Circuit Lower Bounds in Bounded Arithmetics

Ján Pich[1]

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.

Keywords: bounded arithmetic, circuit lower bounds
2000 MSC: 03B70, 03D15, 03F20

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 theo-

Email address: janpich@yahoo.com (Ján Pich)

ries 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:

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

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^1_2 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 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 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 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.

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_i^p, v_j^p, v_k^p where v_i^p 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 $SAT(y,z)$:

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

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.

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 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 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.

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 3.1. *PV_1 proves the following implications*

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

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 $(SAT(y, a) \rightarrow dC(y; w) = 1) \wedge (dC(y; w) = 1 \rightarrow SAT(y, C(y)))$ for any y, a of size $|a| < |y| = n$, hence $\neg LB(SAT, n^k + Kn)$.

Next, in PV_1, if circuit C witnesses $\neg DCE(SAT, n^k)$, 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 $\neg LB(SAT, n^k)$, then it witnesses $\neg DCE(SAT, n^k)$: for any y, a of size $|a| < |y| = n$ for a big enough n, $(C(y; w) = 0 \rightarrow \neg SAT(y, a))$, $C(y(a); w) = 1 \wedge FA(a, y) \rightarrow SAT(y, a)$ and

if $C(y(a); w) = 1 \wedge PA(a, y)$ then for some b extending a $SAT(y, b)$ and thus $C(y(a1); w) = 1 \vee C(y(a0); w) = 1$. □

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.

4. Circuit Lower Bounds in $S_2^1(bit)$

In this section we observe that the provability of circuit lower bounds in $S_2^1(bit)$ 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 $S_2^1(bit)$.

Similar observations appeared already in Buss [3]. More precisely, Proposition 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 4.2.

Proposition 4.1. *If $S_2^1(bit) \vdash LB(SAT, n^k)$, then $LB(SAT, n^k)$ has an S-T protocol with $poly(n)$ rounds. If $S_2^1(bit) \vdash SCE(SAT, n^k)$, then $SCE(SAT, n^k) \in P$. If $S_2^1(bit) \vdash DCE(SAT, n^k)$, 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 $\Sigma_2^b(bit)$-formulas so the first implication follows directly from Theorem 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 $S_2^1(bit)$ that $SCE(SAT, n^k)$ and also $DCE(SAT, n^k)$ can be stated in a $\forall\Sigma_1^b(bit)$ 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 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 $P_{x \in \{0,1\}^{c \log n}}[C_m(''f(\underline{g(x)}) = f(y)'') = g(x)] = 1$. 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 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 4.4.

Proposition 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 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 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 4.3. *If the same hardness assumption as in Proposition 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 $S_1^n, ..., S_{poly(n)}^n$, 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_j^n given 1^n and 1^j without knowing the interaction before round j.

Proof:

By Proposition 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 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_a^1, 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 $y_a^1 = 1$. 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 $y_a^1 = 1$. If $C(y(a1)) = 0$ and $C(y(a0)) = 1$, it sets $y_a^1 = 0$. 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 some 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 (it does not produce sC). 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 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 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 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 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.

[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.

Therefore, we have the following proposition.

Proposition 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.*

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 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 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 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 3.2) expressed positively:

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

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} proves $LB(SAT, n^k)$, 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 .

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 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 sufficiently big 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 6.1. *$T_{NC^1} \not\vdash DCE_\exists(SAT, n^{k+1})$ and $T_{NC^1} \not\vdash LB_\exists(SAT, n^{k+1})$ unless $SIZE(n^k) \subseteq NC^1$. For any sufficiently big k, $V^0 \not\vdash DCE_\exists(SAT, n^k)$ and $V^0 \not\vdash LB_\exists(SAT, n^k)$.*

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 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 6.1.

Theorem 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 $f(x|J_y(A))$. As $n^{2c} + n^{kc} + n^c n^{2c}(2Kc \log n + K) < n^{4kc}$ for $k \geq 1$ and big enough n, the whole circuit computing $(NW_{f,A}(x))_y$ has size $< n^{4kc}$.

Proof(of Theorem 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 and 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 $n^{O(1)}$:

$$(S(x) \notin Tr \rightarrow output = d_0) \wedge (S(x) \in Tr \rightarrow$$
$$((S(x, w_z, b_z) \notin Tr \rightarrow output = d_0) \wedge (...$$
$$(S(x, w_1, b_1, ..., w_t, b_t) \notin Tr \rightarrow output = d_0) \wedge$$
$$(S(x, w_1, b_1, ..., w_t, b_t) \in Tr \rightarrow (output = 0 \leftrightarrow SAT(y_t, b_t)))...)))$$

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 6.2. *$T_{NC^1} \nvdash LB(SAT, n^{4kc})$ and $VNC^1 \nvdash LB(SAT, n^{4kc})$ for $k \geq 1, c \geq 2$ 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 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 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 $LB(SAT, n^{\log n})$:

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

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

7. 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.

References

[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⊆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∩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.