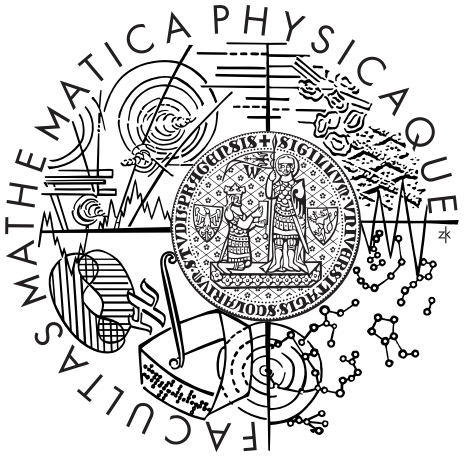
Charles University in Prague
Faculty of Mathematics and Physics

DOCTORAL THESIS



Ján Pich

Complexity Theory in Feasible Mathematics

Department of Algebra

Supervisor of the doctoral thesis:	Prof. RNDr. Jan Krajíček DrSc., MAE
Study programme:	Mathematics
Specialization:	Algebra, Theory of Numbers and Mathematical Logic

Prague 2014

Acknowledgements

Most of all, I would like to thank my supervisor Jan Krajíček for inspirational conversations, far reaching suggestions, careful reading of my manuscripts, patience and support. I am also grateful for a stimulating environment and a guidance provided by members of the Proof Complexity and Computation Complexity group in Prague. That is, in particular, Pavel Pudlák, Neil Thapen, Emil Jeřábek, Jiří Sgall, Michal Koucký, and fellow PhD students Michal, Zi and Sebastian.

It was an enriching experience to meet all participants of Prague Special Semester in Logic and Complexity during the fall of 2011. Here I investigated intuitionistic logic with Kaveh Ghasemloo. As well I learnt extraordinarily many things from a number of researchers attending Semantics and Syntax programme in Cambridge, summer term 2012.

Especially, I thank Albert Atserias and Sam Buss for advise which led to some results presented in the thesis. Further, an anonymous reviewer significantly improved presentation of the first attached article.

Finally, I want to thank my parents for encouragement and support through the whole studies.

My research was financially supported mainly by grant projects GA UK 5732, IAA100190902 GA AV ČR, N-SPP 2011/2012 and SVV-2014-260107.

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In ..... date ..... signature of the author

Názov práce: Teória zložitosti v dosiahnutel'nej matematike

Autor: Ján Pich

Katedra: Katedra Algebry

Vedúci dizertačnej práce: Prof. RNDr. Jan Krajíček, DrSc., MAE

Abstrakt: Skúmame dokázatel'nost' tvrdení z teórie zložitosti v obmedzenej aritmetike. Za istých zložitostných predpokladov ukážeme, že teórie so slabšími dosvedčovacími vlastnost'ami než S^1_2 nemôžu dokázat' spodné odhady vel'kosti n^k na booleovské obvody pre SAT vyjadrené formulou $LB(SAT, n^k)$. Špeciálne, prvorádová teória pravdivých univerzálnych tvrdení v jazyku obsahujúcom symboly pre všetky uniformné NC^1 algoritmy nedokazuje $LB(SAT, n^{4kc})$ pre $k \geq 1, c \geq 2$ predpokladajúc existenciu funkcie $f \in SIZE(n^k)$, ktorá nie je aproximovatel'ná formulami F_n subexponenciálnej vel'kosti $2^{O(n^{1/c})}$ so subexponenciálnou výhodou: $P_{x\in\{0,1\}^n}[F_n(x) = f(x)] \geq 1/2 + 1/2^{O(n^{1/c})}$. Bezpodmienečne, teória V^0 nedokazuje kvazipolynomiálne spodné odhady na booleovské obvody pre SAT. Čo sa týka horných odhadov, dokážeme PCP vetu v Cookovej teórii PV_1. To zahŕňa formalizáciu (n, d, λ)-grafov v PV_1. Ako dôsledok dostaneme polynomiálne krátke Extended Frege dôkazy tautologií vyjdadrujúcich PCP vetu.

Kl'účové slová: booleovské obvody, obmedzená aritmetika, PCP veta

Title: Complexity Theory in Feasible Mathematics

Author: Ján Pich

Department: Department of Algebra

Supervisor: Prof. RNDr. Jan Krajíček, DrSc., MAE

Abstract: We study the provability of statements and conjectures from Complexity Theory in Bounded Arithmetic. First, modulo a hardness assumption, we show that theories weaker in terms of provably total functions than Buss's theory S^1_2 cannot prove n^k-size circuit lower bounds for SAT formalized as a Σ^b_2-formula $LB(SAT, n^k)$. In particular, the true universal first-order theory in the language containing names for all uniform NC^1 algorithms denoted T_{NC^1} does not prove $LB(SAT, n^{4kc})$ where $k \geq 1, c \geq 2$ unless each function $f \in SIZE(n^k)$ can be approximated by formulas F_n of subexponential size $2^{O(n^{1/c})}$ with subexponential advantage: $P_{x\in\{0,1\}^n}[F_n(x) = f(x)] \geq 1/2 + 1/2^{O(n^{1/c})}$. Unconditionally, V^0 does not prove quasipolynomial $n^{\log n}$-size circuit lower bounds for SAT. Considering upper bounds, we prove the PCP theorem in Cook's theory PV_1. This includes a formalization of the (n, d, λ)-graphs in PV_1. A consequence of the result is that Extended Frege proof system admits p-size proofs of tautologies encoding the PCP theorem.

Keywords: Circuit Lower Bounds, Bounded Arithmetic, The PCP theorem

Contents

Introduction

This thesis consists of two articles:

- *Circuit Lower Bounds in Bounded Arithmetics*, to appear in Annals of Pure and Applied Logic, [11],
- *Logical Strength of Complexity Theory and a Formalization of the PCP Theorem in Bounded Arithmetic*, submitted to Logical Methods in Computer Science, [12],

preceded by an introduction providing an extended discussion of their context together with some possible future research directions.

The aim of this thesis is to understand which conjectures from complexity theory can be feasibly true in the sense that they would be provable without declaring the existence of an infeasible object.

Perhaps the best standard approximation of such proofs, which might be called feasible mathematics, is the intuitionistic version of Buss's theory S^1_2 denoted IS^1_2, cf. [3, 4]. As S^1_2 is $\forall\Sigma^b_1$-conservative over IS^1_2, cf. [2], IS^1_2 contains an essential part of Cook's theory PV_1 formalizing a reasoning with p-time concepts. In addition, IS^1_2 admits a stronger form of witnessing than PV_1: for any formula A, $IS^1_2 \vdash \exists y A(x, y)$ implies the existence of a p-time function f such that $A(x, f(x))$, cf. [1]. In S^1_2 and PV_1 this holds only for Σ^b_1 formulas A, cf. [3, 7].

The investigation of Complexity Theory in theories of Bounded Arithmetic is closely related to Propositional Proof Complexity. One reason being that each statement S either cannot be feasibly true simply because it is not witnessed efficiently in the sense that its existential quantifiers are not witnessed by p-time algorithms or otherwise S is equivalent to a Π^b_1-formula S' which can be translated to a sequence of tautologies. If the tautologies are hard for Extended Frege propositional proof system (EF), the statement S' is unprovable in S^1_2. A form of the opposite implication holds too. See [3, 7] for the details.

In the first attached article *Circuit Lower Bounds in Bounded Arithmetics* we study the nonuniform equivalent of the P$\neq$NP conjecture, i.e. SAT $\notin$ P/poly, from the perspective of feasible mathematics.

Polynomial circuit lower bounds for SAT are formalized here as a Σ^b_2-formula $LB(SAT, n^k)$. The article includes an observation that under the assumption of the existence of one-way functions secure against p-size circuits and functions in E hard on average for subexponential circuits, $LB(SAT, n^k)$ is witnessed by a certain p-time protocol, cf. [11, Proposition 1.4.3] . This allows us to express $LB(SAT, n^k)$ as a sequence of tautologies $lb(SAT, n^k)$ that could be even exponentially hard for strong proof systems like EF, cf. Section 0.1 in this introduction. On the contrary, the usual encoding of SAT $\notin$ P/poly by propositional formulas uses the whole truth table of SAT on inputs of each given length. Such formulas are quasi-polynomially easy assuming they are true at all. Nevertheless, they are considered as candidate hard tautologies for strong proof systems.

Interestingly, the relation to Propositional Proof Complexity mentioned above shows that deriving the unprovability of $LB(SAT, n^k)$ in PV_1, which would be a form of consistency of P=NP, could be also seen as an approach to the separation of NP and coNP.

We do not know how to show that $PV_1 \nvdash LB(SAT, n^k)$ but we can obtain the unprovability basically for any weaker theory in terms of provably total functions. In particular, we prove that T_{NC^1}, the true universal first-order theory in the language containing names for all uniform NC^1 algorithms, cannot prove $LB(SAT, n^{4kc})$ where $k \geq 1, c \geq 2$ unless each function $f \in SIZE(n^k)$ can be approximated by formulas F_n of subexponential size $2^{O(n^{1/c})}$ with subexponential advantage: $P_{x\in\{0,1\}^n}[F_n(x) = f(x)] \geq 1/2 + 1/2^{O(n^{1/c})}$, cf. [11, Theorem 1.6.1]. Using Hastad's lower bound on AC^0 circuits computing PARITY, we derive also an unconditional result: the bounded arithmetic theory V^0 cannot prove quasi-polynomial $n^{\log n}$-size circuit lower bounds on SAT, cf. [11, Corollary 1.6.3]. These proofs proceed by showing that a more efficient witnessing of $LB(SAT, n^k)$ than the one by p-time algorithms is impossible unless certain hardness assumption fails.

Since it is hard to obtain an unprovability of $LB(SAT, n^k)$ in PV_1 we discuss also complexity-theoretic statements which could be harder to witness and thus easier to be shown unprovable. Notably, the problem of recognizing the range of a given pseudorandom generator, $LB(Rng(g), n^k)$, cf. Section 0.2 in this introduction.

The second main component of the thesis is the article *Logical Strength of Complexity Theory and a Formalization of the PCP Theorem in Bounded Arithmetic*. It presents known formalizations of theorems from Complexity Theory demonstrating that theories of Bounded Arithmetic like PV_1 are actually very powerful. Our contribution is a PV_1-proof of the PCP theorem [12, Theorem 2.7] which includes a formalization of the (n, d, λ)-graphs, cf. [12, Section 2.6]. In this sense, the PCP theorem is shown to be feasibly true.

A motivation for this formalization was to find a true Π^b_1 complexity-theoretic statement unprovable in PV_1. As this task is too hard it would be good to find at least a Π^b_1 complexity-theoretic statement whose known proofs exceed S^1_2. Our result shows that the PCP theorem which is equivalent to a Π^b_1 sentence is not such case. Consequently, tautologies encoding the PCP theorem have p-size proofs in EF. It remains open if they could be hard for Frege system.

The thesis is organized as follows. Section 0.1 of the introduction discusses circuit lower bounds and tautologies based on feasibly witnessed statements from Complexity Theory. Section 0.2 of the introduction presents a problem of recognizing range of proof complexity generators. Then we attach articles *Circuit Lower Bounds in Bounded Arithmetics* [11] and *Logical Strength of Complexity Theory and a Formalization of the PCP Theorem in Bounded Arithmetic* [12].

In the introductory sections we refrain from defining concepts from Bounded Arithmetic and Propositional Proof Complexity, see Buss [3] Krajíček [7] and Cook-Nguyen [5] for a general introduction into the area. However, some of them can be found in the attached articles. For example, Σ^b_i, Π^b_i hierarchies and a definition of S^1_2 are in [11], and a definition of the theory PV_1 is in [12].

0.1 Circuit Lower Bounds and Complexity - Theoretic Tautologies

In [15] Razborov suggested to investigate propositional formulas $\neg Circuit_t(f_n)$ asserting that the circuit size of Boolean function f_n with n variables is greater than t. The function f_n is represented by its truth table, i.e. a string of 2^n bits. A consequence of one of his conjectures [15, Conjecture 1] is that under certain hardness assumptions, $\neg Circuit_{n^{\omega(1)}}(SAT_n)$ are hard tautologies for Frege system.

Such a hardness for EF would imply the unprovability of a first-order Π^b_1 equivalent of $\neg Circuit_t(SAT_n)$ in S^1_2. Crucially, these formalizations represent SAT_n by its truth table. Hence, if we reason about them inside PV_1 it is as if we could manipulate feasibly with whole truth tables of SAT_n. From the perspective of feasible mathematics it is more natural if SAT is represented by its defining formula. In [11, Section 1.2] we formalize polynomial circuit lower bounds for SAT as such Σ^b_2-formula $LB(SAT, n^k)$ of the following form.

$$\forall 1^n > n_0,\ \forall \textit{ circuit } C \textit{ with } n \textit{ inputs and size } n^k\ \exists\, y, a \textit{ such that}$$
$$(C(y) = 0 \wedge SAT(y, a)) \vee (C(y) = 1 \wedge \forall z \neg SAT(y, z))$$

Here n_0, k are constants, 1^n denotes a string of length n, and $SAT(y, z)$ means that z is a satisfying assignment to the propositional 3CNF formula y. Formally, $LB(SAT, n^k)$ is a set of formulas for all possible n_0's but this should not cause any confusion. Whenever we say that $LB(SAT, n^k)$ is provable in a theory T we mean that it is provable in T for some n_0.

If $\neg Circuit_{n^k}(SAT_n)$ is hard for EF, $LB(SAT, n^k)$ is unprovable in S^1_2. However, it should be "exponentially" harder to reason about $LB(SAT, n^k)$ and thus easier to show that $S^1_2 \nvdash LB(SAT, n^k)$.

By known witnessing theorems [6, 14], $S^1_2 \vdash LB(SAT, n^k)$ implies the existence of p-time functions interactively witnessing the existential quantifiers in $LB(SAT, n^k)$, see [11, Section 1.3.2] for a precise definition. In such case we say that $LB(SAT, n^k)$ has an S-T protocol with $poly(n)$ rounds. Moreover, by witnessing properties of IS^1_2, if $IS^1_2 \vdash LB(SAT, n^k)$, then the existential quantifiers in $LB(SAT, n^k)$ can be witnessed by a single p-time algorithm. Denote the existence of such witnessing by $LB(SAT, n^k) \in P$, see [11, Section 1.3.2].

A way how to derive, at least conditionally, the unprovability of $LB(SAT, n^k)$ in PV_1 is to show that any S-T protocol for $LB(SAT, n^k)$ with $poly(n)$ rounds contradicts some standard hardness assumption. However, such protocols follow from usual cryptographic conjectures, cf. [11, Proposition 1.4.3]. It remains open whether we could similarly obtain $LB(SAT, n^k) \in P$. Nevertheless, it makes sense to consider propositional formulas based on the assumption that $LB(SAT, n^k) \in P$.

$LB(SAT, n^k) \in P$ implies that $LB(SAT, n^k)$ can be rewritten as a Π^b_1-sentence in which the existence of the formula y and its assignment a is witnessed by a specific p-time function whose inputs are n^k-size circuits C. Thus, $LB(SAT, n^k)$ can be expressed as a sequence of tautologies $lb(SAT, n^k)$. The details of such translation can be found in [3, 5].

It seems that these tautologies could be even exponentially hard for EF. Notice that this is not the case with formulas $\neg Circuit_{n^k}(SAT_n)$ which have quasipolynomial-size proofs if they are true.

