# Challenges in computational lower bounds

Emanuele Viola*

August 29, 2013

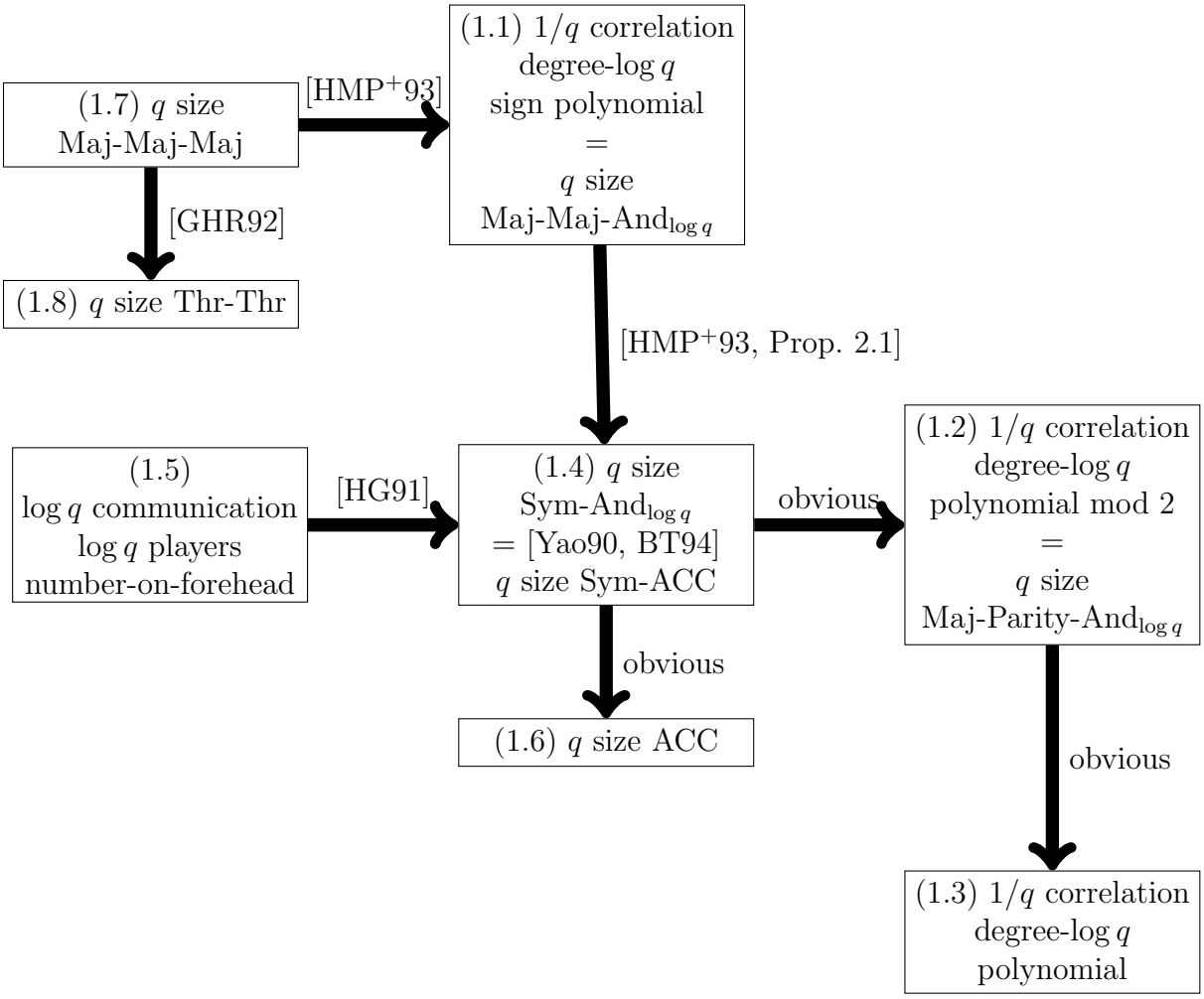
We draw two incomplete, biased maps of challenges in computational complexity lower bounds. Our aim is to put these challenges in perspective, and to present some connections which do not seem widely known.

