1-1-1988

Reliable Computation by Formulas in the Presence of Noise

Nicholas Pippenger
Harvey Mudd College

Reliable Computation by Formulas in the Presence of Noise

NICHOLAS PIPPENGER, SENIOR MEMBER, IEEE

Abstract—**It is shown that if formulas are used to compute Boolean functions in the presence of randomly occurring failures (as has been suggested by von Neumann and others), then 1) there is a limit strictly less than 1/2 to the failure probability per gate that can be tolerated, and 2) formulas that tolerate failures must be deeper (and, therefore, compute more slowly) than those that do not.**

I. Introduction

A COMMON way of computing Boolean functions is by means of Boolean networks or formulas. In 1952, von Neumann [4] showed how this method could be made to work with high probability even when each gate in the network or formula failed independently with some small probability. The main idea in von Neumann's construction is to interleave layers of gates that do computation with layers that do correction in such a way that error probabilities are kept under control. Two striking features of von Neumann's construction are 1) that it works only when the failure probability per gate is sufficiently small (in the case considered by von Neumann, it has to be less than 1/6, though in general it depends on the types of gates available), and 2) that computation proceeds more slowly than in the absence of failures since for a given failure probability a fixed fraction of the layers need to be devoted to correction. (No effort was made by von Neumann to estimate this fraction for this method. Some analysis of what can be achieved by this method will be given in Section III for purposes of comparison.)

The goal of this paper is to show that, at least in the case of formulas, both of these features are essential no matter what types of gates are available and no matter how they are interconnected. (A formula is simply a network in which the output of each gate is used as the input to at most one other gate.) The difficulty of proving this lies in the condition "no matter how they are interconnected." To appreciate this, consider that a formula that works correctly with high probability in the presence of randomly occurring failures need not work at all in the absence of failures. For computation, as distinct from communication, it is possible that noise may be useful as well as detrimental, as is illustrated by the apparent power of randomized algorithms.

Manuscript received October 1, 1986.
The author is with the IBM Almaden Research Laboratory K51-802, 650 Harry Road, San Jose, CA 95120-6099.
IEEE Log Number 8819893.

The heart of our proof is an information-theoretic argument that deals with computation and errors in very general terms. The strength of this argument is that it applies with equal ease no matter what types of gates are available. Its weakness is that it does not seem to predict quantitatively the limiting value of the failure probability or the ratio by which computation proceeds more slowly in the presence of failures. (In the case considered by von Neumann it shows only that the failure probability per gate must be less than 1/3, rather than 1/6. A comparison of computation speeds will be given in Section III.)

It remains an open question whether these features are essential for networks as well as formulas. Our proof breaks down for networks because the independence required at a key point in the proof need not hold. Most previous work (for example, [1], [2], and [5]) has dealt with networks rather than formulas and with size (which corresponds to cost) rather than depth (which corresponds to delay). Tomás Feder, a graduate student in the Computer Science Department at Stanford University, Stanford, CA, has recently extended the results of this paper from formulas to networks.

II. Main Result

Suppose that among the types of gates available there exists one that computes a Boolean function f that depends essentially on $k \geq 2$ arguments but none that depend essentially on more than k arguments. By d-fold composition of f with itself, we obtain a Boolean function f_d that depends essentially on $n = k^d$ arguments.

A formula may be regarded as a k-ary tree in which the leaves are either Boolean constants (0 or 1) or references to arguments $(x_1, x_2, \cdots)$ and in which the internal nodes are gates computing Boolean functions of k arguments. The *depth* of a formula is the number of gates on the longest path from an argument reference to the root; it reflects the delay between the time that the values of the arguments are presented and the time that the value of the function is produced. Clearly, f_d is computed (in the absence of failures) by a formula of depth d. We shall show that any formula that computes f_d with high probability in the presence of failures must be significantly deeper.

