View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Dagstuhl Research Online Publication Server

Brief Announcement: Neighborhood Mutual Remainder and Its Self-Stabilizing Implementation of Look-Compute-Move Robots

Shlomi Dolev
Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel
dolev@cs.bgu.ac.il

Sayaka Kamei
Graduate School of Engineering, Hiroshima University, Japan
s-kamei@se.hiroshima-u.ac.jp

Yoshiaki Katayama
Graduate School of Engineering, Nagoya Institute of Technology, Japan
katayama@nitech.ac.jp

Fukuhito Ooshita
Graduate School of Science and Technology, Nara Institute of Science and Technology, Japan
f-oosita@is.naist.jp

Koichi Wada
Faculty of Science and Engineering, Hosei University, Japan
wada@hosei.ac.jp

Abstract

In this paper, we define a new concept *neighborhood mutual remainder* (NMR). An NMR distributed algorithms should satisfy *global fairness*, *l-exclusion* and *repeated local rendezvous* requirements. We give a simple self-stabilizing algorithm to demonstrate the design paradigm to achieve NMR, and also present applications of NMR to a *Look-Compute-Move* robot system.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases neighborhood mutual remainder, self-stabilization, LCM robot

Digital Object Identifier 10.4230/LIPIcs.DISC.2019.43

Related Version `https://arxiv.org/abs/1903.02843`

Funding This work was supported in part by JSPS KAKENHI No. 17K00019, 18K11167, 19K11823 and 19K11828, Rita Altura Trust Chair in Computer Science, the Ministry of Science and Technology, Israel &Japan Science and Technology Agency (JST) SICORP (Grant#JPMJSC1806) and the German Research Funding Organization (DFG, Grant#8767581199).

1 Introduction

Distributed systems sometimes encounter *mutually exclusive* operations such that, while one operation is executed by a participant, another operation cannot be executed by the participant and its neighboring participants. For example, we can consider a Look-Compute-Move (LCM) robot system, where each robot repeats executing cycles of look, compute, and move phases. Some algorithms assume the move-atomic property, that is, while robot r executes look and compute phases, r's neighbors cannot execute a move phase. In this case, the move operation and the look/compute operations are mutually exclusive.

To execute mutually exclusive operations consistently, participants should schedule the operations carefully. One may think we can apply *mutual exclusion* or *local mutual exclusion* to solve the local synchronization problem. Mutual exclusion (resp., local mutual exclusion)

© Shlomi Dolev, Sayaka Kamei, Yoshiaki Katayama, Fukuhito Ooshita, and Koichi Wada;
licensed under Creative Commons License CC-BY

33rd International Symposium on Distributed Computing (DISC 2019).
Editor: Jukka Suomela; Article No. 43; pp. 43:1–43:3
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

guarantees that no two participants (resp., no two neighboring participants) enter a critical section (CS) at the same time. Indeed, if participants execute mutually exclusive operations only when they are in the CS, they can keep the consistency because no two neighboring participants execute the mutually exclusive operations at the same time. On the other hand, this approach seems expensive because participants execute the operations sequentially despite that they are allowed to execute the same operation simultaneously. Also, to realize local mutual exclusion, participants should achieve symmetry breaking because one participant should be selected to enter the CS. However, in highly-symmetric distributed systems such as the LCM robot system, it is difficult or even impossible to achieve deterministic symmetry breaking and thus achieve local mutual exclusion.

From this motivation, we define the *neighborhood mutual remainder* (NMR) distributed task over a distributed system with a general, non-necessarily complete, communication graph. An NMR distributed algorithm should satisfy *global fairness*, *l-exclusion*, and *repeated local rendezvous* requirements. Global fairness is satisfied when each participant executes the CS infinitely often, l-exclusion is satisfied when at most l neighboring processes enter the CS at the same time, and repeated local rendezvous is satisfied when, for each participant, infinitely often no participant in the closed neighborhood is in the critical or trying sections. Unlike the classical (local) mutual exclusion problem, the NMR allows up to l neighboring participants to be simultaneously in the CS, but requires a guarantee for neighborhood rendezvous in the remainder.

For example, in the LCM robot system, the move-atomic property can be achieved by NMR: Each robot executes look and compute phases when it is in the CS, and executes a move phase only when no robot in its closed neighborhood is in the CS. While some robot executes look and compute phases, none of its neighbors executes a move phase. From the global fairness and local rendezvous properties, all robots execute look, compute, and move phases infinitely often.

Our contributions. In this BA, we formalize the concept of NMR, and give a design paradigm to achieve NMR. To demonstrate the design paradigm, we consider synchronous distributed systems and give a simple self-stabilizing algorithm for NMR. To consider the simplest case, we assume $l = \Delta + 1$, where Δ is the maximum degree, that is, l-exclusion is always satisfied. In this case, the NMR does not require symmetry breaking, however, is still useful for some applications.

