Analysis of Quorum-Based Protocols for Distributed (k + 1)-Exclusion

Divyakant Agrawal, *Member, IEEE Computer Society*, Ömer Eğecioğlu, and Amr El Abbadi, *Member, IEEE Computer Society*

Abstract—A generalization of the majority quorum for the solution of the distributed $(k+1)$-exclusion problem is proposed. This scheme produces a family of quorums of varying sizes and availabilities indexed by integral divisors r of k. The cases $r = 1$ and $r = k$ correspond to known majority based quorum generation algorithms *MAJ* and *DIV*, whereas intermediate values of r interpolate between these two extremes. A cost and availability analysis of the proposed methods is also presented. An interesting implication of this analysis is that in a reasonably reliable environment with a large number of sites, even protocols with low communication costs attain high availability.

Index Terms— Mutual exclusion, fault-tolerance, distributed systems, replicated data.

1 Introduction

The problem of distributed mutual exclusion has been extensively studied and many interesting protocols for its solution have been proposed. Most of these protocols attempt to provide high performance by reducing the number of messages involved or by improving the degree of fault-tolerance and hence improving the chances of achieving mutual exclusion in the presence of site and communication failures. A generalization of the mutual exclusion problem is the *k-mutual exclusion problem*, where no more than k processes are allowed to enter the critical section simultaneously. Since $(k+1)$th process will never be admitted, this problem is also referred to as the *$(k+1)$-exclusion problem*.

In a distributed environment, the $(k+1)$-exclusion problem arises in several interesting applications. For example, it could be used to monitor the number of processes in a distributed system that are allowed to perform a certain action, such as issuing broadcast messages. In such a case, the system may restrict the number of broadcasting processes so as to control the level of congestion. Another application is in the context of replicated databases that allow *bounded ignorance* [11], i.e., when transactions may specify that they do not need to be aware of the k most recent updates to the database. Here also, instead of the traditional database system that uses distributed mutual exclusion to ensure one update to the replicated data at any time, several updates may be permitted simultaneously. Efficient and highly available solutions to the distributed $(k+1)$-exclusion problem would be particularly useful for such applications.

The distributed $(k+1)$-exclusion problem was first solved by Raymond [15], who provided a simple extension to the Ricart and Agrawala's mutual exclusion algorithm [16]. Srimani and Reddy [18] improved on this protocol by using the notion of *privilege* of Suzuki and Kasami [19]. This solution reduces the number of required messages to achieve mutual exclusion. Recently, there has been a significant interest in fault-tolerant methods to solve the $(k+1)$-exclusion problem based on the notion of quorums [8]. Fujita et al. [6] discuss some simple techniques and then propose a scheme with small quorum sizes. Huang et al. [9] propose an alternative method with small quorums and high availability in the presence of failures.

The majority quorum algorithm [20], [8] for distributed mutual exclusion has been widely used to develop quorum-based protocols for mutual exclusion as well as for $(k+1)$-exclusion by a suitable partitioning of the sites. Agrawal and El Abbadi [2] partition the sites in a network to construct majority based quorums defined on hybrid logical structures such as a grid [13] and a tree [3]. Rangarajan and Tripathi [14] partition the sites into N classes and organize the classes to form a finite projective plane. Quorums are then defined using $\sqrt{N}$ classes and within each class a majority of sites is chosen. Kumar [12] uses the partitioning approach to recursively define hierarchical quorums based on the majority rule and quorums. Fujita et al. [6] also use a partitioning approach for $(k+1)$-exclusion in which the sites are partitioned into k clusters and quorums are constructed so that mutual exclusion is ensured within each cluster.

In this paper, we generalize the majority quorum algorithm for constructing quorums for $(k+1)$-exclusion and analyze the resulting methods. This analysis is performed to understand the tradeoff between availability and communication cost for achieving a quorum. In particular, we propose a sequence of algorithms MAJ_r indexed by integral divisors r of k. When $r = 1$, the quorums correspond to those produced by *MAJ* (which partitions the sites into a single class), and when $r = k$, the quorums correspond to

- *The authors are with the Department of Computer Science, University of California, Santa Barbara, CA 93106. E-mail: omer@cs.ucsb.edu.*

Manuscript received 21 Aug. 1995.
For information on obtaining reprints of this article, please send e-mail to: transpdsmputer.org, and reference IEEECS Log Number D95273.

the quorums produced by *DIV* (which partitions the sites into k classes) [6]. For intermediate values of r, a sliding tradeoff between communication cost and availability can be achieved.

