2}$.

□

Note that in the previous proposition, d does not depend on L and b can be chosen arbitrarily big.

7. The PCP theorem in PV_1

The PCP theorem obtained in Arora-Safra [2] and Arora et.al. [3] (see Arora-Barak [1, Chapter 22] for the history of the theorem) is a strengthening of the exponential PCP theorem in which the verifier D uses only $O(\log n)$ random bits. Using these random bits, D asks for at most $O(1)$ bits of the given proof π. Hence, π can be seen as a string of size $poly(n)$. In particular, it can be represented by a binary string in our formalization.

We will follow Dinur's [10] simplified proof of the PCP theorem as it is presented in Arora-Barak [1]. This will go rather smoothly (once we have a suitable formalization of the (n, d, λ)-graphs) because the proof is combinatorial and it needs to count only sets of polynomial size. These are subsets of $\{1, ..., poly(n)\}$ where $n \in Log$ for which we assume to have exact counting in PV_1 defined in a natural way.

Recall the verifier $D^{\pi,w}(x)$ from Definition 3.5. In the standard definition, π would be allowed to be a string of arbitrary length and D would have an oracular access to π, it could ask for any bit of π. Then, for a language L, $L \in PCP(\log n, 1)$ standardly means that there is a p-time algorithm D such that:

1. If $x \in L$, then there is a string π such that D with input x of length n and $O(\log n)$ random bits asks for at most $O(1)$ bits of π and accepts (with probability 1);
2. If $x \notin L$, then for any π, D with input x of length n and $O(\log n)$ random bits asks for at most $O(1)$ bits of π and accepts with probability $\leq 1/2$.

The PCP theorem says that $NP = PCP(\log n, 1)$. In our formalization, proofs π will be represented by p-size strings, hence, the statement of the PCP theorem is modified accordingly. As in the case of the exponential PCP theorem, we could alternatively represent proofs π by oracles which would maybe better reflect the nature of the PCP theorem but then we would need to formalize the PCP theorem in a theory extended by such oracles.

In this Section we use the notion of probability Pr on spaces of polynomial size $poly(n)$ which is assumed to be defined in a natural way using the exact counting of sets of polynomial size in PV_1. This should not be confused with the definition of Pr in APC_1.

First we formalize the easier implication of the PCP theorem: $PCP(\log n, 1) \subseteq NP$.

Theorem 7.1. *Let c, d, k be arbitrary constants, then PV_1 proves that for any kn^k-time algorithm D there exists $2kcn^{2kc}$-time algorithm M such that for each $x \in \{0,1\}^n$:*

$$\exists \pi \in \{0,1\}^{dn^c}\ \forall w < n^c, D^{\pi,w}(x) = 1 \to \exists y \in \{0,1\}^{dn^c}\ M(x,y) = 1$$
$$\forall \pi \in \{0,1\}^{dn^c}\ Pr_{w<n^c}[D^{\pi,w}(x) = 1] \leq 1/2 \to \forall y \in \{0,1\}^{dn^c}\ M(x,y) = 0$$

Proof. Given a kn^k-time algorithm D, define the algorithm M as follows. M accepts x, y if and only if $y = (y_0, ..., y_{n^c-1}) \in \{0,1\}^{dn^c}$ with y_i's in $\{0,1\}^d$ and for all the y_i's the algorithm D on input x, random bits i and with access to π which results in d bits y_i accepts.

Suppose there is $\pi \in \{0,1\}^{dn^c}$ such that for each $w < n^c$, D on input x with bits $r_w \in \{0,1\}^d$ obtained from d-times accessing π accepts. Then for $y = (y_0, ..., y_{n^c-1})$ with $y_w = r_w$ we have that for each $y_i \in y$ the algorithm D on input x and with access to π which results in d bits y_i accepts. Therefore, $M(x,y) = 1$.

Now assume that for any $\pi \in \{0,1\}^{dn^c}$, $Pr_{w<n^c}[D^{\pi,w}(x) = 1] \preceq_0 1/2$. Then for any $y = (y_0, ..., y_{n^c-1})$ with y_i's in $\{0,1\}^d$ there is y_i such that D on x, random bits i, and with access to π resulting in y_i rejects. Otherwise, for some $\pi \in \{0,1\}^{dn^c}$ we have $\{w < n^c | D^{\pi,w}(x) = 1\} = n^c$ contradicting the assumption. Hence, $M(x,y) = 0$. □

As the NP-completeness of SAT is provable in PV_1, the important implication of the PCP theorem, $NP \subseteq PCP(\log n, 1)$, can be stated in PV_1 as Theorem 7.

Theorem 7 (The PCP theorem in PV_1)**.** *There are constants d, k, c and a kn^k-time algorithm D (given as a PV-function) computing as in Definition 3.5 such that PV_1 proves that for any $n \in Log$ and $x \in \{0,1\}^n$, $n \in Log$:*

$$\exists y SAT(x,y) \to \exists \pi \in \{0,1\}^{dn^c}\ \forall w < n^c\ D^{\pi,w}(x) = 1$$
$$\forall y \neg SAT(x,y) \to \forall \pi \in \{0,1\}^{dn^c}\ Pr_{w<n^c}[D^{\pi,w}(x) = 1] \leq 1/2$$

The proof is summarized at the end of this section. It is a sequence of certain reductions between the so called CSP instances (CSP stands for constraint satisfaction problem) so we need to start with a reformulation of Theorem 7 in terms of these reductions.

