Circuit Lower Bounds in Bounded Arithmetics

Ján Pich[1]

Department of Algebra
Faculty of Mathematics and Physics
Charles University in Prague

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^{2kc} where $k \geq 1, c \geq 4$ unless each function $f \in SIZE(n^k)$ can be approximated by formulas $\{F_n\}_{n=1}^{\infty}$ of subexponential size $2^{O(n^{2/c})}$ with subexponenital advantage: $P_{x\in\{0,1\}^n}[F_n(x) = f(x)] \geq 1/2 + 1/2^{O(n^{2/c})}$. 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 (2011 [14]) that certain NW-generators are hard for T_{PV}, the true universal theory in the language containing names for all p-time algorithms.

1. Introduction

We investigate the provability of polynomial circuit lower bounds in weak fragments of arithmetic like $S^1_2(bit)$ or APC_1. These theories are sufficiently strong to prove many important results in Complexity Theory. In fact, they can be considered as formalizations of feasible mathematics. Our motivation behind the investigation of these theories is the general question whether existential quantifiers in complexity-theoretic statements can be witnessed feasibly.

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

[1] 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. I would like to thank Jan Krajíček and Albert Atserias for useful discussions.

Intuitively, if the statement expressing n^k-size circuit lower bounds for SAT was such a feasibly witnessed statement, for any n^k-size circuit with n inputs we could efficiently find a formula of size n on which the circuit fails to decide SAT. We present a natural formalization of n^k-size circuit lower bounds for SAT denoted $LB(SAT, n^k)$ and observe that its provability in $S^1_2(bit)$ gives us such error witnessing. One could hope to use the witnessing algorithm to derive a contradiction with some established hardness assumption, however, Atserias and Krajíček (private communication) noticed that certain cryptographic conjectures imply the same form of witnessing, see Proposition 4.2.

We do not know how to obtain the unprovability of SAT circuit lower bounds in $S^1_2(bit)$ but we can do it basically for any weaker theory with stronger witnessing properties.

In weaker theories the situation is less natural because they cannot fully reason about p-time concepts. In particular, $LB(SAT, n^k)$ is equivalent to a formula $LB_2(SAT, n^k)$ (defined in Section 5) in $S^1_2(bit)$ but not necessarily in weaker theories. Therefore, we need to consider these two formalizations separately. 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.

If T_{NC^1} proves $LB_2(SAT, n^k)$, there are uniform NC^1 circuits which for each n^k-size circuit C with large enough n find a formula y of size n and computation of C on y witnessing that C decides SAT incorrectly on y. It is easy to show that in such case nonuniform NC^1 circuits could simulate $SIZE(n^k)$, see Proposition 6.1. Thus, a conditional unprovability of $LB_2(SAT, n^k)$ in T_{NC^1} follows easily.

To prove $LB(SAT, n^k)$ in T_{NC^1}, the resulting uniform NC^1 circuits would need to output for each n^k-size circuit C with large enough n an error y but they would not need to witness the computation of C on y. In this sense, for T_{NC^1} it is easier to reason about formalization $LB(SAT, n^k)$. We show that even $LB(SAT, n^{2kc})$ for $k \geq 1, c \geq 4$ is unprovable in T_{NC^1} unless each $f \in SIZE(n^k)$ can be approximate by formulas F_n of size $2^{O(n^{2/c})}$ with subexponential advantage: $P_{x\{0,1\}^n}[F_n(x) = f(x)] \geq 1/2 + 1/2^{O(n^{2/c})}$. The proof will be quite generic so, 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 used frequently in Bounded Arithmetic, see Section 5.