2 The Problem Statement

A distributed system consists of a set of sites that communicate with each other by sending messages over a communication network. We assume that every site has the capability to send a message to any other site when there is a communication path between them. The sites are either *fail-stop* or may fail to send or receive messages. Communication links may fail by crashing, or by failing to deliver messages. Although quorum-based protocols are resilient both site and communication failures, our analysis assumes site failures only [14], [10].

Distributed mutual exclusion is a classical technique for providing access to shared resources. We postulate the existence of a resource in the network, which may be accessed by a single process at a time. To access the resource, a process (site) p_i is required to receive permission from a set of sites S_i. If all sites in S_i grant permission to p_i, then it is allowed to access the resource. To ensure mutual exclusion the sets S_i are required to satisfy the *intersection property*: For any i and j, $S_i \cap S_j \neq \phi$. These and related concepts were formalized and analyzed in terms of the notions of *quorums* and *coteries* [8], [13], [7]. In $(k + 1)$-exclusion, up to k processes are allowed to access the resource simultaneously. Thus, if we consider $k + 1$ sets of sites that grant permission to access the resource then there must exist at least two among these $k + 1$ sets with a nonempty intersection. The $(k + 1)$-exclusion problem can now be stated in terms of the requirements 1 and 2 below:

1) **The (k + 1)-Intersection Property**. For any $k + 1$ sets S_1, S_2, $\cdots$, S_{k+1}, there exist two distinct sets S_i and S_j such that $S_i \cap S_j \neq \phi$.

Note that quorums constructed to ensure the traditional mutual exclusion condition also ensure the above property. Hence, in order to eliminate trivial solutions to the $(k + 1)$-exclusion problem, we add an additional restriction [6], [9].

2) **The k-Nonintersection Property.** There exist k sets S_1, S_2, $\cdots$, S_k such that for any two distinct sets S_i and S_j, $S_i \cap S_j = \phi$.

The second property above is desirable for all values of k. When $k = 1$, i.e., in the case of mutual exclusion, it is satisfied vacuously.

3 A General Paradigm for $(k + 1)$-Exclusion

One of the simplest approaches to ensure mutual exclusion in a distributed system is to use majority quorums [8], [20] of size $\lfloor \frac{n}{2} \rfloor + 1$. For three-exclusion, we can reduce the size of the permission sets to $\lfloor \frac{n}{3} \rfloor + 1$. Clearly, the three-intersection property holds since any three sets of sites with size $\lfloor \frac{n}{3} \rfloor + 1$ chosen from n sites will always have two sets with nonempty intersection. Similarly, the two-nonintersection property also holds for $n > 5$, since it is possible to construct two disjoint sets when only one-third the number of sites from n are used for each. For $(k + 1)$-exclusion, it suffices to take the quorum size to be $\lfloor \frac{n}{k+1} \rfloor + 1$. This majority based construction for the $(k + 1)$-exclusion problem is referred to as *MAJ* [6].

Another approach for achieving $(k + 1)$-exclusion is to consider k instances of any mutual exclusion solution. A process wishing $(k + 1)$ exclusive access to a resource acquires permission from any of the k instances. This ensures the $(k + 1)$-intersection property since any $k + 1$ quorums chosen will consist of at least two quorums in the same instance of the mutual exclusion solution, and hence must have a nonempty intersection. Similarly, the k-nonintersection property is satisfied if each of the k processes chooses a quorum from different instances of the mutual exclusion solution. This construction for the $(k + 1)$-exclusion problem is referred to as *DIV* [6].