In the full version [1], to demonstrate applicability of NMR, we implement a self-stabilizing synchronization algorithm for the LCM robot system by using the aforementioned design paradigm. First, we realize the move-atomic property in a self-stabilizing manner on the assumption that robots repeatedly receive clock pulses at the same time. After that, we extend it to the assumption that robots receive clock pulses at different times but the duration between two pulses is identical for all robots. Lastly, on the same assumption, we implement the fully synchronous (FSYNC) model in a self-stabilizing manner. This research presents the first self-stabilizing implementation of the LCM synchronization, allowing the implementation in practice of any self-stabilizing or stateless robot algorithm, where robots possess independent clocks that are advanced in the same speed.

2 Preliminary

A distributed system is represented by an undirected connected graph $G = (V, E)$, where $V = \{v_0, \ldots, v_{k-1}\}$ is a set of processes and $E \subseteq V \times V$ is a set of communication links between processes. Processes are anonymous and identical. Process v_i is a neighbor of v_j if

Algorithm 1 Self-Stabilizing NMR Algorithm for $l = \Delta + 1$. Pseudo-Code for v_i.

```
Upon a global pulse
    $N_i := |N[i]|$; $MaxN_i := \max\{N_j \mid v_j \in N[i]\}$; $Clock_i := (Clock_i + 1) \mod (MaxN_i + 1)$;
    if $Clock_i = 1$ then enter the critical section and leave before the next pulse;
    else if $\forall v_j \in N[i][Clock_j \neq 1]$ then rendezvous in the remainder section;
```

$(v_i, v_j) \in E$ holds. A neighborhood of v_i is denoted by $N(i) = \{v_j \mid (v_i, v_j) \in E\}$, and the degree of v_i is denoted by $\delta(i) = |N(i)|$. Let $\Delta = \max\{\delta(i) \mid v_i \in V\}$. A closed neighborhood of v_i is denoted by $N[i] = N(i) \cup \{v_i\}$.

Each process is a state machine that changes its state by actions. We adopt the *state-reading model* as a communication model. In this model, each v_i can directly read states of all processes in $N[i]$ simultaneously and update its own state. Processes operate synchronously based on global pulses. That is, all processes regularly receive the pulse at the same time, and operate when they receive the pulse. The duration of local computation is sufficiently small so that every process completes the local computation before the next pulse.

▶ **Definition 1** (Neighborhood mutual remainder (NMR)). *The system achieves NMR if the following three properties hold.*

- *Global fairness: Every process infinitely often enters the CS.*
- *l-exclusion: For every process v_i, at most l processes in $N[i]$ enter the CS simultaneously.*
- *Repeated local rendezvous: For every process v_i, infinitely many instants exist such that no process in $N[i]$ is in the critical or trying section.*

3 A self-stabilizing algorithm for neighborhood mutual remainder

In this section, we give a design paradigm to achieve NMR. As an example, we realize a self-stabilizing algorithm to achieve NMR in case of $l = \Delta + 1$. We say an algorithm is self-stabilizing if, from any initial configuration, the system eventually exhibits the desired behavior.

The self-stabilizing NMR algorithm is given in Algorithm 1. Let us consider a simple setting where $|N[i]|$ is identical for any v_i. Every process v_i maintains a clock $Clock_i$ that is incremented by 1 modulo $(|N[i]| + 1)$ in every pulse. The value of $Clock_i$ may differ from the value of $Clock_j$, for a neighbor v_j of v_i. Say, for the sake of simplicity, that v_i may possess the CS only when $Clock_i = 1$. Thus, ensuring that there is a configuration in which all processes in the remainder is equivalent to ensuring that there is a configuration in which the values of all the above clocks are not equal to 1. Using the pigeon-hole principle in every $|N[i]| + 1$ consequence pulse clocks, there must be a configuration in which no clock value of neighboring processes is 1 and at the same time $Clock_i$ is not 1 either. Hence, the NMR must hold. Since $|N[i]| \neq |N[j]|$ may hold for some v_i and v_j, we use $MaxN_i = \max\{|N[j]| \mid v_j \in N[i]\}$ instead of $|N[i]|$. Since every process $v_j \in N[i]$ enters the CS at most once in $MaxN_i + 1$ consecutive pulses, we can still use the pigeon-hole principle and hence the NMR must hold.

References

1 S. Dolev, S. Kamei, Y. Katayama, F. Ooshita, and K. Wada. Neighborhood Mutual Remainder: Self-Stabilizing Implementation of Look Compute Move. *arXiv*, 2019. `arXiv:1903.02843`.