To prove our main claim we firstly observe that by the KPT theorem [15] the provability of $LB(SAT, n^k)$ in universal theories like T_{NC^1} gives us an $O(1)$-round Student-Teacher (S-T) protocol finding errors of n^{2kc}-size circuits attempting to compute SAT. Then, in particular, it works for n^{2kc}-size circuits encoding Nisan-Wigderson (NW) generators based on any functions $f \in SIZE(n^k)$ and suitable design matrices [16]. The interpretation of NW-generators as p-size circuits comes from Razborov [19]. In this situation we apply Krajíček's proof that certain NW-generators are hard for T_{PV} [14] which 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 [18]. Using natural proofs he showed that theory $S^2_2(\alpha)$ cannot prove polynomial circuit lower bounds on SAT unless strong pseudorandom generators do not exist. 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 Robinson's Arithmetic Q. In this respect, our proof applies to much stronger theories, basically to any theory weaker than $S^1_2(bit)$.

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 theory $S^1_2(bit)$, state its properties and in Section 4 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 binary representation of x and # with the intended meaning $x\#y = 2^{|x|\cdot|y|}$. Theories of bounded arithmetic are defined using language $L = \{0, S, +, \cdot, =, \leq, \lfloor x/2 \rfloor, |x|, \#\}$. We will consider also language L_{bit} which contains in addition 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. [2, 12].

Σ^b_0 denotes the set of all formulas in the language L with all quantifiers sharply bounded: $\exists x, x \leq |t|$, $\forall x, x \leq |t|$ where t is a term not containing x. All relations defined by Σ^b_0 formulas are p-time computable. Σ^b_i resp. Π^b_i for $i > 0$ are sets of formulas constructed from sharply bounded formulas by means of $\wedge$, $\vee$, sharply bounded, and existential bounded quantifiers: $\exists y\ y \leq t$ resp. universal bounded quantifiers: $\forall y\ y \leq t$ for x not occurring in t. All NP resp. coNP are representable by Σ^b_1 resp. Π^b_1 formulas, cf. [10, 20, 21].

Define $\Sigma^b_i(bit), \Pi^b_i(bit)$ for $i \geq 0$ as above but in the language L_{bit}. For $i \geq 1$, $\Sigma^b_i(bit)$-formulas are actually equivalent to Σ^b_i-formulas in theory PV_1, cf. [4, 12], see also Section 3. Analogously, for Π^b_i-formulas with $i \geq 1$.

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 $C_{[i,j]}$ for 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 tuple 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:

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

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:

$$\begin{aligned}
&Circ(C,y,w) \land \forall i < |y|\; y_i = w_i \land \forall j,k,l < |w|[\\
&(N_C(j) = (1,0) \land E_C(k,j) = 1 \land E_C(l,j) = 1 \to (w_j = 1 \leftrightarrow w_k = 1 \land w_l = 1))\land\\
&(N_C(j) = (0,1) \land E_C(k,j) = 1 \land E_C(l,j) = 1 \to (w_j = 1 \leftrightarrow w_k = 1 \lor w_l = 1))\land\\
&((N_C(j) = (0,0) \lor N_C(j) = (1,1)) \land E_C(k,j) = 1 \to (w_j = 1 \leftrightarrow w_k = 0))]
\end{aligned}$$

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

Next, let $SAT(y,z)$ be a $\Sigma_0^b(bit)$-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 $SAT(y,z)$:

$$\begin{aligned}
&\forall i,j,k < 2m\; [y_{i,j,k} = 1 \to\\
&\qquad (i,j,k < m \to z_i = 1 \lor z_j = 1 \lor z_k = 1)\land\\
&\qquad (i,j < m \land k \geq m \to z_i = 1 \lor z_j = 1 \lor z_{k-m} = 0)\land\\
&\qquad ...\\
&\qquad (i,j,k \geq m \to z_{i-m} = 0 \lor z_{j-m} = 0 \lor z_{k-m} = 0)]
\end{aligned}$$

Finally, for any k, a hardness of SAT for n^k-size circuits can be expressed as

$LB(SAT, n^k)$

$\forall 1^n > n_0\, \forall C\, \exists y,a\; |a| < |y| = n\, \forall w,z\; |w| \leq n^k, |z| < |y|\; [Comp(C,y,w) \to$
$(C(y;w) = 1 \land \neg SAT(y,z)) \lor (C(y;w) = 0 \land SAT(y,a))]$

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 $|m| = n \land m > n_0$. Therefore, y is feasible in m and for each n_0 and k, $LB(SAT, n^k)$ is universal closure of a $\Sigma_2^b(bit)$ formula.