The two generalizations *MAJ* and *DIV* of majority quorums for the solution of the $(k + 1)$-exclusion problem are at the opposite ends of a spectrum. In *MAJ*, the original mutual exclusion majority solution is generalized whereas in *DIV* the sites in the network are partitioned into k classes with each class using any traditional approach to enforce mutual exclusion. We explore the possibility of enforcing $(k + 1)$-exclusion by varying the number r of classes from 1 to k, and define a quorum generation method MAJ_r for any r dividing k. To simplify the presentation, we assume we are given a set of n sites where $n = kN$ for some $N \geq 1$. In MAJ_r, the $(k + 1)$-exclusion problem is solved by partitioning the sites into r disjoint classes where $r = k/i$ for some integer i. Note that for the envisioned applications of $(k+1)$-exclusion, congestion-control in broadcasting and bounded-ignorance in transaction processing, k is expected to be a reasonably large integer value and k will most likely have several integral divisors. Within each class, we choose the quorums of size q_r which guarantee that at least two sets from any collection of $i + 1$ sets within the class intersect. More precisely MAJ_r denotes the method in which

1) The sites 1, 2, …, n are partitioned into r classes of size $n/r = iN$ each.
2) From each class, all subsets of size

$$q_r = \left\lfloor \frac{iN}{i+1} \right\rfloor + 1 \tag{1}$$

are taken as quorums. Here $r = k/i$.

It should be clear that MAJ_r produces sets that satisfy both **1** and **2**, aside from the trivial cases of small parameters for which $\lfloor \frac{n}{k+r} \rfloor = \lfloor \frac{n}{k+r-1} \rfloor$ (see [21]). Furthermore, $MAJ_1 = MAJ$, and $MAJ_k = DIV$ are special cases of this construction.

4 Availability Measures

The communication cost associated with obtaining mutual exclusion by using the quorum approach is directly proportional to the quorum size. The availability and the fault-tolerance characteristics of a particular method are determined by the number of ways in which a quorum can be

TABLE 1
COMPUTED RANGE OF VALUES $0 < p < p_n$ FOR WHICH
$AV_{DIV}(p) > AV_{MAJ}(p)$

n	12	24	36	48	60	72	84	96
p_n	0.0461	0.0423	0.0408	0.0399	0.0394	0.0391	0.0388	0.0386

constructed from a given number of sites in the network. The total number of sets produced by MAJ_r is

$$r\binom{iN}{q_r}, \tag{2}$$

where q_r is as given in (1). Let $q_1 = q_{MAJ}$ be the quorum size and T_{MAJ} be the number of sets in MAJ. Then

$$q_{MAJ} = \left\lfloor \frac{n}{k+1} \right\rfloor + 1, \quad T_{MAJ} = \binom{n}{q_{MAJ}}. \tag{3}$$

The other extreme of the partitioning approach arises when $r = k$. The quorum size $q_k = q_{DIV}$ and the total number of quorums T_{DIV} for DIV are given by the formulas

$$q_{DIV} = \left\lfloor \frac{n}{2k} \right\rfloor + 1, \quad T_{DIV} = k\binom{\frac{n}{k}}{q_{DIV}}. \tag{4}$$

When MAJ and DIV are evaluated purely in terms of the communication costs incurred to enforce $(k+1)$-exclusion, DIV is preferable due to its smaller-sized quorums. However, if the evaluation criterion includes the number of quorum sets produced, the outcome is not so trivial. Fujita et al. [6] conjectured that the partitioned approach restricts the number of ways a quorum can be selected and hence will provide inferior availability. First, we explore this issue in the context of these two approaches and isolate the instances in which DIV actually performs better than MAJ. Our analysis of availability is based on estimates for truncated binomial sums [4], [14], and the Vandermonde convolution identity [17].

