A Note on Computing the OR and AND Gate by a Probabilistic Depth-2 Circuit

Dejan Zivkovic
Department of Math. and Comp. Science
Savannah State College
Savannah, GA 31404
dzivkov@uscn.cc.uga.edu

September 22, 1994

Keywords: computational complexity, circuits complexity

1 Introduction

In [1] it was shown that every set in uniform-AC^0 can be recognized by a uniform family of depth-3 threshold circuits of size $n^{\log^{O(1)} n}$. The key step in the proof is simulation of the OR and AND gate on n inputs by a uniform depth-2 probabilistic circuits which consist of one PARITY gate at the root and AND gates on the first level. For every positive constant k, the circuit can be constructed so that its size is $n^{\log^{O(1)} n}$, the fan-in of AND gates is $O(\log^4 n)$, the number of random inputs is $O(\log^3 n)$, and, the error probability is less than $1/n^k$. In fact, in [2] the same kind of the circuit was built with the PARITY gate at the root replaced by a MOD_p gate for any prime p.

In this note we generalize the result and show how the simulation can be done with a MOD_m gate at the root of the circuit, for any integer $m \geq 2$. For this we use the idea of randomized polynomials introduced by Tarui [4]. Moreover, our approach gives a slightly simpler argument than the one given in [1] and [2].

2 Preliminaries

This section contains the necessary background and definitions. We assume the reader's familiarity with basic notions of the Boolean function and circuit theory (for more details see, for example, [6]).

If it does not cause confusion, we use the vector notation for a sequence of Boolean variables. For example, $x_{i_1}, \ldots, x_{i_\alpha}$ is denoted by $\vec{x}$ if the indices are clear from the context.

A *Boolean circuit* on n Boolean variables $x_1, \ldots, x_n$ is a directed acyclic graph with the following properties. Each node of fan-in zero, called *input*, is labeled with a variable, the negation of a variable, or the constants 0 or 1. The nodes of fan-in greater than zero, called *gates*, are labeled with (i.e., compute) some Boolean functions. Lastly, there is a single sink node of fan-out zero called *output*. The *size* of a circuit is the number of its gates, and the *depth* is the length of the longest path from an input to the output.

A Boolean circuit C on n inputs $x_1, \ldots, x_n$ computes in a natural way the Boolean function $\{0,1\}^n \to \{0,1\}$ denoted by $C(\vec{x})$. Namely, for every $\vec{b} = (b_1, \ldots, b_n) \in \{0,1\}^n$ we let $C(\vec{b})$ be the result of the output gate when the tuple $(b_1, \ldots, b_n)$ is presented to the corresponding inputs of the circuit C.

A *probabilistic* Boolean circuit C is a generalization of the standard circuit so that, in addition to the regular inputs $x_1, \ldots, x_n$, it has some further inputs $y_1, \ldots, y_r$, $(r \geq 0)$, which take values 0 or 1 according to some probability distribution. For each $\vec{x} \in \{0,1\}^n$ the $C(\vec{x}, \vec{y})$ is a random variable, and we say that the probabilistic circuit computes a Boolean function $f : \{0,1\}^n \to \{0,1\}$ with error ϵ if $Pr[f(\vec{x}) \neq C(\vec{x}, \vec{y})] \leq \epsilon$.

The output of a MOD_m gate is 1 if the sum of its binary inputs is 0 modulo m; otherwise, the output is 0. More precisely, for integers $m \geq 2$ and $\ell \geq 1$,

$$\mathrm{MOD}_m(x_1, \ldots, x_\ell) = \begin{cases} 1 & \text{, if } \sum_{i=1}^{\ell} x_i \equiv 0 \pmod{m} \\ 0 & \text{, otherwise} \end{cases}$$

A circuit family $\{C_n\}_{n \in N}$ is called uniform if the function $n \mapsto C_n$ can be computed in $\mathrm{DSPACE}(\log^{O(1)} n)$.

