La puissance de désaccord d'un adversaire

Carole Delporte-Gallet, Hugues Fauconnier, Rachid Guerraoui, Andreas Tielmann

La puissance de désaccord d'un adversaire

Carole Delporte-Gallet[1] [†] and Hugues Fauconnier[1] and Rachid Guerraoui[2] and Andreas Tielmann[1]

[1] *LIAFA, Université Paris Diderot Paris, France*
[2] *School of Computer and Communication Sciences, EPFL Lausanne, Switzerland*

Un des résultats fondamentaux de l'algorithmique distribuée est que le niveau d'accord qui peut être obtenu en présence de t pannes est exactement $t+1$. Autrement dit un adversaire qui peut mettre en panne n'importe quel sous ensemble d'au plus t processus peut empêcher les autres processus de se mettre d'accord sur t valeurs. Mais quelle est la puissance des $(2^{2^n}-n)$ autres adversaires qui ne peuvent mettre en panne que certaines combinaisons des processus ?

Cet article présente une caractérisation précise d'un adversaire. On y introduit la notion de "puissance de désaccord". La puissance de désaccord d'un adversaire est le plus grand entier k tel que l'accord des processus sur k valeurs soit possible avec cet adversaire. Puis on montre comment *calculer* automatiquement cet entier pour un adversaire donné.

Keywords: Mémoire partagée, tolérance aux pannes

1 Introduction

The theory of distributed computing is largely related to determining what can be computed against a specific adversary. Most results so far have been devoted to *one* specific form of adversaries : those that can control any subset of size t of the processes, i.e., the *t-failures* adversary. In particular, a seminal result in distributed computing says that the level of agreement that can be obtained in a shared memory model where t processes can crash is exactly $t+1$ [2, 11, 13]. In other words, an adversary that can crash any subset of size at most t can prevent the processes from agreeing on t values. In the case of consensus for instance ($t=1$), this translates into FLP [8].

In a sense, these results are very incomplete. Indeed, the t-failures assumption covers only the n "uniform" adversaries in a system of size n. What about the other $(2^{2^n}-(n+1))$ adversaries that can crash certain subsets of processes of a certain size but not others of the same size ? In particular, given any adversary $\mathscr{A}$, for what k does $\mathscr{A}$ prevent k-set agreement [6] ? Beyond intellectual curiosity, the study of adversaries that are "non-uniform" might even be practically motivated by modern multicore architectures where the failures of processes in the same core might all be correlated [1, 9, 12].

This paper characterizes the power of an adversary $\mathscr{A}$, denoted $dis(\mathscr{A})$, by the biggest k for which k-set agreement cannot be solved with $\mathscr{A}$, which we call here the *disagreement power* of $\mathscr{A}$. We show how to automatically compute the disagreement power of an adversary and group adversaries into n equivalence classes in a system of size n. Adversaries within the same class solve the same set of (colorless [‡]) tasks.

As established in [2, 11, 13], it is possible to implement $(k+1)$-set agreement in $\mathscr{B}_k$(the classical k-failure adversary is the adversary for which at most k processes may crash) but it is impossible to implement k-set agreement in $\mathscr{B}_k$. Hence, the disagreement power of $\mathscr{B}_k$is k.

Determining the disagreement power of certain adversaries is trivial. For others, it is not. Consider, in a system of 3 processes, $\{1,2,3\}$, an adversary $\mathscr{A}$ that can fail either no process, both processes 2 and 3, or process 1 i.e. $\mathscr{A}=\{\emptyset, 23^{\S}, 1\}$. It is easy to show that $\mathscr{A}$ can prevent consensus but not 2-set agreement. In this sense, adversary $\mathscr{A}$ has the same disagreement power as the 1-failure adversary, namely, $\mathscr{B}_1$.

[†] Les résultats présentés dans cet article sont issus de [7]. Carole Delporte-Gallet, Hugues Fauconnier et Andreas Tiellmann sont financés par ANR VERSO-SHAMAN et le projet INRIA GANG.

‡. Intuitively, in a colorless task, any process can adopt any other process' input or output value.

§. When appropriate, we will use e.g. 23 as shorthand for the set $\{2,3\}$.

Consider now a more involved scenario : a system of 4 processes and an adversary $\mathscr{A}'$ that can fail any element of $\{\emptyset, 4, 23, 14, 12, 134, 124, 123\}$. What is the disagreement power of $\mathscr{A}'$? We prove in this paper that it is also 1.