Problem 1. *Assume $LB(SAT, n^k) \in P$. Are the resulting tautologies $lb(SAT, n^k)$ hard for EF?*

Instead of $LB(SAT, n^k) \in P$ we could similarly use any S-T protocol for $LB(SAT, n^k)$ with $poly(n)$ rounds. Therefore, assuming the standard cryptographic assumptions we have tautologies expressing polynomial circuit lower bounds for SAT that could be exponentially hard for EF.

Clearly, circuit lower bounds do not play a significant role in the above translation to propositional logic. Essentially any complexity-theoretic statement with existential quantifiers witnessed in p-time (or even by p-size circuits) might serve as a candidate hard tautology.

0.2 Facts that are hard to witness

It would be interesting to find any Π^b_1 complexity-theoretic conjecture, not necessarily circuit lower bounds, whose provability in PV_1 contradicts some standard hypothesis. In fact, the choice of Π^b_1 can be relaxed too as in $LB(SAT, n^k)$. However, for more complex formulas, obtaining a conditional unprovability result might become trivial.

A direct application of Buss's witnessing [3] implies that Σ^b_1 formulas like $\exists x, f(x) = y$ for a PV-function f cannot hold in PV_1 unless the function f is easy to invert. This seems to work for many statements about concepts conjectured to be stronger than P. See also Buss's [3] consistency of NP$\cap$coNP=P with S^1_2 or Pudlák-Krajíček's [10] consistency of p-optimality of EF with S^1_2. Despite that, the witnessing method might lead to interesting unprovability results because for many statements "about p-time concepts" we do not know whether they are likely to be witnessed efficiently or not.

Incomlpleteness-style methods can be used to obtain also an unprovability of Π^b_1-formulas in strong theories like PV_1. Pudlák [13] showed that even the theory S_2 does not prove the so called bounded consistency of S^1_2. See Krajíček [7, Chapter 10.5] for more related results. Unfortunately, no reduction of the bounded consistency to a standard complexity-theoretic statement is known.

We will now describe a problem we find interesting from the perspective of efficient witnessing.

Since the unprovability of $LB(SAT, n^k)$ in PV_1 is hard to obtain, it might be useful to consider similar statements that could be harder to witness. Krajíček (private communication) suggested to investigate formulas $LB(Rng(g), n^k)$ asserting that no n^k-size circuit can recognize the range of a pseudorandom generator g. As we will see, this connects the witnessing method and the theory of proof complexity generators introduced in [1, 8].

Let $g_n : \{0,1\}^n \mapsto \{0,1\}^{n+1}$ be a map such that $g(x) = y$ is a relation defined by a Σ^b_1-formula. Then for constants n_0 and k we have Σ^b_2-formula $LB(Rng(g), n^k)$:

$$\forall 1^n > n_0,\ \forall \text{ circuit } C \text{ with } n \text{ inputs and size } n^k\ \exists\, y < 2^{n+1}, a < 2^n \text{ such that}$$
$$(C(y) = 0 \wedge y = g(a)) \vee (C(y) = 1 \wedge \forall z < 2^n\ y \neq g(z))$$

Problem 2. *Fix a constant k. Is there a p-time algorithm which given a string of length n and any n^k-size circuit C with n inputs finds $y \in \{0,1\}^{n+1}$ and $a \in \{0,1\}^n$ such that*

$$(C(y) = 0 \wedge g(a) = y) \vee (C(y) = 1 \wedge y \notin Rng(g))$$

Shortly, is $LB(Rng(g), n^k) \in P$?

A proof complexity generator $g : \{0,1\}^n \mapsto \{0,1\}^m$ for injective function $m = m(n) > n$ is a map computed by $poly(m)$-size circuits. For any string $b \in \{0,1\}^m$, let $\tau(g)_b$ be a propositional formula asserting that $b \notin Rng(g)$. A generator g is hard for a proof system P if there are no p-size P-proofs of formulas $\tau(g)_b$ for any sequence of different b's. See [1, 8] for more details. It has been indeed conjectured that certain generators g should be hard for systems like EF. A survey of the area can be found in [9].

We will show that formulas $LB(Rng(g), n^k)$ provide a formally simpler version of some problems from the theory of proof complexity generators. The reason is that they do not claim just that a string b is not in the range of given generator g but, in fact, that no small circuit can recognize the range.

Say that a generator $g_n : \{0,1\}^n \mapsto \{0,1\}^{n+1}$ is s-Σ^b_1 definable if the relation $g(x) = y$ is defined by a Σ^b_1 formula in the prenex normal form without negations in the prefix. The s-Σ^b_1 definability of a generator g guarantees that the provability $IS^1_2 \vdash LB(Rng(g), n^k)$ gives us tautologies encoding $LB(Rng(g), n^k)$.

Proposition 0.2.1 (Krajíček (private communication))**.** *If a s-Σ^b_1-definable generator g is hard for EF, then $IS^1_2 \nvdash LB(Rng(g), n^k)$ for any $k \geq 1$.*

Proof. Either $LB(Rng(g), n^k) \notin P$, and so $IS^1_2 \nvdash LB(Rng(g), n^k)$, or otherwise there are tautologies $lb(Rng(g), n^k)$ asserting $LB(Rng(g), n^k)$ which are hard for EF. If they were not, EF would prove $b \notin Rng(g)$ for a string b obtained by substituting a trivial circuit which always outputs 1 to free variables of the tautology $lb(Rng(g), n^k)$. □

Consequently, for each s-Σ^b_1-definable generator g, either $LB(Rng(g), n^k) \notin P$ or there are tautologies $lb(Rng(g), n^k)$ encoding $LB(Rng(g), n^k)$ which are at least as hard as the underlying generator g. In the case of Nisan-Wigderson generators considered in [15], this gives us tautologies $lb(Rng(NW_{f,A}), n^k)$ and thus a potentially easier version of Razborov's conjecture from [15].

Bibliography

[1] Alekhnovich, M., Ben-Sasson, E., Razborov, A., Wigderson, A., *Pseudorandom Generators in Propositional Proof Complexity*. SIAM J. on Comp., 34(1), 2004.

[2] Avigad, J. *Interpreting classical theories in constructive ones*. Journal of Symbolic Logic, 65(4), 2000.

[3] Buss, S.R. *Bounded Arithmetic*. Bibliopolis, Naples, 1986.

[4] Buss, S.R. *The Polynomial Hierarchy and Intuitionistic Bounded Arithmetic*. Structure in Complexity, Lecture Notes in Computer Science, 223:77-103, 1986.

[5] Cook, S.A., Nguyen, P., *Logical Foundations of Proof Complexity*. Cambridge University Press, 2010.

[6] Krajíček, J. *No counter-example interpretation and interactive computation*. Logic from Computer Science, 21:287-293, 1992.

[7] Krajíček, J. *Bounded arithmetic, propositional logic, and complexity theory*. Cambridge University Press, 1995.

[8] Krajíček, J. *On the weak pigeonhole principle*. Fundamenta Mathematicae, 170(1-3):123-140, 2001.

[9] Krajíček, J. *Forcing with random variables and proof complexity*. Cambridge University Press, 2011.

[10] Krajíček, J., Pudlák, P., *Propositional Proof Systems, the Consistency of First Order Theories and the Complexity of Computations*. Journal of Symbolic Logic, 54(3):1063-1079, 1989.

[11] Pich, J. *Circuit Lower Bounds in Bounded Arithmetics*. to appear in Annals of Pure and Applied Logic.

[12] Pich, J. *Logical strength of complexity theory and a formalization of the PCP theorem in bounded arithmetic*. submitted.

[13] Pudlák, P. *A note on bounded arithmetic*. Fundamenta Mathematicae, 136:85-9, 1990.

[14] Pudlák, P. *Some relations between subsystems of arithmetic and complexity theory*. Proc. Conf. Logic from Computer Science, 21:499-519, 1992.

[15] Razborov, A.A. *Pseudorandom Generators Hard for k-DNF Resolution and Polynomial Calculus*. preprint (available at authors webpage), 2002-2003.

1. Attachment:

Circuit Lower Bounds in Bounded Arithmetics

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 prove that T_{NC^1}, the true universal first-order theory in the language containing names for all uniform NC^1 algorithms, cannot prove that for sufficiently large n, SAT is not computable by circuits of size n^{4kc} where $k \geq 1, c \geq 2$ unless each function $f \in SIZE(n^k)$ can be approximated by formulas $\{F_n\}_{n=1}^{\infty}$ of subexponential size $2^{O(n^{1/c})}$ with subexponential advantage:

$$P_{x\in\{0,1\}^n}[F_n(x) = f(x)] \geq 1/2 + 1/2^{O(n^{1/c})}$$

Unconditionally, V^0 cannot prove that for sufficiently large n, SAT does not have circuits of size $n^{\log n}$. The proof is based on an interpretation of Krajíček's proof [J.Krajíček, On the proof complexity of the Nisan-Wigderson generator based on NP∩coNP function, Journal of Mathematical Logic 11(2011) 11-27] that certain NW-generators are hard for T_{PV}, the true universal theory in the language containing names for all p-time algorithms.

1.1 Introduction

We investigate the provability of polynomial circuit lower bounds in weak fragments of arithmetic including Buss's [1] theory S_2^1 and its subsystems. These theories are sufficiently strong to prove many important results in Complexity Theory. In fact, they can be considered as formalizations of feasible mathematics. A motivation behind the investigation of these theories is the general question whether the existential quantifiers in complexity-theoretic statements can be witnessed feasibly and so that to derive the witnessing we do not need to exceed feasible reasoning.

Informally, our formalization of n^k-size circuit lower bounds for SAT, denoted by $LB(SAT, n^k)$, has the following form:

$\forall n > n_0$, $\forall$ *circuit* C *with* n *inputs and size* n^k $\exists\, y, a$ *such that*
$(C(y) = 0 \wedge SAT(y, a)) \vee (C(y) = 1 \wedge \forall z \neg SAT(y, z))$

where n_0, k are constants and $SAT(y, z)$ means that z is a satisfying assignment to the propositional 3CNF formula y, see Section 2.

If S_2^1 proves the formula $LB(SAT, n^k)$ for some constant n_0, then by the usual kind of witnessing, Buss's witnessing [1] or the KPT theorem [12], for any n^k-size circuit with n inputs we can efficiently find a formula of size n on which the circuit fails to solve SAT, see Proposition 1.4.1.

One could hope to use the p-time algorithm to derive a contradiction with some established hardness assumption, however, Atserias and Krajíček noticed that the same p-time algorithm follows from standard cryptographic conjectures, see Proposition 1.4.2. (Actually, as discussed in Section 4, a randomized version of such observations appeared already in Buss [3, Section 4.4] and Cook-Mitchell [6, Section 6].) It is an interesting question to ask how strong theories are needed to derive these conjectures.

We do not know how to obtain the unprovability of SAT circuit lower bounds in S^1_2 but we can do it basically for any weaker theory with stronger witnessing properties. We present it in the case of theory T_{NC^1} which is the true universal first-order theory in the language containing names for all uniform NC^1 algorithms.

In theories weaker than S^1_2, like the theory T_{NC^1}, the situation is less natural because they cannot fully reason about p-time concepts. In particular, some universal quantifiers in $LB(SAT, n^k)$ can be replaced by existential quantifiers without changing the intuitive meaning of the sentence. The resulting formula $LB_\exists(SAT, n^k)$ (defined in Section 5) is equivalent to $LB(SAT, n^k)$ in S^1_2 but not necessarily in T_{NC^1}. This is because $LB_\exists(SAT, n^k)$ asserts among other things the existence of computations of general n^k-size circuits, a fact which may not be T_{NC^1}-provable. Therefore, it is essentially trivial to obtain a conditional unprovability of $LB_\exists(SAT, n^k)$ in T_{NC^1}, see Proposition 1.6.1. This is not the case with the formalization $LB(SAT, n^k)$ and in this sense it is easier and more suitable for the theory T_{NC^1} to reason about $LB(SAT, n^k)$.

The main result of this paper is that we can obtain a conditional unprovability of $LB(SAT, n^k)$ as well. We show that $LB(SAT, n^{4kc})$ for $k \geq 1, c \geq 2$ is unprovable in T_{NC^1} unless each function $f \in SIZE(n^k)$ can be approximated by formulas F_n of size $2^{O(n^{1/c})}$ with subexponential advantage:
$P_{x\{0,1\}^n}[F_n(x) = f(x)] \geq 1/2 + 1/2^{O(n^{1/c})}$. The proof will be quite generic. In particular, using known lower bounds on PARITY function, we will obtain that, unconditionally, V^0 cannot prove quasi polynomial ($n^{\log n}$-size) circuit lower bounds on SAT. Here, V^0 is a second-order theory of bounded arithmetic such that its provably total functions are computable in AC^0, see Section 5.

To prove our main theorem we firstly observe that by the KPT theorem [16] the provability of $LB(SAT, n^{4kc})$ in universal theories like T_{NC^1} gives us an $O(1)$-round Student-Teacher (S-T) protocol finding errors of n^{4kc}-size circuits attempting to compute SAT. Then, in particular, it works for n^{4kc}-size circuits encoding Nisan-Wigderson (NW) generators based on any function $f \in SIZE(n^k)$ and any suitable design matrix [17]. The interpretation of NW-generators as p-size circuits comes from Razborov [20]. In this situation we apply Krajíček's proof from [15] showing that certain NW-generators are hard for the true universal theory T_{PV} in the language containing names for all p-time algorithms. This is the main technique we use. We show that it works in our context as well and allows us to use the S-T protocol to compute f by subexponential formulas with a subexponential advantage.

Perhaps the most significant earlier result of this kind was obtained by Razborov [19]. Using natural proofs he showed that theory $S^2_2(\alpha)$ cannot prove superpolynomial circuit lower bounds on SAT unless strong pseudorandom generators do

not exist. In fact, his proof works even for sufficiently big polynomial circuit lower bounds. The second-order theory $S^2_2(\alpha)$ is however quite weak with respect to the formalization Razborov used. As far as we know his technique does not imply the unprovability of circuit lower bounds (formalized as here, see Section 2) even for V^0. In this respect, our proof applies to much stronger theories, basically to any theory weaker than S^1_2 in terms of provably feasible functions.

The paper is organized as follows. In Section 2 we formalize circuit lower bounds in the language of bounded arithmetic. In Section 3 we define a conservative extension of the theory S^1_2 denoted $S^1_2(bit)$ and state its properties. In Section 4 we discuss the provability of circuit lower bounds in $S^1_2(bit)$. Section 5 defines subtheories of $S^1_2(bit)$ for which we prove our main unprovability results in Section 6.

1.2 Formalization

The usual language of arithmetic contains well known symbols: $0, S, +, \cdot, =, \leq$. To encode reasoning about computation it is natural to consider also symbols $\lfloor \frac{x}{2} \rfloor$, $|x|$ for the length of the binary representation of x and $\#$ with the intended meaning $x\#y = 2^{|x|\cdot|y|}$. Theories of bounded arithmetic are typically defined using the language $L = \{0, S, +, \cdot, =, \leq, \lfloor x/2 \rfloor, |x|, \#\}$, cf. Buss [1]. We will consider also the language L_{bit} which contains in addition the symbol x_i for the i-th bit of the binary representation of x. The basic properties of symbols from L_{bit} are captured by a set of basic axioms $BASIC(bit)$ which we will not spell out, cf. [1, 13], e.g. chapter 5.2 in Krajíček [13] states the axioms for symbols in L and chapter 5.4 in [13] gives a construction of a formula in the language L defining the i-th bit of the binary representation of x which we use here as an axiom.