A randomized polynomial $P(\vec{x}, \vec{y}) = P(x_1, \ldots, x_n, y_1, \ldots, y_r)$ over a ring is a multivariate polynomial in the indeterminates $\vec{x}$ and $\vec{y}$ and coefficients

from the ring. The $x_1, \ldots, x_n$ represent Boolean variables, and $y_1, \ldots, y_r$ random variables which take values 0 or 1 according to some distribution. A randomized polynomial computes a Boolean function $f : \{0,1\}^n \to \{0,1\}$ with one-sided error ϵ if, for each $\vec{x} \in \{0,1\}^n$,

$$\begin{aligned} f(\vec{x}) = 0 &\implies Pr[P(\vec{x}, \vec{y}) \neq 0] \leq \epsilon \text{ , and} \\ f(\vec{x}) = 1 &\implies Pr[P(\vec{x}, \vec{y}) = 1] = 1 \text{ .} \end{aligned}$$

The size of a randomized polynomial is the number of monomials left after applying two operations on the polynomial. First, one blows up the polynomial by multiplying out its terms using the distributivity law as much as possible. Second, one simplifies the blow-up out using the commutativity and idempotency law. The degree of a randomized polynomial is the greatest degree of a monomial in the blow-up. (Observe that the size and degree take into account both regular and random indeterminates.)

The following two easy facts and a lemma from [5] will be needed later.

Fact 1 *For $a, b \in Z$ and $m \geq 2$,*

$$(a + b) \bmod m = (a \bmod m + b \bmod m) \bmod m$$

Fact 2 *For $a \in Z$, $b \in \{0,1\}$, and $m \geq 2$,*

$$(a \cdot b) \bmod m = (a \bmod m) \cdot b$$

Lemma 1 (Valiant and Vazirani) *Consider $\{0,1\}^\ell$ as the ℓ-dimensional vector space over GF(2), and let $*$ denote the inner product over GF(2). Suppose $S \subseteq \{0,1\}^\ell$ is a nonempty set and $v_1, \ldots, v_\ell$ are randomly chosen from $\{0,1\}^\ell$. Let $S_0 = S$ and let $S_i = \{u \in S |\ u * v_1 = \cdots = u * v_i = 0\}$ for each $i = 1, \ldots, \ell$. Then the probability that $|S_i| = 1$ for some $i \in \{0, 1, \ldots, \ell\}$ is at least $1/4$.*

3 Result

In order to avoid repetition, we will describe our result for a NOR gate and then indicate how to do an OR and AND gate.

Simulation of the NOR gate by a probabilistic depth-2 circuit is done in two steps. First, we construct a randomized polynomial which computes

$\mathrm{NOR}(x_1, \ldots, x_n)$ with small error. In the second step, we build the claimed circuit out of such a polynomial.

The following lemma, which is essentially proved in [4], accomplishes the first part of the idea outlined above.

Lemma 2 *For every $k > 0$, the function $NOR(x_1, \ldots, x_n)$ can be computed with one-sided error $1/n^k$ by a randomized polynomial $Q(x_1, \ldots, x_n, y_1, \ldots, y_r)$ over the integers that has size $2^{O(\log^4 n)}$, degree $O(\log^4 n)$, and $r = O(\log^3 n)$.*

Proof: Let $l = \lceil \log n \rceil$. Choose vectors $y_1, \ldots, y_l$ independently and uniformly at random from $\{0,1\}^l$. For each $i = 1, \ldots, l$ and $j = 1, \ldots, n$ define

$$z_{ij} = \begin{cases} 1 & \text{, if } b(j) * y_1 = \cdots = b(j) * y_i = 0 \\ 0 & \text{, otherwise} \end{cases} \tag{1}$$

where $*$ denotes the inner product as in Lemma 1, and $b(j)$ is the standard l-bit binary representation of $j - 1$. Put

$$R'(\vec{x}, \vec{z}) = [1-(x_1+\cdots+x_n)][1-(x_1z_{11}+\cdots+x_nz_{1n})]\cdots[1-(x_1z_{l1}+\cdots+x_nz_{ln})],$$

and

$$Q'(\vec{x}, \vec{z}) = R'_1(\vec{x}, \vec{z}^{(1)}) \cdot R'_2(\vec{x}, \vec{z}^{(2)}) \cdots R'_t(\vec{x}, \vec{z}^{(t)})$$

where $t = \lceil 3k \log n \rceil$, and each $R'_i(\vec{x}, \vec{z}^{(i)})$ is a copy of $R'(\vec{x}, \vec{z})$ with its own set of independent random vectors $y_1^{(i)}, \ldots, y_l^{(i)}$.