Definition 7.1 (in PV_1)**.** *Let q, W be constants, and $n, m \in Log$. A $qCSP_W$ instance ϕ is a collection of circuits $\phi_1, ..., \phi_m$ (called constraints) mapping $[W]^n$ to $\{0,1\}$. Each ϕ_i is encoded by a binary string, it has n inputs which are taking values that are bit strings in $\{0,1\}^{\log W}$ but depends on at most q of them: for every $i \in [m]$ there exist $f_1, ..., f_q \in [n]$ and $f : \{0,1\}^q \mapsto \{0,1\}$ such that $\phi_i(u) = f(u_{f_1}, ..., u_{f_q})$ for every $u \in [W]^n$. We say that q is the arity of ϕ. By $qCSP$ instance we mean a $qCSP$ instance with binary alphabet.*

An assignment $u \in [W]^n$ satisfies ϕ_i if $\phi_i(u) = 1$, and instance ϕ is satisfiable if $val(\phi) := max_{u \in [W]^n} \frac{\Sigma_{i=1}^m \phi_i(u)}{m} = 1$. (We will not need to prove the totality of the function $val(\phi)$ in PV_1. It will be sufficient for us to work with formulas of the form $val(\phi) \leq y$ which are Π_1^b.)

The $qCSP$ instances generalize 3SAT. Another NP-complete problem known as "Graph 3-colorability" can be also seen as a $2CSP_3$ instance where for each edge (i,j) there is a constraint on the variables u_i, u_j that is satisfied if and only if $u_i \neq u_j$. The graph is 3-colorable if and only if there is a way to assign a number in $\{0,1,2\}$ to each variable such that all constraints are satisfied, cf. [1].

Definition 7.2 (in PV_1)**.** *Let q, q', W, W' be arbitrary constants. A p-time function f (given as a PV-function) mapping $qCSP_W$ instances to $q'CSP_{W'}$ instances, abbreviated as $f : qCSP_W \to q'CSP_{W'}$, is a CL-reduction (short for complete linear-blowup reduction) if for every $qCSP_W$ instance ϕ:*

- *Completeness: If ϕ is satisfiable then so is $f(\phi)$.*
- *Linear blowup: If there are m constraints in ϕ, then $f(\phi)$ has at most Cm constraints and alphabet W', where C can depend on q (but not on m or the number of variables in ϕ).*

For a constant k, a function f is CL^k-reduction if it is a CL-reduction computable in time kn^k.

Theorem 7 then follows from the following proposition.

Proposition 7.1. *There are constants $q_0 \geq 3, \epsilon_0 > 0$ and a CL-reduction $f : q_0CSP \to q_0CSP$ such that PV_1 proves that for every q_0CSP instance ϕ, every $\epsilon < \epsilon_0$,*

$$val(\phi) \leq 1 - \epsilon \to val(f(\phi)) \leq 1 - 2\epsilon$$

Proof. (of Theorem 7 from Proposition 7.1) The statement we want to derive is a $\forall\Sigma^b_1$-formula. Hence, we can work in the theory S^1_2. As $q_0 \geq 3$, q_0CSP is a generalization of 3SAT and by the NP-completeness of 3SAT (derived similarly as the NP-completeness of SAT), for some k', there is a $k'n^{k'}$-time function h mapping propositional formulas to q_0CSP instances such that for every $n \in Log$ and $x \in \{0,1\}^n$, $\exists y SAT(x, y) \to val(h(x)) = 1$ and $\forall y \neg SAT(x, y) \to val(h(x)) \leq 1 - 1/m$ where $m \in Log$ is the number of constraints in $h(x)$. Applying Proposition 7.1 we obtain a kn^k-time function $f^{\log m} \circ h$ for some constant k such that

$$\exists y SAT(x, y) \to val(f^{\log m} \circ h(x)) = 1$$
$$\forall y \neg SAT(x, y) \to val(f^{\log m} \circ h(x)) \leq 1 - \epsilon_0$$

Here, we used Π^b_1-LLIND (available in S^1_2) for Π^b_1-formulas $val(f^i(\phi)) \leq 1 - 2^i\epsilon$ where $i \leq |m|$. Therefore, for some constants d', c', and an algorithm D' which given any formula x and proof π accepts if and only if π encodes a satisfying assignment to randomly chosen constraint in $f^{\log m} \circ h(x)$ we have:

$$\exists y SAT(x, y) \to \exists \pi \in \{0,1\}^{d'n^{c'}} \ \forall w D'^{\pi,w}(x) = 1$$
$$\forall y \neg SAT(x, y) \to \forall \pi \in \{0,1\}^{d'n^{c'}} \ Pr_w[D'^{\pi,w}(x) = 1] \leq 1 - \epsilon_0$$

The gap can be amplified to $1/2$ by choosing sufficiently many (but constant number of) constraints in $f^{\log m} \circ h(x)$ and accepting if and only if π encodes satisfying assignments to all of them. This requires Chernoff's bound but only over sets of polynomial size for which we have exact counting in PV_1. □

Proposition 7.1 is an immediate consequence of the following two statements. The first one provides us a CL-reduction producing CSP instances which increase the gap between 0 and the minimal number of unsatisfied constraints. However, the alphabet of the resulting instances increases too. The second statement takes it back to binary while losing just a factor of 3 in the gap.