Theorem: Let F be a formula with the following property: if each leaf of F operates correctly, and if each gate

0018-9448/88/0300-0194$01.00

of F independently operates correctly with probability $1-\epsilon$, where $\epsilon > 0$, then F operates correctly (that is, produces the value of f_d) with probability at least $1-\delta$, where $\delta < 1/2$. Let

$$\Delta = 1 + \delta \log_2 \delta + (1-\delta)\log_2(1-\delta) > 0.$$

If $\epsilon < (k-1)/2k$, then the depth c of F satisfies

$$c \geq Rd - S$$

where

$$R = \frac{\log k}{\log((1-2\epsilon)k)}$$

and

$$S = \frac{\log(1/\Delta)}{\log((1-2\epsilon)k)}.$$

(Since $R > 1$, computation proceeds more slowly in the presence of failures.) If $\epsilon \geq (k-1)/2k$, then

$$d \leq T$$

where

$$T = \frac{\log \Delta}{\log(1-2\epsilon)}.$$

(Thus if ϵ is too large, only relatively simple functions can be computed reliably in the presence of failures.)

The proof is by way of information theory.

For $1 \leq m \leq n$, let F_m denote the following formula. Since f_d depends essentially on x_m, Boolean constants $c_1, \cdots, c_{m-1}, c_{m+1}, \cdots, c_n$ exist such that $f_d(c_1, \cdots, c_{m-1}, x_m, c_{m+1}, \cdots, c_n)$ is either x_m or its complement $\bar{x}_m$. Substitute these constants for all references to arguments other than x_m in F to obtain F_m.

Let X be a Boolean random variable that assumes the values 0 and 1 with equal probabilities. For any formula G, let $Y(G)$ be the Boolean random variable produced by G when X is substituted for all argument references in G, and each gate in G fails with probability ϵ, independently of X and of all other gates. By assumption, either $\Pr(Y(F_m) = X) \geq 1-\delta$ or $\Pr(Y(F_m) = \bar{X}) \geq 1-\delta$. In either case, we have by Fano's lemma

$$I(Y(F_m); X) \geq \Delta. \tag{1}$$

(See [3, theorem 4.1] for Fano's lemma.)

If G is a formula, let $\Phi(G, \xi)$ be the generating function for the paths from argument references to the root in G, classified according to the number of gates they contain. That is, if $L(P)$ denotes the number of gates on the path P, then

$$\Phi(G, \xi) = \sum_P \xi^{L(P)}.$$

Inequality (1) gives a lower bound to $I(Y(F_m); X)$ in terms of δ. To exploit it we shall prove an upper bound in terms of ϵ, namely,

$$I(Y(F_m); X) \leq \Phi(F_m, 1-2\epsilon). \tag{2}$$

For now, let us see how (1) and (2) imply the Theorem.

Combining (1) and (2) yields

$$\Phi(F_m, 1-2\epsilon) \geq \Delta, \tag{3}$$

and summing over m in the range $1 \leq m \leq n$ yields

$$\Phi(F, 1-2\epsilon) \geq n\Delta \tag{4}$$

since

$$\Phi(F, \xi) = \sum_{1 \leq m \leq n} \Phi(F_m, \xi). \tag{5}$$

We shall need the inequality

$$\Phi(F, 1/k) \leq 1. \tag{6}$$

To prove this we shall show that

$$\Phi(G, 1/k) \leq 1 \tag{7}$$

for any k-ary formula G. We shall proceed by induction on the structure of G. If G is a constant leaf, $\Phi(G, \xi) = 0$. If G is an argument reference leaf, $\Phi(G, \xi) = \xi^0 = 1$. For the inductive step, suppose that the root of G is a gate with subformulas $G_1, \cdots, G_k$. Then

$$\Phi(G, \xi) = \xi \sum_{1 \leq j \leq k} \Phi(G_j, \xi). \tag{8}$$

Thus

$$\Phi(G, 1/k) = (1/k) \sum_{1 \leq j \leq k} \Phi(G_j, 1/k).$$

Applying the inductive hypothesis to each term on the right completes the proof of (7). (Inequality (7) is equivalent in content to the Kraft–Szilard inequality; see [3, lemma A.1].) From (5) and (6) it follows that for some m in the range $1 \leq m \leq n$ we have

