A New Characterization of Atomic Registers shared-memory

Antoine Gaillard

A New Characterization of Atomic Registers shared-memory

Antoine Gaillard
Ecole Polytechnique, France
gaillard@lix.polytechnique.fr

Abstract

We study the fundamental communication properties of two major shared-memory models, namely the ones in which processes communicate via Single-Writer/Multi-Reader (SWMR for short) atomic registers and Atomic-Snapshot object, respectively.

In 1998, Gafni already adressed this question. We prove in this paper that the characterization he gave does not match the well-known equivalence between these two models.

The formalism we use is the HO model developed by Charron-Bost and Schiper, in which the features of a specific system (degree of synchrony, failure model...) are encapsulated into a single abstract entity, called a *communication predicate*. In particular, we give a new characterization of SWMR and Atomic-Snapshot models only in terms of the predicates that capture the properties of their communications. Moreover, we prove that our characterization is consistent regarding the equivalence result mentioned above.

KEY WORDS: Round-based model, fault-tolerant distributed systems, shared-memory model, Single-Writer/Multi-Reader atomic registers, Atomic-Snapshot registers.

Introduction

In [9], Gafni defined a new round-based computational model, called *Round-by-Round Fault Detector model* (RRFD for short), for the analysis of distributed systems. In this model, computations are defined to evolve round by round, the properties of communication between processes, either via shared variables or by message passing, being captured by a single module called *Round-by-Round Fault Detector* module. At each round r, and for each process p, the module provides a set of suspected processes from which p will not wait for a message. Hence, synchrony degree and failure model are encapsulated into the same abstract entity.

In [6], Charron-Bost and Schiper combined this approach with the *Transmission Faults model*, introduced by Santoro and Widmayer in [14], designed for synchronous message-passing systems and which locates the failure without specifying their cause. As a result, Charron-Bost and Schiper developed the *Heard-Of model* (HO for short), a computational round-based model suitable for distributed systems subject to benign failures, which is based only on the notion of *transmission failure* and renounce the one of *faulty component*. Computations in this model evolve in rounds. At each round r, each process p sends a message to all the others and waits to receive messages from a subset of them, denoted by $HO(p, r)$, which consists of the processes that p "hears of" at round r. Communication missed at a round is lost, and so rounds are *communication-closed layers*, using the terminology of [7]. A transmission failure from q to p at round r is thus charaterized by the fact that q does not belong to $HO(p, r)$. A crash of some process p can be modeled by the fact that, from some point in the execution, p does not belong to any heard-of set and thus has no impact on the rest of the computation. The features of a particular system (synchrony degree, failure model,...) are captured by a *communication predicate*, which is a predicate over the collections of the $HO(p, r)$'s sets.

The HO model lies at a very high abstraction level. It is thus important to study what communication predicates can be implemented in what system. This allows to characterize and to compare different systems.

Contribution

In [9], Gafni informally studied the capability of the RRFD model to cover various classical types of distributed systems. We analyse here the results he gave regarding two particular types of shared-memory systems, namely the ones in which processes communicate via *SingleWriter/MultiReader atomic registers* (SWMR for short) [10, 12, 11], and *Atomic-Snaphot objects* [1, 2], respectively.

For the first type, each process p is associated with a read/write register R_p such that (i) p is the sole process that is allowed to write into R_p, and (ii) every process can read the value of R_p. Moreover, accesses to a given register are atomic in the sense of [12], i.e., in any execution of the system, there is a way of totally ordering reads and writes so that the values returned by the reads are the same as if the operations had been performed in that order, with no overlapping. An Atomic-Snapshot object consists of an array of SWMR atomic registers, one for each process of the system. The main difference with SWMR registers lies in the fact that a process can atomically take a *snapshot* of the whole array, instead of reading all the registers one after the other. Gafni informally showed the correspondance between Atomic-Snapshot systems and the RRFD module such that at each round and for any two processes p and q, the sets of processes suspected by p and q are ordered by inclusion. As for SWMR systems, he proposed the RRFD module that ensures that at each round there exists some process which is not supected by anyone. Note that any RRFD module can be seen as a communication predicate by defining, for each round, the heard-of set of any process to be the complementary of the set of suspected processes provided by the RRFD module to this process at that round.

The rest of this paper is organized as follow. We give a formal proof of Gafni's result regarding the RRFD module corresponding to Atomic-Snapshot systems in Section 2, and demonstrate that the one he proposed for SWMR systems is strictly weaker. The well-known equivalence between SWMR systems and Atomic-Snaphots ones [1, 2, 8] then leads us to consider an alternative communication predicate. In Section 3, we prove this third predicate to be (1) implementable in SWMR systems and (2) equivalent to the one corresponding to Atomic-Snapshot systems. Moreover, we give a direct proof of the fact that this predicate is strictly stronger than the one corresponding to the RRFD module proposed by Gafni for characterizing SWMR systems. Section 4 draws some conclusions.

1 The HO model

As explained in the Introduction, computations in this model are structured in rounds that are *communication-closed layers* in the sense that a message sent at some round can be received only at that round.

1.1 Heard-Of sets and communication predicates

Let Π be a finite non-empty set of cardinality n, and let M be a set of messages (optionally including a *null* placeholder indicating the empty message). To each p in Π, we associate a *process*, which consists of the following components: a set of states denoted by $states_p$, a subset $init_p$ of initial states, and for each positive integer r called *round number*, a message-sending function S_p^r mapping $states_p \times \Pi$ to a unique message from M, and a state-transition function T_p^r mapping $states_p$ and partial vectors (indexed by Π) of elements of M to $states_p$. The collection of processes is called an *algorithm on* Π.

In each round r, a process p

1. applies S_p^r to the current state, and emits the "messages" to be sent (according to its sending function S_p^r) to each process;
2. applies T_p^r to the partial vector of incoming messages whose support is $HO(p, r)$.

Computations evolves in an infinite sequence of rounds. Each *run* is entirely determined by the initial configuration (i.e., the collection of process initial states), and the collection $(HO(p, r))_{p\in\Pi, r>0}$ of heard-of sets.

A *communication predicate* $\mathcal{P}$ is a predicate over collections of subsets of Π (representing heard-of collections), that is not the constant predicate `false`. As an exemple, the predicate

$$\forall r > 0,\ \forall p \in \Pi\ :\quad |HO(p, r)| \geq n - f$$

models the fact that at each round, each process receives a message from at least $n - f$ distinct processes. A *heard-of machine* (HO machine for short) for a set of processes Π is a pair $(\mathcal{A}, \mathcal{P})$, where $\mathcal{A}$ is an algorithm on Π, and $\mathcal{P}$ is a communication predicate.