We do not survey existing lower bounds, go through the history, or repeat standard definitions. All of this can be found e.g. in the recent book [Juk12], or in the books and surveys [SY10, Lok09, Vio09b, She08, AB09, KN97, Bei93, Raz91, BS90, Hås87].

Each node in the maps represents the challenge of proving that there exists an explicit boolean function that cannot be computed with the resources labeling that node. We take explicit to mean NP, thus excluding most or all of the lower bounds that rely on diagonalization. An arrow from node A to B means that resources A can simulate resources B, and so solving A implies solving B.

*Supported by NSF grant CCF-0845003. Email: viola@ccs.neu.edu

# 1 Circuits with various gates, correlation, and communication




Each occurrence of $q$ stands for a quasipolynomial function $2^{\log^c n}$ for a possibly different constant $c$. For example, Challenge (1.5) asks to exhibit an explicit function $f$ such that for every constants $c$ and $c'$ it holds that for sufficiently large $n$ the function $f$ on inputs of length $n$ cannot be computed by a number-on-forehead protocol among $\log^c n$ players exchanging $\log^{c'} n$ bits.

The picture changes if $q$ stands for a polynomial function $n^c$. In this case the three equalities in (1.1), (1.2), and (1.4) do not hold anymore. Intuitively this is because a polynomial in $n$ variables of degree $\log n$ may have $n^{\Omega(\log n)}$ terms. In fact, Razborov and Wigderson show in [RW93] $n^{\Omega(\log n)}$ lower bounds for Maj-Sym-And circuits, thus resolving one side in each of these equalities. Other than that, every challenge is open even for $q = n^c$. The arrows that are known to hold in this case are the "obvious" arrows (1.4)–(1.6) and (1.2)–(1.3), and the arrow (1.7)–(1.8), labeled [GHR92]. Finally, there are new arrows from (2.6) to (1.7) and to (1.4). For the technique yielding these new arrows see e.g. [Vio09a, Lecture 8].

Both Maj and Thr stand for gates that compute a threshold function, i.e. a function that given input bits $(x_1, \ldots, x_s)$ outputs 1 iff $\sum_i c_i \cdot x_i \geq t$, for fixed integers $c_i$ and $t$. A circuit has *size* $s$ if it has at most $s$ gates and the weights $c_i$ in every majority gate satisfies $|c_i| \leq s$. We do not allow multiple edges. Sym stands for a gate computing a symmetric function. $\text{And}_{\log q}$ is an And gate of fan-in $\log q$. Every other gate has unbounded fan-in. We use standard notation for composing gates. For example Maj-Maj-$\text{And}_{\log q}$ refers to a circuit with output gate Maj taking as input Maj gates taking as input And gates with fan-in $\log q$ taking as input the input bits.

For simplicity all polynomials have integer coefficients. By "$\epsilon$ correlation degree-$d$ polynomials" (1.3) we refer to the set of functions $g : \{0,1\}^n \to \{0,1\}$ such that there exists some distribution $D$ on the inputs, and some polynomial $p$ of type $X$ such that $|\Pr_{x\ D}[p(x) = g(x)] - \Pr_{x\ D}[p(x) \neq g(x)]| \geq \epsilon$. For (1.2) and (1.1) we take the output of the polynomial modulo 2 or, respectively, the sign of the output.

We now elaborate further on some of the challenges:

(1.2) See the survey [Vio09b, Chapter 1]. The equality is obtained as follows. The simulation of polynomials by circuits is proved via boosting [Fre95, Section 2.2] or min-max/linear-programming duality [GHR92, Section 5]. The other direction follows from the "discriminator lemma" of [HMP$^+$93].

(1.1) The equality is obtained by reasoning as for (1.2). Since we are not restricting the magnitude of the polynomial's coefficients this would yield circuits where the middle gate is Thr, not Maj. However [GHR92, Theorem 26] shows that Maj-Thr = Maj-Maj up to a polynomial change in size.