$$\Phi(F_m, 1/k) \leq 1/n. \tag{9}$$

Suppose first that $\epsilon < (k-1)/2k$. Setting $\lambda = \log_{1/k}(1-2\epsilon)$, we have $0 < \lambda < 1$. Hölder's inequality (see [3, lemma C.4]) reads

$$\sum_P A_P B_P \leq \left(\sum_P A_P^{1/\lambda}\right)^{\lambda} \left(\sum_P B_P^{1/(1-\lambda)}\right)^{1-\lambda}.$$

Putting $A_P = (1-2\epsilon)^{L(P)}$ and $B_P = 1$, and letting P run over paths from argument references to the root in F, yields

$$\Phi(F, 1-2\epsilon) \leq \Phi(F, 1/k)^{\lambda} \Phi(F, 1)^{1-\lambda}.$$

Bounding $\Phi(F, 1-2\epsilon)$ below by (4) and bounding $\Phi(F, 1/k)$ above by (6) yields

$$\Phi(F, 1) \geq (n\Delta)^{1/(1-\lambda)}. \tag{10}$$

The left side, $\Phi(F, 1)$, is simply the number of argument reference leaves in F; if F has depth c, this is at most k^c. On the other hand, $n = k^d$. Substituting these relations in (10) and taking logarithms to base k yields the first part of the theorem.

Now suppose that $\epsilon \geq (k-1)/2k$. Setting $\mu = \log_{1-2\epsilon}(1/k)$, we have $0 < \mu \leq 1$. Jensen's inequality (see [3, lemma C.6]) reads

$$\sum_P A_P \geq \left(\sum_P A_P^{1/\mu}\right)^{\mu}.$$

Putting $A_P = (1/k)^{L(P)}$ and letting P run over all paths from argument reference leaves to the root in F_m yields

$$\Phi(F_m, 1/k) \geq \Phi(F_m, 1-2\epsilon)^{\mu}.$$

Bounding $\Phi(F_m, 1/k)$ above by (9) and $\Phi(F_m, 1-2\epsilon)$ below by (3) yields

$$1/n \geq \Delta^{\mu}.$$

Substituting $n = k^d$ and taking logarithms to base k yields the second part of the theorem.

It remains to prove (2). To do this we shall show that

$$I(Y(G); X) \leq \Phi(G, 1-2\epsilon) \tag{11}$$

for any formula G. We shall proceed by induction on the structure of G. If G is a constant leaf, $I(Y(G); X)$ and $\Phi(G, 1-2\epsilon)$ both vanish. If G is an argument reference leaf, then $I(Y(G); X) = H(X)$ and $\Phi(G, 1-2\epsilon) = (1-2\epsilon)^0$ both equal unity. For the inductive step, suppose that the root of G is a gate that computes a Boolean function g of the values produced by the subformulas $G_1, \cdots, G_k$. Since G is a formula, the random variables $Y(G_1), \cdots, Y(G_k)$ are conditionally independent given the value of X (since in this case they depend only on the failures, which are independent in the disjoint subformulas $G_1, \cdots, G_k$). Thus we have

$$I(Y(G_1), \cdots, Y(G_k); X) \leq \sum_{1 \leq j \leq k} I(Y(G_j); X).$$

(See [3, lemma 6.3] for this inequality.) By the data processing theorem,

$$\begin{aligned} I(g(Y(G_1), \cdots, Y(G_k)); X) &\leq I(Y(G_1), \cdots, Y(G_k); X) \\ &\leq \sum_{1 \leq j \leq k} I(Y(G_j); X). \end{aligned}$$