1.2 Translations

Our concern in this paper is to compare different communication predicates and to determine the one corresponding to SWMR systems by using the equivalence between such systems and the Atomic-Snapshot ones. For that, we will use the notion of *equivalence* between communication predicates. We thus have to formalize what it means for an HO machine $(\mathcal{A}, \mathcal{P})$ to emulate a communication predicate $\mathcal{P}'$. This leads us to introduce the notion of *translation* of $\mathcal{P}$ into $\mathcal{P}'$ defined in [6].

Let k be any positive integer, and let $\mathcal{A}$ be an algorithm that maintains a variable $NewHO_p$ at every process p, which contains a subset of Π. We call *macro-round* ρ the sequence of the k consecutive rounds $k(\rho-1)+1,\ldots,k\rho$. The value of $NewHO_p$ at the end of macro-round ρ is denoted $NewHO_p^{(\rho)}$. We say that the HO machine $\mathcal{M}=(\mathcal{A},\mathcal{P})$ *emulates* the communication predicate $\mathcal{P}'$ in k rounds if for any run of $\mathcal{M}$, the following holds:

E1: If process q belongs to $NewHO_p^{(\rho)}$, then there exist an integer $l \in [\![1;k]\!]$, a chain of $l+1$ processes $p_0, p_1, \ldots p_l$ from $p_0=q$ to $p_l=p$ and a subsequence of l rounds $r_1,\ldots,r_l$ in macro-round ρ such that for any index i, $1\leq i\leq l$, we have $p_{i-1}\in HO(p_i,r_i)$.

E2: The collection $\left(NewHO_p^{(\rho)}\right)_{p\in\Pi,\rho>0}$ satisfies predicate $\mathcal{P}'$.

Condition E1 avoid trivial emulations of $\mathcal{P}'$ since it requires that for each macro-round ρ, if some process q belongs to $NewHO_p^{(\rho)}$, then p actually hears of q during this macro-round (possibly not directly but through intermediate processes). If there exists an algorithm $\mathcal{A}$ such that the HO machine emulates $\mathcal{P}'$ in k rounds, then we write $\mathcal{P}\succeq_k\mathcal{P}'$, and we say that $\mathcal{A}$ is a *k-rounds translation* of $\mathcal{P}$ into $\mathcal{P}'$. We shall also say that $\mathcal{P}$ is at least as *strong* as $\mathcal{P}'$. Moreover, predicates $\mathcal{P}$ and $\mathcal{P}'$ are said to be *equivalent* if $\mathcal{P}\succeq_k\mathcal{P}'$ and $\mathcal{P}'\succeq_{k'}\mathcal{P}$ for some integers k and k'.

Note that if $\mathcal{P}\Rightarrow\mathcal{P}'$, the trivial algorithm in which each process p writes the value of $HO(p,r)$ into $NewHO_p$ at the end of each round r is a 1-round translation of $\mathcal{P}$ into $\mathcal{P}'$.

2 Gafni's characterization of SWMR and Atomic-Snapshot systems

As said in the Introduction, Gafni presented in [9] the RRFD model, a round-based model for the analysis of distributed systems subject to benign failures, and informally adressed its expressivity. In particular, he introduced two RRFD modules that he claimed to "naturally" correspond to systems with Atomic-Snaphots objects and SWMR registers, respectively.

Informally speaking, in a system with SWMR registers each process of a set Π is associated with a register R_p that supports two operations: (1) $Write(R_p,v)$, where v is a value drawn from a given set V, and (2) $Read(R_p)$. Each process p can read all the registers but no process $q\neq p$ can write into R_p. Moreover, the registers we consider are *atomic* in the sense of [12], i.e., the *Read* and *Write* operations behave as if they occur in some definite order. In other words, for any execution of the system there exists a way of totally ordering them so that the values returned by the reading operations are the same as if the operations had been performed in that order, with no overlapping.

An Atomic-Snapshot object consists of an array of n atomic SWMR registers, one for each process of the system. Each process can either write a value into its register (it *updates* the object) or take a *snapshot* of the whole array (it performs a *scan*). The operations on the object (either updates or scans) are atomic in the same sense as above.

2.1 Predicates corresponding to Gafni's RRFD modules

The two RRFD modules claimed by Gafni to correspond to SWMR and Atomic-Snapshot systems, respectively, are expressed in the HO model by the two predicates $\mathcal{P}_{Gaf}$ and $\mathcal{P}_{\mathcal{RD}}$, given in Figure 1.

To see that Atomic-Snapshot systems effectively implement $\mathcal{P}_{\mathcal{RD}}$ consider the algorithm *AtSn*, given as Algorithm 1. Roughly speaking, at each round r, all processes can access an Atomic-

$$\mathcal{P}_{Gaf} \;::\; \forall r > 0,\ \bigcap_{p\in\Pi} HO(p,r) \neq \emptyset$$

$$\mathcal{P}_{\mathcal{RD}} \;::\; \forall r > 0, \left| \begin{array}{l} p \in HO(p,r) \\ \forall p,q \in \Pi,\ \big(HO(p,r) \subseteq HO(q,r)\big) \vee \big(HO(q,r) \subseteq HO(p,r)\big) \end{array} \right.$$

Figure 1: Predicates corresponding to Gafni's RRFD modules

Algorithm 1 Algorithm *AtSn*: code for process p

```
1: Initialization
2:     r ∈ ℕ; initially 1
3:     v_p ∈ V, initially v_p = id_p, with id_p the identifier of p
4:     HO_p ⊆ Π

5: Round r:
6:     HO_p := ∅
7:     Write(A_r[p], v_p)
8:     Scan(A_r)
9:     HO_p := {q : Scan(A_r)[q] ≠ ⊥}
10:    r := r + 1
```

Snapshot object A_r. Each process p first updates its own component $A_r[p]$ by writing its identifier into it and then scans the whole array. The following proposition shows that at the end of any round r, for any two processes p and q, the sets of processes' identifiers read by p and q, respectively, at round r are ordered by inclusion.

Proposition 2.1. *Let e be any execution of AtSn in an Atomic-Snapshot system and let r be any integer such that $r \geq 1$. At the end of round r, the collection $\big(HO_p\big)_{p\in\Pi}$ satisfies $\mathcal{P}_{\mathcal{RD}}$.*

Proof: Let r be any integer such that $r \geq 1$. First let p be any process in Π. The code of *AtSn* (lines 7 and 8) directly implies that p belongs to HO_p at the end of round r.