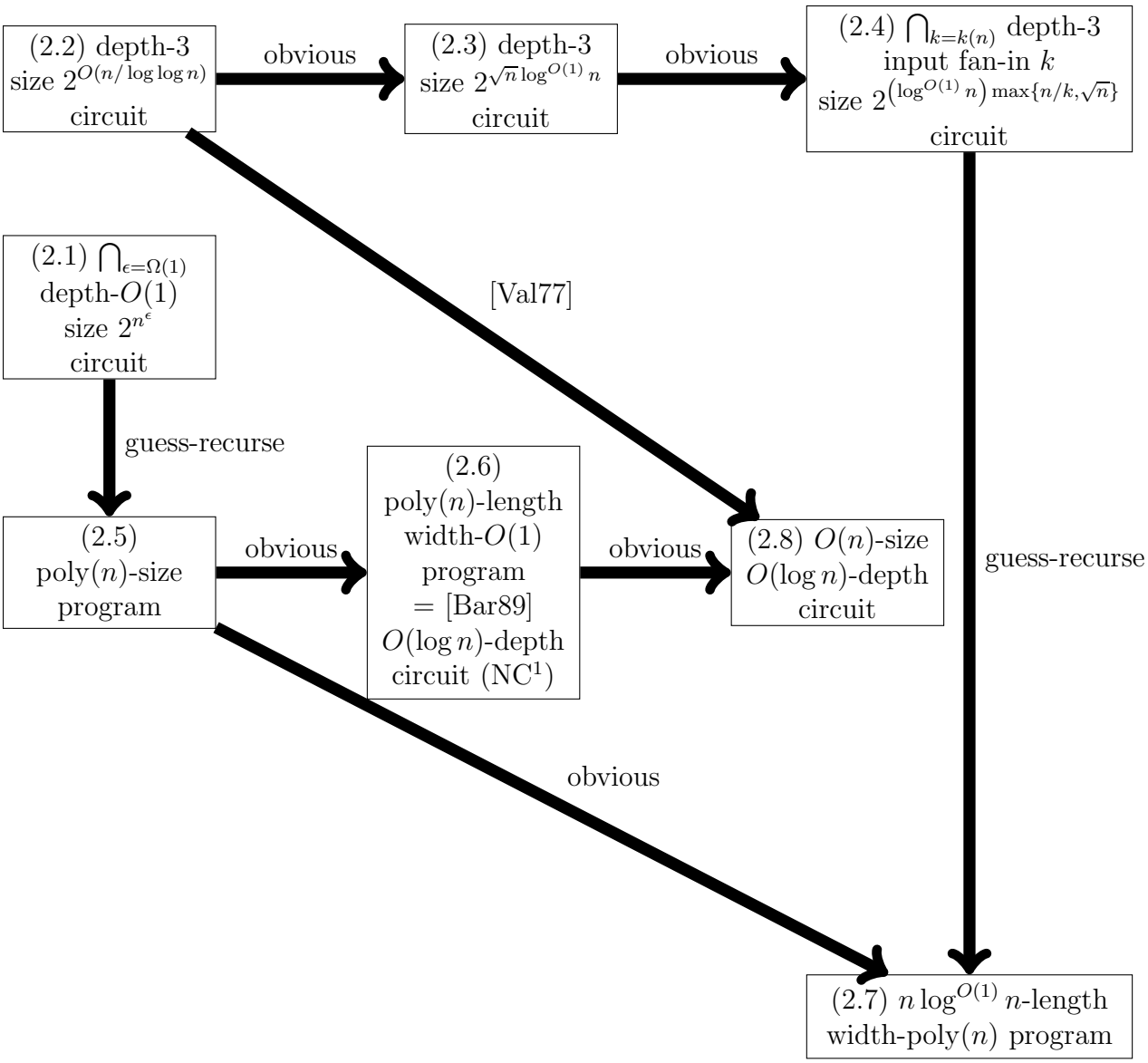
(1.3) For more on this see [RV].

(1.8) For a special case see [HP10].

(1.5) For a special case for which the arrow continues to hold see [BGKL03].

Arrow (1.7)–(1.1), labeled [HMP$^+$93], follows from the techniques in [HMP$^+$93, Lemma 2.4] which give that any Maj-Sym circuit can be turned into a Maj-Maj circuit with a polynomial increase.