The randomized polynomial $Q'(\vec{x}, \vec{z})$ computes $\mathrm{NOR}(x_1, \ldots, x_n)$ with one-sided error $1/n^k$. Indeed, in case $\mathrm{NOR}(x_1, \ldots, x_n) = 1$, i.e., $x_1 = \cdots = x_n = 0$, clearly each $R'_i(\vec{x}, \vec{z}^{(i)}) = 1$, hence $Q'(\vec{x}, \vec{z}) = 1$. If $\mathrm{NOR}(x_1, \ldots, x_n) = 0$, i.e., $J = \{j \in \{1, \ldots, n\} | x_j = 1\} \neq \emptyset$, then by applying Lemma 1 with $S = \{b(j) \in \{0,1\}^l | j \in J\}$ we see that $Pr[R'_i(\vec{x}, \vec{z}^{(i)}) = 0] \geq 1/4$, for each $i = 1, \ldots, t$. Thus,

$$\begin{aligned} Pr[Q'(\vec{x}, \vec{z}) \neq 0] &= Pr[\forall i, (1 \leq i \leq t),\ R'_i(\vec{x}, \vec{z}^{(i)}) \neq 0] \\ &= \prod_{i=1}^{t} Pr[R'_i(\vec{x}, \vec{z}^{(i)}) \neq 0] \end{aligned}$$

$$\leq \left(\frac{3}{4}\right)^t \leq \left(\frac{3}{4}\right)^{3k\log n} = \left(\frac{27}{64}\right)^{k\log n}$$
$$< \left(\frac{1}{2}\right)^{k\log n} = \frac{1}{n^k} .$$

It can be easily seen that the polynomial $Q'(\vec{x},\vec{z})$ also satisfies the other requirements, except that the number of random variables $\vec{z}$ is too big—$O(n\log^2 n)$. A way to get around this problem is to observe that the random variables z_{ij} appear in $R'(\vec{x},\vec{z})$ just as place holders, and they actually depend on only l^2 random bits corresponding to the components of $y_1,\ldots,y_l$. If we denote the components by $y_{11},\ldots,y_{1l},\ldots,y_{l1},\ldots,y_{ll}$, then, for each (fixed) i and j, z_{ij} is the Boolean function $z_{ij}(y_{11},\ldots,y_{1l},\ldots,y_{l1},\ldots,y_{ll})$ whose value for a particular setting of the l^2 variables is given by (1). Thus, we can compute the DNF for each z_{ij} and write the polynomial corresponding in the obvious way to the DNF which exactly computes z_{ij}.

Now, if we replace each z_{ij} in $R'(\vec{x},\vec{z})$ by the polynomial corresponding to the z_{ij}'s DNF, then we get a new polynomial $R(\vec{x},\vec{y})$ which has only l^2 random variables. Also, if $Q(\vec{x},\vec{y})$ is as before made out of t independent copies of such $R(\vec{x},\vec{y})$, we achieve the claimed number $r = O(t\cdot l^2) = O(\log^3 n)$ of random variables in $Q(\vec{x},\vec{y})$. Furthermore, it is straightforward to check that the size and degree of $Q(\vec{x},\vec{y})$ is $2^{O(\log^4 n)}$ and $O(\log^4 n)$, respectively. ■

Next we use the randomized polynomial constructed in the previous lemma and show how to build the depth-2 probabilistic circuit.

Theorem *For every $k > 0$ and integer $m \geq 2$, the $NOR(x_1,\ldots,x_n)$ can be computed by a uniform depth-2 probabilistic circuit C consisting of one MOD_m-gate at the root and $2^{O(\log^4 n)}$ AND-gates with fan-in $O(\log^4 n)$, and having $O(\log^3 n)$ random inputs and error $1/n^k$.*

Proof: Let $Q(x_1,\ldots,x_n,y_1,\ldots,y_r)$ be the randomized polynomial as in Lemma 2. Denote by $P'(\vec{x},\vec{y})$ the polynomial which is obtained from the blown-up version of $Q(\vec{x},\vec{y})$ by algebraic simplifications. Let $P(\vec{x},\vec{y})$ be the polynomial which is obtained from the $P'(\vec{x},\vec{y})$ by reducing coefficients of its monomials modulo m. For each $\vec{x}\in\{0,1\}^n$ and $\vec{y}\in\{0,1\}^r$ we have