Now let q be any process distinct from q. By the atomicity of the Atomic-Snapshot object, either p or q is the first process to scan A_r. Since the values written cannot be deleted, we deduce that all the identifiers read by the first process that scans A_r are also read by the second one. Hence, at the end of round r, we have either $HO_p \subseteq HO_q$ or $HO_q \subseteq HO_p$. □

The fact that an Atomic-Snaphot object can be implemented in a system whose executions satisfy $\mathcal{P}_{\mathcal{RD}}$ is a simple corollary of [5]. Combining this result with Proposition 2.1, we derive the following theorem:

Theorem 2.2. *The communication predicate $\mathcal{P}_{\mathcal{RD}}$ entirely captures the communication properties of Atomic-Snapshot systems in the benign case.*

For the SWMR systems, in which at each round each process writes into its own register and then reads all the registers, Gafni noticed that, by the atomicity assumption, the first process to write into its register at some round is necessarily heard by all the others at that round, and so belongs to all the heard-of sets.

At this point, regarding the equivalence between the two considered types of systems and Theorem 2.2, it seems natural to check whether the two corresponding predicates $\mathcal{P}_{\mathcal{RD}}$ and $\mathcal{P}_{Gaf}$ are actually equivalent, according to the definition given in Section 1.2. Although, the two following propositions show that $\mathcal{P}_{\mathcal{RD}}$ is actually strictly stronger than $\mathcal{P}_{Gaf}$.

Proposition 2.3. *The communication predicate $\mathcal{P}_{\mathcal{RD}}$ is at least as strong as $\mathcal{P}_{Gaf}$.*

$$\mathcal{P}_{sym} \;::\; \forall r > 0,\ \forall p, q \in \Pi \;:\; \big(p \in HO(q,r)\big) \vee \big(q \in HO(p,r)\big)$$

Figure 2: Predicate $\mathcal{P}_{sym}$, our candidate for SWMR atomic registers

Proof: We are going to show that $\mathcal{P}_{\mathcal{RD}}$ in fact directly implies $\mathcal{P}_{Gaf}$. Let $\big(HO(p,r)\big)_{p\in\Pi, r>0}$ be a collection of heard-of sets which satisfies $\mathcal{P}_{\mathcal{RD}}$ and let $r > 0$ be any round number.

The second part of $\mathcal{P}_{\mathcal{RD}}$ implies that the collection of all the heard-of sets of round r is ordered by the inclusion, while the first one ensures that each of them is nonempty. It thus follows that their intersection is nonempty. We therefore conclude that $\big(HO(p,r)\big)_{p\in\Pi, r>0}$ satisfies $\mathcal{P}_{Gaf}$. □

Proposition 2.4. *The communication predicate $\mathcal{P}_{Gaf}$ is not at least as strong as $\mathcal{P}_{\mathcal{RD}}$.*

Proof: We proceed by contradiction. Assume that there exists an algorithm $\mathcal{A}$ that translates $\mathcal{P}_{Gaf}$ into $\mathcal{P}_{\mathcal{RD}}$ in k rounds, for some positive integer k.

Let e be the execution of the HO machine $(\mathcal{A}, \mathcal{P}_{Gaf})$ such that

$$\exists q \in \Pi,\ \forall r > 0,\ \forall p \in \Pi \;:\; HO(p,r) = \{q\}.$$

Let ρ be a macro-round of e and let p be some process, other than q. Since $\mathcal{A}$ is a translation of $\mathcal{P}_{Gaf}$ into $\mathcal{P}_{\mathcal{RD}}$, condition E2 of the definition of a translation ensures that the collection $\big(NewHO_p^{(\rho)}\big)_{p\in\Pi,\rho>0}$ satisfies $\mathcal{P}_{\mathcal{RD}}$, and so that p belongs to $NewHO_p^{(\rho)}$. From condition E1, we then deduce that p hears of itself (possibly through intermediate processes) during macro-round ρ. In particular, this implies that there exist some round r of macro-round ρ and some process $p' \in \Pi$ (possibly $p' = p$) such that $p \in HO(p', r)$, which contradicts the fact that for all rounds r of e and for all processes $p' \in \Pi$, $HO(p', r) = \{q\}$, whith $q \neq p$. Hence $\mathcal{A}$ is not a translation of $\mathcal{P}_{Gaf}$ into $\mathcal{P}_{\mathcal{RD}}$. □

By combining Propositions 2.3 and 2.4, we derive the following theorem:

Theorem 2.5. *The communication predicate $\mathcal{P}_{\mathcal{RD}}$ is strictly stronger than $\mathcal{P}_{Gaf}$.*

This result points out that, since $\mathcal{P}_{\mathcal{RD}}$ has been shown to be the predicate which entirely characterizes Atomic-Snapshot systems, we cannot morally consider that $\mathcal{P}_{Gaf}$ effectively corresponds to SWMR systems.

3 In search for equivalence

As shown in the above section, the communication predicate $\mathcal{P}_{Gaf}$ appears to be too weak for characterizing SWMR systems. We consider here an alternative predicate, called $\mathcal{P}_{sym}$ and given in Figure 2, that we claim to be (i) guaranteed by such systems, and (ii) equivalent to $\mathcal{P}_{\mathcal{RD}}$.

For the first point, we introduce the algorithm *SWMR*, given as Algorithm 2. In this algorithm, at each round $r \geq 1$, each process p writes its own identifier into a SWMR atomic register R_p^r and then reads all the R_q^r's. The following proposition shows that at each round r, for any two processes p and q, either p belongs to HO_q or q belongs to HO_p.

