Probabilistic Evaluation of Common Functions on Rings of Known Size**

Karl Abrahamson*
Andrew Adler†
Lisa Higham‡
David Kirkpatrick‡
Technical Report 88-15
June 1988

* Department of Computer Science
Washington State University
Pullman, WA. 99164 1210
U.S.A.

†Department of Mathematics
‡Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1W5
Canada

** This research was supported in part by the Natural Sciences and Engineering Research Council of Canada.

Abstract

In [5], Duris and Galil prove an $\Omega(n \log n)$ lower bound for the average number of messages required by any *deterministic* algorithm which elects a leader on an asynchronous ring with distinct identifiers where ring size n is known and is a power of 2. Their results imply the same lower bound for the expected complexity of any *randomized* leader election algorithm for an anonymous ring of known size $n = 2^k$. If their new techniques are used to achieve the randomized result directly, the resulting proof is significantly simplier than the original deterministic one. This simplicity facilitates extension of the result in two directions; namely, for arbitrary known ring size, and for algorithms that permit error with probability at most ϵ. Specifically, we prove that the expected message complexity of any probabilistic algorithm that selects a leader with probability at least $1 - \epsilon$ on an anonymous ring of known size n, is $\Omega(n \min(\log n, \log\log(1/\epsilon)))$. A number of common function evaluation problems (including AND, OR, PARITY, and SUM) on rings of known size, are shown to inherit this complexity bound and that the bound is tight to within a constant factor.

1 Introduction and background

Leader election is the problem of distinguishing a single processor from among all the processors of a distributed network. This problem has been recognized as a fundamental problem in distributed computing because it forms a building block for many more involved algorithms. The problem has been extensively studied, with particular attention being paid to leader election on an asynchronous ring of processors. Within this model, many versions of the problem arise depending upon whether or not the processors have distinct identifiers and whether or not the ring size is known.

On an (even unidirectional) ring of size n with distinct identifiers taken from a set ID, a leader can be elected by a deterministic algorithm using $O(n \log n)$ messages [4,8]. Furthermore, $\Omega(n \log n)$ messages are required in the worst case [3] for a large enough ID set. Pachl showed in [7] that this lower bound extends to the average case (averaging over all possible ID sequences), as long as the algorithm is required to work for a large range of ring sizes. Since Pachl proceeds by constructing new rings of various sizes and insisting that the algorithm still works, his proof does not apply if the ring size is fixed. In [5], Duris and Galil address this remaining situation. They show that if ring size is known and is a power of two and the identifier set is "large enough", then the complexity of deterministic leader election, averaged

over all labelings of the ring with distinct identifiers, remains $\Omega(n \log n)$ messages.

Without any identifiers, (i.e., an *anonymous* ring), no deterministic algorithm can elect a leader. However, when ring size is known to within a factor of two, randomization can be employed to elect a leader on an anonymous ring in $O(n \log n)$ expected bits, [1]. Furthermore, $\Omega(n \log n)$ bits are required even in the best case when ring size is only known to within a constant factor [1].

In [1] it is shown that leader election can be decomposed into two problems called attrition and solitude verification. Attrition is the problem of reducing the original collection of contenders for leadership to exactly one contender. Solitude verification is the problem of confirming that only one contender remains. The $\Omega(n \log n)$ bit lower bound for randomized leader election in [1] is actually achieved by proving this lower bound for solitude verification. But when ring size is known exactly, the complexity of solitude verification drops to $\Theta(n\sqrt{\log n})$ bits [2]. The techniques of [2] yield lower bounds that hold even for best case computations. Since there are randomized procedures for attrition that have bit complexity at most $O(n \log \log n)$ in the best case when ring size is known, [6], these techniques cannot provide an $\Omega(n \log n)$ lower bound for attrition or leader election on rings of known size. Hence a gap remains between the upper and lower bounds for the expected communication complexity of randomized leader election when ring size is fixed.