We say that 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: $\exists y,\ y \leq t$, or universal bounded: $\forall y,\ y \leq t$ where y is not occuring in t. $\Sigma^b_0 (= \Pi^b_0)$ denotes the set of all formulas in the language L with all quantifiers sharply bounded. Note that all relations defined by Σ^b_0 formulas are p-time computable. 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 the language L (formulas with all quantifiers bounded) is defined by counting the number of alternations of bounded quantifiers, ignoring the sharply bounded ones.

All NP resp. coNP properties are representable by Σ^b_1 resp. Π^b_1 formulas, cf. [11, 21, 22].

Define $\Sigma^b_i(bit), \Pi^b_i(bit)$ for $i \geq 0$ as above but in the language L_{bit}. For $i \geq 1$, $\Sigma^b_i(bit)$ resp. $\Pi^b_i(bit)$ formulas are actually equivalent to Σ^b_i resp. Π^b_i formulas in

the theory called PV_1, cf. [4, 13], see also Section 3.

We will now express circuit lower bounds in L_{bit}.

Firstly, denote by $Comp(C, y, w)$ a $\Sigma^b_0(bit)$-formula saying that w is a computation of circuit C on input y. Such a formula can be constructed in many ways and our results work for any $\Sigma^b_0(bit)$ formalization. For simplicity, we present here a less efficient one where C represents a directed graph on $|w|$ vertices.

Let $E_C(i, j)$ be $C_{[i,j]}$, the $[i, j]$th bit of C, where $[i, j]$ is the pairing function $[i, j] = (i + j)(i + j + 1)/2 + i$. $E_C(i, j) = 1$, $i, j < |w|$, means that there is an edge in circuit C going from the i-th vertex to the j-th vertex. For $k < |w|$, let $N_C(k)$ be the pair of bits $(C_{[|w|,|w|]+2k}, C_{[|w|,|w|]+2k+1})$ encoding the connective in the k-th node of circuit C, say $(0, 1)$ be $\wedge$, $(1, 0)$ be $\vee$, and $(1, 1)$ and $(0, 0)$ be $\neg$. Therefore, $|C| = [|w|, |w|] + 2|w|$. Then let $Circ(C, y, w)$ be the formula stating that C encodes a $|w|$-size circuit with $|y|$ inputs:

$$
\begin{array}{l}
\forall j < |w|, j \geq |y|, \\
\quad (N_C(j) = (1,0) \vee N_C(j) = (0,1) \rightarrow \exists i, k < j,\ i \neq k,\ \forall l < j,\ l \neq k,\ l \neq i, \\
\qquad (E_C(i,j) = 1 \wedge E_C(k,j) = 1 \wedge E_C(l,j) = 0)) \wedge \\
\quad (N_C(j) = (1,1) \vee N_C(j) = (0,0) \rightarrow \exists i < j,\ \forall l < j,\ l \neq i, \\
\qquad (E_C(i,j) = 1 \wedge E_C(l,j) = 0))
\end{array}
$$

which means that if the j-th node of C is $\wedge$ or $\vee$, there are exactly two previous nodes i, k of C with edges going from i and k to j, if the j-th node of C is $\neg$, there is exactly one previous node i with an edge going from i to j.

$Comp(C, y, w)$ says that for each $i < |y|$ the value of w_i is the value of the i-th input bit of y and each w_j is an evaluation of the j-th node of circuit C given w_k's evaluating nodes connected to the j-th node:

$$
\begin{array}{l}
Circ(C, y, w) \wedge \forall i < |y|,\ y_i = w_i \wedge \forall j, k, l < |w|,\ k \neq l,\ [\\
(N_C(j) = (0,1) \wedge E_C(k,j) = 1 \wedge E_C(l,j) = 1 \rightarrow (w_j = 1 \leftrightarrow w_k = 1 \wedge w_l = 1)) \wedge \\
(N_C(j) = (1,0) \wedge E_C(k,j) = 1 \wedge E_C(l,j) = 1 \rightarrow (w_j = 1 \leftrightarrow w_k = 1 \vee w_l = 1)) \wedge \\
((N_C(j) = (0,0) \vee N_C(j) = (1,1)) \wedge E_C(k,j) = 1 \rightarrow (w_j = 1 \leftrightarrow w_k = 0))]
\end{array}
$$

Formula $C(y; w) = 1$ stating that w is an accepting computation of circuit C on input y will be $Comp(C, y, w) \wedge w_{|w|-1} = 1$. Similarly for $C(y; w) = 0$.

Next, let $SAT(y, z)$ be a $\Sigma^b_0(bit)$-formula saying that z is a satisfying assignment to the propositional 3-CNF formula y.

To define it explicitly for each $i, j, k < 2m$ we let $y_{[i,j,k]} = 1$ if and only if the 3-CNF encoded in y contains a clause of variables v^p_i, v^p_j, v^p_k where v^p_i is v_i if $i < m$ and $\neg v_{i-m}$ if $i \geq m$. Here also $[i, j, k] = [i, [j, k]]$. Hence, the 3-CNF encoded in y has m variables $v_0, ..., v_{m-1}$ and $|y| = [2m - 1, 2m - 1, 2m - 1] + 1$. We use m implicitly given by y in the formula $SAT(y, z)$:

$$
\begin{array}{l}
\forall i, j, k < 2m,\ [y_{i,j,k} = 1 \rightarrow \\
\qquad (i, j, k < m \rightarrow z_i = 1 \vee z_j = 1 \vee z_k = 1) \wedge \\
\qquad (i, j < m \wedge k \geq m \rightarrow z_i = 1 \vee z_j = 1 \vee z_{k-m} = 0) \wedge \\
\qquad ...
\end{array}
$$

$$(i,j,k \geq m \rightarrow z_{i-m} = 0 \vee z_{j-m} = 0 \vee z_{k-m} = 0)]$$

Finally, for any k, hardness of SAT for n^k-size circuits can be expressed as the following $\forall\Sigma^b_2(bit)$ sentence

$$\begin{aligned}
&LB(SAT, n^k) : \\
&\forall 1^n > n_0,\ \forall C,\ \exists y, a,\ |a| < |y| = n,\ \forall w, z,\ |w| \leq n^k,\ |z| < |y|, \\
&\quad [Comp(C, y, w) \rightarrow \\
&\qquad (C(y; w) = 1 \wedge \neg SAT(y, z)) \vee (C(y; w) = 0 \wedge SAT(y, a))]
\end{aligned}$$

Here n_0 is a fixed constant which is not indicated in $LB(SAT, n^k)$. This should not cause any confusion. Whenever we say that $LB(SAT, n^k)$ is provable in a theory T we mean that it is provable in T for some n_0. Further, $\forall 1^n > n_0$ is a shortcut for $\forall m, n$ such that $|m| = n \wedge m > n_0$. Therefore, y is feasible in m and for each n_0 and k, $LB(SAT, n^k)$ is universal closure of a $\Sigma^b_2(bit)$ formula.

We use the formalization of circuit lower bounds which is essentially a family of statements parametrized by n_0 instead of the formalization of the form $\exists n_0,\ LB(SAT, n^k)$ because the latter would result in a formula with higher quantifier complexity and the witnessing necessary in our proofs would not work. A similar problem would arise if we used lower bounds of the form "$\forall 1^{n_0}, \exists 1^n > 1^{n_0}, \forall C, \exists y, a$ ...". Moreover, it seems natural to avoid situations in which $\exists n_0,\ LB(SAT, n^k)$ is provable but not for any specific n_0.

Note also that, strictly speaking, for fixed k, $LB(SAT, n^k)$ might not be equivalent to lower bounds with different encodings of SAT formulas. For instance, our encoding of 3CNF's makes the formula size (the n) always cubic in the number of variables. However, the choice of our encoding is rather arbitrary and our results apply analogously for any efficient encoding of 3CNF's. On the other hand, if we used general SAT formulas instead of 3CNF's, the predicate $SAT(x, y)$ would not be in AC^0 anymore what would cause problems in results concerning the provability in theory V^0. Then, we would need to decide what is the right formalization of circuit lower bounds in the case of V^0 and modify the proof accordingly which we want to avoid.

1.3 Feasible Mathematics

If we obtain n^k-size circuit lower bounds for SAT but do not find any efficient method how to witness errors of potential n^k-size circuits for SAT, some of these circuits might work in practice like correct ones. We will now define theories of feasible mathematics where provability of n^k-size circuit lower bound for SAT implies the existence of such an error witnessing.

Perhaps, the most prominent one is S^1_2 introduced by Buss [1]. We will use its conservative extension $S^1_2(bit)$. The theory $S^1_2(bit)$ is defined in the language L_{bit} and its axioms consist of $BASIC(bit)$ and polynomial induction for $\Sigma^b_1(bit)$-formulas A:

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

(S^1_2 is defined in the language L and its axioms consist of polynomial induction

for Σ^b_1-formulas and $BASIC(bit)$ except the defining axioms of x_i.) An important property of $S^1_2(bit)$ is Buss's witnessing theorem:

Theorem 1.3.1 (Buss [1]). *If $S^1_2(bit) \vdash \exists y A(x,y)$ for $\Sigma^b_0(bit)$-formula A, then there is a p-time function f such that $A(x, f(x))$ holds for any x.*

$S^1_2(bit)$ admits also a useful kind of witnessing for $\Sigma^b_2(bit)$-formulas which was obtained by using a direct method in Pudlák [18], and by using Herbrand functions in Krajíček [12].

Theorem 1.3.2 (Pudlák [18], Krajíček [12]). *If $S^1_2(bit) \vdash \exists y\ \forall z \leq t\ A(x,y,z)$ for $\Sigma^b_0(bit)$-formula A and term t depending only on x, y, then there is p-time algorithm S such that for any x either $\forall z \leq t\ A(x, S(x), z)$ or for some z_1, $\neg A(x, S(x), z_1)$. In the latter case, either $\forall z \leq t\ A(x, S(x, z_1), z)$ or there is z_2 such that $\neg A(x, S(x, z_1), z_2)$. However after $k \leq poly(|x|)$ rounds of this kind, $\forall z \leq t\ A(x, S(x, z_1, ..., z_k), z)$ holds for any x.*

Another theory with similar witnessing properties is PV_1 which is an extension of a theory PV introduced by Cook [4], see also [13]. The language of PV_1 consists of symbols for all functions given by a Cobham-like inductive definition of p-time functions (hence it contains L_{bit}). PV_1 defined in Krajíček-Pudlák-Takeuti [16] is then a first-order theory axiomatized by equations defining all the function symbols and a derivation rule similar to polynomial induction for open formulas. It is a universal theory, i.e. it has an axiomatization by purely universal sentences, and since all function symbols of PV_1 have well-behaved Σ^b_1 and Π^b_1 definitions in $S^1_2(bit)$, PV_1 is contained in the extension of $S^1_2(bit)$ by these definitions. We denote the extension also $S^1_2(bit)$.

Let $\Sigma^b_0(PV)$-formulas be defined as Σ^b_0-formulas but in the language of PV_1. PV_1 proves induction:

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

for $\Sigma^b_0(PV)$-formulas A.

Theories $S^1_2(bit)$ and PV_1 are weak fragments of arithmetic but they are sufficiently strong to prove many things. We can interpret provability in PV_1 and S^1_2 as capturing the idea of what can be demonstrated when our reasoning is restricted to manipulations of p-time objects.

1.3.1 More formalizations of circuit lower bounds for SAT

$LB(SAT, n^k)$ is not the only way to express circuit lower bounds for SAT. For example, for given n_0 and k, we can define formula $SCE(SAT, n^k)$ stating that for each $1^n > n_0$ and each n^k-size circuit there is a satisfiable formula of size n such that the circuit will not find its satisfying assignment.

$$SCE(SAT, n^k):$$
$$\forall 1^n > n_0,\ \forall C,\ \exists y, a,\ |a| < |y| = n,\ \forall w, z,\ |w| \leq n^k, |z| < |y|$$
$$[SAT(y,a) \wedge (C(y;w) = z \rightarrow \neg SAT(y,z))]$$

where $C(y;w) = z$ means that w is a computation of circuit C on input y with output bits z. Formally, $Comp(C,y,w) \wedge \forall i < |z|(w_{|w|-i-1} = 1 \leftrightarrow z_i = 1)$. SCE in $SCE(SAT, n^k)$ refers to "search SAT counterexample".

A different formalization of circuit lower bounds is given by the following formula $DCE(SAT, n^k)$ where DCE refers to "decision SAT counterexample". In $DCE(SAT, n^k)$ circuits C attempting to solve SAT have again just one output but using self-reducibility they are used to search for satisfying assignments of propositional formulas: If C says that a formula y is satisfiable, we can set the first free variable in y firstly to 1 and then to 0, and use C to decide in which of these cases the resulting formula is satisfiable and in the same manner continue searching for the full satisfying assignment. $DCE(SAT, n^k)$ states that for each n^k-size circuit C there is a formula y and a possibly partial assignment to its variables a such that either 1.) $SAT(y,a)$ and C says that y is unsatisfiable, or 2.) $\neg SAT(y,a)$ for a full assignment a of y and C says that a satisfies y, or 3.) it happens that C gets into a local inconsistency: for a partial assignment a of y C claims that y under the assignment a is satisfiable but when we extend a by setting the first of the remaining free variables by 1 and 0 in both cases C claims that the resulting formula is unsatisfiable. Formally,

$$
\begin{aligned}
&DCE(SAT, n^k):\\
&\forall 1^n > n_0,\ \forall C,\ \exists y, a,\ |a| < |y| = n,\ \forall w^0, ..., w^4,\ |w^0|, ..., |w^4| \leq n^k,\ [\\
&\quad (Comp(C, y, w^0) \rightarrow (C(y; w^0) = 0 \wedge SAT(y, a)))\vee\\
&\quad (Comp(C, y(a), w^1) \rightarrow (C(y(a); w^1) = 1 \wedge FA(a, y) \wedge \neg SAT(y, a)))\vee\\
&\quad (Comp(C, y(a), w^2) \rightarrow (C(y(a); w^2) = 1 \wedge PA(a, y)\wedge\\
&\qquad\qquad (Comp(C, y(a1), w^3) \rightarrow C(y(a1); w^3) = 0)\wedge\\
&\qquad\qquad (Comp(C, y(a0), w^4) \rightarrow C(y(a0); w^4) = 0)))]
\end{aligned}
$$

where $y(a)$ encodes formula y under the assignment a, $FA(a,y)$ resp. $PA(a,y)$ means that a is full resp. partial assignment to variables in y and $y(a1)$ resp. $y(a0)$ is y under the assignment which is the extension of a that sets the first unassigned variable in y to 1 resp. 0. We leave details of these encodings to the reader.

The formalizations $LB(SAT, n^k), SCE(SAT, n^k), DCE(SAT, n^k)$ are (essentially) equivalent modulo slight changes to the size parameter. For example, $SCE(SAT, Kn^{k+1}) \rightarrow LB(SAT, n^k)$ and $LB(SAT, n^k + Kn) \rightarrow SCE(SAT, n^k)$, where $SCE(SAT, Kn^{k+1})$ is defined as $SCE(SAT, n^k)$ but with $|w|$ bounded by Kn^{k+1}. Similarly for $LB(SAT, n^k + Kn)$. Here, K is a sufficiently big constant and n_0 is arbitrary but the same constant in the assumption and in the conclusion of each implication. We claim that this is provable already in PV_1.