Proposition 3.1. *Let e be any execution of SWMR in a SWMR system. At the end of any round r, we have:*

$$p \in HO_q \quad \vee \quad q \in HO_p.$$

Algorithm 2 Algorithm *SWMR*: code for process p

1: **Initialization**
2: $r \in \mathbb{N}$; initially 1
3: $v_p \in V$, initially $v_p = id_p$, whith id_p being the identifier of process p
4: $HO_p \subseteq \Pi$

5: **Round** r**:**
6: $HO_p := \emptyset$
7: $Write(R_p^r, v_p)$
8: Pour tout $q \in \Pi$, $Read(R_q^r)$
9: $HO_p := \{q \ : \ Read(R_q^r) \neq \perp\}$
10: $r := r + 1$

Proof: Let p and q be two processes in Π. First assume that $p = q$. The code of *SWMR* trivially implies that p reads its own identifier at round r, and so $p \in HO_p$. Now assume that p and q are two distinct processes and that $p \notin HO_q$ at the end of round r. By the atomicity of the registers, this implies that q started executing line 8 of its code before p executed line 7. The code of *SWMR* therefore ensures that p started reading the registers after q has written its identifier, and so $q \in HO_p$ at the end of round r. □

3.1 $\mathcal{P}_{sym}$ and $\mathcal{P}_{\mathcal{RD}}$ are equivalent

The result of the previous section shows that $\mathcal{P}_{sym}$ is guaranteed by SWMR systems in the benign case. We give in this section a rigorous proof of the equivalence between $\mathcal{P}_{sym}$ and $\mathcal{P}_{\mathcal{RD}}$ which, thanks to Theorem 2.2 and the equivalence between SWMR systems and Atomic-Snapshot ones, demonstrate that $\mathcal{P}_{sym}$ entirely characterizes SWMR systems in the benign case.

We start our demonstration by showing that $\mathcal{P}_{\mathcal{RD}}$ is at least as strong as $\mathcal{P}_{sym}$.

Theorem 3.2. *The communication predicate $\mathcal{P}_{\mathcal{RD}}$ implies $\mathcal{P}_{sym}$.*

Proof: Let $\big(HO(p,r)\big)_{p\in\Pi,r>0}$ be a collection of heard-of sets that satisfies $\mathcal{P}_{\mathcal{RD}}$. We show that $\big(HO(p,r)\big)_{p\in\Pi,r>0}$ also satisfies $\mathcal{P}_{sym}$.

Let $r > 0$ be any round number and let p and q be any two processes in Π. We have to demonstrate that either $p \in HO(q,r)$ or $q \in HO(p,r)$.

- If $p = q$, then the first part of $\mathcal{P}_{\mathcal{RD}}$ implies that $p \in HO(p,r)$.
- Now assume that p and q are distinct and $p \notin HO(q,r)$. The first part of $\mathcal{P}_{\mathcal{RD}}$ implies that $p \in HO(p,r)$, and therefore that $HO(p,r) \not\subseteq HO(q,r)$. The second part of $\mathcal{P}_{\mathcal{RD}}$ then ensures that $HO(q,r) \subseteq HO(p,r)$. Since, by the first part of $\mathcal{P}_{\mathcal{RD}}$, q belongs to $HO(q,r)$, we finally conclude that $p \in HO(q,r)$.

Hence, the collection $\big(HO(p,r)\big)_{p\in\Pi,r>0}$ satisfies $\mathcal{P}_{sym}$, as needed. □

It remains to show that $\mathcal{P}_{sym}$ is at least as strong as $\mathcal{P}_{\mathcal{RD}}$, i.e., there exists a translation of $\mathcal{P}_{sym}$ into $\mathcal{P}_{\mathcal{RD}}$. We present here the algorithm $\mathcal{SRD}$, given as Algorithm 3, which we prove to be such a translation. Informally speaking, at each macro-round ρ, each process p maintains a variable D_p consisting of the processes that p hears of, directly or through intermediate processes, during ρ. If D_p does not change for a given number of rounds, then p defines $NewHO_p$ to be the set of all processes it heard of during ρ. This scheme is inspired by the double-collect used in many implementations (e.g. [2], [3]) of Atomic-Snapshot objects by SWMR atomic registers.

Algorithm 3 The $\mathcal{SRD}$ algorithm

```
 1: Initialization:
 2:     D_p ⊆ Π; initially {p}
 3:     NewHO_p ⊆ Π; initially ∅
 4:     Known_p ⊆ Π; initially ∅
 5:     k, N_p ∈ ℕ; initially 0

 6: S_p^r :
 7:     send ⟨D_p⟩ to all processes

 8: T_p^r :
 9:     if ∃l > 0 such that r = (n^2 + n) · l + 1 then
10:         NewHO_p := ∅;
11:         Known_p := ∅;
12:         D_p := {p}
13:     (endif)
14: if NewHO_p = ∅ then
15:     Known_p := ⋃{D_q : q ∈ HO(p, r)}

16:     if Known_p ⊆ D_p then
17:         k := k + 1
18:     else
19:         k := 0

20:     D_p := D_p ∪ Known_p

21:     if k = 1 then
22:         N_p := |HO(p, r)|
23:     if k = 2 then
24:         k := k − 1; N_p := N_p − 1

25:     if N_p = 0 then
26:         NewHO_p := D_p
```

Before starting our correctness proof, we introduce some piece of notation which we will use in the sequel. Let p be any process, and x_p be some variable local to p. For each round $r > 0$, we denote by $x_p^{(r)}$ the value of x_p at the end of round r.

We first show that during any macro-round, each process p sets its variable $NewHO_p$ to another value than $\emptyset$. In other word, the guard in line 25 is well defined.

Lemma 3.3. *Let e be an execution of the HO machine $(\mathcal{SRD}, \mathcal{P}_{sym})$ and let ρ be any macro-round of e. Let $r_1, r_2, \ldots, r_{n+1}$ be any sequence of $n+1$ consecutive rounds of ρ and let p be some process in Π.*

If $NewHO_p^{(r_1-1)} = \emptyset$, then there exists some index $i \in [\![1; n+1]\!]$ such that

$$D_p^{(r_i)} \neq D_p^{(r_i-1)} \quad \vee \quad NewHO_p^{(r_i)} \neq \emptyset.$$

Proof: Let e be an execution of $(\mathcal{SRD}, \mathcal{P}_{sym})$ and let $\rho > 0$ be any macro-round of e. Let $r_1, r_2, \ldots, r_{n+1}$ be a sequence of $n+1$ consecutive rounds of ρ. We proceed by contradiction.

Assume that there exists some process p such that $NewHO_p^{(r_1-1)} = \emptyset$ and, for all indices $i \in [\![1; n+1]\!]$,

$$D_p^{(r_i)} = D_p^{(r_i-1)} \quad \wedge \quad NewHO_p^{(r_i)} = \emptyset.$$

The code of $\mathcal{SRD}$ ensures that p executes the line 24 during each round $r_2, \ldots, r_{n+1}$. It follows that $N_p^{(r_{n+1})} \leq N_p^{(r_2)} - n$. Since $N_p^{(r_2)} = |HO(p, r_2)|$, we deduce that $N_p^{(r_{n+1})} \leq 0$. Therefore, there exists an index $j \leq n+1$ such that $N_p^{(r_j)} = 0$, and so such that p executes the line 26 at round r_j.