Proposition 7.2 (Gap amplification in PV_1)**.** *For every l, q there are W, ϵ_0 and a CL-reduction $g_{l,q} : qCSP \to 2CSP_W$ such that PV_1 proves that for every $qCSP$ instance ϕ and for every $\epsilon < \epsilon_0$*

$$val(\phi) \leq 1 - \epsilon \to val(g_{l,q}(\phi)) \leq 1 - l\epsilon$$

Proposition 7.3 (Alphabet reduction in PV_1)**.** *There is d such that for any W there is a CL-reduction $h : 2CSP_W \to dCSP$ such that PV_1 proves that for every $2CSP_W$ instance ϕ, and for each ϵ*

$$val(\phi) \leq 1 - \epsilon \to val(h(\phi)) \leq 1 - \epsilon/3$$

Proposition 7.1 can be obtained from previous two propositions by taking $l = 6$ in Proposition 7.2 and $q = max\{d, 3\}$ for d from Proposition 7.3.

We firstly derive Proposition 7.3 using the following application of the exponential PCP theorem which is scaled down so that we need to reason only about sets of constant size.

Proposition 7.4. *There are constants d, k' and an algorithm D such that for every s, PV_1 proves: given any s-size circuit C with $2n_1$ inputs, D runs in time $s^{k'}$, examines $\leq d$ bits in the provided strings and*

1. *If $C(u_1, u_2) = 1$ for $u_1, u_2 \in \{0,1\}^{n_1}$, there is a string π_3 of size $2^{s^{k'}}$ such that $\forall w < 2^{s^{k'}}$ $D^{(WH(u_1),WH(u_2),\pi_3),w}(C) = 1$.*
2. *For bit strings π_1, π_2, π_3 where $\pi_1, \pi_2 \in \{0,1\}^{2^{n_1}}$, $\pi_3 \in \{0,1\}^{2^{s^{k'}}}$*
   if $Pr_{w<2^{s^{k'}}}[D^{(\pi_1,\pi_2,\pi_3),w}(C) = 1] \geq 1/2$, then
   $Pr_{w<2^{n_1}}[(\pi_1)_w = WH(u_1)(w)] \geq 0.99$ and
   $Pr_{w<2^{n_1}}[(\pi_2)_w = WH(u_2)(w)] \geq 0.99$
   for some $u_1, u_2 \in \{0,1\}^{n_1}$ such that $C(u_1, u_2) = 1$.
   (Note that all the probabilities are defined on sets of constant size, i.e. of size which is quantified outside of the theory PV_1.)

Proof. (of Proposition 7.3 from Proposition 7.4) The CL-reduction h works as follows. Let ϕ be a $2CSP_W$ instance with constraints $\phi_1, \phi_2, ..., \phi_m$ on variables $u_1, ..., u_n$ which are taking values that are in $\{0,1\}^{\log W}$. Each constraint $\phi_S(u_i, u_j)$ is a circuit applied to the bit strings representing u_i, u_j. Without loss of generality $s \leq 2^{4 \log W}$ is an upper bound on the size of this circuit.

Given such ϕ, h replaces each variable u_i by a sequence $U_i = (U_{i,1}, ..., U_{i,W})$ of W binary variables (U_i is long enough to represent $WH(u_i)$). Then, for each constraint $\phi_S(u_i, u_j)$ it applies Proposition 7.4 where $\phi_S(u_i, u_j)$ is the circuit whose assignment is being verified. The resulting $s^{k'}$-time algorithm D can be represented as a $2^{s^{O(1)}}$-size $dCSP$ instance $\psi_S(U_i, U_j, \Pi_S)$ where U_i, U_j play the role of π_1, π_2 and $2^{s^{k'}}$ new binary variables Π_S play the role of π_3. The arity d of $\psi_S(U_i, U_j, \Pi_S)$ is the number of bits D reads in the proof which is a fixed constant independent of W and ϵ. The instance $\psi_S(U_i, U_j, \Pi_S)$ contains one constraint for each possible random string in D, so the fraction of its satisfied constraints is the acceptance probability of D. The CL-reduction h thus maps $2CSP_W$ instances ϕ to $dCSP$ instances ψ where each $\phi_S(u_i, u_j)$ is replaced by a $dCSP$ instance $\psi_S(U_i, U_j, \Pi_S)$. As $2^{s^{O(1)}}$ is a constant independent of m and n, linear blowup is preserved.

If ϕ is satisfiable, then by property 1 in Proposition 7.4 so is ψ. We want to show that if some assignment satisfies more than $1 - \epsilon/3$ fraction of the constraints in ψ, then we can

construct an assignment for ϕ satisfying more then $1-\epsilon$ fraction of its constraints: For each i, if U_i is 0.99-close to some linear function $WH(a_i)$, i.e. $Pr_x[U_{i,x} = WH(a_i)(x)] \geq 0.99$, then use (the determined) a_i as the assignment for u_i, and otherwise use arbitrary string. The algorithm is p-time because the size of each U_i is constant. If the decodings a_i, a_j of U_i, U_j do not satisfy $\phi_S(u_i, u_j)$, then by property 2 in Proposition 7.4 at least half of constraints in ψ_S is not satisfied. Hence, the fraction of unsatisfied constraints in ϕ is $< 2\epsilon/3$.

□

Proof. (of Proposition 7.4) PV_1 can prove the statement from Proposition 7.4 simply by examining all possible cases of which there is a constant number. Hence, the provability of the statement follows from it being true. Nevertheless, we present also the standard proof itself.

