J. Parallel Distrib. Comput. 87 (2016) 1–12

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Counting-based impossibility proofs for set agreement and renaming☆

Hagit Attiya *, Ami Paz

Department of Computer Science, Technion, Israel

HIGHLIGHTS

- We consider the wait-free solvability of tasks in shared memory systems.
- Only combinatorial and number-theoretic arguments are used.
- $(n-1)$-set agreement is not wait-free solvable.
- Adaptive $(2n-2)$-renaming and strong symmetry breaking are not wait-free solvable.
- Nonadaptive $(2n-2)$-renaming and weak symmetry breaking are not wait-free solvable, when n is the power of a prime number.

ARTICLE INFO

Article history:
Received 18 March 2015
Accepted 8 September 2015
Available online 30 September 2015

Keywords:
Lower bounds
Wait-free algorithms
Adaptive and nonadaptive renaming
Weak and strong symmetry breaking

ABSTRACT

Set agreement and renaming are two tasks that allow processes to coordinate, even when agreement is impossible. In *k-set agreement*, n processes must decide on at most k of their input values. While n-set agreement is trivially wait-free solvable by each process deciding on its input, $(n-1)$-set agreement is not wait-free solvable. In *M-renaming*, processes must decide on distinct names in a range of size M. For any number n of processes, $(2n-1)$-renaming is wait-free solvable, but surprisingly, $(2n-2)$-renaming is wait-free solvable if and only if n is not a prime power; the only previous lower bound on the number of names necessary for renaming, when n is not a prime power, is $n+1$. In *adaptive* renaming, M decreases when the number p of participants in the execution decreases. It is known that $(2p-1)$-adaptive renaming is wait-free solvable, while $(2p-\lceil p/n-1\rceil)$-adaptive renaming is not.

This paper presents counting-based proofs for the above mentioned impossibility results: n processes can wait-free solve neither $(n-1)$-set agreement nor $(2p-\lceil p/n-1\rceil)$-adaptive renaming; if n is a prime power, n processes cannot wait-free solve $(2n-2)$-renaming. For an arbitrary number of processes, we give a lower bound for renaming, by reduction from renaming for a different number of processes, and relying on the distribution of prime numbers.

Our proofs combine simple operational properties of a restricted set of executions with elementary counting arguments to show the existence of an execution violating the task's conditions. This makes the proofs easier to understand, verify, and, we hope, extend.

1. Introduction

The capabilities and limitations of a distributed system can be investigated by studying simple coordination tasks. In a *task*, each process starts with an input value, communicates with other processes, decides on a value, and terminates. We consider algorithms for an asynchronous system, where n processes communicate by reading and writing to a shared memory. A process must decide on a value within a finite number of steps, regardless of the steps by other processes. Algorithms of that kind are called *wait-free*, and we say that a task is *wait-free solvable* if it has a wait-free algorithm.

The wait-free solvability of two tasks, k-set agreement and M-renaming, was intensively studied.

In *k-set agreement*, each process starts with an arbitrary input and has to decide on an input value of a (possibly different) process, such that at most k different values are decided. As the value of k decreases, so does the level of disagreement; when $k=1$, 1-set agreement is the well-known *consensus* task [19]. Chaudhuri [16] defined k-set agreement and showed it is wait-free solvable in a system where at most $f < k$ processes may crash. Three independent papers [9,26,30] showed that $(n-1)$-set agreement is

☆ This research is supported in part by Yad-HaNadiv foundation and the Israel Science Foundation (grant number 1227/10). A preliminary version of this paper appeared in [5].

* Corresponding author.
E-mail addresses: hagit@cs.technion.ac.il (H. Attiya), amipaz@cs.technion.ac.il (A. Paz).

http://dx.doi.org/10.1016/j.jpdc.2015.09.002
0743-7315/

not wait-free solvable, all using variants of Sperner's lemma. This implies that $(f-1)$-set agreement is not solvable when f process may crash, due to a reduction showing that $(n-1)$-set agreement is wait-free solvable if $(f-1)$-set agreement is solvable with f crash failures [9,12].

In *M-renaming*, each process has to decide on a unique value in the range $\{1,\ldots,M\}$. Renaming captures situations in which processes have to allocate a limited number of resources; the smaller the number of names M is, the more efficient the algorithm is. This task has two variants: in *M-nonadaptive renaming*, M depends only on n, the number of processes in the system, but not on p, the number of processes participating in the specific execution; in *M-adaptive renaming*, M may depend both on n and on p.

The renaming problem was defined by Attiya et al. [2], who showed a $(2p-1)$-adaptive renaming algorithm.[1] A harder variant of adaptive renaming is $(2p-\lceil p/n-1\rceil)$-adaptive renaming, where if all processes participate then they decide on at most $2n-2$ values, and when $p < n$ processes participate, they decide on at most $2p-1$ values. $(2p-\lceil p/n-1\rceil)$-adaptive renaming can be used to solve *strong symmetry breaking* (SSB). In SSB, each process should output a single bit, so that in every execution at least one process outputs 1, and if all processes output then not all output the same value. A reduction from SSB to $(n-1)$-set agreement [9,20] shows that $(2p-\lceil p/n-1\rceil)$-adaptive renaming is not wait-free solvable.

The impossibility results for $(n-1)$-set agreement, SSB, and $(2p-\lceil p/n-1\rceil)$-adaptive renaming hold even if the algorithm executed by a process p_i depends on its *identifier*, i; nonadaptive renaming, on the other hand, can be easily solved by letting process p_i decide on i. Trivial solutions of this sort can be ruled out by considering *symmetric* algorithms [14], in which processes may only compare their identifiers, i.e., use their relative ranks.