Hence $NewHO_p^{(r_j)} = D_p^{(r_j)}$. However, for all rounds $r > 0$, process p belongs to $D_p^{(r)}$, since $\mathcal{P}_{sym}$ is satisfied in e. We thus conclude that $NewHO_p^{(r_j)} \neq \emptyset$, a contradiction. □

As a corollary of the previous lemma, we derive the following proposition:

Proposition 3.4. *Let e be any execution of the HO machine $(\mathcal{SRD}, \mathcal{P}_{sym})$. For any macro-round ρ of e and for any process p, we have $NewHO_p^{(\rho)} \neq \emptyset$.*

Proof: Let e be an execution of $(\mathcal{SRD}, \mathcal{P}_{sym})$ and let $\rho > 0$ be any macro-round of e.

Let p be any process of Π. Since $D_p = \{p\}$ at the beginning of macro-round ρ and $D_p^{(r)} \subseteq \Pi$ for all rouds $r > 0$, we deduce that there exist at most $n-1$ rounds $r_1, r_2, \ldots, r_{n-1}$ of macro-round ρ such that, for all indices $i \in [\![1, ; n-1]\!]$, we have $D_p^{(r_i)} \neq D_p^{(r_i-1)}$.

Since ρ consists of $n^2 + n$ rounds, Lemma 3.3 implies that p necessarily executes the line 26 during ρ, and so $NewHO_p^{(\rho)} = D_p^{(\rho)}$. The result finally follows from the fact that, since $\mathcal{P}_{sym}$ is satisfied in e, for all rounds $r > 0$, we have $D_p^{(r)} \neq \emptyset$.

□

We are now in position to show that for any execution of the HO machine $(\mathcal{SRD}, \mathcal{P}_{sym})$, the collection $\left(NewHO_p^{(\rho)}\right)_{p\in\Pi,\rho>0}$ satisfies $\mathcal{P}_{\mathcal{RD}}$. The following proposition shows that every process p belongs to its own set $NewHO_p$ at the end of every macro-round ρ.

Proposition 3.5. *Let e be any execution of the HO machine $(\mathcal{SRD}, \mathcal{P}_{sym})$ and let $\rho > 0$ be any macro-round of e. For all processes $p \in \Pi$ we have $p \in NewHO_p^{(\rho)}$.*

Proof: Let e be any execution of the HO machine $(\mathcal{SRD}, \mathcal{P}_{sym})$ and let $\rho > 0$ be any macro-round of e.

Since $\mathcal{P}_{sym}$ is satisfied in e we have, for all rounds $r > 0$, $p \in D_p^{(r)}$. Moreover, Proposition 3.4 ensures that every process p executes the line 26 during ρ, and so that $NewHO_p^{(\rho)} = D_p^{(\rho)}$. We thus conclude that, for every process p, we have $p \in NewHO_p^{(\rho)}$. □

It remains to show that at the end of any macro-round ρ, for any two processes p and q, the sets $NewHO_p^{(\rho)}$ and $NewHO_q^{(\rho)}$ are ordered by inclusion.

Lemma 3.6. *Let e be any execution of the HO machine $(\mathcal{SRD}, \mathcal{P}_{sym})$, let $\rho > 0$ be any macro-round of e and let r_0 be any round of ρ.*

If p and q are two distinct processes that both execute line 26 at round r_0, then

$$NewHO_p^{(\rho)} \subseteq NewHO_q^{(\rho)} \quad \vee \quad NewHO_q^{(\rho)} \subseteq NewHO_p^{(\rho)}.$$

Proof: Let e be any execution of the HO machine $(\mathcal{SRD}, \mathcal{P}_{sym})$ and let $\rho > 0$ be any macro-round of e. Assume that there exist two distinct processes p and q that both execute line 26 at some round r_0 of macro-round ρ.

Since $\mathcal{P}_{sym}$ is satisfied in e, we have either $p \in HO(q, r_0)$ or $q \in HO(p, r_0)$. The code of $\mathcal{SRD}$ (line 15) then ensures that $D_p^{(r_0-1)} \subseteq Known_q^{(r_0)}$ or $D_q^{(r_0-1)} \subseteq Known_p^{(r_0)}$, respectively, which implies that $D_p^{(r_0-1)} \subseteq D_q^{(r_0)}$ or $D_q^{(r_0-1)} \subseteq D_p^{(r_0)}$

Since p and q both execute line 26 at round r_0, we necessarily have $D_p^{(r_0-1)} = D_p^{(r_0)} = NewHO_p^{(\rho)}$ and $D_q^{(r_0-1)} = D_q^{(r_0)} = NewHO_q^{(\rho)}$. Hence, we finally deduce that either $NewHO_p^{(\rho)} \subseteq NewHO_q^{(\rho)}$ or $NewHO_q^{(\rho)} \subseteq NewHO_p^{(\rho)}$. □

We now extend the result to the case in which p and q determine $NewHO_p$ and $NewHO_q$ during two distinct rounds.

Lemma 3.7. *Let e be any execution of the HO machine $(\mathcal{SRD}, \mathcal{P}_{sym})$, let $\rho > 0$ be any macro-round of e and let r_0 be any round of ρ.*

If p and q are two distinct processes that execute line 26 at rounds r_0 and r_1 respectively, with $r_0 \neq r_1$, then

$$NewHO_p^{(\rho)} \subseteq NewHO_q^{(\rho)} \quad \vee \quad NewHO_q^{(\rho)} \subseteq NewHO_p^{(\rho)}.$$

Proof: Let e be any execution of the HO machine $(\mathcal{SRD}, \mathcal{P}_{sym})$ and let $\rho > 0$ be any macro-round of e. Assume that there exist two distinct processes p and q that execute line 26 at rounds r_0 and r_1 respectively, with $r_0 \neq r_1$.

We proceed by contradiction and assume that

$$NewHO_p^{(\rho)} \not\subseteq NewHO_q^{(\rho)} \quad \wedge \quad NewHO_q^{(\rho)} \not\subseteq NewHO_p^{(\rho)}.$$

Since $\mathcal{P}_{sym}$ is satisfied in e, we have either $p \in HO(q, r_0)$ or $q \in HO(p, r_0)$. If we assume $p \in HO(q, r_0)$, then the code of $\mathcal{SRD}$ ensures that $D_p^{(r_0-1)} \subseteq D_q^{(r_0)}$. By the definition of r_0, we thus have $D_p^{(r_0-1)} = NewHO_p^{(\rho)}$. Moreover, since $D_q^{(r_0)} \subseteq NewHO_q^{(\rho)}$, it follows that $NewHO_p^{(\rho)} \subseteq NewHO_q^{(\rho)}$, a contradiction. We thus deduce that $p \notin HO(q, r_0)$, which implies $q \in HO(p, r_0)$, and so $D_q^{(r_0-1)} \subseteq D_p^{(r_0)} = NewHO_p^{(\rho)}$.