(See [3, theorem 4.2] for this theorem.) Now $Y(G) = g(Y(G_1), \cdots, Y(G_k)) + W$, where "+" denotes the sum modulo 2 ("exclusive or"), and W is a Boolean random variable that assumes the value 1 with probability ϵ and the value 0 with probability $1-\epsilon$, independently of X and $g(Y(G_1), \cdots, Y(G_k))$. By the following lemma we have

$$\begin{aligned} I(Y(G); X) &\leq (1-2\epsilon) I(g(Y(G_1), \cdots, Y(G_k)); X) \\ &\leq (1-2\epsilon) \sum_{1 \leq j \leq k} I(Y(G_j); X). \end{aligned}$$

Applying the inductive hypothesis to each term on the right and using (8) completes the proof of (11). It remains to prove the following lemma.

Lemma: Let X and V be Boolean random variables, and let $Y = V + W$, where W is a Boolean random variable that assumes the value 1 with probability ϵ and the value 0 with probability $1-\epsilon$, independently of X and V. Then

$$I(Y; X) \leq (1-2\epsilon) I(V; X).$$

Proof: Let U be a Boolean random variable that assumes the values 0 and 1 with equal probabilities, independently of X and V. Let W' be a Boolean random variable that assumes the value 1 with probability 2ϵ and the value 0 with probability $1-2\epsilon$, independently of X, V, and U. Let the random variable Y' equal U if $W' = 1$ and equal V if $W' = 0$. Then X, V, and Y' have the same joint distribution as X, V, and Y, so it will suffice to show that

$$I(Y'; X) \leq (1-2\epsilon) I(V; X).$$

This is done by an easy calculation:

$$\begin{aligned} I(Y'; X) &\leq I((Y', W'); X) \\ &= I(Y'; X|W') + I(W'; X) \\ &= \Pr(W' = 0) I(V; X) \\ &\quad + \Pr(W' = 1) I(U; X) + I(W'; X). \end{aligned}$$

The first term is $(1-2\epsilon) I(V; X)$, and the remaining two terms vanish since U and W' are independent of X.

III. Conclusion

We have shown that if k-ary formulas are to compute arbitrarily complex functions with error probability bounded below $1/2$, the failure probability per gate must be bounded below $(k-1)/2k < 1/2$. This may be surprising considering that a binary symmetric channel (which corresponds to a unary identity gate in our model) with failure probability ϵ has capacity $1 + \epsilon \log_2 \epsilon + (1-\epsilon) \log_2 (1-\epsilon)$, which remains positive for $\epsilon < 1/2$. The reason for this discrepancy is evidently that when discussing reliable communication in the presence of noise, the definition of capacity allows unlimited computation without errors by the encoder and decoder; for reliable computation in the presence of noise, however, any encoding or decoding must be done by components that are themselves subject to failures. Only when the errors introduced by a gate can be offset by the computation performed by that gate can reliable computation in the presence of noise be done.

Finally, we shall give a quantitative comparison of the negative results in this paper with the positive results implicit in von Neumann's paper. The case considered throughout the greater part of von Neumann's paper is that in which errors are corrected by three-way voting using a three-argument majority gate. If the inputs to such a gate are independently incorrect with probability ξ and if the gate fails with probability ϵ, then the output is incorrect with probability $f(\xi)$, where

$$f(\xi) = \epsilon + (1-2\epsilon)(3\xi^2 - 2\xi^3).$$

The error probability δ must be greater than the smallest solution $q(\epsilon)$ of the equation $f(\xi) = \xi$. We have $q(\epsilon) < 1/2$ if and only if $\epsilon < 1/6$. For $k = 3$, however, our result only gives $\epsilon < 1/3$, and it remains an open question whether any scheme can achieve $\delta < 1/2$ for $1/6 \leq \epsilon < 1/3$.

