# On the (Non) NP-Hardness of Computing Circuit Complexity

Cody D. Murray and Ryan Williams

presented by Radek Hušek

## Complexity ZOO

<table>
<thead>
<tr>
<th>Complexity class</th>
<th>Characterization</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>polytime deterministic algorithms</td>
</tr>
<tr>
<td>$RP$</td>
<td>polytime randomized algorithms with bounded one-size error(^1)</td>
</tr>
<tr>
<td>$BPP$</td>
<td>polytime randomized algorithms with bounded two-size error</td>
</tr>
<tr>
<td>$ZPP$</td>
<td>randomized algorithms with average polytime complexity</td>
</tr>
<tr>
<td>$AC^0$</td>
<td>polysize circuits with unbounded fan-in and constant depth(^2)</td>
</tr>
<tr>
<td>$AC^0[m]$</td>
<td>$AC^0 + \text{“mod } m\text{” gates}$</td>
</tr>
<tr>
<td>$E$</td>
<td>$\text{TIME}(2^{O(n)})$</td>
</tr>
<tr>
<td>$EXP$</td>
<td>$\text{TIME}(2^{n^{O(1)}})$ deterministic algorithms</td>
</tr>
<tr>
<td>$P_{/\text{poly}}$</td>
<td>polytime with polynomial advise</td>
</tr>
</tbody>
</table>

The “N” prefix denotes non-deterministic variant of given complexity class: Input of non-deterministic algorithm is (except instance of given problem) a “certificate”. For every YES-instance there exists certificate which makes algorithm answer yes, and for NO-instance no certificate can convince algorithm to answer yes.

Given complexity class $C$, language $L$ belongs into class i.o.-$C$ (infinitely often) iff $L \cap \{0, 1\}^n = L' \cap \{0, 1\}^n$ for some $L' \in C$ and infinitely many $n$, and $\text{co} C := \{ L : \bar{L} \in C \}$.

## Minimum Circuit Size Problem Complexity

**Definition 1.** The Minimum Circuit Size Problem (MCSP):

*Input is $(T, k)$ where $T \in \{0, 1\}^n$ is truth-table of boolean function on log$_2 n$ variables and $k \in \mathbb{N}$ (encoded binary or unary). Output is YES if there is circuit of complexity\(^3\) at most $k$ which evaluates function $T$, and NO otherwise.*

We’re encoding MCSP as string $Tx$, where $|T| = \max_{n \in \mathbb{N}} \{ 2^n < |Tx| \}$ and $x$ is binary encoding of parameter $k$\(^4\).

We will use machine model with random access to input such as random-access Turing machine.

---

\(^1\)Only false-negatives.

\(^2\)We allow only AND, OR and NOT gates.

\(^3\)Complexity of is circuit is number of its gates and we’re allowed to use AND, OR and NOT gates with fan-in at most 2.

\(^4\)This encoding limits possible values of $k$ but it’s not a problem because every Boolean function on $n$ variables has circuit complexity at most $(1 + o(1))2^n / n$ (Lupanov 59).
Definition 2. An algorithm $R : \Sigma^* \times \Sigma^* \to \{0, 1, *\}$ is \textbf{TIME($t(n)$) reduction} from $L$ to $L'$ if there is constant $c \geq 0$ such that $\forall x \in \Sigma^*$:

- $R(x, i)$ runs in $O(t(|x|))$ for all $i \in \{0, 1\}^{2c\log_2|x|}$,
- There is an $l_x \leq |x|^c + c$ such that $R(x, i) \in \{0, 1\}$ for all $i \leq l_x$ and $R(x, i) = *$ for all $i > l_x$, and
- $x \in L \iff R(x, 1)R(x, 2)\ldots R(x, l_x) \in L'$.

Proposition 3 (Skyum & Valiant 85; Papadimitriou & Yannakakis 86). SAT, Vertex Cover, Independent Set, Hamiltonian Path and 3-Coloring are NP-complete under \textbf{TIME(poly(log(n)))} reductions.

Theorem 4. For every $\delta < \frac{1}{2}$, there is no \textbf{TIME($n^\delta$)} reduction from Parity to MCSP. Hence MCSP is not AC0[2]-hard under \textbf{TIME($n^\delta$)} reductions.

Theorem 5. If MCSP is NP-hard under polytime reductions, then EXP $\neq$ NP $\cap$ P/poly. Consequently EXP $\neq$ ZPP.

Theorem 6. If MCSP is NP-hard under logspace reductions, then PSPACE $\neq$ ZPP.

Theorem 7. If MCSP is NP-hard under logtime-uniform AC0 reductions, then NP $\not\subset$ P/poly and E $\not\subset$ i.o.-SIZE($2^{\delta n}$) for some $\delta > 0$. As consequence P = BPP.

Proofs

Lemma 8 (Williams 2013). There is a universal $c \geq 1$ such that for any binary string $T$ and any substring $S$ of $T$, $CC(f_S) \leq CC(f_T) + c\log |T|$.

Theorem 9 (Håstad 86). For every $k \geq 2$, Parity cannot be computed by circuits with AND, OR and NOT gates of depth $k$ and size $2^{o(n^{1/(k-1)})}$.

Definition 10 (Cabanets & Cai 2000). A reduction from language $L$ to MCSP is \textbf{natural} if the size of all output instances and the size parameters $k$ depend only on length of the input to the reduction.

Claim 11. Let $\varepsilon > 0$. If there is \textbf{TIME($n^{1-\varepsilon}$)} reduction from Parity to MCSP, then there is \textbf{TIME($n^{1-\varepsilon} \log^2 n$)} natural reduction from Parity to MCSP. Furthermore, the value of $k$ in this natural reduction is $O(n^{1-\varepsilon}\text{poly}(\log(n)))$.

Claim 12. If there is a \textbf{TIME($n^{1-\varepsilon}$)} reduction from Parity to MCSP, then there is a $\Sigma_2 \text{TIME}(n^{1-\varepsilon}\text{poly}(\log(n)))$ algorithm for Parity.

Theorem 13. If every sparse language in NP has polytime reduction to MCSP, then EXP $\subseteq$ P/poly $\Rightarrow$ EXP = NEXP.