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 [2]. We will use its conservative extension $S^1_2(bit)$ which consists of $BASIC(bit)$ and polynomial induction for $\Sigma^b_1(bit)$-formulas A:

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

An important property of $S^1_2(bit)$ is Buss's witnessing theorem:

Theorem 3.1 (Buss [2])**.** *If $S^1_2(bit) \vdash \exists y A(x,y)$ for $\Sigma^b_0(bit)$-formula A, then there is a p-time functions 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.

Theorem 3.2 (Krajíček [11])**.** *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 defined by Cook [4], see also [12]. The language of PV_1 consist 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 [15] 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 because 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 which we denote also $S^1_2(bit)$. PV_1 proves induction and polynomial induction for $\Sigma^b_0(PV)$-formulas defined similarly as Σ^b_0-formulas but in the language of PV_1. There is also an interesting theory APC_1 introduced by Jeřábek [9] which is an extension of $S^1_2(bit)$ capturing a subclass of BPP similarly as $S^1_2(bit)$ captures P.

Theories $S^1_2(bit), PV_1$ and APC_1 are weak fragments of arithmetic but they are sufficiently strong to prove many important things. In [12, chap. 15] it is shown how to prove PARITY $\notin AC^0$ in APC_1. Razborov[17] argued that $S^1_2(bit)$ is the right theory capturing techniques from circuit complexity in 1995. We expect that APC_1 captures feasible reasoning so well that any provable statement about feasible concepts is provable in APC_1 assuming that feasible concepts intuitively correspond to BPP concepts . Of course, this does not contain, for instance, Shannon's argument if it is formalized so that it manipulates with exponentially big objects.

3.1. Equivalent formalizations of $LB(SAT, n^k)$

There are more possible formalizations of circuit lower bounds that are essentially equivalent to $LB(SAT, n^k)$. For example, $SCE(SAT, n^k)$ meaning that for each n^k-size circuit there is a satisfiable formula of size n such that the circuit will not find its satisfying assignment.

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

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 counter example".

Another formalization of circuit lower bounds is given by the following formula $DCE(SAT, n^k)$ where DCE refers to "decision SAT counter example". Now circuits C have again just one output but using self-reducibility they can be used to search for satisfying assignments of propositional formulas: If C says that 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, then in the same manner continue searching for the full satisfying assignment. If no such C can be used to find satisfying assignments of satisfiable propositional formulas, for each such C there is a formula y and a possibly partial assignment to its variables a such that either $SAT(y, a)$ and C says that y is unsatisfiable or $\neg SAT(y, a)$ for full assignment a of y and C says that a satisfies y or it happens that C gets into a local inconsistency: for a partial assignment a of y it claims that y assigned by 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:

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

where $y(a)$ encodes formula y assigned by 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 assigned by extension of a which set the first unassigned variable in y by 1 resp. by 0. We leave details of these encodings to kind reader.

$LB(SAT, n^k), SCE(SAT, n^k), DCE(SAT, n^k)$ are basically equivalent. We claim that this is provable already in PV_1 and hence also in $S^1_2(bit)$.

Proposition 3.1. *PV_1 proves the following implications*

$$\begin{array}{c}
SCE(SAT, n^{2k}) \to LB(SAT, n^k) \\
LB(SAT, n^{2k}) \to SCE(SAT, n^k) \\
LB(SAT, n^k) \to DCE(SAT, n^k) \\
DCE(SAT, n^k) \to LB(SAT, n^k)
\end{array}$$

where n_0 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 an n^{2k}-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 $\neg SCE(SAT, n^{2k})$.