Proposition 1.3.1. *PV_1 proves the following implications*

$$
\begin{aligned}
SCE(SAT, Kn^{k+1}) &\rightarrow LB(SAT, n^k)\\
LB(SAT, n^k + Kn) &\rightarrow SCE(SAT, n^k)\\
LB(SAT, n^k) &\rightarrow DCE(SAT, n^k)\\
DCE(SAT, n^k) &\rightarrow LB(SAT, n^k)
\end{aligned}
$$

where K is a sufficiently big constant and n_0 is arbitrary but the same constant in the assumption and the conclusion of each implication.

Proof: The first implication was observed in [5]: Assume $\neg LB(SAT, n^k)$, i.e. for a big enough n there is an n^k-size circuit C deciding SAT on instances of size n. Then there is a p-time function which given a circuit C witnessing $\neg LB(SAT, n^k)$ produces a Kn^{k+1}-size circuit sC which outputs a satisfying assignment $sC(y)$ for every satisfiable formula y of size n. For each i, the circuit sC finds the i-th bit of the satisfying assignment by asking C whether y remains satisfiable if the i-th variable is set to 1, given the values it has previously found for the first $i-1$ variables. Then (assuming $\neg LB(SAT, n^k)$ and $SAT(y, a)$) PV_1 proves by $\Sigma^b_0(PV)$ induction on i that y instantiated by the first i truth values is satisfiable according to C and hence $\neg SCE(SAT, Kn^{k+1})$.

Concerning the second implication: If $\neg SCE(SAT, n^k)$, i.e. for a big enough n there is an n^k-size circuit C which outputs a satisfying assignment $C(y)$ for every satisfiable formula of size n, then there is a p-time function which given any such circuit C produces an $(n^k + Kn)$-size circuit dC which decides SAT on instances of size n. Given a formula y, dC outputs 1 if and only if $C(y)$ satisfies y. Assuming $\neg SCE(SAT, n^k)$ it follows in PV_1 that for any y, a of size $|a| < |y| = n$, $(SAT(y, a) \rightarrow dC(y; w) = 1) \wedge (dC(y; w) = 1 \rightarrow SAT(y, C(y)))$, hence $\neg LB(SAT, n^k + Kn)$.

Next, in PV_1, if circuit C witnesses formula $\neg DCE(SAT, n^k)$, then it witnesses also $\neg LB(SAT, n^k)$: for any y, a of size $|a| < |y| = n$ for a big enough n, $C(y; w) = 0 \rightarrow \neg SAT(y, a)$ and if $C(y; w) = 1$ then by $\Sigma^b_0(PV)$-induction (as in the first implication) $C(y(b); w) = 1$ for a full assignment b of y for which $SAT(y, b)$ holds.

Finally, in PV_1, if circuit C witnesses formula $\neg LB(SAT, n^k)$, then it witnesses $\neg DCE(SAT, n^k)$: for any y, a of size $|a| < |y| = n$ for a sufficiently large n, $(C(y; w) = 0 \rightarrow \neg SAT(y, a))$, $C(y(a); w) = 1 \wedge FA(a, y) \rightarrow SAT(y, a)$ and if $C(y(a); w) = 1 \wedge PA(a, y)$ then for some b extending a $SAT(y, b)$ and thus $C(y(a1); w) = 1 \vee C(y(a0); w) = 1$. □

1.3.2 Witnessing errors of p-size circuits

Using $LB(SAT, n^k), SCE(SAT, n^k)$ and $DCE(SAT, n^k)$ we can define several types of error witnessing of p-size circuits claiming to solve SAT.

We say somewhat informally that $LB(SAT, n^k) \in$ P if there is a p-time algorithm A which for any sufficiently big n and any n^k-size circuit C with n inputs finds out y, a such that $LB(C, y, a)$:

$$C(y) = 0 \wedge SAT(y, a) \quad or \quad C(y) = 1 \wedge \forall z \neg SAT(y, z)$$

Intuitively, A witnesses the important existential quantifiers in $LB(SAT, n^k)$.

We say that $LB(SAT, n^k)$ has an S-T protocol with l rounds if there is a p-time algorithm S such that for any function T and any sufficiently big n, whenever S is given n^k-size circuit C, S outputs y_1, a_1 such that either $LB(C, y_1, a_1)$ or otherwise T sends to S w_1, z_1 certifying $\neg LB(C, y_1, a_1)$. Then S uses C, w_1, z_1 to produce y_2, a_2 and the protocol continues in the same way, S possibly using all counter-examples T sent in earlier rounds. But after at most l rounds S outputs y, a such that $LB(C, y, a)$.

Analogously, $DCE(SAT, n^k) \in$ P if there is a p-time algorithm A which for any sufficiently big n and any n^k-size circuit C with n inputs finds out y, a such that $DCE(C, y, a)$:

$C(y) = 0 \wedge SAT(y, a)$ or $C(y(a)) = 1 \wedge FA(a, y) \wedge \neg SAT(y, a)$ or
$C(y(a)) = 1 \wedge PA(a, y) \wedge (C(y(a0)) = 0 \wedge C(y(a1)) = 0)$

Finally, $SCE(SAT, n^k) \in$ P if there is a p-time algorithm A which for any sufficiently big n and any n^k-size circuit C with n inputs and n outputs finds out y, a such that $SAT(y, a) \wedge \neg SAT(y, C(y))$.

The phrase that $DCE(SAT, n^k)$ resp. $SCE(SAT, n^k)$ has an S-T protocol with l rounds could be defined similarly but notice that in this case T's advice would consist only of computations w of given circuit C which can be produced by S itself as it has C as input.

In practice, if we want to witness that no small circuit solves SAT, it does not seem sufficient to have a p-time algorithm for $LB(SAT, n^k)$ because such an algorithm could output a tautology but we would not have an apriori way to certify that it is indeed a tautology and hence a correctly witnessed error. Therefore, it seems that practically more appropriate error witnessing is defined by $DCE(SAT, n^k)$ or $SCE(SAT, n^k)$ in which we actually force given circuits to claim inconsistent statements. We discuss it in more detail in the next section.

1.4 Circuit Lower Bounds in $S^1_2(bit)$

In this section we observe that the provability of circuit lower bounds in $S^1_2(bit)$ would give us an efficient witnessing of errors of p-size circuits for SAT described in the previous section. Then we show that certain hardness assumptions imply the same efficient witnessing of errors. Consequently it seems that the first result itself cannot be used to show the unprovability of $LB(SAT, n^k)$ in $S^1_2(bit)$.

Similar observations appeared already in Buss [3]. More precisely, Proposition 1.4.1 is a folklore and Buss [3, Section 4.4] described also a witnessing of $SCE(SAT, n^k)$ by non-uniform p-size circuits based on the existence of strong pseudorandom generators which is analogous to the one from Proposition 1.4.2.

Proposition 1.4.1. *If $S^1_2(bit) \vdash LB(SAT, n^k)$, then $LB(SAT, n^k)$ has an S-T protocol with $poly(n)$ rounds. If $S^1_2(bit) \vdash SCE(SAT, n^k)$, then $SCE(SAT, n^k) \in P$. If $S^1_2(bit) \vdash DCE(SAT, n^k)$, then $DCE(SAT, n^k) \in P$.*

Proof: $LB(SAT, n^k), DCE(SAT, n^k)$ and $SCE(SAT, n^k)$ are universal closures of $\Sigma^b_2(bit)$-formulas so the first implication follows directly from Theorem 1.3.2. In case of $SCE(SAT, n^k)$ and $DCE(SAT, n^k)$ T's advice in the resulting S-T protocol consist just of computations of given circuit C. This can be, however, produced by S itself as it has C as input.

Alternatively, one could show in $S^1_2(bit)$ that $SCE(SAT, n^k)$, $DCE(SAT, n^k)$ can be stated in a $\forall\Sigma^b_1(bit)$ way and apply directly Buss's witnessing. □

An efficient witnessing of errors of p-time SAT algorithms can be performed in the following way.

If f is a one-way function, we can secretly produce $a \in \{0,1\}^n$ and ask the algorithm claiming to solve SAT whether the statement $f(a) = f(x)$ encoded as a $poly(|a|)$-size formula with free variables $x = x_1, ..., x_n$ is satisfiable (the formula might also contain some auxiliary variables used to express computation of f such that their value can be efficiently determined given any assignment to x), see Cook-Mitchell [6]. The algorithm is forced to say that the formula is satisfiable and by the choice of f, with high probability it will not find its satisfying assignment.

Atserias (private communication) suggested to derandomize this construction and Krajíček made the following observation.

Proposition 1.4.2. *If there exists a one-way permutation f computable in p-time and secure against p-size circuits, i.e. for any p-size circuits C_n there is a function $\epsilon(n) = n^{-\omega(1)}$ such that for large enough n,*

$$P_{x\in\{0,1\}^n}[C_n(f(x)) = x] \leq \epsilon(n)$$

and if there exists $h \in E$ hard on average for subexponential circuits, i.e. there is $\delta > 0$ such that for all circuits C_n of size $\leq 2^{\delta n}$ and large enough n,

$$P_{x\in\{0,1\}^n}[C_n(x) = h(x)] \leq 1/2 + 1/2^{\delta n}$$

then for each k, $SCE(SAT, n^k) \in P$.

Proof: If there is $h \in E$ hard on average for subexponential circuits, by [17] for each l there is c and NW-generator $g : \{0,1\}^{c \log n} \mapsto \{0,1\}^n$ such that g is $poly(n)$-time computable and for any n^l-size circuits D_n,

$$|P_{x\in\{0,1\}^{c\log n}}[D_n(g(x)) = 1] - P_{x\in\{0,1\}^n}[D_n(x) = 1]| \leq 1/n$$

This generator allows us to derandomize the construction above: Let f be a one-way permutation secure against p-size circuits. Take l such that for each $((n+1)^d)^k$-size circuits $C_{(n+1)^d}$ with $(n+1)^d$ inputs, the following predicate $C_{(n+1)^d}(''f(\underline{x}) = f(y)'') = x$ with input $x \in \{0,1\}^n$ can be computed by n^l-size circuits. Here, $''f(\underline{x}) = f(y)''$ is a 3CNF formula expressing the fact that $f(x) = f(y)$. The formula has free variables $y = y_1, ..., y_n$ together with auxiliary variables used to express the computation of f. On the other hand, $\underline{x}$'s in $''f(\underline{x}) = f(y)''$ are constants denoting $x \in \{0,1\}^n$. The size of $''f(\underline{x}) = f(y)''$ is n^d for an absolute constant d (but $''f(\underline{x}) = f(y)''$ can be seen also as a formula of size $(n+1)^d$). For the chosen l there is c and NW-generator g as mentioned above.

Now, we will describe the algorithm witnessing $SCE(SAT, n^k) \in P$. For sufficiently big n, given m^k-size circuit C_m with m inputs, $n^d \leq m < (n+1)^d$, consider the one-way function f on n inputs. Formulas of the form $''\underline{e} = f(y)''$ where $e \in \{0,1\}^n$ can be seen as formulas of size m. By exhaustive search find $b \in \{0,1\}^{c\log n}$ such that $C_m(''f(\underline{g(b)}) = f(y)'') \neq g(b)$. If such b did not exist, then $P_{x\in\{0,1\}^{c\log n}}[C_m(''f(\underline{g(x)}) = \overline{f(y)}'') = g(x)] = 1$. This would break g because by definition of f, $P_{x\in\{0,1\}^n}[C_m(''f(\underline{x}) = f(y)'') = x]$ is small. The failure of C_m is thus witnessed in p-time by the formula $''f(\underline{g(b)}) = f(y)''$ and its assignment $g(b)$. □

Proposition 1.4.2 says that under certain hardness assumptions we can witness circuit lower bounds for SAT in p-time. It is natural to ask now for a p-time witnessing of these assumptions. What we already know is that by Jeřábek [9, Corollary 3.6] the existence of a function $h \in E$ hard for subexponential circuits in S^1_2 would imply that S^1_2 proves tha so-called dual weak pigeonhole principle for PV-functions $dWPHP(PV)$. In this case, S^1_2 could formalize randomized algorithms as described in Jeřábek [10]. Krajíček observed that a witnessing of $LB(SAT, n^k)$ is also possible assuming just that $LB(SAT, n^k)$ holds but the witnessing is non-constructive and only by nonuniform p-size circuits, see Proposition 1.4.4.

Proposition 1.4.2 seems to imply that for proving $S^1_2(bit) \nvdash SCE(SAT, n^k)$ we need to use other properties than $SCE(SAT, n^k) \in$ P. Moreover, assumptions of Proposition 1.4.2 give us an S-T protocol for $LB(SAT, n^k)$ too. Informally, any n^k-size circuit C claiming to decide SAT can be used to search for satisfying assignments of propositional formulas. Using the algorithm from Proposition 1.4.2, S can produce y, a, such that $SAT(y, a)$ but C will not find any satisfying assignment of y. This means that either C claims that y is unsatisfiable or the assignment it finds does not satisfy y or while searching for a satisfying assignment it gets into a local inconsistency which is the only case when S needs to ask for an advice of T, a satisfying assignment of y extending the partial assignment found by C.

Proposition 1.4.3. *If the same hardness assumption as in Proposition 1.4.2 holds, then $LB(SAT, n^k)$ has an S-T protocol with 1 round (i.e. 1 advice of T) where S is a p-time algorithm, and $LB(SAT, n^k)$ has also an S-T protocol with $poly(n)$ rounds where S is in uniform AC^0. Here, "S in uniform AC^0" means that for each n, there are $poly(n)$ circuits $S^n_1, ..., S^n_{poly(n)}$, one for each round of the interaction of the S-T protocol, and the uniformity means that there is a p-time algorithm which produces S^n_j given 1^n and 1^j without knowing the interaction before round j.*

Proof:

By Proposition 1.4.2 we have a p-time algorithm A solving $SCE(SAT, n^{2k})$. Firstly, we show that $LB(SAT, n^k)$ has an S-T protocol with 1 round and p-time S.

For each n^k-size circuit C with one output bit, there is a circuit sC of size $\leq Kn^{k+1}$, for a sufficiently big K, searching for satisfying assignments of given formulas as in Proposition 1.3.1. Here we give a more detailed description: For each formula y, let a be a partial assignment of y produced by sC so far (empty at the beginning) and denote by $y(a)$ the formula y under the assignment a. If $C(y(a)) = 0$, sC outputs an assignment of y full of zeros. If $C(y(a)) = 1$, it assigns y^1_a, the first free variable in $y(a)$, firstly by 1 and then by 0. Denote the resulting formula $y(a1)$ resp. $y(a0)$. If $C(y(a1)) = C(y(a0)) = 1$, sC sets $y^1_a = 1$. If $C(y(a1)) = C(y(a0)) = 0$, sC outputs an assignment of y full of zeros. If $C(y(a1)) = 1$ and $C(y(a0)) = 0$, sC sets $y^1_a = 1$. If $C(y(a1)) = 0$ and $C(y(a0)) = 1$, it sets $y^1_a = 0$. In this way sC sets all variables in y.

Given C, S can produce sC in p-time and use A to find y, a_1 such that $SAT(y, a_1)$ but $\neg SAT(y, sC(y))$.

If $C(y) = 0$, S outputs y, a_1. Else, S simulates sC on input y. If it never happens that $C(y(a1)) = C(y(a0)) = 0$ for any partial assignment a produced by sC, S outputs $y(sC(y))$. Otherwise, for a partial assignment a of y, $C(y(a)) = 1$ and $C(y(a1)) = C(y(a0)) = 0$. In such case S outputs $y(a), a_2$ where a_2 is a full assignment of y extending a with all zeros. If this is not a correct answer, T replies with a_3 extending a and satisfying y. Then S outputs $y(ab), a_3$ where $b \in \{0, 1\}$ such that ab is consistent with a_3.