Notice that this gap is closed by the Duris and Galil result if ring size is a power of two. Let α be a randomized leader election algorithm for an anonymous ring of size $n = 2^k$. Suppose that with very high probability no processor uses more than $f(n)$ random bits when running α. Now consider the class $\mathcal{R}$ of rings of size n with distinct identifiers taken from the interval $[1, 2^{f(n)}]$. A deterministic algorithm β for $\mathcal{R}$ can be constructed from α by using the bits of the processor identifiers in place of the random bits. In the rare event that some processor requires more pseudo-random bits than provided by the identifiers, β proceeds by running any $O(n \log n)$ messages deterministic leader election algorithm. Thus α must have complexity $\Omega(n \log n)$ expected messages since otherwise algorithm β would contradict the $\Omega(n \log n)$ messages lower bound for deterministic leader election.

Alternatively, an $\Omega(n \log n)$ expected message complexity for randomized attrition can be proved directly using the techniques introduced by Duris and Galil. In the deterministic model, counting arguments are used to ensure that the characteristic of distinct identifiers is maintained. In the randomized model these combinatorial techniques are not needed. As a consequence, the proof of a lower bound of $\Omega(n \log n)$ expected messages for randomized attrition when n is a power of two, is substantially simpler than the corresponding proof of a lower bound of $\Omega(n \log n)$ messages on average for deterministic leader election. This simplicity facilitates extension of the result in two directions; namely, for known ring sizes that are not necessarily a power of two, and for algorithms that permit error with probability at most ϵ. Denote an attrition procedure that deadlocks with probability at most ϵ by ϵ-*attrition*. It is shown in section 3 that every ϵ-attrition procedure for rings of known size n has expected complexity $\Omega\left(n \min\left\{\log n, \log\log\left(1/\epsilon\right)\right\}\right)$messages.

Section 2 outlines the model of computation used for the lower bound result and reviews the definitions of relevant terms.

Attrition is a subproblem of a number of common functions such as AND, OR, PARITY, and SUM as well as of leader election. For example, a probabilistic AND algorithm that errs with probability at most ϵ can be converted into a ϵ-attrition procedure. Section 4 contains a reduction from ϵ-attrition to probabilistic AND that permits the ϵ-attrition lower bound to extend to AND. Similar reductions extend the lower bound to several other functions. Section 4 also describes probabilistic algorithms for these problems, which demonstrate that the lower bound is tight. A large number of problems including AND, OR, PARITY, SUM and leader election are thus shown to have expected complexity $\Theta\left(n \min\left\{\log n, \log\log\left(1/\epsilon\right)\right\}\right)$messages on rings of known size n.

2 Model

An attrition procedure is required to reduce the original n contenders on a ring to exactly one while ensuring that all contenders are not eliminated. The definition of attrition can be generalized to permit deadlock with low probability. A procedure is an ϵ-*attrition* procedure if

it deadlocks with probability at most ϵ and all computations that do not deadlock eventually reduce the number of contenders on the ring to exactly one. Nondeadlocking computations of ϵ-attrition do not terminate. Therefore the complexity of ϵ-attrition is defined to be the expectation over all nondeadlocking computations of the number of messages sent until exactly one contender remains.

In the natural description of randomized ϵ-attrition for an anonymous ring, each process runs the same randomized process. A processor's next state and next output message is determined by its current state, its last input message, and the result of a random experiment. Random choices occur throughout the run of the algorithm. But these random choices can be simulated by a single random choice by each processor at the beginning of the algorithm. A processor randomly chooses a function from "internal state, input message" pairs to "internal state, output message" pairs. (Essentially, a processor preselects all its random coin tosses.) The resulting model pushes all the randomization to a single random experiment by each processor at the beginning of the computation. The rest of the computation proceeds deterministically. Hence, a randomized distributed procedure for ϵ-attrition on an anonymous ring is modelled as a probability space of deterministic processes available for assignment to processors on the ring.

3 Lower Bounds for ϵ-attrition

This section bounds the expected message complexity of ϵ-attrition. It will be shown that any ϵ-attrition procedure has expected message complexity $\Omega\left(n \min\left\{\log n, \log\log\left(1/\epsilon\right)\right\}\right)$. The expected message complexity of nondeadlocking attrition, $(\Omega(n \log n))$, is derived from the general result by setting the allowable probability of deadlock, ϵ, to less than $1/2^n$.

The proof uses two techniques that are adapted from those introduced by Duris and Galil in [5]. The first technique argues that expected computation for parts of the ϵ-attrition procedure cannot be too low because otherwise deadlock will occur, under a specified scheduler, with intolerably high probability. This is the essence of lemma 3.1. The second technique, used here in theorem 3.2, sums these expected message complexities for disjoint parts of the ϵ-

attrition procedure to get a lower bound on total expected computation. Since the techniques are applicable to bidirectional rings, the lower bound is presented in that generality.