By assumption, $NewHO_p^{(\rho)} \not\subseteq NewHO_q^{(\rho)}$. Since $D_q^{(r_0-1)} \subseteq NewHO_q^{(\rho)}$, we then obtain $D_q^{(r_0-1)} \subset NewHO_p^{(\rho)}$. We also assumed that $NewHO_q^{(\rho)} \not\subseteq NewHO_p^{(\rho)}$. Hence there exists some process $q_0 \notin NewHO_p^{(\rho)}$ which belongs to $NewHO_q^{(\rho)}$.

Let r_q be the round of ρ such that $q_0 \in D_q^{(r_q)} \setminus D_q^{(r_q-1)}$. Since $D_q^{(r_0-1)} \subset NewHO_p^{(\rho)}$ and $q_0 \notin NewHO_p^{(\rho)}$, we necessarily have $r_q \geq r_0$.

Assume $r_q = r_0$. Then there exists some process q_1 distinct from p such that

$$q_0 \in D_{q_1}^{(r_0-1)} \quad \wedge \quad NewHO_p^{(\rho)} \not\subseteq D_{q_1}^{(r_0-1)} \quad \wedge \quad q_1 \in HO(q, r_0).$$

Now assume that $q_0 \in D_{q_1}^{(r_0-2)}$. This implies that $q_1 \notin HO(p, r_0 - 1)$ and therefore, under $\mathcal{P}_{sym}$, we have $p \in HO(q_1, r_0 - 1)$. By the definition of r_0, it follows that $NewHO_p^{(\rho)} \subseteq D_{q_1}^{(r_0-1)}$, and so $NewHO_p^{(\rho)} \subseteq D_q^{(r_0)}$, a contradiction. Hence, $q_0 \in D_{q_1}^{(r_0-1)} \setminus D_{q_1}^{(r_0-2)}$.

The same argument as above shows that there exist $N_p^{(r_0)}$ processes $q_1, \ldots, q_{N_p^{(r_0)}}$, each distinct from p, such that for all indices $l \in [\![1; N_p^{(r_0)}]\!]$ we have

$$q_0 \in D_{q_l}^{(r_0-l)} \setminus D_{q_l}^{(r_0-l-1)} \quad \wedge \quad NewHO_p^{(\rho)} \not\subseteq D_{q_l}^{(r_0-l)},$$

which implies that for all indices $l \in [\![1; N_p^{(r_0)}]\!]$, we have $NewHO_p^{(\rho)} \not\subseteq D_{q_l}^{(r_0-N_p^{(r_0)})}$.

However, since $\mathcal{P}_{sym}$ is satisfied, the definition of $N_p^{(r_0)}$ ensures that there exist at most $N_p^{(r_0)} - 1$ processes q' distinct from p such that $NewHO_p^{(\rho)} \not\subseteq D_{q'}^{(r_0-N_p^{(r_0)})}$, a contradiction. Hence $r_q > r_0$.

By a similar argument, we show that for any round r' of ρ such that $r' > r_0$, we have $r_q \geq r'$, a contradiction since ρ consists of a finite number of rounds. We thus conclude that either $NewHO_p^{(\rho)} \subseteq NewHO_q^{(\rho)}$ or $NewHO_q^{(\rho)} \subseteq NewHO_p^{(\rho)}$.

□

As a last step in our argumentation, we demonstrate that $Psym$ is at least as strong as $\mathcal{P}_{\mathcal{RD}}$:

Theorem 3.8. *Algorithm $\mathcal{SRD}$ is a translation of $\mathcal{P}_{sym}$ into $\mathcal{P}_{\mathcal{RD}}$.*

Proof: Let e be any execution of the HO machine $(\mathcal{SRD}, \mathcal{P}_{sym})$.

We first argue condition E2, which requires that the collection $\left(NewHO_p^{(\rho)}\right)_{p \in \Pi, \rho > 0}$ satisfies $\mathcal{P}_{\mathcal{RD}}$. Proposition 3.5 ensures that every process p belongs to $NewHO_p$ at the end of each macro-round ρ of e, while the combination of Lemmas 3.6 and 3.7 implies that for any two distinct processes p and q, and for any macro-round ρ, the sets $NewHO_p^{(\rho)}$ and $NewHO_q^{(\rho)}$ are ordered by inclusion. Therefore, condition E2 is satisfied.

Condition E1 directly follows from the code of $\mathcal{SRD}$ (lines 15, 20 and 26) and from the fact that under $\mathcal{P}_{sym}$, each process p belongs to $HO(p, r)$ at each round r. □

Combining Theorems 3.2 and 3.8, we derive our main result:

Theorem 3.9. *Predicates $\mathcal{P}_{sym}$ and $\mathcal{P}_{\mathcal{RD}}$ are equivalent.*

3.2 A 2-rounds translation of $\mathcal{P}_{sym}$ into $\mathcal{P}_{Gaf}$

We have shown in the above section that (i) $\mathcal{P}_{\mathcal{RD}}$ implies $\mathcal{P}_{Gaf}$, and (ii) algorithm $\mathcal{SRD}$ is a (n^2+n)-rounds translation of $\mathcal{P}_{sym}$ into $\mathcal{P}_{\mathcal{RD}}$. It trivially follows that $\mathcal{SRD}$ is a (n^2+n)-rounds translation of $\mathcal{P}_{sym}$ into $\mathcal{P}_{Gaf}$.