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^{2k}-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 $(SAT(y, a) \to dC(y; w) = 1) \wedge (dC(y; w) = 1 \to SAT(y, C(y)))$ for any y, a of size $|a| < |y| = n$, hence $\neg LB(SAT, n^{2k})$.

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

Finally, in PV_1, if C witnesses $\neg LB(SAT, n^k)$, it witnesses $\neg DCE(SAT, n^k)$: for any y, a of size $|a| < |y| = n$ for a big enough n, $(C(y; w) = 0 \to \neg SAT(y, a))$, $C(y(a); w) = 1 \wedge FA(a, y) \to SAT(y, a)$ 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 large n and n^k-size circuit C with n inputs finds out y, a such that $LB(C, y, a)$: $C(y) = 1 \wedge SAT(y, a)$ or $C(y) = 0 \wedge \forall z \neg SAT(y, z)$. 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 large 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 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

$C(y(a)) = 1 \wedge PA(a, y) \wedge (C(y(a0)) = 0 \vee C(y(a1)) = 0)$

$SCE(SAT, n^k) \in$ P if there is a p-time algorithm A which for 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 advise 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 details in the next section.

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

The provability of circuit lower bounds in $S^1_2(bit)$ gives us an efficient witnessing errors of p-size circuits for SAT described in the previous section.

Proposition 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 Krajíček's witnessing theorem. In case of $SCE(SAT, n^k)$ and $DCE(SAT, n^k)$ T's advise 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. □

An efficient witnessing errors of p-time SAT algorithms follows also from instance checkers for SAT, see [1, chap. 8]. If we want to check only n^k-time algorithms, the instance checker is p-time itself:

Theorem 4.1. *There is a p-time algorithm that given any n^k-time algorithm M and a formula y of size n accepts if M solves SAT on all instances, and* *rejects with probability $\geq 1 - 1/2^n$ if M does not decide satisfiability of y correctly.*

Therefore, any n^k-time algorithm M claiming to solve SAT can be tested by checking it on formula $F_M(y, a)$ encoding the statement "a satisfies formula y but M fails to find a satisfying assignment of y (in the same way as C fails to find it in $DCE(SAT, n^k)$)". If $M(F_m(y, a)) = 1$, by self-reducibility M will be forced to find a satisfying assignment of F_M which is an error of M or it will end up in a local inconsistency which is also error. If $F_M(y, a)$ is unsatisfiable, the checker will use an interactive protocol with M as a Prover to verify that.

In practice, we can test whether a given algorithm M proves theorems efficiently also by taking a statement we consider hard to prove and refute it instead of F_M .

Furthermore, if f is one-way function, we can also secretly produce $a \in \{0,1\}^n$ and ask the algorithm whether the statement $f(a) = f(x)$ encoded as a *poly*$(|a|)$-size formula with free variables $x = x_1, ..., x_n$ is satisfiable, see [6]. In this case, we do not need to use interactive protocols because 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 one-way permutation f 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 [16] 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 one-way permutation secure against p-size circuits. Take l such that for each n^k-size circuits C_n predicate $C_n(f(x)) \neq x$ for $x \in \{0,1\}^n$ can be computed by n^l-size circuits. Now, for any n^k-size circuit C_n with sufficiently big n, for each $x \in \{0,1\}^{c \log n}$ find out whether $C_n(f(g(x))) \neq g(x)$ holds. This can be done in $poly(n)$-time. If we did not succeed at least once, $P_{x\in\{0,1\}^{c \log n}}[C_n(f(g(x))) = g(x)] = 1$, and that would break g. □

In [13] Krajíček also observed that in order to show $SCE(SAT, n^k) \in$ *P/poly*, it suffices to assume that SAT $\notin SIZE(n^{2k})$. It uses a well known combinatorial principle: 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 "if y is a satisfiable formula of size n, circuit x does not find a satisfying assignment of y". 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, for any satisfiable y at least one of the n fixed circuits would find a satisfying assignment of y. 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 $SCE(SAT, n^k)$.