Definitions and Notation:

Let α be an ϵ-attrition procedure for anonymous rings of known size n. Let $\mathcal{P}$ be the probability space of deterministic processes associated with α. Let $\mathcal{R}$ be a ring of n processes from $\mathcal{P}$ and let x be any subsequence of processes in $\mathcal{R}$. Imagine blocking the two links on either end of subsequence x, and running α on $\mathcal{R}$. Denote by $c(x)$, the total number of messages that are sent over links internal to x while the two links bounding x remain blocked. Let x and y be two adjacent sequences on $\mathcal{R}$, with blocks placed before sequence x, between x and y, and after sequence y. In an extension of the above notation, let $c(x|y)$ denote the number of *additional* messages sent on sequence xy after removal of the block between x and y while the links at either end of xy remain blocked. Let $|x|$ denote the length of subsequence x. If $|x| = l$, then x is called an *l-process*.

$\mathcal{P}^l$ denotes the product space formed from l copies of $\mathcal{P}$ together with the induced product probabilities.

Lemma 3.1: Let x and y be two random l-processes from $\mathcal{P}^l$ and let z and w be two random $(l+1)$-processes from $\mathcal{P}^{l+1}$, where l is an integer satisfying $n = 2dl + r$ for some integers d and $0 \leq r \leq 2d - 1$. Then one of the following is true.

$$\mathrm{E}\left(c(x|y)\right) \geq \left(1 - \epsilon^{l/n}\right)^2 \frac{l}{32} \tag{3.1}$$

$$\mathrm{E}\left(c(x|z)\right) \geq \left(1 - \epsilon^{l/n}\right)^2 \frac{l}{32} \tag{3.2}$$

$$\mathrm{E}\left(c(z|x)\right) \geq \left(1 - \epsilon^{l/n}\right)^2 \frac{l}{32} \tag{3.3}$$

$$\mathrm{E}\left(c(z|w)\right) \geq \left(1 - \epsilon^{l/n}\right)^2 \frac{l}{32} \tag{3.4}$$

Proof: There are two cases, depending on the size of the remainder, r.

Case 1: $r \leq d$. It will be shown that one of the following is true:

$$\mathrm{E}\left(c(x|y)\right) \geq \left(1 - \epsilon^{l/n}\right)^2 \frac{l}{32}$$

$$\mathrm{E}\left(c(x|z)\right) \geq \left(1-\epsilon^{l/n}\right)^2 \frac{l}{32}$$

$$\mathrm{E}\left(c(z|x)\right) \geq \left(1-\epsilon^{l/n}\right)^2 \frac{l}{32}$$

Let $\lambda = 1 - \min\left\{\Pr(c(x|y) \leq \frac{l}{2}), \Pr(c(x|z) \leq \frac{l}{2}), \Pr(c(z|x) \leq \frac{l}{2})\right\}$. Then $\Pr(c(x|y) \leq \frac{l}{2}) \geq 1-\lambda$ and $\Pr(c(x|z) \leq \frac{l}{2}) \geq 1-\lambda$ and $\Pr(c(z|x) \leq \frac{l}{2}) \geq 1-\lambda$. Let $S_1 = \{s | s \in A^l \wedge \Pr(c(s|y) \leq \frac{l}{2}) > 1-\lambda^{1/2}\}$. Then $\Pr(S_1) \geq 1-\lambda^{1/2}$ since otherwise $\Pr(c(x|y) \leq \frac{l}{2}) < (1-\lambda^{1/2}) + \lambda^{1/2}(1-\lambda^{1/2}) = 1-\lambda$. Similarly, let:

$$S_2 = \{s | s \in A^l \wedge \Pr(c(y|s) \leq \frac{l}{2}) > 1-\lambda^{1/2}\}$$

$$S_3 = \{s | s \in A^l \wedge \Pr(c(s|z) \leq \frac{l}{2}) > 1-\lambda^{1/2}\}$$

$$S_4 = \{s | s \in A^l \wedge \Pr(c(z|s) \leq \frac{l}{2}) > 1-\lambda^{1/2}\}$$

Then, in the same way, $\Pr(S_i) \geq 1-\lambda^{1/2}$ for $i = 2, 3$ and 4. Let $C = S_1 \cap S_2 \cap S_3 \cap S_4$. Then $\Pr(C) \geq 1 - 4\lambda^{1/2}$.