Algorithm 4 Algorithme $\mathcal{STN}$): code of process p

1: **Initialisation:**
2: $NewHO_p \in 2^{\Pi}$, initially empty
3: id_p is the identifier of process p

4: **Round $r = 2\rho - 1$**
5: S_p^r :
6: Send $\langle\, id_p \,\rangle$ to all

7: T_p^r :
8: $NewHO_p := \emptyset$

9: **Round $r = 2\rho$**
10: S_p^r :
11: Send $\langle\, HO(p, r-1) \,\rangle$ to all

12: T_p^r :
13: $NewHO_p := \bigcup_{q \in HO(p,r)} HO(q, r-1)$

We present here the algorithm $\mathcal{STN}$, given as Algorithm 4, which we prove to be a 2-rounds translation of $\mathcal{P}_{sym}$ into $\mathcal{P}_{Gaf}$. For that, we use a purely combinatorial result stated by the following lemma:

Lemma 3.10. *Let n be an integer such that $n \geq 2$, and let $A = (a_{i,j})_{i,j \in [\![1;n]\!]}$ and $B = (b_{i,j})_{i,j \in [\![1;n]\!]}$ be two matrices in $\mathcal{M}_{n\times n}(\{0,1\})$.*

If A and B verify

$* \ \forall i, j \in [\![1;n]\!], \quad a_{i,j} + a_{j,i} > 0$

$* \ \forall i, j \in [\![1;n]\!], \quad b_{i,j} + b_{j,i} > 0$

and if $M = (m_{i,j})_{i,j \in [\![1;n]\!]}$ is defined by $M = A \times B$, then M verifies the following condition:

$$\exists i \in [\![1;n]\!], \quad \forall j \in [\![1;n]\!] \qquad m_{i,j} > 0.$$

Proof: We procced by induction on the size n ($n \geq 2$) of matrices A and B.

• If $n = 2$. Let $A = (a_{i,j})_{i,j \in \{1,2\}}$ and $B = (b_{i,j})_{i,j \in \{1,2\}}$ in $\mathcal{M}_{2\times 2}(\{0,1\})$.
If A is such that

$$\forall i, j \in \{1,2\}, \ a_{i,j} + a_{j,i} > 0,$$

then there exists $i_0 \in \{1,2\}$ such that $a_{i_0,1} > 0$ and $a_{i_0,2} > 0$. Moreover, if B is such that

$$\forall i, j \in \{1,2\}^2, \ b_{i,j} + b_{j,i} > 0$$

then we have $b_{1,1} > 0$ and $b_{2,2} > 0$. It follows that if M is defined by $M = A \times B$ then we have $m_{i_0,1} > 0$ and $m_{i_0,2} > 0$.

• Now assume that the result holds for matrices of size $n-1$.
Let $A = (a_{i,j})_{i,j \in [\![1;n]\!]}$ and $B = (b_{i,j})_{i,j \in [\![1;n]\!]}$ be two matrices in $\mathcal{M}_{n\times n}(\{0,1\})$ that satisfy

- $\forall i, j \in [\![1;n]\!], \quad a_{i,j} + a_{j,i} > 0$

- $\forall i, j \in [\![1;n]\!], \quad b_{i,j} + b_{j,i} > 0$

Let $M = (m_{i,j})_{i,j\in[\![1;n]\!]}$ be the matrix defined by $M = A \times B$. By contradiction assume that M satisfies

$$\forall i \in [\![1;n]\!],\ \exists j \in [\![1;n]\!] : \quad m_{i,j} = 0.$$

For all indices $k = 1, \cdots, n$, let A_k, B_k in $\mathcal{M}_{(n-1)\times(n-1)}(\{0,1\})$ and M_k in $\mathcal{M}_{(n-1)\times(n-1)}(\mathbb{N})$ be the matrices defined by

$$A_k = (a_{i,j})_{i,j\neq k},\ B_k = (b_{i,j})_{i,j\neq k},\ M_k = (m_{i,j})_{i,j\neq k}.$$

By the definition of A and B, for all indices $k = 1, \cdots, n$, matrices A_k and B_k satisfy

- $\forall i, j \in (\{1, \cdots, n\} \setminus \{k\}), \quad a_{i,j} + a_{j,i} > 0$

- $\forall i, j \in (\{1, \cdots, n\} \setminus \{k\}), \quad b_{i,j} + b_{j,i} > 0$

Consider A_1 and B_1. The recurrence assumption implies that there exists an index $i_1 \neq 1$ such that, for all indices $j = 2, \cdots, n$, we have

$$\sum_{l=2}^{n} a_{i_1,l} \cdot b_{l,j} > 0.$$

Moreover, for all indices $j = 2, \cdots, n$

$$m_{i_1,j} = \sum_{l=2}^{n} a_{i_1,l} \cdot b_{l,j}.$$

It follows that, for all indices $j = 2, \cdots, n$,

$$m_{i_1,j} \geq \sum_{l=2}^{n} a_{i_1,l} \cdot b_{l,j} \quad > 0$$

Since we have assumed that

$$\forall i \in [\![1;n]\!],\ \exists j \in [\![1;n]\!], \quad m_{i,j} = 0,$$

we thus conclude that $m_{i_1,1} = 0$.

Now consider A_2, B_2 and M_2. By the same argument, we show that there exists an index $i_2 \neq 2$ such that $m_{i_2,2} = 0$ and $\forall j \neq 2$, $m_{i_2,j} > 0$. We have shown that $m_{i_1,1} = 0$, so we deduce that $i_2 \neq i_1$.

Repeating the same argument for A_k, B_k and M_k, $k = 3, \cdots, n$, we show that

- for all $k = 1, \cdots, n$ there exists an index $i_k \in \{1, \cdots, n\}$ such that

$$m_{i_k,k} = 0 \quad \text{and} \quad \forall l \in \{1, \cdots, n\} \setminus \{k\},\ \ m_{i_k,l} > 0$$

- $\forall l, k \in [\![1;n]\!], \quad k \neq l \ \Rightarrow\ i_k \neq i_l$

Hence we can define a mapping $i : [\![1;n]\!] \longrightarrow [\![1;n]\!]$ such that, for all indices $j = 1, \cdots, n$, $i(j) = i_j$. By construction, the mapping i is bijective.

Now, let $k \in \{1, \cdots, n\}$. Since $i : \ j \to i(j)$ is bijective, we have $m_{i(k),i(i^{-1}(k))} = 0$. The fact that $b_{i(i^{-1}(k)),i(i^{-1}(k))} > 0$ then implies $a_{i(k),i(i^{-1}(k))} = 0$.

Moreover, for all $i, j \in [\![1; n]\!]$, $a_{i,j} + a_{j,i} > 0$ and so $a_{i(i^{-1}(k)),i(k)} > 0$.

By the definition of the mapping i, we have $m_{i(i^{-1}(k)),i^{-1}(k)} = 0$, i.e., $a_{i(i^{-1}(k)),i(k)} \cdot b_{i(k),i^{-1}(k)} = 0$.

We then deduce that $b_{i(k),i^{-1}(k)} = 0$ which, by the definition of B, implies that $b_{i^{-1}(k),i(k)} > 0$.