The algorithm D firstly reduces the problem of satisfiability of the given circuit C with s wires (inputs are considered as wires in the circuit) to the question of solvability of a set of quadratic equations with $t = s^{O(1)}$ variables similarly as in the proof of the exponential PCP theorem. D expects π_3 to contain linear functions f, g which are $WH(z)$ and $WH(z \otimes z)$ respectively for $z \in \{0,1\}^t$ satisfying the set of quadratic equations and checks these functions as in the exponential PCP theorem. Moreover, D checks that π_1 and π_2 are 0.99-close to some linear functions. That is, if D accepts π_1, π_2, π_3 with probability $\geq 1/2$, it is because the set of quadratic equations is satisfiable and $Pr_w[(\pi_1)_w = WH(u_1)(w)] \geq 0.99$, $Pr_w[(\pi_2)_w = WH(u_2)(w)] \geq 0.99$ for some $u_1, u_2 \in \{0,1\}^{n_1}$.

Finally, D checks that π_1, π_2 encode strings whose concatenation is the same as the first $2n_1$ bits of the string encoded by f (without loss of generality the first $2n_1$ bits encode satisfying assignement for C) by performing the following concatenation test:

Pick random $x, y \in \{0,1\}^{n_1}$ and denote by $XY \in \{0,1\}^t$ the string whose first n_1 bits are x, the next n_1 bits are y and the remaining bits are all 0. Accept if and only if $f(XY) = \pi_1(x) + \pi_2(y)$.

The algorithm D runs in time $s^{k'}$ and examines $\leq d$ bits in π_1, π_2, π_3 for some constants k', d. It satisfies the first property from Proposition 7.4. Moreover, assuming that $\pi_1 = WH(u), \pi_2 = WH(v)$ and z is the string encoded by a linear function f, the concatenation test rejects with probability $1/2$ if u, v differs from the first $2n_1$ bits of z. Hence, if D accepts π_1, π_2, π_2 with probability $\geq 1/2$, it is because π_1, π_2 are 0.99-close to linear functions encoding u_1, u_2 such that $C(u_1, u_2) = 1$. □

In the rest of this section we derive Proposition 7.2. To do this, we will need two facts about probability:

Proposition 7.5. *1. Let t be a square and S_t be the binomial distribution over t fair coins, i.e. $Pr[S_t = k] = t!/((t-k)!k!)2^{-t}$. Then for $i \in \{0,1\}$ and any δ such that $0 \leq \delta < 1$, PV_1 proves:*

$$\Sigma_k |Pr[S_t = k] - Pr[S_{t+(-1)^i \lfloor \delta\sqrt{t} \rfloor} = k]| \leq 20\delta$$

2. For any k, PV_1 proves that for each $n \in Log$, if V is a nonnegative random variable defined on a sample space of size n^k, then $Pr[V > 0] \geq E[V]^2/E[V^2]$.

The first part of Proposition 7.5 is an estimation of a so called statistical distance of two binomial distributions which is known to hold (see [1] page 469) and as all its parameters are quantified outside of the theory PV_1, it is trivially provable by an explicit "brute force"

enumeration. That is, to prove the inequality PV_1 just sums up finitely many rational numbers and compares the result to 20δ.

The second part is obtained from a simple expansion:

$$(E[X])^2 = (E[X \cdot 1_{X>0}])^2 \leq E[X^2]E[(1_{X>0})^2] = E[X^2]Pr[X > 0]$$

where we used a form of Cauchy-Schwarz inequality $E[XY]^2 \leq E[X^2]E[Y^2]$ which can be derived in the same way as our Cauchy-Schwarz inequality from Section 6 but with $\langle x, y\rangle := E[XY]$.

The proof of Proposition 7.2 is divided into two parts. The first part shows how to reduce any $qCSP$ instance into a $2CSP_W$ instance which is nice (in a sense defined below) and the second part gives us a CL-reduction from nice instances which amplifies the gap as it is required in Proposition 7.2.

Definition 7.3. *(in PV_1)*

1. Let ϕ be a $2CSP_W$ instance mapping $[W]^n$ to $\{0,1\}$. The constraint graph of ϕ is the graph G with vertex set $[n]$ where for every constraint ϕ depending on the variables u_i, u_j, the graph G has the edge (i,j). G is allowed to have parallel edges and self-loops. Then G is d-regular for some constant d independent of W, and at every node, at least half the edges incident to it are self-loops.

2. A $qCSP_W$ instance ϕ is nice if $q = 2$ and the constraint graph of ϕ denoted G satisfies $\lambda(G) \leq 0.9$

The reduction into nice instances which we need is a consequence of the following three Propositions.

Proposition 7.6. *There is a constant k such that for every q there is a CL^k-reduction $h : qCSP \to 2CSP_{2^q}$ such that PV_1 proves that for any $qCSP$ instance ϕ and any ϵ*

$$val(\phi) \leq 1 - \epsilon \to val(h(\phi)) \leq 1 - \epsilon/q$$

Proof. The CL^k reduction works as follows. Given $qCSP$ instance ϕ over n variables $u_1, ..., u_n$ with m constraints, it produces $2CSP_{2^q}$ instance ψ over the variables $u_1, ..., u_n$, $y_1, ..., y_m$ such that for each ϕ_i in ϕ depending on the variables $u_{f_1}, ..., u_{f_q}$, ψ contains q constraints $\psi_{i,j}, j = 1, ..., q$ where $\psi_{i,j}(y_i, u_{f_j})$ is true iff y_i encodes an assignment to $u_{f_1}, ..., u_{f_q}$ satisfying ϕ_i and $u_{f_j} \in \{0,1\}$ agrees with the assignment y_i.