Suppose p is the probability that a site is up. If $q = q_r$ is the quorum size given by (1), then the probability $C_r(p)$ that a quorum set is available in a given class for the method MAJ_r is a polynomial $C_r(p)$ of degree iN in p (recall that there are a total of r classes of iN sites each):

$$C_r(p) = \sum_{j=0}^{iN-q} \binom{iN}{q+j} p^{q+j}(1-p)^{iN-q-j}. \tag{5}$$

Let $AV_r(p)$ denote the probability that a quorum set is available when the method MAJ_r is used. For the extreme cases of MAJ and DIV, we also use the notation $AV_{MAJ}(p)$ for $AV_1(p)$, and $AV_{DIV}(p)$ for $AV_k(p)$. The probability that none of the r classes of the partition has a quorum set available is $(1 - C_r(p))^r$, and therefore

$$AV_r(p) = 1 - (1 - C_r(p))^r. \tag{6}$$

EXAMPLE 1. Suppose $k = 2$ and $n = 10$. Then for $i = 2$, we get $r = 1$ and $MAJ_1 = MAJ$. From (5),

$$C_{MAJ}(p) = 210p^4(1-p)^6 + 252p^5(1-p)^5 + 210p^6(1-p)^4 + 120p^7(1-p)^3 + 45p^8(1-p)^2 + 10p^9(1-p) + p^{10}.$$

For $i = 1$, $r = 2$ and $MAJ_2 = DIV$. In this case $C_{DIV}(p) = 10p^3(1-p)^2 + 5p^4(1-p) + p^5$. Therefore, from (6)

$$AV_{MAJ}(p) = C_{MAJ}(p),$$
$$AV_{DIV}(p) = 20p^3 - 30p^4 + 12p^5 - 100p^6 + 300p^7 - 345p^8 + 180p^9 - 36p^{10}.$$

Since $AV_{DIV}(p) - AV_{MAJ}(p) = 20p^3(1-p)^6(1-6p)$, setting $AV_{DIV}(p) > AV_{MAJ}(p)$ and solving for p, we find that whenever $p < 1/6$, DIV provides better quorum availability than MAJ.

The above example can be generalized by deriving asymptotic results for the availabilities of MAJ_r for $r = 1$ (MAJ) and $r = k$ (DIV). For simplicity of exposition, we first discuss the three-exclusion case ($k = 2$). Since in this case

$$q_1 = q_{MAJ} = \left\lfloor \frac{n}{3} \right\rfloor + 1, \quad q_2 = q_{DIV} = \left\lfloor \frac{n}{4} \right\rfloor + 1$$

from (1), it is convenient to assume that $n = 12m$ for some $m \geq 1$. Then $q_{MAJ} = 4m + 1$, $q_{DIV} = 3m + 1$ with

$$AV_{MAJ}(p) = \sum_{j=4m+1}^{12m} \binom{12m}{j} p^j (1-p)^{12m-j},$$

$$AV_{DIV}(p) = 1 - \left[1 - \sum_{j=3m+1}^{6m} \binom{6m}{j} p^j (1-p)^{6m-j}\right]^2.$$

The values of p in $0 < p < p_n$ for which $AV_{DIV}(p) > AV_{MAJ}(p)$ for small values of $n = 12m$ are tabulated in Table 1 (computed with the aid of MACSYMA). Even though the numbers p_n are decreasing, the limit of p_n as n gets large is nonzero. In fact, it can be shown that for three-exclusion, DIV provides better availability for large n than MAJ when the probability of a site being up is less than 0.0299 [1]. This result can be generalized in two directions, both valid for large n. First, it is possible to compare analytically the availability of MAJ and DIV for arbitrary k and derive a constant $b_k = O(1/k^2)$ such that whenever $0 < p < b_k$, the availability of DIV is greater than that of MAJ. Alternately, given k, it is possible to compute the range of values of p (as a function of k and r) for which MAJ_r provides better availability than MAJ. However, these generalizations are of theoretical interest as they are valid only for very small values of p [1].

5 AVAILABILITY OF MAJ_r QUORUMS

In general, MAJ has better availability than DIV as well as all the other MAJ_r, except for systems with high site failure probability. On the other hand, from the formula in (1), MAJ quorums have twice the size of DIV, and are always larger than MAJ_r quorums for $r > 1$.