Repeating the same argument, we obtain $a_{i(i(k)),i^{-1}(k)} = 0$ and then $a_{i^{-1}(k),i(i(k))} > 0$.

Therefore, $b_{i(i(k)),i^{-2}(k)} = 0$ which implies that $b_{i^{-2}(k),i(i(k))} > 0$ and then $a_{i^3(k),i^{-2}(k)} = 0$.

Generalizing we obtain

$$\forall l, \ a_{i^{-l}(k),i^{l+1}(k)} > 0 \tag{1}$$

where i^r is the r-th iterated of i.

Since $i : \{1, \cdots, n\} \to \{1, \cdots, n\}$ is bijective, there exists an integer $l_k \geq 1$ such that $i^{l_k}(k) = k$. Setting $l_k = l + 1$ in equation (1), we obtain $a_{i(k),k} > 0$, a contradiction. □

Theorem 3.11. *Algorithm $\mathcal{STN}$ is a 2-rounds translation of $\mathcal{P}_{sym}$ into $\mathcal{P}_{Gaf}$.*

Proof: Let e be an execution of the HO machine and let $\rho > 0$ be any macro-round of e.

The fact that e satisfies condition E1 of the definition of a translation is a straightforward consequence of the way $\mathcal{STN}$ works. Indeed, the code of $\mathcal{STN}$ (line 13) ensures that if some process q belongs to the set $NewHO_p^{(\rho)}$ of some process p, then p actually heard of q, possibly through an intermediate process, during macro-round ρ.

We now argue condition E2. Let $A_\rho = (a_{i,j})_{i,j \in [\![1;n]\!]}$, $B_\rho = (b_{i,j})_{i,j \in [\![1;n]\!]}$ and $M_\rho = (m_{i,j})_{i,j \in [\![1;n]\!]}$ be the matrices defined by:

* $a_{i,j} = 1$ if $p_i \in HO(p_j, 2\rho - 1)$, and 0 otherwise

* $b_{i,j} = 1$ if $p_i \in HO(p_j, 2\rho)$, and 0 otherwise

* $M_\rho = A_\rho \times B_\rho$

By these definitions, it is obvious to see that, for all indices $i, j \in [\![1; n]\!]$, we have $m_{i,j} > 0$ if and only if process p_i belongs to $NewHO_{p_j}^{(\rho)}$. Hence, e satisfies condition E2 if and only if M_ρ verifies the following condition:

$$C \quad : \quad \exists i \in [\![1; n]\!], \ \forall j \in [\![1; n]\!] \quad m_{i,j} > 0.$$

Since $\mathcal{P}_{sym}$ is satisfied in e, matrices A_ρ and B_ρ verify

* $\forall i, j \in [\![1; n]\!], \quad a_{i,j} + a_{j,i} > 0$

* $\forall i, j \in [\![1; n]\!], \quad b_{i,j} + b_{j,i} > 0$

Lemma 3.10 then ensures that $M_\rho = A_\rho \times B_\rho$ verifies condition C, which ends to show that $\mathcal{STN}$ translates $\mathcal{P}_{sym}$ into $\mathcal{P}_{Gaf}$. □

4 Conclusions and future work

In this paper, we adress the expressivity of the HO model. We give a first answer regarding two major types of models for distributed computing in the presence of benign failures. In particular, we present the first formal characterization of classical models only in terms of predicates that capture the properties of their communications. Moreover, we show how it is possible to compare and hierarchize such predicates.

In [4], Charron-Bost *et al.* generalized the HO model to cope with *value failures*. This extended model covers both the *Byzantine failures* [13] and the dynamic transmission faults of [14]. The HO model thus appears to be suitable for systems with any type of failures.

Future works may try to apply the techniques presented in this paper to rigorously determine the predicates corresponding to other existing models, either shared-memory or message passing, with benign failures or value failures. This would provide a formal unified framework for the analysis of fault-tolerant distributed systems and may give new insights about questions such : (i) what communication properties are really crucial, or (ii) what problems are solvable in what systems.

References

[1] Yehuda Afek, Hagit Attiya, Danny Dolev, Eli Gafni, Michael Merritt, and Nir Shavit. Atomic snapshots of shared memory. *J. ACM*, 40(4):873–890, 1993.

[2] Hagit Attiya and Ophir Rachman. Atomic snapshots in o (n log n) operations. *SIAM J. Comput.*, 27(2):319–340, 1998.

[3] Hagit Attiya and Jennifer Welch. *Distributed Computing*. Wiley, 2004.

[4] Martin Biely, Josef Widder, Bernadette Charron-Bost, Antoine Gaillard, Martin Hutle, and André Schiper. Tolerating corrupted communication. In *PODC '07: Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, pages 244–253, New York, NY, USA, 2007. ACM.

[5] Elizabeth Borowsky and Eli Gafni. A simple algorithmically reasoned characterization of wait-free computations (extended abstract). In *PODC*, pages 189–198, 1997.

[6] Bernadette Charron-Bost and André Schiper. The Heard-Of Model: Computing in Distributed Systems with Benign Failures. Technical report, 2007. Replaces TR-2006: The Heard-Of Model: Unifying all Benign Failures.

[7] Tzilla Elrad and Nissim Francez. Decomposition of distributed programs into communication-closed layers. *Sci. Comput. Program.*, 2(3):155–173, 1982.

[8] Faith Fich. How hard is it to take a snapshot? In M. Bielikova et al., editor, *Proceedings of 31st Annual Conference on Current Trends in Theory and Practice of Informatics (SOFSEM)*, volume 3381 of *Lecture Notes on Computer Science*, pages 28–37, 2005.

[9] E. Gafni. Rounds-by-rounds fault detectors: unifying synchrony and asynchrony. In *Proceedings of the Seventeenth ACM Symposium on Principles of Distributed Computing*, pages 143–152, August 1998.

[10] M. P. Herlihy. Wait-free synchronization. *ACM Transactions on Programming Languages and Systems*, 13(1):123–149, January 1991.

[11] Clyde P. Kruskal, Larry Rudolph, and Marc Snir. Efficient synchronization of multiprocessors with shared memory. *ACM Trans. Program. Lang. Syst.*, 10(4):579–601, 1988.

[12] Leslie Lamport. On interprocess communication. part ii: Algorithms. *Distributed Computing*, 1(2):86–101, 1986.

[13] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980.

[14] N. Santoro and P. Widmayer. Time is not a healer. In *Proceedings of the 6th Symposium on Theor. Aspects of Computer Science*, pages 304–313, Paderborn, Germany, 1989.