$$(P(\vec{x},\vec{y})) \bmod m = (Q(\vec{x},\vec{y})) \bmod m \qquad (2)$$

This is so because

$$
\begin{aligned}
(Q(\vec{x},\vec{y})) \bmod m &= (P'(\vec{x},\vec{y})) \bmod m \\
&= \Big(\sum c_\alpha \cdot y_{i_1} \cdots y_{i_\beta} \cdot x_{j_1} \cdots x_{j_\gamma}\Big) \bmod m \\
&= \Big(\sum \Big(c_\alpha \cdot y_{i_1} \cdots y_{i_\beta} \cdot x_{j_1} \cdots x_{j_\gamma}\Big) \bmod m\Big) \bmod m \\
&= \Big(\sum (c_\alpha \bmod m) \cdot y_{i_1} \cdots y_{i_\beta} \cdot x_{j_1} \cdots x_{j_\gamma}\Big) \bmod m \\
&= (P(\vec{x},\vec{y})) \bmod m
\end{aligned}
$$

The third equality follows from Fact 1 and the forth one from Fact 2; the others are by definition.

Now, the probabilistic circuit C for computing $\mathrm{NOR}(x_1,\ldots,x_n)$ is built as follows. Make an AND gate corresponding to each monomial in $P(\vec{x},\vec{y})$ and connect it to the MOD_m gate with the number of wires equal to the coefficient of monomial. Also, add $m-1$ extra wires to the MOD_m gate connected to the constant 1.

Clearly, the circuit C has all properties regarding the size, fan-in of AND gates, and number of random inputs as desired. Let us verify that the circuit C computes $\mathrm{NOR}(\vec{x})$ with the small error as well. If $\mathrm{NOR}(\vec{x}) = 1$, then $Q(\vec{x},\vec{y}) = 1$ by Lemma 2. So $Q(\vec{x},\vec{y}) \equiv 1 \pmod m$, hence $P(\vec{x},\vec{y}) \equiv 1 \pmod m$ by (2). Thus, the MOD_m gate outputs 1, i.e., $C(\vec{x},\vec{y}) = 1$ with probability 1. When $\mathrm{NOR}(\vec{x}) = 0$ we have

$$
\begin{aligned}
Pr[C(\vec{x},\vec{y}) = 1] &= Pr[\mathrm{MOD}_m \text{ gate outputs } 1] \\
&= Pr[P(\vec{x},\vec{y}) \equiv 1 \pmod m] \\
&= Pr[Q(\vec{x},\vec{y}) \equiv 1 \pmod m] \text{ , by (2)} \\
&\le Pr[Q(\vec{x},\vec{y}) \neq 0] \\
&\le 1/n^k \text{ , by Lemma 2.}
\end{aligned}
$$

As far as the uniformity is concerned, it may be easily checked that the above construction of the circuit C can be done in a polylogarithmic-space-uniform way. ■

Finally, we note that for computing $\mathrm{OR}(x_1,\ldots,x_n) = \neg\mathrm{NOR}(x_1,\ldots,x_n)$ one can use the polynomial $1-Q(\vec{x},\vec{y})$, and for computing $\mathrm{AND}(x_1,\ldots,x_n) = \mathrm{NOR}(\bar{x}_1,\ldots,\bar{x}_n)$ the polynomial $Q(1-x_1,\ldots,1-x_n,\vec{y})$.

4 Acknowledgment

I would like to thank Sorin Istrail for his encouragement and very useful discussions about the problem.

References

[1] E. Allender, "A Note on the Power of Threshold Circuits", *Proceedings of 30th IEEE Symposium on Foundations of Computer Science*, pp. 580–584, 1989.

[2] E. Allender and U. Hertrampf, "On the Power of Uniform Families of Constant Depth Threshold Circuits", *Proceedings of Symposium on Mathematical Foundations of Computer Science*, Springer-Verlag Lecture Notes in Computer Science vol. 452, pp. 158–164, 1990.

[3] R. Beigel and J. Tarui, "On ACC", *Proceedings of 32nd IEEE Symposium on Foundations of Computer Science*, pp. 783–792, 1991.

[4] J. Tarui, "Randomized Polynomials, Threshold Circuits, and the Polynomial Hierarchy", *Proceedings of Symposium on Theoretical Aspects of Computer Science*, Springer-Verlag Lecture Notes in Computer Science vol. 480, 1991.

[5] L. Valiant and V. Vazirani, "NP is as easy as detecting unique solutions", *Theoretical Computer Science 47*, pp. 85–93, 1986.

[6] I. Wegener, "The Complexity of Boolean Functions", *John Wiley & Sons*, 1987.