Similarly, it works for $DCE(SAT, n^k)$ because checking $E(x,y)$, i.e. whether circuit x (with one output) can be used to find the satisfying assignment, is efficient. For $LB(SAT, n^k)$ it could however happen that the search for the satisfying assignment ends in a local inconsistency.

Proposition 4.3 (Krajíček [13])**.** *If SAT $\notin SIZE(n^{2k})$, then $SCE(SAT, n^k)$ and $DCE(SAT, n^k)$ are in P/poly.*

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 4.4. *If the same hardness assumption as in Proposition 4.2 holds, then $LB(SAT, n^k)$ has an S-T protocol with $poly(n)$ rounds where S is in uniform AC^0, and it has also an S-T protocol with 1 round (i.e. 1 advice of T) where S is a p-time algorithm.*

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 $\leq n^{2k}$ searching for satisfying assignments of given formulas: 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 assigned by 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. 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 $C(y(a0)) = C(y(a0)) = 0$. In such case S outputs y, 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 $\leq n^c$ elements such that each n^{2k}-size circuit fails on at least one of them. It suffices to use instead of A the set B, i.e. to try all elements from B in place of a_1. Moreover, whenever S needs to simulate circuit C on input y it can output y with an arbitrary assignment c of y. If this is not a correct answer, T will reply either with a satisfying assignment d of y or with the computation of C on y which can be verified by a uniform constant-depth formula. In the former case S outputs y, d and this time it gets what it wants. □

This also shows 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. [3]. 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.

5. Theories weaker than PV_1

In this section we consider 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. Analogously, T_{PV} resp. T_{AC^0} is the true universal theory in the language L_{PV} resp. L_{AC^0} containing names for all p-time algorithms resp. uniform families of AC^0 circuits.*

These theories are universal so they admit the KPT theorem from [15]:

Theorem 5.1 ([15]). *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

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

Analogously for T_{AC^0} and T_{PV} for which the resulting functions are in uniform AC^0 resp. in P.

In the field of Bounded Arithmetic there are also standard theories corresponding to uniform AC^0, NC^1 and other complexity classes, cf. [7]. Typically, they are presented as two-sorted theories having one sort of variables representing numbers and the second sort of variables representing bounded sets of numbers. 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 which holds for a number n and set X if and only if 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 number term which does not involve X. Σ_0^B 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 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_0^B-formulas stating that for any Σ_0^B 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_0^B-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_0^B-formula A, there are finitely many functions $f_1, ..., f_k$ in uniform NC^1 such that

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

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

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

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

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_{AC^0}. L_{AC^0} contains symbols for $SAT(y, z), Comp(C, y, w)$ and all the predicates we explicitly defined as $\Sigma_0^b(bit)$-formulas because they are not just p-time but in fact constant-depth formulas. Moreover, even if multiplication is not in L_{AC^0} (but in L_{NC^1}) we may assume that the L_{AC^0} functions $Comp(C, y, w), C(y; w) = 1/0$ contain the bound $|w| \leq |y|^k$. For simplicity, whenever we speak about $LB(SAT, n^k)$ in T_{AC^0} we mean its formalization where instead of the $\Sigma_0^b(bit)$-formulas we have the respective symbols of L_{AC^0}. Similarly for $SCE(SAT, n^k)$, $DCE(SAT, n^k)$ and T_{NC^1}. Therefore, $LB(SAT, n^k)$, $SCE(SAT, n^k)$ and $DCE(SAT, n^k)$ in T_{AC^0} and 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 $T_{NC^1}, VNC^1, T_{AC^0}$... is less natural because they cannot fully reason about p-time concepts. In particular, there is a formula $LB_2(SAT, n^k)$ which is equivalent to $LB(SAT, n^k)$ in $S_2^1(bit)$ but not necessarily in T_{NC^1}. $LB_2(SAT, n^k)$ is like $LB(SAT, n^k)$ but with $LB(C, y, a)$ expressed positively:

$LB_2(SAT, n^k)$

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

Analogously define $DCE_2(SAT, n^k)$, $SCE_2(SAT, n^k)$ and their two-sorted and L_{AC^0} formulations.