In this section, we show that for large n, and p close to 1, the increase in availability provided by MAJ itself as compared to MAJ_r is actually quite small. This is significant since in most current systems we expect sites to have a low probability of failure. Hence in these cases we can use the smaller-sized quorums of MAJ_r without losing much on availability, while reducing the communication overhead by up to a factor of two.

Let $D_r(p) = | AV_{MAJ}(p) - AV_r(p) |$ denote the magnitude of the absolute error in availability made when MAJ_r instead of MAJ itself is used. Then $D_r(p)$ is given by

$$\left| \sum_{j=0}^{q_{MAJ}-1} \binom{n}{j} p^j (1-p)^{n-j} - \left[\sum_{j=0}^{q_r-1} \binom{n/r}{j} p^j (1-p)^{n/r-j} \right]^r \right|$$

Factoring $(1-p)^n$ and letting $q = p/(1-p)$, we can write

$$D_r(p) = (1-p)^n \left| \sum_{j=0}^{q_{MAJ}-1} \binom{n}{j} q^j - \left[\sum_{j=0}^{q_r-1} \binom{n/r}{j} q^j \right]^r \right|$$

Consider the coefficients c_j in the expansion

$$\left[\sum_{j=0}^{q_r-1} \binom{n/r}{j} q^j \right]^r = \sum_{j=0}^{r(q_r-1)} c_j q^j.$$

By the Vandermonde convolution identity [5], [17], $c_j \le \binom{n}{j}$ with equality in the range $0 \le j \le q_r - 1$. Therefore,

$$D_r(p) = (1-p)^n \left| \sum_{j=q_r}^{q_{MAJ}-1} \binom{n}{j} q^j - \sum_{j=q_r}^{r(q_r-1)} c_j q^j \right|$$

$$\le (1-p)^n \sum_{j=q_r}^{r(q_r-1)} \binom{n}{j} q^j.$$

However, q^j is an increasing function of j whenever $p \ge 0.5$. Thus for $p \ge 0.5$

$$D_r(p) \le p^{r(q_r-1)} (1-p)^{n-r(q_r-1)} \sum_{j=q_r}^{r(q_r-1)} \binom{n}{j}.$$

But since $1 \le r \le k$, $r(q_r - 1) = (r/(k + r)), n \le n/2$, and therefore

$$\sum_{j=q_r}^{r(q_r-1)} \binom{n}{j} \le 2^{n-1}.$$

Furthermore, $n - (r/(k + r))\, n = (k/(k + r))n$, and 1 is an upper bound for $p^{r(q_r-1)}$. Therefore,

$$D_r(p) \le \frac{1}{2} \left[2(1-p)^{\frac{k}{k+r}} \right]^n.$$

If p is close enough to 1 so that $2(1-p)^{\frac{k}{k+r}} < 1$, or equivalently when

$$p > 1 - \left(\frac{1}{2} \right)^{\frac{k+r}{k}}, \quad (7)$$

the error $D_r(p)$ goes to zero as n gets large. To summarize,

THEOREM 1. *If p is in the range given by (7), then*

$$\left| AV_{MAJ}(p) - AV_r(p) \right| \le \frac{1}{2} \left[2(1-p)^{\frac{k}{k+r}} \right]^n, \quad (8)$$

and the right hand side of (8) goes to zero as n gets large.

To get a rough idea of the magnitude of the difference in availability which will hold for all algorithms MAJ_r at once regardless of r, we note that $(k + r)/k \le 2$ since r is a divisor of k. Therefore, whenever $p > 1 - (1/2)^2 = 0.75$, the error in availability satisfies

$$\left| AV_{MAJ}(p) - AV_r(p) \right| \le \frac{1}{2} \left(2\sqrt{1-p} \right)^n, \quad (9)$$

independent of the value of r.

EXAMPLE 2. Consider a system with $n = 100$ sites in which the probability of a site being up is $p = 0.9$. From (9), the difference between the availabilities of MAJ and DIV for any $(k + 1)$-exclusion is about 6.3×10^{-21}, a negligible amount. On the other hand, when $k = 9$, MAJ requires quorums of size $q_{MAJ} = 11$, while DIV requires quorums of size only $q_{DIV} = 6$.