The number of constraints in ψ is qm and if ψ is satisfiable, then so is ψ. Suppose that $val(\phi) \leq 1 - \epsilon$ and let $u_1, ..., u_n, y_1, ..., y_m$ be any assignment to ψ. By the assumtion, there is a set $S \subseteq [m]$ of size $\geq \epsilon m$ such that all constraints $\phi_i, i \in S$ are violated by $u_1, ..., u_n$. Then, for any $i \in S$ there is $j \in [q]$ such that $\psi_{i,j}$ is violated. □

Proposition 7.7. *There are constants d, e, k such that for every W there is a CL^k-reduction $h : 2CSP_W \to 2CSP_W$ such that PV_1 proves that for any $2CSP_W$ instance ϕ, and any ϵ*

$$val(\phi) \leq 1 - \epsilon \to val(h(\phi)) \leq 1 - \epsilon/(100Wed)$$

and the constraint graph of $h(\phi)$ is d-regular.

Proof. By Proposition 6.10 and Proposition 6.3 there are constants d, e such that for each $e^t, t \in LogLog$, there is a d-regular graph G_{e^t} which for any $S \subseteq V, |V| = e^t, |S| \leq e^t/2$ satisfies $|E(S, V - S)| \geq d|S|/4 - 1/8$. In particular, for each W and $S \subseteq V$, $|S| \leq e^t/2$, we have $(*)$: $|E(S, V - S)| \geq |S|/(10W)$.

The CL^k-reduction h works as follows.

Let ϕ be a $2CSP_W$ instance. First, erase variables in ϕ that do not appear in any constraint. Suppose next that u_l is a variable that appears in $c' \geq 1$ constraints. Put $c := e^t$ for the smallest natural t such that $c' \leq e^t$. Replace u_l by c variables $y_l^1, ..., y_l^c$ so that in each constraint u_l originally appeared in we have different y_l^f (different c's might be needed for each u_l). Add a constraint requiring that $y_l^j \leftrightarrow y_l^{j'}$ for every edge (j, j') in the graph G_c. Do this for every variable in ϕ until each variable appears in $d+1$ constraints, d equality constraints and one original constraint resp. a null constraint that always accepts which is added if necessary. Denote the resulting $2CSP_W$ instance as ψ $(= h(\phi))$.

If ϕ has m constraints, ψ has $\leq m + 2dem + 2em$ constraints (m original constraints, $\leq 2em$ null constraints and $\leq 2dem$ "$y_l^j \leftrightarrow y_l^{j'}$" constraints). If ϕ is satisfiable, then so is ψ. Suppose that $val(\phi) \leq 1 - \epsilon$ and let y be any assignment to ψ. Consider then the plurality assignment u to ϕ's variables: u_i gets the most likely value that is claimed for it by $y_i^1, ..., y_i^c$. Define t_i to be the number of y_i^j's that disagree with the plurality value of u_i.

If $\Sigma_{i=1}^n t_i \geq \epsilon m/2$, then by $(*)$ there are $\geq \epsilon m/(20W)$ equality constraints violated in ψ.

Suppose that $\Sigma_{i=1}^n t_i < \epsilon m/2$. Since $val(\phi) \leq 1 - \epsilon$, there are $\geq \epsilon m$ constraints in ϕ violated by u. All of these constraints are also present in ψ. If more than $\epsilon m/2$ of them were assigned a different value by y than by u, then $\Sigma_i^n t_i \geq \epsilon m/2$. Thus y violates $\geq \epsilon m/2$ constraints in ψ.

Note that all the sets we counted had polynomial size so we had exact counting for them in PV_1.

□

Proposition 7.8. *There are constants d, e, k such that for any d', W there is a CL^k-reduction $h : 2CSP_W \to 2CSP_W$ such that PV_1 proves that for any $2CSP_W$ instance ϕ with d'-regular constraint graph for $d \geq d'$ and for any ϵ,*

$$val(\phi) \leq 1 - \epsilon \to val(h(\phi)) \leq 1 - \epsilon/(10de)$$

Moreover, the constraint graph G of $h(\phi)$ is $4d$-regular with at least half the edges coming out of each vertex being self-loops and $\lambda(G) \leq 0.9$.

Proof. By Proposition 6.10 there are constants d, e such that for each e^t where $t \in LogLog$, there is a d-regular graph G_{e^t} in PV_1 with $\lambda(G_{e^t}) \leq 0.1$. The CL^k-reduction h works as follows.

Let ϕ be a $2CSP_W$-instance with n variables, m constraints, and d'-regular constraint graph G' for $d' \leq d$. Without loss of generality $2m \geq n$. Otherwise, ϕ contains variables that are not in any constraint so $d' = 0$ and ϕ is empty. Add new vertices and self-loops to G' so that it becomes d-regular with e^t vertices for the smallest $e^t \geq n$. For each of these new vertices add new variables and for the new self-loops add null constraints that always accept. Then add null constraints for every edge in the graph G_{e^t}. Finally, add $2d$ null constraints forming self-loops for each vertex in G_{e^t}.

The resulting instance $\psi(=h(\phi))$ has $4d$-regular constraint graph with $\leq 2den$ constraints, and at least half the edges coming out of each vertex being self-loops. Assuming $val(\phi) < 1 - \epsilon$, there are $\geq \epsilon m \geq \epsilon 2den/(4de)$ violated constraints in ψ.