In all cases S succeeds after asking for at most 1 advice of T.

To get S in uniform AC^0 note that A actually produces a set B of $\leq n^c$ propositional formulas of the form $f(Y) = s$ and their satisfying assignments such that each Kn^{k+1}-size circuit fails on at least one of them. It suffices to use instead of A the set B, i.e. AC^0 S will try all of the formulas $f(Y) = s$ with their satisfying assignments in place of y, a_1. Recall that the AC^0 S is actually a sequence of polynomially many uniform AC^0 circuits in the sense that every reply of T is managed by a different AC^0 circuit.

Given C, S will firstly try some y, a_1 from B. If y, a_1 does not witness that C does not solve SAT as in $LB(SAT, n^k)$, T replies with the computation of C witnessing that $C(y) = 1$. S then finds out if $C(y(1)) = C(y(0)) = 0$ using the following general protocol. Whenever S needs to simulate given circuit C on input z, it outputs z with its arbitrary assignment r. If z, r does not witness that C fails to solve SAT, T replies either with a satisfying assignment d of z or with the computation of C on input z which can be verified by a uniform constant-depth formula. In the former case, S (but a different AC^0 circuit than the one which produced z, r) outputs z, d and this time it either witnesses that C fails to solve SAT or it gets the computation of C. In this way S finds out if $C(y(1)) = C(y(0)) = 0$ and continues to simulate sC and the S-T protocol with p-time S.

If the protocol above using y, a_1 does not witness failure of C, S tries another element from B in place of y, a_1. By the definition of B, at least one of them works. □

Note that the uniformity of the AC^0 S-T protocol described in Proposition 1.4.3 is not DLOGTIME because to produce the respective AC^0 circuits we need to compute a function $h \in E$ on log-sized inputs which is hard for subexponential circuits.

Further, while Proposition 1.4.3 says that uniform AC^0 S-T protocols for $LB(SAT, n^k)$ with $poly(n)$ rounds are likely to exist, in Theorem 1.6.1 we will show that under a hardness assumption $LB(SAT, n^k)$ has no AC^0 S-T protocols with $O(1)$ rounds.

The proof of Proposition 1.4.3 shows also that if $SCE(SAT, n^k) \in$ P, then $DCE(SAT, n^k) \in$ P. All in all, Buss's witnessing does not seem to help us to obtain the unprovability of $LB(SAT, n^k)$ in PV_1 or $S^1_2(bit)$. Maybe it could work for intuitionistic S^1_2 where the witnessing holds for arbitrarily complex formulas, cf. Buss [2]. The situation is different in case of weaker theories where we have more efficient witnessing. This will allow us to reduce to some hardness assumptions.

Before considering weaker theories let us also mention that in order to show $SCE(SAT, n^k) \in P/poly$, it suffices to assume that for any sufficiently big n, SAT restricted to instances of length n has no circuit of size n^{2k}. This was observed by Krajíček in [14] but unlike Buss's [3, Section 4.4] proof of $SCE(SAT, n^k) \in P/poly$ which assumes the existence of strong pseudorandom generators, this method is not constructive in the sense that it does not tell us what could be the hard SAT instances.

Krajíček's observation uses a well known combinatorial principle[1]:
Let $E \subseteq X \times Y$ be a bipartite graph, $|X| = 2^{n^k}, |Y| = 2^n$. Then

$$\forall x_1, ..., x_n \in X \; \exists y \in Y \bigwedge_{i=1,...,n} E(x_i, y) \Rightarrow$$

$$\exists y_1, ..., y_{n^k} \in Y \; \forall x \in X \bigvee_{i=1,...,n^k} E(x, y_i)$$

Now take as X the set of all $n^{k/2}$-size circuits and interpret $E(x, y)$ as "y is a satisfiable formula of size n and circuit x does not find a satisfying assignment of y". Assume n is big enough. If SAT restricted to instances of size n does not have n^k-size circuits, then for every n circuits $C_1, ..., C_n$ of size $n^{k/2}$ there is y such that $\bigwedge_{i=1,...,n} E(C_i, y)$. Else, there is a specific sequence of n circuits such that for any satisfiable y at least one of these n circuits finds a satisfying assignment of y and this yields a single n^k-size circuit solving SAT at length n, contradicting the assumption. By the principle above, there are then $y_1, ..., y_{n^k}$ such that for each $n^{k/2}$-size circuit C, $\bigvee_{i=1,...,n^k} E(C, y_i)$. Therefore there is an n^{2k}-size circuit which for each $x \in X$ finds y such that $E(x, y)$ by trying $E(x, y_i)$ for $i = 1, ..., n^k$ and thus using additional satisfying assignments $a_1, ..., a_{n^k}$ of respective y's as advice solves $SCE(SAT, n^{k/2})$.

Analogously, we can show that $DCE(SAT, n^k) \in P/poly$ by considering $E(x, y) =$ "circuit x rejects formula y which is satisfiable or circuit x accepts y but if it is used to find a satisfying assignment of y it ends up in the same inconsistent situation as in $DCE(x, y, a)$ for some a". Such $E(x, y)$ is a p-time relation.

It is not clear how to apply this technique in the case of $LB(SAT, n^k)$. Straightforwardly defining $E(x, y)$ as "circuit x rejects formula y which is satisfiable or circuit x accepts unsatisfiable y" does not work because then for each y, $\neg E(1, y) \vee \neg E(0, y)$ where 1 resp. 0 is a trivial circuit which outputs always 1 resp. always 0.

Therefore, we have the following proposition.

Proposition 1.4.4 (Krajíček [14]). *If for any sufficiently big n, SAT restricted to instances of length n has no circuit of size n^{2k}, then $SCE(SAT, n^k)$ and $DCE(SAT, n^k)$ are in P/poly.*

[1] To see that the principle holds note that by a counting argument whenever r x's from X remain unconnected to any of already chosen y's there is another $y \in Y$ connected to at least $r/2$ of these r x's.

1.5 Theories weaker than PV_1

We will now present some theories weaker than PV_1 like T_{NC^1} for which we will show the unprovability of circuit lower bounds. We could however similarly define a general theory T_C corresponding to a standard complexity class C and our results would work analogously.

Definition 1.5.1. *T_{NC^1} is the first-order theory of all universal L_{NC^1} statements true in the standard model of natural numbers where L_{NC^1} is the language containing names for all uniform NC^1 algorithms.*

T_{NC^1} is a universal theory so it admits the KPT theorem from [16]:

Theorem 1.5.1 (Krajíček-Pudlák-Takeuti [16])**.** *If $T_{NC^1} \vdash \exists y A(x,y)$ for open formula A, then there is a function f in uniform NC^1 such that $A(x, f(x))$ holds for any x.*

If $T_{NC^1} \vdash \exists y \forall z A(x,y,z)$ for open formula A, there are finitely many functions $f_1, ..., f_k$ in uniform NC^1 such that

$$T_{NC^1} \vdash A(x, f_1(x), z_1) \vee A(x, f_2(x, z_1), z_2) \vee ... \vee A(x, f(x, z_1, ..., z_{k-1}), z_k)$$

There are also two-sorted theories of Bounded Arithmetic corresponding to uniform AC^0, NC^1 and other complexity classes, cf. Cook-Nguyen [7]. The first-sort (number) variables are denoted by lower case letters $x, y, z, ...$ and the second-sort (set) variables by capital letters $X, Y, Z, ...$ The underlying language includes the symbols $+, \cdot, =, \leq, 0, 1$ of first-order arithmetic. In addition it contains symbol $=_2$ interpreted as equality between bounded sets of numbers, $|X|$ for the function mapping an element X of the set sort to the largest number in X plus one, and $\in$ for the relation $n \in X$ meaning that n is an element of X.

Bounded quantifiers for sets have the form $\exists X \leq t\ \phi$ which stands for $\exists X\ (|X| \leq t \wedge \phi)$ or $\forall X \leq t\ \phi$ for $\forall X\ (|X| \leq t \rightarrow \phi)$. Here t is a number term which does not involve X. Σ^B_0 formulas are formulas without bounded quantifiers for sets but may have bounded number quantifiers. Each bounded set $X \leq t$ can be seen also as a finite binary string of size $\leq t$ which has 1 in the i-th position iff $i \in X$. When we say that a function $f(x, X)$ mapping bounded sets and numbers to bounded sets is in AC^0 or NC^1 we mean that the corresponding function on finite binary strings X and unary representation of x is in AC^0 or NC^1.

The base theory we will consider is V^0 consisting of a set of basic axioms capturing the properties of symbols in the two-sorted language and a comprehension axiom schema for Σ^B_0-formulas stating that for any Σ^B_0 formula there exists a set containing exactly the elements that satisfy the formula, cf. [7]. Further, Cook and Nguyen define theory VNC^1 as V^0 extended by the axiom that every monotone formula has an evaluation, see [7].

Theorem 1.5.2 (Cook-Nguyen [7])**.** *If $VNC^1 \vdash \forall x\ \forall X\ \exists Y\ A(x, X, Y)$ for Σ^B_0-formula A, there is a function f in uniform NC^1 such that $A(x, X, f(x, X))$ holds for any x, X.*

If $VNC^1 \vdash \forall x\ \forall X\ \exists Y\ \forall Z\ A(x, X, Y, Z)$ for Σ^B_0-formula A, there are finitely many functions $f_1, ..., f_k$ in uniform NC^1 such that $\forall x, X, Z_1, Z_2, ..., Z_k$

$$A(x, X, f_1(x, X), Z_1) \vee A(x, X, f_2(x, X, Z_1), Z_2) \vee ... \\ ... \vee A(x, X, , f(x, X, Z_1, ..., Z_{k-1}), Z_k)$$

Analogously for V^0 with the resulting functions in uniform AC^0.

$LB(SAT, n^k)$ translates to the two-sorted language as follows

$$\forall n > n_0,\ \forall C,\ \exists Y \leq n,\ \exists A \leq n,\ \forall W \leq n^k,\ \forall Z \leq n,\ [Comp(C, Y, W) \rightarrow \\ (C(Y; W) = 1 \wedge \neg SAT(Y, Z)) \vee (C(Y; W) = 0 \wedge\ SAT(Y, A))]$$

where k, n_0 are constants as before and $Comp(C, Y, W), C(Y; W) = 0/1, SAT(Y, Z)$ are defined as their first-order counterparts but function x_i is replaced by $i \in X$.

Similarly, we obtain the two-sorted $SCE(SAT, n^k), DCE(SAT, n^k)$.

Let us also specify the formalization of $LB(SAT, n^k)$ in T_{NC^1}. L_{NC^1} contains symbols for $SAT(y, z), Comp(C, y, w)$ and all the predicates we explicitly defined as $\Sigma^b_0(bit)$-formulas because they are not just p-time but in fact uniform NC^1. For simplicity, whenever we speak about $LB(SAT, n^k)$ in T_{NC^1} we mean its formalization where instead of the $\Sigma^b_0(bit)$-formulas we have the respective symbols of L_{NC^1}. Similarly for $SCE(SAT, n^k)$, $DCE(SAT, n^k)$. Therefore, $LB(SAT, n^k)$, $SCE(SAT, n^k)$ and $DCE(SAT, n^k)$ in T_{NC^1} have the form $\exists y \forall z\ A(x, y, z)$ for an open formula A (i.e. A has no quantifiers).

The situation with the provability of polynomial circuit lower bounds in weak theories like T_{NC^1} is less natural because they cannot fully reason about p-time concepts. In particular, there is a formula $LB_\exists(SAT, n^k)$ which is equivalent to $LB(SAT, n^k)$ in $S^1_2(bit)$ but not necessarily in T_{NC^1}. $LB_\exists(SAT, n^k)$ is like $LB(SAT, n^k)$ but with $LB(C, y, a)$ (defined in Section 1.3.2) expressed positively:

$$LB_\exists(SAT, n^k) : \\ \forall 1^n > n_0,\ \forall C,\ \exists y, a, w,\ |a| < |y| = n,\ |w| \leq n^k,\ \forall z,\ |z| < |y|, \\ [\neg Circ(C, y, w) \vee \\ (C(y; w) = 0 \wedge SAT(y, a)) \vee (C(y; w) = 1 \wedge \neg SAT(y, z))]$$

Analogously define $DCE_\exists(SAT, n^k)$, $SCE_\exists(SAT, n^k)$ and their two-sorted and L_{NC^1} formulations.

By the witnessing theorem above, if $T_{NC^1} \vdash LB(SAT, n^k)$, then $LB(SAT, n^k)$ has an NC^1 S-T protocol with $O(1)$ rounds which is S-T protocol with $O(1)$ rounds and S in uniform NC^1. If $T_{NC^1} \vdash LB_\exists(SAT, n^k)$, then $LB_\exists(SAT, n^k)$ has an NC^1 S-T protocol with $O(1)$ rounds which is defined analogously as for $LB(SAT, n^k)$ but with S producing also computations w of given circuits. As $DCE_\exists(SAT, n^k)$ has the form $\exists y A(x, y)$ for an open A in L_{NC^1}, its provability in T_{NC^1} implies $DCE_\exists(SAT, n^k) \in NC^1$. Here again, $DCE_\exists(SAT, n^k) \in NC^1$ is defined as $DCE(SAT, n^k) \in NC^1$ but with the witnessing algorithm producing also computations w of given circuits. Analogously for theories V^0, VNC^1 .

1.6 Unprovability of circuit lower bounds in subtheories of PV_1

To prove that VNC^1 or T_{NC^1} do not prove $LB(SAT, n^k)$ it suffices to show that $LB(SAT, n^k)$ has no S-T protocol with $O(1)$ rounds where S is in uniform NC^1. For the unprovability of $LB_\exists(SAT, n^k)$ it however suffices to refute the existence of S-T protocols with $O(1)$ rounds where S $\in NC^1$ produces w's (computations of given circuits) itself. This is essentially trivial since in such case, NC^1 circuits could produce computations of general circuits of similar size:

Proposition 1.6.1. *$LB(SAT, n^{k+1}) \notin NC^1$, $DCE_\exists(SAT, n^{k+1}) \notin NC^1$ and $LB_\exists(SAT, n^{k+1})$ has no NC^1 S-T protocol with $poly(n)$ rounds unless $SIZE(n^k) \subseteq NC^1$. Unconditionally, for any big enough k, $LB(SAT, n^k) \notin AC^0$, $DCE_\exists(SAT, n^k) \notin AC^0$ and $LB_\exists(SAT, n^k)$ has no AC^0 S-T protocol with $poly(n)$ rounds.*

Proof: Assume first that $LB(SAT, n^{k+1}) \in NC^1$, i.e. there are NC^1 circuits $D_m(x)$ such that for sufficiently big n whenever $x \in \{0,1\}^m$ for $m = poly(n)$ encodes an n^{k+1}-size circuit C_n with n inputs, $D_m(x)$ outputs y, a such that

$$C_n(y) = 0 \wedge SAT(y, a) \quad or \quad C_n(y) = 1 \wedge \forall z \neg SAT(y, z)$$

Now any n^k-size circuits B_n with n inputs can be simulated by NC^1 circuits: For $b \in \{0,1\}^n$ and $z = (z_1, ..., z_n)$ denote $R[B_n, b, z]$ the circuit with n inputs z but computing as B_n on b, i.e. it does not use inputs z at all. The size of $R[B_n, b, z]$ is $(n^k + n)$. Let $E_n(b)$ be an AC^0 circuit which uses description of B_n's as advice and maps $b \in \{0,1\}^n$ to $x \in \{0,1\}^m$ encoding $R[B_n, b, z]$.