6 DISCUSSION

In this paper, we proposed a family of quorum-based protocols MAJ_r that generalize majority quorums for distributed $(k+1)$-exclusion. These protocols are indexed by integral divisors r of k, with $MAJ_1 = MAJ$ and $MAJ_k = DIV$. In addition, we considered the whole spectrum of resulting protocols with respect to availability and communication cost. Recently, Kakugawa et al. [10] analyzed MAJ versus a centralized site solution adapted for $(k+1)$-exclusion called SGL, and showed that for networks with low probability of site failure, MAJ provides optimal availability performance over *all* quorum-based mechanisms, whereas for sites with a high probability of failures, SGL gives higher availability than MAJ. Our analysis is similar except that our quorums generalize majority, and instead of the two extremes, we consider a whole spectrum of protocols. Each protocol MAJ_r, $r > 1$ considered here, provides higher availability than MAJ in a large network where the probability of a site being up is sufficiently small [1]. On the other end of the probability spectrum in which the network has a low probability of site failure, the increase in availability achieved by MAJ over any member of the family MAJ_r, $r > 1$ decreases rapidly as the number of sites gets large. Since the communication overhead of solving the $(k+1)$-exclusion problem is directly proportional to the quorum sizes, reductions up to a factor of two in communication overhead can be achieved without significant sacrifice in availability. Thus for highly available geographically dispersed distributed systems with a large number of sites, MAJ_r for $r > 1$ may be preferable over MAJ. Even when individual site availability is greater than about 75%, the increased availability offered by MAJ becomes inconsequential when compared even to DIV, which falls on the other end of the spectrum. Intermediate values of r serve to further interpolate between communication cost and availability.

ACKNOWLEDGMENT

This research was partially supported by NASA under contract number NAGW-3888.

REFERENCES

[1] D. Agrawal, Ö. Eğecioğlu, and A. El Abbadi, "Analysis of Quorum-Based Protocols for Distributed $(k+1)$-Exclusion," Technical Report TRCS96-11, Dept. of Computer Science, Univ. of California, Santa Barbara, 1996.

[2] D. Agrawal and A. El Abbadi, "Exploiting Logical Structures of Replicated Databases," *Information Processing Letters*, vol. 33, no. 5, pp. 255–260, Jan. 1990.

[3] D. Agrawal and A. El Abbadi, "An Efficient and Fault-Tolerant Solution for Distributed Mutual Exclusion," *ACM Trans. Computer Systems*, pp. 1–20, Feb. 1991.

[4] M. Ahamad and M.H. Ammar, "Performance Characterization of Quorum-Consensus Algorithms for Replicated Data," *IEEE Trans. Software Eng.*, vol. 15, no. 4, pp. 492–495, Apr. 1989.

[5] D.I.A. Cohen, *Basic Techniques of Combinatorial Theory*. New York: John Wiley and Sons, 1978.

[6] S. Fujita, M. Yamashita, and T. Ae, "Distributed k-Mutual Exclusion Problem and k-Coteries," *Proc. Symp. Algorithms*, pp. 22–31, 1991.

[7] H. Garcia-Molina and D. Barbara, "How to Assign Votes in a Distributed System," *J. ACM*, vol. 32, no. 4, pp. 841–860, Oct. 1985.

[8] D.K. Gifford, "Weighted Voting for Replicated Data," *Proc. Seventh ACM Symp. Operating Systems Principles*, pp. 150–159, Dec. 1979.

[9] S. Huang, J. Jiang, and Y. Kuo, "K-coteries for Fault-Tolerant k Entries to a Critical Section," *Proc. 13th Int'l Conf. Distributed Computing Systems*, pp. 74–81, June 1993.

[10] H. Kakugawa, S. Fujita, M. Yamashita, and T. Ae, "Availability of *k*-Coteries," *IEEE Trans. Computers*, vol. 42, no. 5, pp. 553–558, May 1993.

[11] N. Krishnakumar and A.J. Bernstein, "Bounded Ignorance in Replicated Systems," *Proc. 10th ACM Symp. Principles of Database Systems*, pp. 63–74, May 1991.