Now consider the class of rings, B, with length n, defined by:

$$B = \Bigg\{ x_1, \ldots, x_{2d} \,\Big|\, \begin{array}{l} x_i \in C \text{ for } i = 1, 3, \ldots, 2d-1, \\ x_i \in A^{l+1} \text{ for } i = 2, 4, \ldots, 2r, \\ x_i \in A^l \text{ for } i = 2r+2, , \ldots, 2d, \\ \text{and } c(x_i|x_{i+1}) \leq \frac{l}{2}, \text{ and } c(x_{2d}|x_1) \leq \frac{l}{2} \end{array} \Bigg\}$$

Then $\Pr(B)$ in the product space $\mathcal{P}^n$ can be bounded as follows. Since a random l-process is in C with probability at least $1 - 4\lambda^{1/2}$, all x_i, for $i = 1, 3, \ldots, 2d-1$, are in C with probability at least $(1-4\lambda^{1/2})^d$. Given $x_i \in C$ for $i = 1, 3, \ldots, 2d-1$, $c(x_i|x_{i+1}) \leq \frac{l}{2}$ with probability at least $1-\lambda^{1/2}$ and, $c(x_{i+1}|x_{i+2}) \leq \frac{l}{2}$ with probability at least $1-\lambda^{1/2}$. Hence for a fixed $i = 2, 4, \ldots, 2d$, the conditions on B are met with probability at least $1-2\lambda^{1/2}$. Hence $\Pr(B) \geq (1-4\lambda^{1/2})^d(1-2\lambda^{1/2})^d > (1-4\lambda^{1/2})^{\lfloor n/l \rfloor}$.

Imagine the following scheduler on any element of B. Block the links between each adjacent pair (x_i, x_{i+1}) and (x_{2d}, x_1), and run α until all remaining messages are queued at

the blocks. The removal of any block accounts for at most $l/2$ additional messages. It is therefore impossible for messages generated by the removal of any pair of blocks to interact. Hence after removal of all blocks there are at most $(l/2)(n/l)$ additional message before all message traffic ceases. So elements of B produce deadlocking ϵ-attrition computations under this scheduler.

Since deadlock occurs with probability at most ϵ, $\Pr(B) \leq \epsilon$, which implies that $(1 - 4\lambda^{1/2})^{\lfloor n/l \rfloor} < \epsilon$. Thus $\lambda > (1 - \epsilon^{l/n})^2/16$. But from the definition of λ, either $\Pr(c(x|y) \leq \frac{l}{2}) = 1 - \lambda$ or $\Pr(c(x|z) \leq \frac{l}{2}) = 1 - \lambda$ or $\Pr(c(z|x) \leq \frac{l}{2}) = 1 - \lambda$. Hence either $\mathrm{E}(c(x|y)) \geq \lambda \cdot \frac{l}{2} > (1 - \epsilon^{l/n})^2 \frac{l}{32}$ or $\mathrm{E}(c(x|z)) > (1 - \epsilon^{l/n})^2 \frac{l}{32}$ or $\mathrm{E}(c(z|x)) > (1 - \epsilon^{l/n})^2 \frac{l}{32}$.

Case 2: $r \geq d$. In this case one of the following is true:

$$\mathrm{E}(c(z|w)) \geq \left(1 - \epsilon^{l/n}\right)^2 \frac{l}{32}$$

$$\mathrm{E}(c(z|x)) \geq \left(1 - \epsilon^{l/n}\right)^2 \frac{l}{32}$$

$$\mathrm{E}(c(x|z)) \geq \left(1 - \epsilon^{l/n}\right)^2 \frac{l}{32}$$

In case 1, the set C is composed of l-processes that with high probability communicate few additional messages when juxtaposed on the left or the right with a random l-process or a random $l+1$-process. A member of the set B of deadlocking rings of size n is constructed from r alternations of elements of C and appropriate $l+1$ processes followed by $d-r$ alternations of elements of C and appropriate l-processes. Thus the remainder, r, in $n = 2dl + r$ where $r \leq d$, is absorbed by the r $l+1$-processes. In case 2, the set corresponding to C is composed of $l+1$-processes that with high probability communicate few additional messages when juxtaposed on the left or the right with a random l-process or a random $l+1$-process. A member of the set corresponding to B is constructed from $r-d$ alternations of elements of C and appropriate $l+1$ processes followed by $2d - r$ alternations of elements of C and appropriate l-processes. Thus the remainder, r, in $n = 2dl + r$ where $r \geq d$, is absorbed by the d elements of C and the additional $r - d$ $l+1$-processes. The details of the proof of this case mimic those is case 1, and are therefore omitted. ∎