Let G be ψ's constraint graph and A its random-walk matrix. Then $A = 3/4B + C/4$ for C the random-walk matrix of G_{e^t} and B the random walk matrix of a $3d$-regular graph. In Section 6.3, we observed that for any $x \in Q^n/m$, $||Ax|| \leq 3/4||Bx|| + 1/4||Cx|| + 1/L^{1/2}$

and by Proposition 6.2, for any $\delta > 0$, $\lambda(B) \leq 1 + \delta + 1/L$. Thus, assuming δ is sufficiently small and L sufficiently big, $\lambda(G) \leq 3/4(1 + \delta + 1/L^{1/2}) + 1/4\lambda(G_{e^t}) + 1/L \leq 0.9$.

□

Note that the constant d from Proposition 7.8 can be chosen so that it is bigger than the constant d from Proposition 7.7. Therefore, Propositions 7.6, 7.7 and 7.8 show that there are constants d, e, k such that for any q (and $W = 2^q$) there is a CL^k-reduction $h : qCSP \to 2CSP_{2^q}$ such that PV_1 proves that h maps any $qCSP$ instance into an instance which is nice with the constraint graph being d-regular while the fraction of violated constraints is reduced by a factor at most $1/(1000We^2d^2q)$. This shows that to derive Proposition 7.2 it suffices to prove the following powering proposition:

Proposition 7.9. *There is k such that for any $W > 0$ and sufficiently big square $t \geq 1$ there is an algorithm A with properties described below such that PV_1 proves that for any nice $2CSP_W$ instance ψ with n variables with $n \in Log$ the algorithm A produces a $2CSP_{W'}$ instance ψ^t such that:*

1. $W' \leq W^{d^{5t}}$, where d is the degree of ψ's constraint graph. The instance ψ^t has $\leq d^{5t}n$ constraints.

2. If ψ is satisfiable, then so is ψ^t.

3. For every $\epsilon < 1/(d\sqrt{t})$,

$$val(\psi) \leq 1 - \epsilon \to val(\psi^t) \leq 1 - \epsilon\sqrt{t}/(10^6 dW^5)$$

4. The formula ψ^t is produced from ψ (by A) in time $(nd)^k W^{kd^{5t}}$.

Proof. (It might be helpful to the reader to consult the proof we present here in conjunction with the exposition from [1, Lemma 22.9] where some concepts are explained with additional details.)

Let ψ be a $2CSP_W$ instance with n variables $u_1, ..., u_n$ and $m \leq nd/2$ constraints and let G denote the constraint graph of ψ.

The formula ψ^t will have n variables $y_1, ..., y_n$ over an alphabet of size $W' = W^{d^{5t}}$. A value of a variable y_i is a d^{5t}-tuple of values in $\{0, ..., W-1\}$ and we will think of it as giving a value $y_i(u_j)$ in $\{0, ..., W-1\}$ to every variable u_j in ψ where j can be reached from i using a path of $\leq t + \sqrt{t}$ steps in G. Since G is d-regular the number of such nodes is $\leq d^{t+\sqrt{t}+1} \leq d^{5t}$.

For every path $p = \langle i_1, ..., i_{2t+2} \rangle$ in G we will have a constraint C_p in ψ^t depending on variables y_{i_1} and $y_{i_{2t+2}}$ which outputs 0 if and only if there is some $j \in [2t+1]$ such that

1. i_j can be reached from i_1 using a path of $\leq t + \sqrt{t}$ steps in G
2. i_{j+1} can be reached from i_{2t+2} using a path of $\leq t + \sqrt{t}$ steps n G
3. $y_{i_1}(u_{i_j}), y_{i_{2t+2}}(u_{i_{j+1}})$ violate the constraint in ψ depending on u_{i_j} and $u_{i_{j+1}}$

The $2CSP_{W'}$ instance ψ^t can be produced in time $(nd)^k W^{kd^{5t}}$ and has $\leq d^{5t}n$ constraints. Any assignment $u_1, ..., u_n$ satisfying ψ induces an assignment $y_1, ..., y_n$ satisfying ψ^t: each y_i encodes values u_j for j's that can be reached from i by $\leq t+\sqrt{t}$ steps in G. Therefore, it remains to show that for $\epsilon < 1/(d\sqrt{t})$, $val(\psi) \leq 1-\epsilon \to val(\psi^t) \leq 1-\epsilon\sqrt{t}/(10^6 dW^5)$.

Every assignment y for ψ^t induces the so called plurality assignment u for ψ: u_i gets the value $\sigma y(u_i)$ which is the most likely value $y_k(u_i)$ for y_k's where k is obtained by taking a t-step random walk from i in G. If more than one value is most likely, take the lexicographically first one.

Suppose that $val(\psi) \leq 1 - \epsilon$, then there is a set F of ϵm constraints violated by the plurality assignment.

Pick a random path $p = \langle i_1, ..., i_{2t+2} \rangle$ in G. For $j \in \{1, ..., 2t+1\}$ we say that the edge (i_j, i_{j+1}) in p is truthful if $y_{i_1}(u_{i_j}) = \sigma y(u_{i_j})$ and $y_{i_{2t+2}}(u_{i_{j+1}}) = \sigma y(u_{i_{j+1}})$. Let $\delta = 1/(1000W)$ and denote by V the number of edges in $\left\langle i_t, ..., i_{t+\lfloor \delta\sqrt{t} \rfloor + 1} \right\rangle$ that are truthful and in F. That is, V is a nonnegative random variable defined on a sample space of size $poly(n)$. If there is at least one such edge, the corresponding constraint in ψ^t is unsatisfied so we want to show that $Pr_p[V > 0] \geq \epsilon\sqrt{t}/(10^6 dW^5)$.

