Brief Announcement: Specification, Implementation and Application of Randomized Regular Registers

Hyunyoung Lee and Jennifer L. Welch*

This paper presents a definition of a randomized regular register, shows that the definition is implemented by the probabilistic quorum algorithm of [3], and shows how to program with such registers using the framework of [4]. Consequently, existing iterative algorithms for a large class of problems (including solving systems of linear equations, finding shortest paths, etc.) will converge with high probability if executed in a system in which the shared data is implemented with registers satisfying the new condition. A modified definition is presented and its expected time for convergence is calculated and compared experimentally with that for the original definition.

We define a **random regular register** to satisfy: every operation invocation has a matching response; every read reads from a write that starts before the read ends; and for every finite execution that ends in a write invocation, the probability that this write W is read from infinitely often is 0, if an infinite number of writes are performed in the extension. We show in [1] that the probabilistic quorum algorithm of [2, 3] implements a random regular register, by showing that the probability that at least one replica from W's quorum survives ℓ subsequent writes is at most $k(\frac{n-k}{n})^{\ell}$, where n is the number of processes and k is the quorum size.

The class of algorithms considered in [4] are those in which a function is applied repeatedly to a vector to produce another vector. In typical applications, each vector component may be computed by a separate process, based on that process' current best estimate of the values of all the vector components – estimates which might be out of date. Üresin and Dubois show that if the function satisfies certain properties and if the outdatedness of the vector component estimates is not too extreme, then this iterative procedure will eventually converge to the fixed point of the function, in increments called **pseudocycles**. Functions satisfying the desired properties are called **asynchronously contracting operators** (ACOs). In [1] we prove: *If* $\mathbf{F}$ *is an ACO, then in every execution using random regular registers, the computed vector eventually converges to the fixed point of* $\mathbf{F}$ *with probability 1.*

Our second definition requires the register to be **monotone**, meaning that if a read reads from a certain write, then no subsequent read by the same process reads from an earlier write. Additionally, let Y be a random variable whose value is the number of reads by a process after a write W until W or a later write is read from by that process. Then there must exist p, $0 < p \leq 1$, such that for all $r \geq 1$, $\Pr(Y = r) \leq (1-p)^{r-1} \cdot p$.

In [1] we define a **round** to be a minimal length (contiguous) subsequence of an execution in which each process performs *at least one* update of its vector components, and we prove the following: *In every execution using monotone random regular registers with constant p, the expected number of rounds per pseudocycle is at most* $\frac{1}{p}$. Thus the expected number of rounds for an ACO requiring M pseudocycles for convergence is at most M/p.

In [1] we prove that a simple modification to the probabilistic quorum algorithm for n processes with quorum size k implements a monotone random regular register with $p = 1 - \binom{n-k}{k}/\binom{n}{k}$. This implies that the expected number of rounds per pseudocycle is at most $\frac{1}{1-(\frac{n-k}{n})^k}$.

[1] H. Lee and J. L. Welch, "Specification, Implementation and Application of Randomized Regular Registers," TR00-012, Department of Computer Science, Texas A&M University, March 2000.

[2] D. Malkhi and M. Reiter, "Byzantine Quorum Systems," In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 569–578, May 1997.

[3] D. Malkhi, M. Reiter, and R. Wright, "Probabilistic Quorum Systems," In *Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing*, pages 267–273, Aug. 1997.

[4] A. Üresin and M. Dubois, "Parallel Asynchronous Algorithms for Discrete Data," *J. ACM,* Vol. 37, No. 3, pages 558-606, July 1990.

[1] Department of Computer Science, Texas A&M University, College Station, TX 77843. e-mail: {hlee, welch}@cs.tamu.edu.