Theorem 3.2: Every ϵ-attrition procedure for rings of fixed size n has expected message complexity $\Omega(n \min\{\log n, \log\log(1/\epsilon)\})$ on rings of size n.

Proof: Let α be an ϵ-attrition procedure for rings of fixed size n and let $\mathcal{P}$ be the probability space of processes available to α. A schedule for α is constructed which proceeds in rounds. Each round includes message traffic between adjacent segments of the ring of length l such that l satisfies the conditions of lemma 3.1. Hence at least one of the four equations (3.1) through (3.4) holds. Each segment participating in round i, is subdivided in such a way that it contains two round $i-1$ segments. The total expected communication is calculated by summing, over all rounds, the expected communication in each round.

Let $\mathcal{R} = \pi_1, \ldots, \pi_n$ be a random element of $\mathcal{P}^n$. Place one *level 0 block* between processors π_n and π_1. Define the set of *level 0 segments* to contain just the single sequence $\pi_1, \ldots, \pi_n$. Let $d_1 = 2$ and $d_i = 2d_{i-1} + 1$ for $i \geq 2$. From each level $i-1$ segment, $\pi_g, \ldots, \pi_h$, create two level i segments as follows. Set $l_i = \lfloor n/(2d_i) \rfloor$. Then $n = 2d_i l_i + r$ where $0 \leq r \leq 2d_i - 1$. So at least one of equations (3.1), (3.2), (3.3), or (3.4) from lemma 3.1, holds for $l = l_i$. If (3.1) holds, set $a_i = b_i = l_i$. Similarly, set

$$\begin{aligned} a_i = l_i \text{ and } b_i = l_i + 1 \quad & \text{if equation (3.2) holds} \\ a_i = l_i + 1 \text{ and } b_i = l_i \quad & \text{if equation (3.3) holds} \\ a_i = l_i + 1 \text{ and } b_i = l_i + 1 \quad & \text{if equation (3.4) holds} \end{aligned}$$

Place a *level i block* between processors π_{g+a_i-1} and π_{g+a_i}. Place a *level i-1 block* between processors $\pi_{g+a_i+b_i-1}$ and $\pi_{g+a_i+b_i}$. The sequences $\pi_g, \ldots, \pi_{g+a_i-1}$ and $\pi_{g+a_i}, \ldots, \pi_{g+a_i+b_i-1}$ are the level i sequences derived from $\pi_g, \ldots, \pi_h$. Henceforth ignore the subsequence $\pi_{g+a_i+b_i}$ through π_h.

Continue to create two higher level segments within each current level segment until some segment has length less than or equal to two.

Since $d_1 = 2$ and $d_i = 2d_{i-1} + 1$, it follows that $d_i = 3 \cdot 2^{i-1} - 1$. So $l_i = \left\lfloor \frac{n}{2 \cdot d_i} \right\rfloor$ implying $l_i > \frac{n}{2^{i+3}}$, and there are at least $\log n - 4$ levels. Since the number of segments doubles at each level, and level 1 has one pair of segments, level i has 2^{i-1} pairs of segments.