[12] A. Kumar, "Hierarchical Quorum Consensus—A New Algorithm for Managing Replicated Data," *IEEE Trans. Computers*, vol. 39, no. 9, pp. 996–1,004, Sept. 1991.

[13] M. Maekawa, "A $\sqrt{n}$ Algorithm for Mutual Exclusion in Decentralized Systems," *ACM Trans. Computer Systems*, vol. 3, no. 2, pp. 145–159, May 1985.

[14] S. Ranagarajan and S. Tripathi, "Robust Distributed Mutual Exclusion Algorithms," *Proc. Fifth Int'l Workshop Distributed Algorithms*, pp. 295–308. *Lecture Notes in Computer Science, 579*, Springer-Verlag, Oct. 1991.

[15] K. Raymond, "A Distributed Algorithm for Multiple Entries to a Critical Section," *Information Processing Letters*, vol. 30, no. 4, pp. 189–193, Feb. 1989.

[16] G. Ricart and A.K. Agrawala, "An Optimal Algorithm for Mutual Exclusion in Computer Networks," *Comm. ACM*, vol. 24, no. 1, pp. 9–17, Jan. 1981.

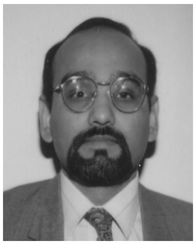
[17] J. Riordan, *An Introduction to Combinatorial Analysis*. Princeton, N.J.: Princeton Univ. Press, 1978.

[18] P.K. Srimani and R.L.N. Reddy, "Another Distributed Algorithm for Multiple Entries to a Critical Section," *Information Processing Letters*, vol. 41, no. 1, pp. 51–57, Jan. 1992.

[19] I. Suzuki and T. Kasami, "A Distributed Mutual Exclusion Algorithm," *ACM Trans. Computer Systems*, vol. 3, no. 4, pp. 344–349, Nov. 1985.

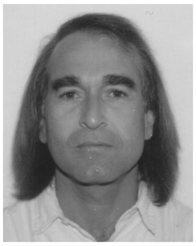
[20] R.H. Thomas, "A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases," *ACM Trans. Database Systems*, vol. 4, no. 2, pp. 180–209, June 1979.

[21] S.M. Yuan and H.M. Chang, Comments on "Availability of *k*-Coterie," *IEEE Trans. Computers*, vol. 43, no. 12, p. 1,457, Dec. 1994.

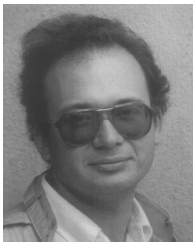


Divyakant Agrawal received his BE degree from Birla Institute of Technology and Science, Pilani, India, in 1980 and his MS and PhD degrees in computer science from the State University of New York, Stony Brook in 1984 and 1987. He is currently an associate professor in the Department of Computer Science at the University of California, Santa Barbara. His research interests include design of algorithms for concurrent and fault-tolerant systems.

Dr. Agrawal is a member of the Association for Computing Machinery and the IEEE Computer Society.



Ömer Eğecioğlu obtained his PhD degree in mathematics from the University of California, San Diego in 1984. At present, he is an associate professor in the Computer Science Department of the University of California, Santa Barbara, where he has been on the faculty since 1985. His principal areas of research are parallel algorithms, bijective and enumerative combinatorics, and combinatorial algorithms. His current interest in parallel algorithms involves approximation and numerical techniques on distributed memory systems while his combinatorial interests center around computational geometry, bijective methods, and ranking algorithms for combinatorial structures.



Amr El Abbadi received his BEng in computer engineering from Alexandria University, Egypt, in 1980, and his MS and PhD degrees in computer science from Cornell University in 1984 and 1987, respectively. In August 1987, he joined the University of California, Santa Barbara, where he is currently an associate professor. His current interests are in the field of distributed systems and distributed databases. He is especially interested in the design of protocols for highly available fault-tolerant systems.

Dr. El Abbadi is a member of the Association for Computing Machinery SIGACT and the IEEE Computer Society.