To compare bounds on the speed of computation, we must determine the fraction of layers that must be devoted to correction. This fraction depends on what is done by the layers devoted to computation. The simplest assumption is that the gates in the layers devoted to computation are three-argument parity gates (a discussion of the general case will be given later). If the inputs to such a gate are independently incorrect with probability ξ and the gate fails with probability ϵ, then the output is incorrect with

probability $g(\xi)$, where

$$g(\xi) = \epsilon + (1-2\epsilon)(3\xi - 6\xi^2 + 4\xi^3).$$

Let us consider first the case $\epsilon \to 0$. The order of magnitude of $q(\epsilon)$ is ϵ. The effect of applying f is roughly to square its argument (it also triples the result and adds ϵ, but these effects are less important). Thus a correction should be applied when the error probability has order of magnitude $\epsilon^{1/2}$. The effect of applying g is roughly to triple its argument (again, it also adds ϵ, but this effect is less important). Thus the number of layers of computation that can be put between one correction layer (after which the error probability has order of magnitude ϵ) and another (before which it has order of magnitude $\epsilon^{1/2}$) is asymptotic to $\log_3(\epsilon^{1/2}/\epsilon) = (1/2)\log_3(1/\epsilon)$. Thus a fraction asymptotic to $2/\log_3(1/\epsilon)$ of the layers must be devoted to correction. Our result says only that this fraction must be at least asymptotic to $2\epsilon/\ln 3$, which vanishes much more rapidly as $\epsilon \to 0$.

Let us now consider the case $\epsilon \to 1/6$. To do this we write $\epsilon = 1/6 - \vartheta$ and let $\vartheta \to 0$. In this case $q(\epsilon) \to 1/2$, so we write $h(\xi) = 1/2 - \xi$ and find that $h(q(\epsilon))$ has order of magnitude $\vartheta^{1/2}$. The effect of applying $h \circ g \circ h$ is roughly to cube its argument, so after a computation layer the error probability has order of magnitude $\vartheta^{3/2}$. The effect of applying $h \circ f \circ h$ is roughly to multiply its argument by $f'(1/2) = 1 + 3\vartheta$. Thus the number of correction layers that need to be put between one computation layer (after which the error probability has order of magnitude $\vartheta^{3/2}$) and another (before which it has order of magnitude $\vartheta^{1/2}$) is asymptotic to $\log_{1+3\vartheta}(\vartheta^{1/2}/\vartheta^{3/2}) \sim (1/3\vartheta)\ln(1/\vartheta)$. Thus only a fraction asymptotic to $3\vartheta/\ln(1/\vartheta)$ of the layers can be devoted to computation. Our result says only that at most a fraction $\log_3 2$ of the layers can be devoted to computation.

The analysis just given can be extended to cases in which gates other than parity gates appear in the layers devoted to computation, but some care is needed. For parity gates, increasing the error probability of an input increases the error probability of the output, but in general this need not be the case. In the absence of such monotonicity, one must maintain lower bounds to error probability as well as upper bounds. When this is done, however, it turns out that parity constitutes the worst case, even when all three-argument gates are available. In other special cases (for example, when only majority and minority gates are available) more favorable bounds can be obtained.

It remains an open problem to obtain stronger negative results than those given here. In particular, it would be interesting to prove either 1) that reliable computation of arbitrarily complex functions cannot be done by three-argument gates if $\epsilon \geq 1/6$, or 2) that the fraction of layers that must be devoted to correction must have order of magnitude $1/\ln(1/\epsilon)$ for small ϵ.

REFERENCES

[1] R. L. Dobrushin and S. I. Ortyukov, "Upper bound for the redundancy of self-correcting arrangements of unreliable functional elements," *Prob. Inform. Transm.*, vol. 13, pp. 203–218, 1977.

[2] ——, "Lower bound for the redundancy of self-correcting arrangements of unreliable functional elements," *Prob. Inform. Transm.*, vol. 13, pp. 59–65, 1977.

[3] F. Jelinek, *Probabilistic Information Theory*. New York: McGraw-Hill, 1968.

[4] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies*, C. E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, 1956, pp. 43–98.

[5] N. Pippenger, "On networks of noisy gates," in *Proc. IEEE Symp. Found. of Comp. Sci.*, vol. 26, 1985, pp. 30–38.