In each round, the scheduler removes all blocks with the currently highest level number and runs α until all messages are queued at the remaining blocks, before proceeding to the next round. Each round is charged for the messages delivered by the end of that round and after the completion of the previous round. By lemma 3.1, the expected number of messages used by α, denoted $\mathrm{E}(\mathrm{complexity}_\alpha)$, is bounded by:

$$\begin{aligned}
\mathrm{E}(\mathrm{complexity}_\alpha) &> \sum_{i=1}^{\log n-4} \left(1-\epsilon^{l_i/n}\right)^2 \frac{l_i}{32}\cdot 2^{i-1} \\
&> \sum_{i=1}^{\log n-4} \left(1-\epsilon^{2^{-i-3}}\right)^2 \frac{n}{2^{i+3}}\frac{1}{32}\cdot 2^{i-1} \\
&= \frac{n}{2^9}\sum_{i=1}^{\log n-4} \left(1-\epsilon^{2^{-i-3}}\right)^2
\end{aligned}$$

Notice that $(1-\epsilon^{2^{-x-3}})^2 \geq \frac{1}{4}$ as long as $x \leq \log\log(1/\epsilon) - 3$. Hence

$$\begin{aligned}
\mathrm{E}(\mathrm{complexity}_\alpha) &> \frac{n}{2^9}\sum_{i=1}^{\min\left\{\log n-4,\log\log\frac{1}{\epsilon}-3\right\}} \left(1-\epsilon^{2^{-i-3}}\right)^2 \\
&= \frac{n}{2^9}\frac{1}{4}\min\left\{\log n-4, \log\log\frac{1}{\epsilon}-3\right\} \\
&= \Omega\left(n\min\left\{\log n, \log\log\frac{1}{\epsilon}\right\}\right)
\end{aligned}$$

■

4 Related results

An $O(n\log c)$ expected messages randomized attrition procedure for an anonymous ring with $c \geq 1$ initial contenders is described in [1]. This randomized attrition can be converted to an ϵ-attrition procedure which shows that the $\Omega(n\min(\log n, \log\log(1/\epsilon)))$ expected messages bound of section 3 is tight to within a constant factor. Let $\lambda = \min(n, \log(1/\epsilon))$. Processors first choose to be contenders with probability λ/n and the contenders run randomized attrition. Since λ contenders are expected, the resulting ϵ-attrition has the desired complexity. The only way the algorithm can err is if $\log(1/\epsilon) < n$ and no processor chooses to be a contender. This happens with probability $(1-\lambda/n)^n \leq e^{-\lambda} = \epsilon$.

Lower bounds on ϵ-attrition imply lower bounds on probabilistic leader election. Thus leader election that errs with probability at most ϵ inherits a lower bound of $\Omega(n \min(\log n, \log\log(1/\epsilon)))$ expected messages on rings of fixed size. This lower bound is tight to within a constant factor. In [1] it is shown that a leader election algorithm can be assembled from attrition and solitude verification. This relationship between the three problems extends to the probabilistic case where error with probability at most ϵ is tolerated. But probabilistic solitude verification has complexity $\Theta\left(n \min\left(\sqrt{\log n}, \sqrt{\log\log(1/\epsilon)} + \log\nu(n), \log\log(1/\epsilon)\right)\right)$ expected bits on rings of known size, where $\nu(n)$ is the smallest non-divisor of n, [2]. Hence there is a leader election algorithm that errs with probability at most ϵ and has complexity $O(n \min(\log n, \log\log(1/\epsilon)))$ expected bits (and messages).

The preceding discussion illustrates that attrition is an essential and dominant part of leader election in the sense that the complexities of the two problems are equivalent (even in the probabilistic case which permits error with low probability). A number of other common problems have the same relationship to attrition. We show here that the ϵ-attrition bound extends to computing OR with probability of error at most ϵ on rings of know size.

Theorem 4.1: The complexity of any algorithm that with probability at least $1 - \epsilon$ computes OR on an anonymous ring of fixed size n is $\Omega(n \min(\log n, \log\log(1/\epsilon)))$ expected messages.

Proof: Let α be an algorithm for OR which errs with probability at most ϵ. Let $f(n, \epsilon)$ be the expected message complexity of α. Let γ be any algorithm for attrition on a ring of size n with any non-empty subset of initial contenders such that the expected complexity of γ is $O(n \log c)$ messages, where c is the actual number of contenders (not necessarily known). ([1] describes one such attrition algorithm.) Let $\lambda = \min(\log n, \log\log(1/\epsilon))$. (Throughout this proof "log" refers to the natural logarithm.) Define β to be the distributed algorithm where each processor on an anonymous ring executes the following:

β: 1. generate a random bit, *myflip* $\in \{0, 1\}$ such that $\Pr(1) = \lambda/n$.

2. if *myflip* = 1 become a contender and initiate γ. Henceforth participate in γ and discard all α messages.