# 2 Circuits and branching programs




"Program" stands for "branching program." Specifically we consider layered branching programs of width $w$ (i.e., space $\log w$) and length $t$. The size is $w \cdot t$. Each node is labeled with an input variable. The challenges remain open for the model of oblivious branching programs where the label on each node depends only on the layer. Recall that Nechiporuk's argument [Nec66] gives bounds of the form $\geq n^2/\log^{O(1)} n$ on the size. This bound gives $t = n^2/\log^{O(1)} n$ for constant width $w = O(1)$; it gives nothing for polynomial width $w = n^{O(1)}$. For polynomial or even sub-exponential width the state-of-the-art is due to Beame, Saks, Sun, and Vee [BSSV03]. For sub-exponential width they obtain $t \geq \Omega(n\sqrt{\log n/\log\log n})$.

All circuits are over the basis And, Or, and Not, with negations at the input level only. For circuits of depth $O(1)$ the fan-in of Or and And gates is unbounded; for circuits of depth

$\Omega(\log n)$ the fan-in of these gates is 2. The *size* of a circuit is its number of edges. Recall that for every constant $d$ the state-of-the-art lower bounds are of the form $\geq 2^{cn^{1/(d-1)}}$ for a constant $c$, see e.g. [Hås87]. Challenge (2.1) asks to exhibit an $\epsilon > 0$ such that for every $d$ a lower bound $2^{n^\epsilon}$ holds. Note for $d = 3$ the state-of-the-art gives $2^{c\sqrt{n}}$. Challenge (2.3) asks to improve this. For a recent approach, see [GW13]. Further parameterized by the input fan-in $k$ of the circuit, the available lower bounds for $d = 3$ are no better than $2^{c \max(n/k),\sqrt{n}}$ for a constant $c$. Challenge (2.4) asks to break this tradeoff.

The arrows (2.1)–(2.5) and (2.4)–(2.7), labeled "guess-recurse," are obtained via a technique attributed to Nepomnjaščiĭ [Nep70]. The arrow (2.1)–(2.5) continues to hold if (2.5) is replaced with the functions that for every $\epsilon > 0$ are computable by non-deterministic branching programs of length $\text{poly}(n)$ and width $2^{n^\epsilon}$, a class containing NL.

We give the details for the (2.4)–(2.7) arrow.

**Claim 2.9.** Let $f : \{0,1\}^n \to \{0,1\}$ be computable by a branching program with width $w$ and time $t$. Then $f$ is computable by a depth-3 circuit with $\leq 2^{\sqrt{t \log w}} \cdot t$ wires. More generally, for any parameter $b$ one can have a depth-3 circuit with

$$2^{b \log w + t/b + \log t}$$

wires, output fan-in $w^b$, and input fan-in $t/b$.

The (2.4)–(2.7) arrow corresponds to the setting $t = n \cdot \log^{O(1)}$ and $w = \text{poly}(n)$. It is obtained as follows. If $k \geq \sqrt{n}$ (infinitely often) the arrow follows immediately. If $k < \sqrt{n}$ set $b := t/k$ and note that the lemma gives a circuit with input fan-in $k$ and size $\leq 2^{(\log^{O(1)} n)n/k + k + O(\log n)} \leq 2^{(\log^{O(1)} n)n/k}$.

*Proof.* On an input $x$, guess $b$ middle points on the branching program's computation path, at fixed times $t/b, 2t/b, \ldots, t$. Since the times are fixed, this is a choice out of $w^b$. Then verify the computation of each of the corresponding $b$ intervals is correct.

Each interval involves paths of length $\leq t/b$. The computation can be written as a decision tree of the same depth. In turn, this is a CNF with $\leq 2^{t/b} t/b$ wires.

Collapsing adjacent layers of And gates we obtain a circuit with size

$$\leq w^b \cdot b \cdot \leq 2^{t/b} t/b = 2^{b \log w + t/b + \log t}$$

wires.

Setting $b := \sqrt{t/\log w}$ yields size

$$2^{\sqrt{t \log w} + \log t}.$$

Moreover, by construction this circuit has output fan-in $w^b$ and input fan-in $t/b$. □

For an exposition of the arrow (2.2)–(2.8), labeled [Val77], see e.g. [Vio09b, Chapter 3].