For each edge e of G and each $j \in \{1, 2, ..., 2t+1\}$, $Pr_p[e = (i_j, i_{j+1})] = 1/m$, i.e. each edge has the same probability to be the j-th edge in p.

Claim.: For any edge e of G and any $j \in \{t, ..., t + \lfloor \delta\sqrt{t} \rfloor\}$,

$$Pr_p[(i_j, i_{j+1}) \text{ is truthful} \mid e = (i_j, i_{j+1})] \geq 1/(2W^2)$$

To prove the claim, let i_1 be the endpoint of a random walk p_1 of length j out of i_j and i_{2t+2} be the endpoint of a random walk p_2 of length $2t - j$ out of i_{j+1}. We need to show that

$$Pr_{p_1}[y_{i_1}(u_{i_j}) = \sigma y(u_{i_j})] Pr_{p_2}[y_{i_{2t+2}}(u_{i_{j+1}}) = \sigma y(u_{i_{j+1}})] \geq 1/(2W^2)$$

Since half of the edges incident to each vertex are self-loops, we can see an l-step random walk from a vertex i as follows:

1. throw l fair coins and let S_l denote the number of "heads";
2. take S_l non-self-loop steps on the graph.

Denote by $l(p)$ the length of a path p not counting self-loops. Then,

$$\begin{aligned}
&Pr_{p_1}[y_{i_1}(u_{i_j}) = \sigma y(u_{i_j})] = \\
&= \Sigma_l Pr[S_j = l] Pr_{p_1}[l(p_1) = l \wedge y_{i_1}(u_{i_j}) = \sigma y(u_{i_j})] \\
&\geq \Sigma_l Pr[S_t = l] Pr_{p_1}[l(p_1) = l \wedge y_{i_1}(u_{i_j}) = \sigma y(u_{i_j})] - 20\delta \quad \textit{by Proposition } 7.5 \\
&\geq 1/W - 20\delta
\end{aligned}$$

where the last inequality follows from the definition of the plurality assignment which implies that for $j = t$, $Pr_{p_1}[y_{i_1}(u_{i_j}) = \sigma y(u_{i_j})] \geq 1/W$. Similarly we obtain $Pr_{p_2}[y_{i_{2t+2}}(u_{i_{j+1}}) = \sigma y(u_{i_{j+1}})] \geq (1/W - 20\delta)$. This proves our claim.

The claim implies $Pr_p[(i_j, i_{j+1}) \text{ is truthful and in } F] \geq |F|/(m2W^2)$ for any j from $\{t, ..., t + \lfloor \delta\sqrt{t} \rfloor\}$. Without a loss of generality, $|\{t, ..., t + \lfloor \delta\sqrt{t} \rfloor\}|$ is $\lceil \delta\sqrt{t} \rceil$. Thus by linearity of expectation,

$$E[V] \geq \epsilon \lceil \delta\sqrt{t} \rceil / (2W^2)$$

By Proposition 7.5 2., $Pr[V > 0] \geq E[V]^2/E[V^2]$, so to conclude the proof it suffices to show that $E[V^2] \leq 50d\epsilon\lceil \delta\sqrt{t} \rceil$.

Denote by V' the number of edges in $\left\langle i_t, ..., i_{t+\lfloor \delta\sqrt{t} \rfloor + 1} \right\rangle$ that are in F. For any j from $\{t, ..., t + \lfloor \delta\sqrt{t} \rfloor\}$ put $I_j := 1$ iff $(i_j, i_{j+1}) \in F$. Further, let S be the set of vertices contained

in an edge from F. Then, assuming that the constant L from our definition of $\lambda(G)$ satisfies $L > d$ and $L > \delta\sqrt{t}$,

$$\begin{aligned}
&E[V^2] \leq E[V'^2] = E[\Sigma_{j,j'} I_j I_{j'}] = E[\Sigma_j I_j^2] + E[\Sigma_{j \neq j'} I_j I_{j'}] \\
&= \epsilon\lceil\delta\sqrt{t}\rceil + 2\Sigma_{j<j'} Pr_p[(i_j, i_{j+1}) \in F \wedge (i_{j'}, i_{j'+1}) \in F] \\
&\leq \epsilon\lceil\delta\sqrt{t}\rceil + 2\Sigma_{j<j'} Pr_{(i_j,i_{j'})\in G^{j'-j}}[i_j \in S \wedge i_{j'} \in S] \\
&\leq \epsilon\lceil\delta\sqrt{t}\rceil + 2\Sigma_{j<j'} \epsilon d(\epsilon d + 2 \cdot 0.9^{j'-j}) && \textit{by Proposition } 6.4 \\
&\leq \epsilon\lceil\delta\sqrt{t}\rceil + 2\epsilon^2 d^2 \lceil\delta\sqrt{t}\rceil^2 + 40\epsilon d\lceil\delta\sqrt{t}\rceil \leq 50\epsilon d\lceil\delta\sqrt{t}\rceil && \textit{using } \epsilon < 1/(d\sqrt{t})
\end{aligned}$$