By the witnessing theorem above, if T_{NC^1} proves $LB(SAT, n^k)$, $LB(SAT, n^k)$ has an NC^1 S-T protocol with $O(1)$ rounds which is S-T protocol with

$O(1)$ rounds and uniform NC^1 S. If $T_{NC^1} \vdash LB_2(SAT, n^k)$, $LB_2(SAT, n^k)$ 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 $DCE_2(SAT, n^k)$ has the form $\exists y A(x, y)$ for an open A in L_{AC^0}, its provability in T_{NC^1} implies $DCE_2(SAT, n^k) \in NC^1$. Here again, $DCE_2(SAT, n^k) \in NC^1$ 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 T_{AC^0}, 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 $LB_2(SAT, n^k)$ 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 quite simple:

Proposition 6.1. *$LB(SAT, n^{k+1}) \notin NC^1$, $DCE_2(SAT, n^{k+1}) \notin NC^1$ and $LB_2(SAT, n^{k+1})$ has no NC^1 S-T protocol with $poly(n)$ rounds unless $SIZE(n^k) \subseteq NC^1$. Unconditionally, for any sufficiently big k, $LB(SAT, n^k) \notin AC^0$, $DCE_2(SAT, n^k) \notin NC^1$ and $LB_2(SAT, n^k)$ 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

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

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

This construction works analogously for $DCE_2(SAT, n^k)$ and as well for $LB_2(SAT, n^k)$ because if there was some NC^1 S-T protocol for $LB_2(SAT, n^k)$ S would be forced to produce computations w of given circuits. □

Corollary 6.1. *$T_{NC^1} \nvdash DCE_2(SAT, n^{k+1})$ and $T_{NC^1} \nvdash LB_2(SAT, n^k)$ unless $SIZE(n^k) \subseteq NC^1$. For any sufficiently big k, $V^0 \nvdash DCE_2(SAT, n^k)$ and $V^0 \nvdash LB_2(SAT, n^k)$.*

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 4.4 $LB(SAT, n^k)$ 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 [19] and Krajíček's proof of a hardness of certain NW-generators for T_{PV} [14]. It actually seems to be a relatively straightforward modification of the previous simple observation.

Theorem 6.1. *If there is $f \in SIZE(n^k)$ such that for all formulas F_n of size $2^{O(n^{2/c})}$, $P_{x\in\{0,1\}^n}[F_n(x) = f(x)] < 1/2 + 1/2^{O(n^{2/c})}$ for infinitely many n's, then $LB(SAT, n^{2kc})$ has no NC^1 S-T protocol with $O(1)$ rounds.*

To prove the theorem we will use Nisan-Wigdewrson (NW) generators with specific design properties. Let $A = \{a_{i,j}\}^{i=1,...,m}_{j=1,...,n}$ 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 $c \geq 4$, Nisan and Wigderson [16] constructed $2^n \times n^c$ 0-1 matrix A with $n^{c/2}$ ones per row which is also $(n, n^{c/2})$-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 there are n^c-size circuits which given $i \in \{0,1\}^n$ compute the set $J_i(A)$. Therefore, as it was observed by Razborov [19], if f is in addition computable by n^k-size circuits, for any $x \in \{0,1\}^{n^c}$, $(NW_{f,A}(x))_x$ is a function on n inputs y computable by circuits of size n^{2kc}.