3. if *myflip* = 0 initiate α. Participate in α only as long as no γ message arrives. Upon receipt of a γ message, participate in γ as a non-contender and discard all subsequent α messages. If α confirms "all 0's" then restart the algorithm at step 1.

Step 2 performs attrition on an expected small number of contenders. Step 3 alerts the processors to try again in the event that there were no contenders. Thus β is an attrition algorithm that deadlocks with small probability.

Error: The only way that the attrition algorithm, β, deadlocks is if all processors flip 0 and the OR algorithm α fails to confirm all 0's. Therefore the probability of deadlock of β is at most $\epsilon \sum_{i=1}^{\infty} \left(\left(1-\frac{\lambda}{n}\right)^n\right)^i \leq \frac{\epsilon}{e^{\lambda}-1} < \epsilon$ as long as $n > 2$ and $\epsilon < 0.135 < 1/e^2$.

Complexity: Let random variable C be the number of processors with *myflip* = 1. The expected number of messages sent by β, denoted $\mathrm{E}(\mathrm{complexity}_\beta)$, is given by

$$\begin{aligned}
\mathrm{E}\left(\mathrm{complexity}_\beta\right) &= \mathrm{E}\left(\mathrm{complexity}_\beta | \text{all 0's}\right) \Pr(\text{all 0's}) + \\
&\quad \mathrm{E}\left(\mathrm{complexity}_\beta | \text{at least one 1}\right) \Pr(\text{at least one 1}) \\
&\leq \left(f(n,\epsilon) + \mathrm{E}(\mathrm{complexity}_\beta)\right)\left(1-\frac{\lambda}{n}\right)^n + \\
&\quad \left(f(n,\epsilon) + \mathrm{E}(n \log C)\right)\left(1-\left(1-\frac{\lambda}{n}\right)^n\right) \\
&\leq \frac{f(n,\epsilon) + n\log\lambda\left(1-\left(1-\frac{\lambda}{n}\right)^n\right)}{1-\left(1-\frac{\lambda}{n}\right)^n} \\
&< 2f(n,\epsilon) + n \min\left(\log\log n, \log\log\log\frac{1}{\epsilon}\right) \\
&\quad (\text{if } n > 2 \text{ and } \epsilon < 0.135)
\end{aligned}$$

Since β is an ϵ-attrition algorithm, β has complexity $\Omega\left(n \min\left(\log n, \log\log\left(1/\epsilon\right)\right)\right)$. Hence $f(n,\epsilon)$ has order $\Omega\left(n \min\left(\log n, \log\log\left(1/\epsilon\right)\right)\right)$. ■

It is easily verified that ϵ-attrition also reduces to other functions such as AND, PARITY and SUM.

Note that this lower bound for OR (and similarly for AND, SUM, and PARITY) is tight to within a constant factor because OR can be computed on a ring of size n by expending an additional $O(n)$ bits after electing a leader.

References

[1] Karl Abrahamson, Andrew Adler, Rachel Gelbart, Lisa Higham, and David Kirkpatrick. The bit complexity of randomized leader election on a ring. *SIAM Journal on Computing*, 1988. in press.

[2] Karl Abrahamson, Andrew Adler, Lisa Higham, and David Kirkpatrick. *Probabilistic Solitude Detection II: Rings Size Known Exactly.* Technical Report 86-26, University of British Columbia, 1986. submitted for publication.

[3] J. Burns. *A Formal Model for Message Passing Systems.* Technical Report TR-91, Indiana University, 1980.

[4] Danny Dolev, Maria Klawe, and Michael Rodeh. An $O(n \log n)$ unidirectional distributed algorithm for extrema finding in a circle. *J. Algorithms*, 3(3):245–260, 1982.

[5] Pavol Duris and Zvi Galil. Two lower bounds in asynchronous distributed computation. In *FOCS87*, pages 326–330, 1987. preliminary version.

[6] Lisa Higham. Ph.d. thesis. in preparation.

[7] Jan Pachl. *A Lower Bound for Probabilistic Distributed Algorithms.* Technical Report CS-85-25, University of Waterloo, Waterloo, Ontario, 1985.

[8] Gary Peterson. An $O(n \log n)$ algorithm for the circular extrema problem. *ACM Trans. on Prog. Lang. and Systems*, 4(4):758–752, 1982.