For each $b \in \{0,1\}^n$, use $D_m(E_n(b))$ to find y, a and output 0 iff $SAT(y, a)$.

Deciding $SAT(y, a)$ is by our formalization doable by constant-depth formulas. Therefore, for each b, we predict $B_n(b)$ with an NC^1 circuit.

If $LB(SAT, n^k) \in AC^0$ for sufficiently big k, we would obtain AC^0 circuits for PARITY, which is impossible.

This construction works analogously for $DCE_\exists(SAT, n^{k+1})$ and as well for $LB_\exists(SAT, n^{k+1})$. If $LB_\exists(SAT, n^{k+1})$ has an NC^1 S-T protocol, then for given n^{k+1}-size circuit C, S does not have to produce w, y, a such that w is a computation of C on input y but then T can reply 0 and S is thus eventually forced to produce a computation of circuit C which means that NC^1 S can simulate any n^k-size circuit as in the case of $LB(SAT, n^{k+1})$. □

Corollary 1.6.1. *$T_{NC^1} \nvdash DCE_\exists(SAT, n^{k+1})$ and $T_{NC^1} \nvdash LB_\exists(SAT, n^{k+1})$ unless $SIZE(n^k) \subseteq NC^1$. For any sufficiently big k, $V^0 \nvdash DCE_\exists(SAT, n^k)$ and $V^0 \nvdash LB_\exists(SAT, n^k)$.*

This simple observation does not work if we want to refute that $LB(SAT, n^k)$ has NC^1 S-T protocols because T can send to S a computation of the artificially attached circuit. Indeed by Proposition 1.4.3, $LB(SAT, n^k)$ has a uniform AC^0 S-T protocol with $poly(n)$ rounds under a plausible assumption.

We can however show that $LB(SAT, n^k)$ has no NC^1 S-T protocols with $O(1)$ rounds under a hardness assumption. To show this we will use an interpretation

of suitable NW-generators as p-size circuits which is due to Razborov [20] and Krajíček's proof of a hardness of certain NW-generators for theory T_{PV} which is defined as T_{NC^1} but in the language containing names for all p-time algorithms, cf. [15]. Actually, the proof of the following theorem seems to be a natural modification of the proof of Proposition 1.6.1.

Theorem 1.6.1. *Let $c \geq 2, k \geq 1$. If there is $f \in SIZE(n^k)$ such that for all formulas F_n of size $2^{O(n^{1/c})}$, $P_{x\in\{0,1\}^n}[F_n(x) = f(x)] < 1/2 + 1/2^{O(n^{1/c})}$ for infinitely many n's, then $LB(SAT, n^{4kc})$ has no NC^1 S-T protocol with $O(1)$ rounds.*

To prove the theorem we will use Nisan-Wigderson (NW) generators with specific design properties. Let $A = \{a_{i,j}\}_{j=1,...,n}^{i=1,...,m}$ be an $m \times n$ 0-1 matrix with l ones per row. $J_i(A) := \{j \in \{1, ..., n\}; a_{i,j} = 1\}$ and $f : \{0,1\}^l \mapsto \{0,1\}$. Then define NW-generator based on f and A, $NW_{f,A} : \{0,1\}^n \mapsto \{0,1\}^m$ as

$$(NW_{f,A}(x))_i = f(x|J_i(A))$$

where $x|J_i(A)$ are x_j's such that $j \in J_i(A)$.

For any $c \geq 2$, Nisan and Wigderson [17] constructed $2^n \times n^{2c}$ 0-1 matrix A with n^c ones per row which is also (n, n^c)-design meaning that for each $i \neq j$, $|J_i(A) \cap J_j(A)| \leq n$. Moreover, the matrix A has such a property that for big enough n there are n^{2c}-size circuits which given $i \in \{0,1\}^n$ compute the set $J_i(A)$, more precisely, given input $i \in \{0,1\}^n$ they output n^c indices in $J_i(A)$ where each index is described by $2c \log n$ output bits. Therefore, as it was observed by Razborov [20], if f is in addition computable by n^k-size circuits, for any $x \in \{0,1\}^{n^{2c}}$, $(NW_{f,A}(x))_y$ is a function on n inputs y which is for sufficiently big n computable by circuits of size n^{4kc}.

To see this, note that for any given $y \in \{0,1\}^n$ an n^{2c}-size circuit produces n^c indices of $J_y(A)$ where the r-th index is described by $2c \log n$ bits $J_{r,1}, ..., J_{r,2c\log n}$. Then a circuit of size $\leq n^c n^{2c}(2Kc \log n + K)$, with an absolute constant K, which has the form

$$\bigwedge_{r\in\{1,...,n^c\}} \bigwedge_{s\in\{0,1\}^{2c\log n}} ((\bigwedge_{t\in\{1,...,2c\log n\}} (J_{r,t} \leftrightarrow s_t)) \rightarrow (r\text{-th output bit} \leftrightarrow x_s))$$

specifies n^c bits in x on which an n^{ck}-size circuit computes $f(x|J_y(A))$. Since $n^{2c} + n^{kc} + n^c n^{2c}(2Kc \log n + K) < n^{4kc}$ for $k \geq 1$ and big enough n, the whole circuit computing $(NW_{f,A}(x))_y$ has size $< n^{4kc}$.

Proof(of Theorem 1.6.1): Let $f \in SIZE(n^k)$ and A be a $2^n \times n^{2c}$ (n, n^c)-design defined above so for any sufficiently big n and any x, $(NW_{f,A}(x))_y$ can be computed from y by an n^{4kc}-size circuit. Assume that $LB(SAT, n^{4kc})$ has an NC^1 S-T protocol with $O(1)$ rounds. In particular, for sufficiently big n and each n^{4kc}-size circuit $C(y)$ computing $(NW_{f,A}(x))_y$, S either finds out the value of $C(y_1)$ by deciding (in AC^0) $SAT(y_1, a_1)$ for y_1, a_1 it produced itself or T will send to S counterexamples w_1, b_1 such that

$$(C(y_1; w_1) = 1 \vee \neg SAT(y_1, a_1)) \wedge (C(y_1; w_1) = 0 \vee SAT(y_1, b_1))$$

In the latter case, S continues with its second try y_2, a_2. After at most $t \leq l$ rounds for some fixed constant l, S will successfully predict $C(y_t)$.

Let $E_{n^{2c}}(x)$ be AC^0 circuits mapping $x \in \{0,1\}^{n^{2c}}$ to a description of an n^{4kc}-size circuit with n inputs y computing the function $(NW_{f,A}(x))_y$, so $E_{n^{2c}}$ just substitutes given x to a description of $(NW_{f,A}(x))_y$ which is otherwise fixed. Moreover, without loss of generality, for any y, x_1, x_2 such that $x_1|J_y(A) = x_2|J_y(A)$ the computation of $E_{n^{2c}}(x_1)$ on input y is the same as the computation of $E_{n^{2c}}(x_2)$ on input y up to the specific bits of x_1 resp. x_2 where x_1 and x_2 differ. We denote the invariant part of the computation of $E_{n^{2c}}(x)$ on input y as its *relevant* part. To be precise, it is the computation of $E_{n^{2c}}(x)$ on input y with bits $x_j, j \notin J_y(A)$ replaced by 0's.

We will consider our S-T protocol only on inputs of the form $E_{n^{2c}}(x)$.

Krajíček [15] showed that if f is in NP∩coNP with unique witnesses such S-T protocol allows us to approximate f by a p-size circuit. We will inspect that his proof works also for f in *P/poly* and NC^1 S-T protocols. In addition we will assume that T in our S-T protocol operates as follows: whenever S outputs y with some a, T answers with the lexicographically first assignment b satisfying y and the unique relevant part w of the computation of given circuit on input y. If there is no such b, T replies with a string of zeroes instead of b (and the unique relevant part w of the computation of given circuit on input y). This should replace the uniqueness property assumed in [15]. Note that S can recover the full computation of given circuit on input y just from its relevant part.

For $u \in \{0,1\}^{n^c}$ and $v \in \{0,1\}^{n^{2c}-n^c}$ define $r_y(u,v) \in \{0,1\}^{n^{2c}}$ by putting bits of u into positions $J_y(A)$ and filling the remaining bits by v (in the natural order). For each x there is a trace $tr(x) = y_1, a_1, ..., y_t, a_t, t \leq l$ of the S-T communication.

Claim 1. *There is a trace $Tr = y_1, a_1, ..., y_t, a_t, t \leq l$ and $p \in \{0,1\}^{n^{2c}-n^c}$ such that $Tr = tr(r_{y_t}(u,p))$ for at least a fraction of $2/(3(2^{2n}))^t$ of all u's.*

Tr and p can be constructed inductively. There are at most 2^{2n} pairs y_j, a_i, hence there is y_1, a_1 such that at least $1/2^{2n}$ traces begin with it. Either there is $p \in \{0,1\}^{n^{2c}-n^c}$ such that $y_1, a_1 = tr(r_{y_1}(u,p))$ for at least $2/(3(2^{2n}))$ of all u's or we can find y_2, a_2 such that at least $1/(3(2^{2n})^2)$ traces begin with y_1, a_1, y_2, a_2. For the induction step assume we have a trace $y_1, a_1, ..., y_i, a_i$ such that at least $1/(3^{i-1}(2^{2n})^i)$ traces begin with it. Either there is $p \in \{0,1\}^{n^{2c}-n^c}$ such that $y_1, a_1, ..., y_i, a_i = tr(r_{y_i}(u,p))$ for at least $2/(3^i(2^{2n})^i)$ of all u's or we can find y_{i+1}, a_{i+1} such that at least $1/(3^i(2^{2n})^{i+1})$ traces begin with $y_1, a_1, ..., y_{i+1}, a_{i+1}$. This proves the claim.

Fix now Tr and p from the previous claim.

Because A is (n, n^c)-design, for any row $y \neq y_t$ at most n x_j's with $j \in J_y(A)$ are not set by p. Hence there are at most 2^n assignments z to x_j's with $j \in J_y(A)$ not set by p. For each such z let w_z, b_z be the T's advice after S outputs y, a_i on any x containing the assignment given by z and p. By our choice of T, b_z depends only on y and w_z is uniquely determined by z (and p which is fixed). Let $Y_y, y \neq y_t$ be the set of all these witnesses w_z, b_z for all possible z's. The size of each such Y_y is $2^{O(n)}$ (including the sizes of the witnesses w_z, b_z).

Now we define a formula F that attempts to compute f and uses as advice Tr, p and some t sets Y_y. For each $u \in \{0,1\}^{n^c}$ produce $r_{y_t}(u,p)$ (this is in AC^0). Let V be the set of those inputs u for which $tr(r_{y_t}(u,p))$ either is Tr or extends Tr and let U be the complement of V. Define d_0 to be the majority

value of f on U. Then use S to produce y'_1, a'_1. If y'_1, a'_1 is different from Tr output d_0. Otherwise, find the unique T's advice in Y_{y_1}. Again, this is doable by a constant depth formula of size $2^{O(n)}$ which has $poly(n)$ output bits. It has the form $\bigwedge_{z\in\{0,1\}^n}(z = r_{y_t}(u,p)|(J_{y_1}(A) \cap J_{y_t}(A)) \rightarrow output = w_z \in Y_{y_1})$. In the same manner continue until S produces y'_t, a'_t. If y'_t, a'_t differs from Tr output d_0. Otherwise, output 0 iff $SAT(y_t, a_t)$.

F is a formula with n^c inputs and size $2^{O(n)}$ because producing $r_{y_t}(u,p)$ is in AC^0, searching for T's advice in Y_{y_i}'s is doable by constant-depth $2^{O(n)}$-size formulas, S is in NC^1 and the structure of S-T protocol can be described by a constant-depth formula of size $n^{O(1)}$:

$$
\begin{array}{l}
(S(x) \notin Tr \rightarrow output = d_0) \wedge (S(x) \in Tr \rightarrow \\
\quad ((S(x, w_z, b_z) \notin Tr \rightarrow output = d_0) \wedge (... \\
\qquad (S(x, w_1, b_1, ..., w_t, b_t) \notin Tr \rightarrow output = d_0) \wedge \\
\qquad (S(x, w_1, b_1, ..., w_t, b_t) \in Tr \rightarrow (output = 0 \leftrightarrow SAT(y_t, b_t)))...)))
\end{array}
$$

By the choice of Tr, for at least a fraction $2/(3(2^n))^t$ of all $u \in \{0,1\}^{n^c}$, we have that $u \in V$ and F will successfully predict $f(u)$. Moreover, by the choice of Tr in the proof of Claim 1, at most $1/(3(2^n))^t$ of all traces $tr(r_{y_t}(u,p))$ properly extend Tr. Since d_0 is the correct value on at least half of $u \in U$, F will successfully predict $f(u)$ on at least half of U, half of V and $1/2(1/(3^t 2^{nt}))$ of all u's. That is, $P_{u\in\{0,1\}^{n^c}}[F(u) = f(u)] \geq 1/2 + 1/(3^t 2^{nt+1})$. □

Corollary 1.6.2. *$T_{NC^1} \nvdash LB(SAT, n^{4kc})$ and $VNC^1 \nvdash LB(SAT, n^{4kc})$ where $k \geq 1, c \geq 2$ unless for each $f \in SIZE(n^k)$ there are formulas F_n of size $2^{O(n^{1/c})}$ such that for sufficiently big n's, $P_{x\in\{0,1\}^n}[F_n(x) = f(x)] \geq 1/2 + 1/2^{O(n^{1/c})}$.*

To obtain an unconditional unprovability of circuit lower bounds we can use Hastad's lower bound for constant depth circuits computing the parity function.

Theorem 1.6.2 (Hastad [8])**.** *For any depth d circuits C_n of size $2^{n^{1/(d+1)}}$ and large enough n, $P_{x\in\{0,1\}^n}[C_n(x) = PARITY(x)] \leq 1/2 + 1/2^{n^{1/(d+1)}}$.*

If $V^0 \vdash LB(SAT, n^k)$, then $LB(SAT, n^k)$ has an AC^0 S-T protocol with $O(1)$ rounds so the resulting formula F in the proof of Theorem 1.6.1 would be actually a constant-depth circuit and PARITY could be approximated by constant depth circuits of size $2^{O(n^{1/c})}$ with advantage $1/2^{O(n^{1/c})}$. This is not enough for the contradiction with Hastad's theorem. Nevertheless, it is sufficient if we replace polynomial circuit lower bounds $LB(SAT, n^k)$ by quasi polynomial lower bounds $LB(SAT, n^{\log n})$:

$$
\begin{array}{l}
\forall m > n_0,\ \forall C,\ \exists y, a,\ |a| < |y| = n,\ \forall w,\ |w| \leq n^{\log n} = m, \\
\quad [Comp(C, y, w) \rightarrow \\
\qquad (C(y; w) = 0 \wedge SAT(y, a)) \vee (C(y; w) = 1 \wedge \forall z \neg SAT(y, z))]
\end{array}
$$

where n is the number of inputs to C and m represents $n^{\log n}$ (or simply $|m| = |n|^2$).

If $V^0 \vdash LB(SAT, n^{\log n})$, then in the proof of Theorem 1.6.1 we can use instead of n^{4kc}-size circuits of the form $(NW_{f,A}(x))_y$ with $x \in \{0,1\}^{n^{2c}}$ say $n^{4k\lfloor \log\log n\rfloor}$-size circuits $(NW_{f,A}(x))_y$ with x of size $n^{2\lfloor \log\log n\rfloor}$ and big enough k. The proof

works for big enough n even if $c = \log\log n$. The size of the resulting constant-depth circuit F is then $2^{O(n^{1/\lfloor \log\log n \rfloor})}$ with advantage $1/2^{O(n^{1/\lfloor \log\log n \rfloor})}$ contradicting Hastad's theorem.