□

This concludes our formalization of the PCP theorem in the theory PV_1. It can be briefly summarized as follows. In Theorem 7 we formulated the PCP theorem as a $\forall\Sigma^b_1$-formula. Thus, by $\forall\Sigma^b_1$-conservativity of S^1_2 over PV_1 we could afford to work instead in the theory S^1_2. Specifically, we used Π^b_1-LLIND induction available in S^1_2 to show that the PCP theorem is a consequence of a statement about CSP instances, Proposition 7.1. Then we observed that the CSP formulation of the PCP theorem is a collorary of two propositions, Gap amplification 7.2 and Alphabet reduction 7.3. The latter one was an application of the exponential PCP theorem in a scaled-down setting where we needed to count only sets of constant size, hence it was provable already in PV_1. The gap amplification was a consequence of a CL-reduction into nice CSP instances and Powering proposition 7.9. The reduction to nice instances used the (n, d, λ)-graphs which we constructed in Section 6. Section 6 contained the most challenging part where we needed to employ certain approximating tools to reason about algebraic definitions of pseudorandom constructions in PV_1. In the remaining part of the proof of the PCP theorem, including the powering proposition, we were mainly verifying step by step that the reasoning used in the standard proof does not exceed the possibilities of the theory PV_1.

8. Acknowledgement

I would like to thank Jan Krajíček for many constructive discussions during the development of the paper, Sam Buss for detailed comments and suggestions which improved the quality of the manuscript and an anonymous reviewer for further useful remarks. I would also like to thank Neil Thapen, Pavel Pudlák and Emil Jeřábek for comments and suggestions during its seminar presentation. This research was supported by grants GA UK 5732/2014 and SVV-2014-260107.

References

[1] Arora S., Barak B.; Computational Complexity: A Modern Approach, Cambridge University Press, 2009.
[2] Arora S., Safra S.; Probabilistic checking of proofs: A new characterization of NP, J. ACM, 45(1):70-122, 1998. Preliminary version FOCS 1992.
[3] Arora S., Lund C., Motwani R., Sudan M., Szegedy M.; Proof verification and the hardness of approximation problems, J. ACM, 45(3):501-555, 1998. Preliminary version FOCS 1992.
[4] Buss S.R.; Bounded Arithmetic, Bibliopolis, Naples, 1986.
[5] Buss S.R., Kołodziejczyk L.A., Zdanowski K.; Collapsing Modular Counting in Bounded Arithmetic and Constant Depth Propositional Proofs, To appear in Transactions of the AMS.
[6] Cai J.; $S^P_2 \subseteq ZPP^{NP}$, Journal of Computer and System Sciences, 73(1):25-35, 2007.

[7] Cobham A.; The intrinsic computational difficulty of functions, Proceedings of the 2nd International Congress of Logic, Methodology and Philosophy of Science, North Holland, pp. 24-30, 1965.
[8] Cook S.A.; Feasibly constructive proofs and the propositional calculus, Proceedings of the 7th Annual ACM Symposium on Theory of Computing, ACM Press, pp. 83-97, 1975.
[9] Cook S.A., Krajíček J.; Consequences of the Provability of NP⊆P/poly, Journal of Symbolic Logic, 72:1353-1357, 2007.
[10] Dinur I.; The PCP theorem by gap amplification, J. ACM, 54(3), 2007.
[11] Dai Tri Man Le; Bounded arithmetic and formalizing probabilistic proofs, Ph.D. thesis, University of Toronto, 2014.
[12] Imagliazzo R., Wigderson A.; P=BPP unless E has subexponential circuits: Derandomizing the XOR Lemma, Proceedings of the 29th Annual ACM Symposium on Theory of Computing, pp. 220-229, 1997.
[13] Jeřábek E.; Dual weak pigeonhole principle, Boolean complexity and derandomization, Annals of Pure and Applied Logic, 129:1-37, 2004.
[14] Jeřábek E.; Weak pigeonhole principle, and randomized computation; Ph.D. thesis, Faculty of Mathematics and Physics, Charles University, Prague, 2005.
[15] Jeřábek E.; Approximate counting in bounded arithmetic, Journal of Symbolic Logic, 72:959-993, 2007.
[16] Jeřábek E.; On independence of variants of the weak pigeonhole principle, Journal of Logic and Computation, 17:587-604, 2007.
[17] Jeřábek E.; Approximate counting by hashing in bounded arithmetic, Journal of Symbolic Logic, 74:829-860, 2009.
[18] Krajíček J.; Bounded arithmetic, propositional logic, and complexity theory, Cambridge University Press, 1995.
[19] Krajíček J.; Dual weak pigeonhole principle, pseudo-surjective functions and provability of circuit lower bounds, Journal of Symbolic Logic, 69(1):265-286, 2004.
[20] Krajíček J., Pudlák P., Takeuti G.; Bounded arithmetic and the polynomial hierarchy, Annals of Pure and Applied Logic, 52:143-153, 1991.
[21] Moshkovitz D.; Lecture notes: PCP and Hardness of Approximations, http://people.csail.mit.edu/dmoshkov/courses/pcp-mit/4-linearity-test.pdf.
[22] Nisan N., Wigderson A.; Hardness vs. randomness, Journal of Computer and System Sciences, 49(2):149-167, 1994.
[23] Parikh, R.; Existence and feasibility in arithmetic, Journal of Symbolic Logic, 36: 494-508, 1971.
[24] Pich J.; Circuit lower bounds in bounded arithmetics, Annals of Pure and Applied Logic, 166(1), 2015.
[25] Razborov A.A.; Bounded Arithmetic and Lower Bounds in Boolean Complexity, Feasible Mathematics II, pp. 344-386, 1995.
[26] Razborov A.A; Pseudorandom Generators Hard for k-DNF Resolution and Polynomial Calculus, Annals of Mathematics, 181(2):415-472, 2015.