# References


[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity*. Cambridge University Press, 2009. A modern approach.

[Bar89] David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in $NC^1$. *J. of Computer and System Sciences*, 38(1):150–164, 1989.

[Bei93] Richard Beigel. The polynomial method in circuit complexity. In *8th Structure in Complexity Theory Conference*, pages 82–95. IEEE, 1993.

[BGKL03] László Babai, Anna Gál, Peter G. Kimmel, and Satyanarayana V. Lokam. Communication complexity of simultaneous messages. *SIAM J. Comput.*, 33(1):137–166, 2003.

[BS90] Ravi B. Boppana and Michael Sipser. The complexity of finite functions. In *Handbook of theoretical computer science, Vol. A*, pages 757–804. Elsevier, Amsterdam, 1990.

[BSSV03] Paul Beame, Michael Saks, Xiaodong Sun, and Erik Vee. Time-space trade-off lower bounds for randomized computation of decision problems. *J. of the ACM*, 50(2):154–195, 2003.

[BT94] Richard Beigel and Jun Tarui. On ACC. *Computational Complexity*, 4(4):350–366, 1994.

[Fre95] Yoav Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

[GHR92] Mikael Goldmann, Johan Håstad, and Alexander A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2:277–300, 1992.

[GW13] Oded Goldreich and Avi Wigderson. On the size of depth-three boolean circuits for computing multilinear functions. *Electronic Coll. on Computational Complexity (ECCC)*, 20:43, 2013.

[Hås87] Johan Håstad. *Computational limitations of small-depth circuits*. MIT Press, 1987.

[HG91] Johan Håstad and Mikael Goldmann. On the power of small-depth threshold circuits. *Comput. Complexity*, 1(2):113–129, 1991.

[HMP+93] András Hajnal, Wolfgang Maass, Pavel Pudlák, Mário Szegedy, and György Turán. Threshold circuits of bounded depth. *J. of Computer and System Sciences*, 46(2):129–154, 1993.

[HP10] Kristoffer Arnsfelt Hansen and Vladimir V. Podolskii. Exact threshold circuits. In *IEEE Conf. on Computational Complexity (CCC)*, pages 270–279, 2010.

[Juk12] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*. Springer, 2012.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

[Lok09] Satyanarayana V. Lokam. Complexity lower bounds using linear algebra. *Foundations and Trends in Theoretical Computer Science*, 4(1-2):1–155, 2009.

[Nec66] E. I. Nechiporuk. A boolean function. *Soviet Mathematics-Doklady*, 169(4):765–766, 1966.

[Nep70] Valery A. Nepomnjaščiĭ. Rudimentary predicates and Turing calculations. *Soviet Mathematics-Doklady*, 11(6):1462–1465, 1970.

[Raz91] Alexander A. Razborov. Lower bounds for deterministic and nondeterministic branching programs. In *Fundamentals of Computation Theory (FCT)*, pages 47–60, 1991.

[RV] Alexander Razborov and Emanuele Viola. Real advantage. *ACM Trans. Computation*

*Theory*.

[RW93] Alexander Razborov and Avi Wigderson. $n^{\Omega(\log n)}$ lower bounds on the size of depth-3 threshold circuits with AND gates at the bottom. *Information Processing Letters*, 45(6):303–307, 1993.

[She08] Alexander A. Sherstov. Communication lower bounds using dual polynomials. *Bulletin of the EATCS*, 95:59–93, 2008.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *6th Symposium on Mathematical Foundations of Computer Science*, volume 53 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 1977.

[Vio09a] Emanuele Viola. Gems of theoretical computer science. Lecture notes of the class taught at Northeastern University. Available at http://www.ccs.neu.edu/home/viola/classes/gems-08/index.html, 2009.

[Vio09b] Emanuele Viola. On the power of small-depth computation. *Foundations and Trends in Theoretical Computer Science*, 5(1):1–72, 2009.

[Yao90] Andrew Chi-Chih Yao. On ACC and threshold circuits. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 619–627, 1990.