Corollary 1.6.3. $V^0 \nvdash LB(SAT, n^{\log n})$.

Acknowledgement

I would like to thank Jan Krajíček, Albert Atserias, Sam Buss and an anonymous reviewer for many useful discussions, comments and suggestions. This research was supported by grant GAUK 5732/2012 and partially by grants IAA100190902 of GA AV ČR and SVV-2012-267317. A part of this research was done while I was a visiting fellow at the Isaac Newton Institute in Cambridge in Spring 2012 supported by grant N-SPP 2011/2012.

Bibliography

[1] Buss S.R.; Bounded Arithmetic, Bibliopolis, Naples, 1986.

[2] Buss S.R.; The Polynomial Hierarchy and Intuitionistic Bounded Arithmetic, Structure in Complexity, Lecture Notes in Computer Science #223, 1986, 77-103.

[3] Buss S.R.; Bounded arithmetic, cryptography and complexity, Theoria, 63 (1997), 147-167.

[4] Cook S.A.; Feasibly constructive proofs and the propositional calculus, Proceedings of the 7th Annual ACM Symposium on Theory of Computing, ACM Press, 1975, 83-97.

[5] Cook S.A., Krajíček J.; Consequences of the Provability of NP$\subseteq$P/poly, J. of Symbolic Logic, 72 (2007), 1353-1357.

[6] Cook S.A., Mitchell D.G.; Finding Hard Instances of the Satisfiability problem: A survey, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1997.

[7] Cook S.A., Nguyen P.; Logical Foundations of Proof Complexity, Cambridge University Press, 2010.

[8] Hastad J.; Computational limitations for small depth circuits, PhD thesis, M.I.T. press, 1986.

[9] Jeřábek E.; Dual weak pigeonhole principle, Boolean complexity and derandomization, Annals of Pure and Applied Logic, 129 (2004) 1-37.

[10] Jeřábek E.; Approximate counting in bounded arithmetic, Journal of Symbolic Logic, 72 (2007), 959-993.

[11] Kent C.F., and Hodgson B.R.; An arithmetic characterization of NP, Theoretical Comput. Sci., 21 (1982), 255-267.

[12] Krajíček J.; No counter-example interpretation and interactive computation, Logic from Computer Science, 21 (1992), 287-293.

[13] Krajíček J.; Bounded arithmetic, propositional logic, and complexity theory, Cambridge University Press, 1995.

[14] Krajíček J.; Extensions of models of PV, Logic Colloquium '95, ASL Springer Series Lecture Notes in Logic, 11 (1998), 104-114.

[15] Krajíček J.; On the proof complexity of the Nisan-Wigderson generator based on NP$\cap$coNP function, J. of Mathematical Logic, 11 (2011), 11-27.

[16] Krajíček J., Pudlák P., Takeuti G.; Bounded arithmetic and the polynomial hierarchy, Annals of Pure and Applied Logic, 52 (1991), 143-153.

[17] Nisan N., Wigderson A.; Hardness vs. Randomness, J. Comput. System Sci., 49 (1994), 149-167.

[18] Pudlák P.; Some relations between subsystems of arithmetic and complexity theory, Proc. Conf. Logic from Computer Science, 21 (1992), 499-519.

[19] Razborov A.A; Unprovability of Lower Bounds on the Circuit Size in Certain Fragments of Bounded Arithmetic, Izvestiya of the Russian Academy of Science, 59 (1995), 201-224.

[20] Razborov A.A; Pseudorandom Generators Hard for k-DNF Resolution and Polynomial Calculus, preprint, 2002-2003.

[21] Stockmayer L.J.; The polynomial-time hierarchy, Theoretical Comput. Sci., 3 (1976), 1-22.

[22] Wrathall C.; Complete sets and the polynomial-time hierarchy. Theoretical Comput. Sci., 3 (1976), 23-33.

2. Attachment:

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.

2.1 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 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 2.2.

Generally, it would be interesting to arrive at a complexity-theoretic statement, not necessarilly 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 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 2.4 as it gives us the opportunity to introduce some notions. For more details concerning the list see Section 2.3.

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. Nevertheless, although the proof of the exponential version is used in the proof of the PCP theorem, its formalization in APC_1 is different. In PV_1 it is applied so that we need to reason only about sets of constant size, while in APC_1 it is performed 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 2.2 we describe general properties of our formalizations and define theories of bounded arithmetic in which these formalizations take place. In Section 2.3 we discuss theorems that have been already formalized in bounded arithmetic as well as the new ones obtained in this paper. Section 2.4 illustrates a formalization of the Cook-Levin theorem in PV_1. In Section 2.5 we prove the exponential PCP theorem in APC_1. Section 2.6 formalizes pseudorandom constructions in PV_1 which are then used in Section 2.7 to formalize the PCP theorem in PV_1.

Theory	Theorem	Reference
PV_1	Cook-Levin's theorem	Section 2.4
	(n, d, λ)-graphs	Section 2.6
	the PCP theorem	Section 2.7
$PV_1 + WPHP(PV_1)$	PARITY $\notin AC^0$	[18]
APC_1	BPP, ZPP, AM,...	[15]
	Goldreich-Levin's theorem	[11]
	the exponential PCP theorem	Section 2.5
$HARD_\epsilon$	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.

2.2 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 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 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 necessarilly 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 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].

2.2.1 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 (described below). 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 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 in 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.

Definition 2.2.1. *A function f is defined from functions g, h_0, h_1 and l by limited recursion on notation if:*

1. $f(x,0) = g(x)$
2. $f(x, s_i(y)) = h_i(x, y, f(x,y))$, *for* $i = 0, 1$
3. $f(x,y) \leq l(x,y)$

where $s_0(y), s_1(y)$ are functions adding 0, resp. 1, to the right of the binary representation of y, i.e. $s_0(y) := 2y$, $s_1(y) = 2y + 1$.

Theorem 2.2.1 (Cobham [7])**.** *The set of polynomial time functions is the smallest set of functions containing constant 0, functions $s_0(y), s_1(y), x\#y$, and closed under:*

1. *permutation and renaming of variables,*
2. *composition of functions,*
3. *limited recursion on notation.*

Definition 2.2.2 (Cook [8])**.** *We simultaneously define function symbols of rank k and PV-derivations of rank k, $k = 0, 1, ...$ The language of PV will then consist of all function symbols of any rank, and a PV-derivation will be a PV-derivation of any rank.*

1. Function symbols of rank 0 are constant 0, unary $s_0(y), s_1(y)$ and $Tr(x)$; and binary $x \frown y, x\#y$, and $Less(x,y)$

2. Defining equations of rank 0 are:

$Tr(0) = 0$
$Tr(s_i(x)) = x$, $i = 0, 1$
$x \frown 0 = x$
$x \frown (s_i(y)) = s_i(x \frown y)$, $i = 0, 1$
$x\#0 = 0$
$x\#s_i(y) = x \frown (x\#y)$, $i = 0, 1$
$Less(x, 0) = x$
$Less(x, s_i(y)) = Tr(Less(x,y))$, $i = 0, 1$

$Tr(x)$ deletes the rightmost bit, $x \frown y$ is the concatenation, $x\#y$ is $|y|$ concatenated copies of x, and $Less(x,y)$ is x with $|y|$ right bits deleted

3. PV rules are as follows. Let $t, u, v, t_1, u_1, ..., t_k, u_k, f, f_1, f_2$ be function symbols.

R1. from $t = u$ derive $u = t$
R2. from $t = u, u = v$ derive $t = v$
R3. from $t_1 = u_1, ..., t_k = u_k$ derive $f(t_1, ..., t_k) = f(u_1, ..., u_k)$
R4. from $t = u$ derive $t(x/v) = u(x/v)$

R5. Let $E_1, ..., E_6$ be the equations (1-3) from the definition of the limited recursion on notation: three for f_1 and three for f_2 in place of f. Then from $E_1, ..., E_6$ PV can derive $f_1(x, y) = f_2(x, y)$

4. PV-derivations of rank k are sequences of equations $E_1, ..., E_t$ in which every function symbol is of rank $\leq k$ and every E_i is either a defining equation of rank $\leq k$ or derived from some earlier equations by one of the PV-rules

5. Let t be a term consisting of function symbols of rank $\leq k$. Then f_t is a function symbol of rank $k + 1$ and $f_t = t$ is a defining equation of rank $k + 1$.

6. Other function symbols of rank $k + 1$ are obtained as follows. Whenever g, h_0, h_1, l_0, l_1 are function symbols of rank $\leq k$ and π_0, π_1 are PV-derivations of rank k of equality $Less(h_i(x, y, z), z \frown l_i(x, y)) = 0$ then $f = f_{(g,h_0,h_1,l_0,l_1,\pi_0,\pi_1)}$ is a function symbol of rank $k + 1$, and the equations defining f from g, h_i by limited recursion on notation are defining equations of rank $k + 1$.

PV_1 is a theory in the first-order predicate logic which consists of all equations provable in PV but has also a form of induction axiom: For an open formula $\psi(x)$ define a function $h(b, y)$ by

(a) $h(b, 0) = (0, b)$
(b) if $h(b, \lfloor u/2 \rfloor) = (x, y)$ and $u > 0$ then

$$\begin{aligned} h(b, u) &:= (\lceil x + y/2 \rceil, y) \text{ if } \lceil x + y/2 \rceil < y \wedge \psi(\lceil x + y/2 \rceil) \\ &:= (x, \lceil x + y/2 \rceil) \text{ if } x < \lceil x + y/2 \rceil \wedge \neg\psi(\lceil x + y/2 \rceil) \\ &:= (x, y) \text{ otherwise} \end{aligned}$$

Then PV_1 contains the universal axiom

$$(\psi(0) \wedge \neg\psi(b) \wedge h(b, b) = (x, y)) \rightarrow (x + 1 = y \wedge \psi(x) \wedge \neg\psi(y))$$

Note that PV_1 is a universal theory.

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

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

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,

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

In Proposition 2.6.9, 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. For any $\Pi^b_1(PV)$-formula A:

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

2.2.2 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

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

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

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 $n, m \in Log$, $C : 2^n \rightarrow 2^m$ 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 2.2.3 (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 2.2.4 (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 2.2.5 (in PV_1)**.** *Let $f : 2^k \mapsto 2$ be a truth-table of a Boolean function (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 2.2.1 (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 in definition 2.2.2. This means that the language of $PV_1(\alpha)$, denoted also $PV(\alpha)$, contains symbols for all p-time oracle algorithms.

Definition 2.2.6 (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 2.2.2 (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 2.2.7 (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 2.2.2. *(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 2.2.3. *(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.

2.3 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.

2.3.1 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 2.3.1 (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 2.4.

2.3.2 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 explicitely 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 2.3.1 (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 signallizing 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.

Definition 2.3.2 (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 2.3.2 (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 $T^1_2 + rWPHP(PV_2)$. Here, 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 $dWPHP(f)$.

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 $|x|$.

Theorem 2.3.3 (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 $x \in L \Leftrightarrow P(x)$.*

2.3.3 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

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

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 but distinct over PV_1, see [16]. However, the theory $PV_1 + dWPHP(PV)$ is $\forall\Sigma^b_1$-conservative over $PV_1 + \{\exists y < a \# a \ \forall x < a, \ f(x) \neq y |$ for PV-functions $f\}$ (noted in Jeřábek [16] as a corollary of earlier results).

Theorem 2.3.4 (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 explicitely 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.

2.3.4 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 2.3.3 (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 2.3.5 (Jeřábek [17])**.** *(in APC_2) Graph nonisomorphism is in AM.*

2.3.5 Cryptography

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

Theorem 2.3.6 (Dai Tri Man Le [11])**.** *(in APC_1) Let $f : \{0,1\}^n \rightarrow \{0,1\}^n$ 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

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

2.3.6 Complexity of counting

In [5], Buss, Kolodziejczyk 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 2.3.7 (Buss, Kolodziejczyk, Zdanowski [5])**.** *$APC_2^{\oplus_p P}$ proves that any $\Sigma_\infty^b(\oplus_p)$ formula defines a property in $BP{\cdot}\oplus_p P$.*

2.3.7 Derandomization

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

Theorem 2.3.8 (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 2.3.9 (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.

2.3.8 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 2.3.4. *(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 2.5.

Theorem 2.5 (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 2.3.4 such that APC_1 proves that for any $x \in \{0,1\}^n$, $n \in Log$:*

$$\exists y SAT(x,y) \rightarrow \exists k' n^{k'} \textit{size circuit } \pi \ \forall w < 2^{kn^k}, D^{\pi,w}(x) = 1$$

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

We also formalize pseudorandom constructions involving the (n, d, λ)-graphs in PV_1 but leave the presentation of these results to Section 2.6 as it would require to introduce 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 2.3.5. *(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 2.3.5 we abuse the notation and use the shortcut $D^{\pi,w}(x)$ in diferent meaning than in Definition 2.3.4. This should not lead into confusion.

Theorem 2.7 (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 2.3.5 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. Nevertheless, although the proof of the exponential version is used in the proof of the PCP theorem, its formalization in APC_1 is different. In PV_1 it is applied so that we need to reason only about sets of constant size, while in APC_1 it is performed with p-time definable sets. Hence, the APC_1 proof shows different tools to be available in low fragments of arithmetic.

2.4 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 2.4.1. *(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. Note firstly that PV can introduce functions using conditional definitions:

$f(x) := g_i(x)$ if $P_i(x), i = 0, 1$

where g_0, g_1 are functions already defined in PV and P_0, P_1 are disjoint and exhaustive open PV-formulas. This is because such P_0, P_1 define p-time truth functions which can be introduced as PV-functions P'_0, P'_1 and f can be then defined as
$f(x) := P'_0(x)g_0(x) + P'_1(x)g_1(x)$.

Now, we show that for some PV-function f, PV_1 proves $(*)$:

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

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 $f(M, x)$ 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.

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 $|y| \leq |x| \wedge x_{|y|} = i$, $i = 0, 1$

For the sake of brevity we ignored the case $|y| > |x|$, it is meant that in such case $f_{input}(x, s_i(y)) = f(x, y)$. Moreover, in the definition of f_{input}, we are again abusing the notation: $A \wedge B$ denotes a function which given A and B produces a code of the conjunction of propositional formulas encoded in A and B. In definitions of other functions in this proof we will use similarly also more complex propositional formulas.

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 $(T_{t',l',m'} \rightarrow \neg T_{t',l'',m'})$ for all $l' \neq l''$ and t', m' such that $[t', l', m']$,

$[t', l'', m'] \leq [t, l, m]$. This guarantees that cell $t' \leq t$ contains only one symbol at step $m' \leq m$.

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

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 $t', t'' \leq t; l', l'' \leq l; m' \leq m$

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, $f(M, x) := f_{trans}(M, x, |x|^k) \wedge \bigvee_{r \in F, t \leq |x|^k} Q_{r,t}$.

This defines a PV-function f. To see that $(*)$ holds, given M, x, w, we define y as follows:

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

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

□

2.5 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 2.3.4 introducing the predicate $D^{\pi,w}(x)$. The algorithm D will represent the so called verifier of probabilistically checkable proofs π. The verifier is standardly 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 2.5 as the right formulation of the exponential PCP theorem in APC_1.