Proof(of Theorem 6.1): Let $f \in SIZE(n^k)$ and A be a $2^n \times n^c$ $(n, n^{c/2})$-design defined above so for any x, $(NW_{f,A}(x))_y$ can be computed from y by an n^{2kc}-size circuit. Assume that $LB(SAT, n^{2kc})$ has an NC^1 S-T protocol with $O(1)$ rounds. In particular, for each n^{2kc}-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 w_1, b_1 such that

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

In the later 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 $E_{n^c}(x)$ be AC^0 circuits mapping $x \in \{0,1\}^{n^c}$ to a description of an n^{2kc}-size circuit with n inputs y computing the function $(NW_{f,A}(x))_y$. We will consider our S-T protocol only on inputs of the form $E_{n^c}(x)$.

Krajíček [14] 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 satisfying assignment b to y and the unique computation w of given circuit y. If there is no such b, T replies with a string of zeros. This should replace the uniqueness property assumed in [14].

For $u \in \{0,1\}^{n^{c/2}}$ and $v \in \{0,1\}^{n^c - n^{c/2}}$ define $r_y(u,v) \in \{0,1\}^{n^c}$ 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 $a \in \{0,1\}^{n^c - n^{c/2}}$ such that $Tr = tr(r_{y_t}(u,a))$ for at least a fraction of $2/(3(2^{2n}))^t$ of all u's.*

Tr and a can be constructed inductively. There are at most 2^{2n} tuples 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 $a \in \{0,1\}^{n^c - n^{c/2}}$ such that $y_1, a_1 = tr(r_{y_1}(u,a))$ 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 $a \in$

$\{0,1\}^{n^c-n^{c/2}}$ such that $y_1, a_1, ..., y_i, a_i = tr(r_{y_i}(u,a))$ 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 a from the previous claim.

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

Now we define a formula F that attempts to compute f and uses as advice Tr, a and some t sets Y_y. For each $u \in \{0,1\}^{n^{c/2}}$ produce $r_{y_t}(u,a)$ (this is in AC^0). Let V be the set of those inputs u for which $tr(r_{y_t}(u,a))$ either is Tr or starts as 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^n which has $poly(n)$ output bits. It has the form $\bigvee_{z\in\{0,1\}^n}(z = r_{y_t}(u,a)|(J_{y_1}(A) \cap J_{y_t}(A)) \rightarrow output = w_z \in Y_{y_1})$. 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 $n^{c/2}$ inputs and size $2^{O(n)}$ because producing $r_{y_t}(u,a)$ is in AC^0, searching for T's advice in 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 $u \in \{0,1\}^{n^{c/2}}$ F will successfully predict $f(u)$. Moreover, at most $1/(3(2^n))^t$ of all traces $tr(r_{y_t}(u,a))$ extend Tr. Because d_0 is the correct value on at least half of $u \in U$, $P_u[F(u) = f(u)] \geq 1/2 + 1/(3^t 2^{nt+1})$ □

Corollary 6.2. *$T_{NC^1} \nvdash LB(SAT, n^{2kc})$ and $VNC^1 \nvdash LB(SAT, n^{2kc})$ for $k \geq 1, c \geq 4$ unless for each $f \in SIZE(n^k)$ there are formulas F_n of size $2^{O(n^{2/c})}$ such that for sufficiently big n's, $P_{x\in\{0,1\}^n}[F_n(x) = f(x)] \geq 1/2 + 1/2^{O(n^{2/c})}$.*

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 $V^0 \vdash LB(SAT, n^k)$, $LB(SAT, n^k)$ 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 $2^{O(n^{2/c})}$ with advantage $1/2^{O(n^{2/c})}$. 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 $|m| = |n|^2$). Analogously, define the two-sorted and L_{AC^0} version of $LB(SAT, n^{\log n})$.

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

References

[1] Arora S., Barak B.; Computational Complexity: A Modern Approach, Cambridge University Press, 2009.

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

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

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

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

[11] Krajíček J.; Fragments of Bounded Arithmetic and Bounded Query Classes, Transactions of the AMS, 338 (1993), 587-598.

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

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

[14] 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.

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

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

[17] Razborov A.A.; Bounded Arithmetic and Lower Bounds in Boolean Complexity, Feasible Mathematics II, 1995, pp. 344-386.

[18] 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.

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

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

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