Several papers [6,25,26] claimed to prove that there is no symmetric wait-free algorithm for $(2n-2)$-nonadaptive renaming. All these proofs were by reduction to *weak symmetry breaking* (WSB). In WSB, each process should output a single bit, such that if all processes output, then not all of them output the same value. All these papers used closely related topological lemmas to prove that WSB is not wait-free solvable.

A few years ago, however, Castañeda and Rajsbaum showed that these lemmas are incorrect when the binomial coefficients $\binom{n}{1},\ldots,\binom{n}{n-1}$ are relatively prime. For these values of n, they proved that a wait-free symmetric WSB algorithm exists [15]. For all other values of n, they proved the impossibility of solving WSB [14] using symmetric wait-free algorithms. Both results of Castañeda and Rajsbaum use nontrivial topological tools on *oriented manifolds*. The lower bound result was later reproved using arguments from algebraic topology [13], and the upper bound was recently simplified [4].

Our contribution. This paper proves that there are no wait-free algorithms for solving $(n-1)$-set agreement, $(2p-\lceil p/n-1\rceil)$-adaptive renaming and $(2n-2)$-nonadaptive renaming, by using elementary tools. We also derive a new lower bound on the number of names needed for renaming, for arbitrary values of n.

We prove that $(n-1)$-set agreement is not wait-free solvable using a simple counting argument, inspired by a similar proof of Sperner's Lemma [27, Section 5a]. The proof considers a wait-free algorithm, in which each process starts with its identifier as input, and decides on the input of some (possibly other) process. We prove that such an algorithm has an *n-valued* execution, in which n different values are decided. This is done by counting the number of n-valued executions within a subset of the possible executions of the algorithm. This shows that no wait-free algorithm can solve $(n-1)$-set agreement.

To prove that SSB, and hence, $(2p-\lceil p/n-1\rceil)$-adaptive renaming, are not wait-free solvable, we consider a wait-free algorithm in which a process has no input, and it has to decide on a single bit. We show that the algorithm has a *univalued* execution, in which only 0 or only 1 are decided, within the subset of executions considered above. This is done by assigning a $+1$ or -1 sign to each execution, and showing that the count of univalued executions according to these signs is nonzero, showing that the algorithm has a univalued execution.

Similar ideas are used to prove the impossibility of solving WSB, and hence, $(2n-2)$-nonadaptive renaming, *when n is a power of a prime number*. We consider a symmetric, wait-free algorithm in which a process has no input, and it has to decide on a single bit. As for SSB, we show that the set of univalued executions is nonempty, by summing their signs. We use the symmetry of the algorithm in order to partition its executions into equivalence classes, and prove that the size of each class is a binomial coefficient of n. We use combinatorial properties of the binomial coefficient of n, when n is a prime power, to show that the sum of signs is nonzero. This implies that the algorithm has a univalued execution and hence, it cannot solve WSB.

All previous impossibility proofs for nonadaptive renaming use nontrivial topological tools and notions in an innovative way. While providing important intuition, the interaction between the topological and the operational arguments is difficult to understand, making the proofs less accessible to many researchers, and more prone to mistakes. Although inspired by them, our proofs do not use topological notions (see a detailed discussion in Section 6).

It can be proved (Appendix) that n is a prime power if and only if $\binom{n}{1},\ldots,\binom{n}{n-1}$ are not relatively prime. Hence, WSB is wait-free solvable when n is not a prime power [4,15]. This means that WSB is not wait-free solvable only for a small fraction of the possible values of n, since the fraction of prime powers in the set $\{1,\ldots,N\}$ tends to 0 as N goes to infinity. (In $\{1,\ldots,N\}$ there are asymptotically $\Theta\left(\frac{N}{\log N}\right)$ primes and $O\left(\sqrt{N}\log N\right)$ powers of primes with exponent greater than 1 [23, pp. 27–28].)

When n is not a prime power, let n' be the largest prime smaller than n. An M-renaming algorithm for n processes also solves M-renaming for n' processes, so $M \geq 2n'-1$. The currently best known bound on the distribution of prime numbers [8] implies the impossibility of $(2n-2n^{0.525}-2)$-nonadaptive renaming. If Cramér's conjecture [17] holds, the lower bound can be improved to $2n-\omega(\log^2 n)$. The only previously-known lower bound for arbitrary n was $n+1$ [2].

Organization. The rest of the paper is organized as follows. Section 2 describes the basic model and specifically, *block executions* and their properties. Section 3 presents the impossibility result for k-set agreement. Section 4 defines the (strong and weak) symmetry breaking tasks and their relation to renaming, while Section 5 presents the related impossibility results. We conclude with a discussion of the results and their relation to previous work, in Section 6. The Appendix shows that our characterization is equivalent to the one of Castañeda and Rajsbaum [14,15].

2. Preliminaries

2.1. Computational model

We use a standard model of an asynchronous shared-memory system [3,7]. A system consists of a set of n *processes*, denoted $\mathbb{P} = \{p_0,\ldots,p_{n-1}\}$, each of which is a (possibly infinite) deterministic

[1] The algorithm was originally presented for the message-passing model, but it can be extended to the shared-memory model. The bound on the number of names was stated differently, but it can be shown to be $(2p-1)$-adaptive.