Theorem 2.5 (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 2.3.4 such that APC_1 proves that for any $x \in \{0,1\}^n$, $n \in Log$:*

$$\exists y SAT(x,y) \rightarrow \exists k' n^{k'} \textit{size circuit } \pi \; \forall w < 2^{kn^k}, D^{\pi,w}(x) = 1$$

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

Proof. 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:

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

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

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 $\exists y SAT(x,y)$, 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:

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

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 2.5.1.

Proposition 2.5.1 (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 2 (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 2.2.3 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. $g_l = WH(u \otimes u)$.

Claim 3. *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 $\{r_2 \in 2^{n_1}|Ur_2 \neq Wr_2\}$.

Similarly, if $U \neq W$, 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 2.2.3 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

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

which is $\preceq_{28\epsilon} 15/16(2^{2n_1})$ by Claim 3, 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 2.2.3 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.

2.5.1 Test of linearity in APC_1

In this section we prove Proposition 2.5.1 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].

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 $x < 2^n$, $P_x := Pr_{y<2^x}[g_\epsilon(x) = f(y) + f(x+y)]_\epsilon \geq 1/2$. Hence, $g_\epsilon(x)$ is the major value of the expression $f(y) + f(x+y)$ for possible y's.

We will now derive three claims that can be combined into a proof of Proposition 2.5.1.

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 2.2.2*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, $2^n/2 \preceq_\epsilon T_x$.

Hence, by Proposition 2.2.3 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 2.2.2 *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 Proposition 2.2.3 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 2.2.2 *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 2.2.3 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 2.2.2 *i*) we have $2^{2n} - (C \cap 2^{2n}) \cup (D \cap 2^{2n}) \preceq_0 (A \cup B) \cap 2^{2n}$.

By 2.2.3 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 2.5.1. 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, $Pr_{x,y}[f(x+y) = f(x) + f(y)]_\epsilon \preceq_{11\epsilon+13\epsilon^2+2\epsilon^3} 1/2 + p/2$.

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].

2.6.1 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 2.6.3) 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 and only if $x = (x_1, ..., x_n)$ and each element 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 property: If $r = \frac{c \cdot c \cdot e}{d \cdot d \cdot f} \in Q/m$ where c, d have the form $\Sigma_i 2^i y_i, y_i \in \{0, 1\}$, 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 $c \in [L^7 m^6]$ such that $SQRT(r)$ which is $\frac{ec}{2fLqm^4} \in Q/(Lm)^7$ satisfies $0 \leq (\frac{ec}{2fLqm^4})^2 - r \leq \frac{1}{L}$.

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 the product $\langle x, x\rangle$ 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^b_1. 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 $||x|| = 1$, $||Ax|| \leq y$ are computable in time $n^{O(1)}$.

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 2.6.1. *(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,

$$||x||^2 = (\tfrac{\langle x,y\rangle}{\langle y,y\rangle})^2 ||y||^2 + ||z||^2 = \tfrac{\langle x,y\rangle^2}{||y||^2} + ||z||^2 \geq \tfrac{\langle x,y\rangle^2}{||y||^2}.$$ □

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 2.6.2. *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
$||Ax||^4 = \langle Ax, Ax\rangle^2 = \langle A^2 x, x\rangle^2 \leq ||A^2 x||^2$ where A^2 is the random-walk matrix of G^2, so also $||A^2 x||^2 \leq n$ and $||Ax||^4 \leq n$. This shows that

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

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 2.6.3. *(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 2.6.4. *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, we have $A^{l-1} \in Q^{n\times n}/d^{l-1}$ 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 2.6.3:

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

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

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

□

2.6.2 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 2.6.5. *(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, we have $|\frac{x_i}{||x||'}| \leq 1$ so we can find c_i (i.e. PV_1 can prove the existence of c_i) 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 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 2.6.6. *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 2.6.5, 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,

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

2.6.3 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.

Definition 2.6.1. *(in PV_1) If $A = \{a_{i,j}\}_{i,j=1,\ldots,n}$ 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 that in PV_1 for every two $n \times n$ matrices A, B and $x \in Q^n/m$ where $n \in Log$, Cauchy-Schwarz inequality implies,

$$||(A + B)x||^2 = \langle (A + B)x, (A + B)x \rangle = ||Ax||^2 + 2\langle Ax, Bx \rangle + ||Bx||^2 \leq$$
$$\leq ||Ax||^2 + 2||Ax||||Bx|| + ||Bx||^2 \leq (||Ax|| + ||Bx||)^2$$

and so $||(A + B)x|| \leq ||Ax|| + ||Bx|| + 1/L^{1/2}$.

Proposition 2.6.7. *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 2.6.6 A is $(1-\lambda)J_n + \lambda C$ 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 distributions $(A+B) \otimes C = A \otimes C + B \otimes C$ and $A \otimes (B+C) = A \otimes B + A \otimes C$, for any $x \in Q^{nn'}/(Lnn')^{(Lnn')^L}$ we have $(*)$:

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

If $\Sigma_i x_i = 0$, then $J_n \otimes J_{n'}x = 0$. Moreover, if $x \in Q^n/(Ln)^{(Ln)^L}$, then $||J_n x||^2 = \frac{1}{n}(\Sigma_i x_i)^2 \leq ||x||^2$ 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 $A \in Q^{n \times n}/(2L^2 n^5 d)$, $n' \times n'$ matrix $B \in Q^{n' \times n'}/(2L^2 (n')^5 d)$ 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 $j \in [n]$,

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

Then, $||(A \otimes B)x||^2 = \Sigma_{i \in [n']} ||Ax^i||^2$ and as by Proposition 2.6.5 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 $j \in [n]$,

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

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

□

2.6.4 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 diagonal.

This means that $G \oslash G'$ has a copy of G' for every vertex in G (including edges) 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 2.6.8. *(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 2.6.8, 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 $(G \oslash H)^3$ 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

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

By Proposition 2.6.6, $B = \delta J + (1-\delta)C$ for some C with $||C|| \leq 1$ and $D \times D$ all $1/D$ matrix J. Therefore,

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

Since $||C||, ||I|| \leq 1$, 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 2.6.7. 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 $\delta \in [0,1]$,

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

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 $x, ||x|| = 1$,

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

It remains to observe that $(I \otimes J)\hat{A}(I \otimes J) = A \otimes J$. Then, by Proposition 2.6.7, for any $x, ||x|| = 1$ such that Σ_i, x_i (and so $Jx = 0$) we have:

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

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

2.6.5 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 2.6.3 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 2.6.9. *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 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].
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 2.6.7, we have $\lambda(G \otimes G) \leq 2/2^e$. Hence, by Proposition 2.6.8,

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

and $\lambda(((G \otimes G) \oslash H)^b) \leq 1/2^e$.

The last inequality is a consequence of the fact that the assumption $\lambda(G) \leq \lambda$ implies $\lambda(G^b) \leq \lambda^b(1 + 4/L) + 5d^b/L^{1/2}$. To see this, note that as in Proposition 2.6.4, assuming $\lambda(G) \leq \lambda$, for any $x \in Q^n/((Ln^7)^n n)$ 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. If $x \notin Q^n/((Ln^7)^n n), ||x|| = 1$, $\langle x, 1 \rangle = 0$, we can approximate x by vector $c \in Q^n/((Ln^7)^n n)$: for each i, $|x_i| \leq 1$ so we can find $c_i \in Q/(Ln^7)$ such that $|x_i - c_i| \leq 1/(Ln^6)$ and $\langle c, 1 \rangle = 0$. Then $||c||^2 \leq 1 + 3/(Ln^5)$ and for each j, $|A^b_{j,i} x - A^b_{j,i} c| \leq d^b/(Ln^6)$. Hence, $|(A^b x)_j - (A^b c)_j| \leq d^b/(Ln^5)$. Since $(A^b x)_j, (A^b c)_j \leq 3d^b n$, we have $|||A^b x||^2 - ||A^b c||^2| \leq 12d^{2b}/(Ln^3)$ and $||A^b x||^2$ is at most $\lambda^{2b}(||c||^2 + 1/L) + 12d^{2b}/L \leq \lambda^{2b}(1 + 4/L) + 12d^{2b}/L$. Thus, $||A^b x|| \leq \lambda^b(1 + 4/L) + 5d^b/L^{1/2}$.

□

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

2.7 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 2.3.5. 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 2.7.1. *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 $x \in \{0,1\}^n$:*

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

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

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, $PCP(\log n, 1) \subseteq NP$, can be stated in PV_1 as Theorem 2.7.

Theorem 2.7 (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 2.3.5 such that PV_1 proves that for any $n \in Log$ and $x \in \{0,1\}^n$, $n \in Log$:*

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

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

The proof is summarized at the end of this section. It is a sequence of certain reductions between the so called CSP instances so we need to start with a reformulation of Theorem 2.7 in terms of these reductions.

Definition 2.7.1 (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 $val(\phi) := max_{u \in [W]^n} \frac{\sum_{i=1}^m \phi_i(u)}{m} = 1$.

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^b_1.

Definition 2.7.2 (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 and W' 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 2.7 then follows from the following proposition.

Proposition 2.7.1. *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 2.7 from Proposition 2.7.1) 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 any $x \in \{0,1\}^n, n \in Log$ $\exists y SAT(x, y) \to val(h(x)) = 1$ 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 2.7.1 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 2.7.1 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 loosing just a factor of 3 in the gap.

Proposition 2.7.2 (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 2.7.3 (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 2.7.1 can be obtained from previous two propositions by taking $l = 6$ in Proposition 2.7.2 and $q = max\{d, 3\}$ for d from Proposition 2.7.3.

We firstly derive Proposition 2.7.3 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 2.7.4. *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
   $Pr_{w<2^{2^{n_1}}}[(\pi_1)_w = WH(u_1)(w)] \geq 0.99$ and
   $Pr_{w<2^{2^{n_1}}}[(\pi_2)_w = WH(u_2)(w)] \geq 0.99$
   for some $u_1, u_2 \in \{0,1\}^{n_1}$ such that $C(u_1, u_2) = 1$.

Proof. (of Proposition 2.7.3 from Proposition 2.7.4) 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 $U_i = (U_{i,1}, ..., U_{i,2^W})$ of 2^W binary variables. Then, for each constraint $\phi_S(u_i, u_j)$ it applies Proposition 2.7.4 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 2.7.4 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 2.7.4 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 2.7.4) PV_1 can prove the statement from Proposition 2.7.4 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 the algorithm D accepts π_1, π_2, π_3 with probability at least $1/2$, 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 2.7.4. 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 2.7.2. To do this, we will need two facts about probability:

Proposition 2.7.5. *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 2.7.5 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.

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 2.6 but with $\langle x, y \rangle := E[XY]$.

The proof of Proposition 2.7.2 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 2.7.2.

Definition 2.7.3. *(in PV_1) A $qCSP_W$ instance ϕ is nice if $q = 2$ and the following holds,*

1. Let the constraint graph of ϕ be 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. The constraint graph of ϕ satisfies $\lambda(G) \leq 0.9$

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

Proposition 2.7.6. *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 $u_1, ..., u_q$, ψ contains q constraints $\psi_{i,j}, j = 1, ..., q$ where $\psi_{i,j}(y_i, u_j)$ is true iff y_i encodes an assignment to $u_1, ..., u_q$ satisfying ϕ_i and $u_j \in \{0, 1\}$ 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 2.7.7. *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 Propositions 2.6.9 and 2.6.3 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 $S \subseteq V, |S| \leq e^t$ 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. 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. 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 2.7.8. *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 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 2.6.9 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 istance $\psi(=h(\phi))$ 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 is $3/4B + C/4$ for C the random-walk matrix of G_{e^t} and B the random walk matrix of a $3d$-regular graph. In Section 2.6.3, 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 2.6.2, 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 2.7.8 can be chosen so that it is bigger than the constant d from Proposition 2.7.7. Therefore, Propositions 2.7.6, 2.7.7 and 2.7.8 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 2.7.2 it suffices to prove the following powering proposition:

Proposition 2.7.9. *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 $\epsilon < 1/(d\sqrt{t})$,

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

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

Proof. 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 $\epsilon < 1/(d\sqrt{t})$, $val(\psi) \leq 1 - \epsilon \rightarrow val(\psi^t) \leq 1 - \epsilon\sqrt{t}/(10^6 dW^5)$.

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,

$$\begin{aligned}
&Pr_{p_1}[y_{i_1}(u_{i_j}) = \sigma y(u_{i_j})] = \\
&= \Sigma_l Pr[S_j = l]Pr_{p_1}[l(p_1) = l \wedge y_{i_1}(u_{i_j}) = \sigma y(u_{i_j})] \\
&\geq \Sigma_l Pr[S_t = l]Pr_{p_1}[l(p_1) = l \wedge y_{i_1}(u_{i_j}) = \sigma y(u_{i_j})] - 20\delta \quad \textit{by Proposition 2.7.5} \\
&\geq 1/W - 20\delta
\end{aligned}$$

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 2.7.5 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 $L > \delta\sqrt{t}$,

$$\begin{aligned}
&E[V^2] \leq E[V'^2] = E[\Sigma_{j,j'} I_j I_{j'}] = E[\Sigma_j I_j^2] + E[\Sigma_{j \neq j'} I_j I_{j'}] \\
&= \epsilon\lceil\delta\sqrt{t}\rceil + 2\Sigma_{j<j'} Pr_p[(i_j, i_{j+1}) \in F \wedge (i_{j'}, i_{j'+1}) \in F] \\
&\leq \epsilon\lceil\delta\sqrt{t}\rceil + 2\Sigma_{j<j'} Pr_{(i_j, i_{j'}) \in G^{j'-j}}[i_j \in S \wedge i_{j'} \in S] \\
&\leq \epsilon\lceil\delta\sqrt{t}\rceil + 2\Sigma_{j<j'} \epsilon d(\epsilon d + 2 \cdot 0.9^{j'-j}) \qquad \textit{by Proposition 2.6.4} \\
&\leq \epsilon\lceil\delta\sqrt{t}\rceil + 2\epsilon^2 d^2 \lceil\delta\sqrt{t}\rceil^2 + 40\epsilon d\lceil\delta\sqrt{t}\rceil \leq 50\epsilon d\lceil\delta\sqrt{t}\rceil \qquad \textit{using } \epsilon < 1/(d\sqrt{t})
\end{aligned}$$

□

This concludes our formalization of the PCP theorem in the theory PV_1. It can be briefly summarized as follows. In Theorem 2.7 we formulated the PCP theorem as a $\forall\Sigma_1^b$-formula. Thus, by $\forall\Sigma_1^b$-conservativity of S_2^1 over PV_1 we could afford to work instead in the theory S_2^1. Specifically, we used Π_1^b-LLIND induction available in S_2^1 to show that the PCP theorem is a consequence of a statement about CSP instances, Proposition 2.7.1. Then we observed that the CSP formulation of the PCP theorem is a collorary of two propositions, Gap amplification 2.7.2 and Alphabet reduction 2.7.3. The latter one was an application of the

exponential PCP theorem in a scale-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 2.7.9. The reduction to nice instances used the (n, d, λ)-graphs which we constructed in Section 2.6. Section 2.6 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.

Acknowledgement

I would like to thank Jan Krajíček for many constructive discussions during the development of the paper. 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.

Bibliography

[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., Kolodziejczyk L.A., Zdanowski K.; Collapsing Modular Counting in Bounded Arithmetic and Constant Depth Propositional Proofs, preprint (available at author's webpage), 2012.

[6] Cai J.; $S_2^P \subseteq ZPP^{NP}$, 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, to appear in Annals of Pure and Applied Logic, 2013.

[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, preprint (available at author's webpage), 2002-2003.