2 Domination property

We give a general characterization of adversaries that enables to automatically compute their disagreement power. Namely, we introduce a *structural predicate* on adversaries, parameterized by an integer k, and which, intuitively, checks that for any set of faulty processes of size less or equal k, there is some adequate *matching* set of the adversary. More formally :

Definition 1 *Let $\mathscr{A}$ and $\mathscr{B}$ be any two adversaries. We say that $a \in \mathscr{A}$* dominates *$b \in \mathscr{B}$, if*

$$(a \supseteq b) \quad \textit{and}$$
$$(\forall b' \in \mathscr{B}, b' \supsetneq b, \exists a' \in \mathscr{A}, a' \supseteq a : D(a', \mathscr{A}, b', \mathscr{B})).$$

Definition 2 *We say that an adversary $\mathscr{A}$* dominates *an adversary $\mathscr{B}$ (denoted $D(\mathscr{A}, \mathscr{B})$) if and only if the following property is satisfied :*

$$\forall b \in \mathscr{B}, \exists a \in \mathscr{A} : D(a, \mathscr{A}, b, \mathscr{B}).$$

We give an example in which the domination property holds for two adversaries. Assume $n = 4$ and consider the following adversaries (we use $ij\ldots$ as a shorthand for the set $\{p_i, p_j, \ldots\}$) :

$$\begin{aligned} \mathscr{A} &= \{\emptyset, 12, 34, 123, 124, 134, 234\} \\ \mathscr{B}_2 &= \{\emptyset, 1, 2, 3, 4, 12, 13, 14, 23, 24, 34\} \end{aligned}$$

In this example, $D(\mathscr{A}, \mathscr{B}_2)$, i.e. for every $b \in \mathscr{B}_2$ there exists an $a \in \mathscr{A}$ such that $D(a,b)$ (e.g. $D(\emptyset, \emptyset)$, $D(12,2)$ and $D(123,13)$).

This property is intricate. One may think that if for all $b_0 \subsetneq b_1 \ldots \subsetneq b_x$ in $\mathscr{B}$ there exist $a_0 \subseteq a_1 \ldots \subseteq a_x$ in $\mathscr{A}$ such that $b_i \subseteq a_i$ for all i then $D(\mathscr{A}, \mathscr{B})$. But this is not the case :

Consider now a slightly different example with $n = 3$:

$$\begin{aligned} \mathscr{A} &= \{\emptyset, 2, 12, 13, 23\} \\ \mathscr{B}_2 &= \{\emptyset, 1, 2, 3, 12, 13, 23\} \end{aligned}$$

In this example, for all $b_0 \subsetneq b_1 \subsetneq b_2$ there exist $a_0 \subseteq a_1 \subseteq a_2$ and $b_i \subseteq a_i$. But $\neg D(\mathscr{A}, \mathscr{B}_2)$, because for all $a \in \mathscr{A}$, $\neg D(a, 3)$.

3 Results

The existence of such a matching set means that the structure of the adversary is in some sense similar to the structure of the uniform t-failures adversary. We prove that any adversary that satisfies the predicate $D(\mathscr{A}, \mathscr{B}_k)$ and not $D(\mathscr{A}, \mathscr{B}_{k+1})$ has disagreement power k.

We show (sufficient condition) that if k-set agreement can be solved with some adversary that satisfies the predicate $D(\mathscr{A}, \mathscr{B}_k)$ for some k, then k-set agreement can be solved with the k-failures adversary which, in turn, is known to be impossible [2, 11, 13]. Hence, an adversary that satisfies the predicate $D(\mathscr{A}, \mathscr{B}_k)$ has disagreement power at least k. For this, we use a new simulation between adversaries, which we call the *conservative back-off simulation*, and which we believe is interesting in its own right. The idea underlying our simulation is the following : a process backs-off and skips its simulation step if the process thinks that it is faulty in some set where the simulated algorithm is known to work.

Conversely (necessary condition), we show how to solve k-set agreement with any adversary $\mathscr{A}$ that does not satisfy the predicate $D(\mathscr{A}, \mathscr{B}_k)$ for some k. We do this by showing how to implement failure detector k-anti-Ω [14], known to implement k-set agreement. (Each query to k-anti-Ω returns $n-k$ process ids ; the specification ensures that there is a correct process whose id is eventually never output.)

We then use our characterization to split the set of all adversaries into n disjoint *equivalence* classes, one for every level of disagreement : we show that for any two adversaries with the same disagreement power, exactly the same set of (colorless) tasks can be solved. Intuitively, in a colorless task [3, 10] any process can adopt any input or output value of any other process without violating the task specification. The key to our proof of the equivalence is that, for every adversary with disagreement power k, it is possible to simulate a wait-free system of $k+1$ processes. This can simulate every other k-failure adversary [3, 4]. Technically, this is achieved by implementing $(k+1)$-anti-Ω for the adversary and translating it into a vector of $k+1$ Ω failure detectors [5] of which at least one is a "real" Ω (i.e. it outputs eventually everywhere the same correct process). Then, each of the $k+1$ simulated processes can be associated with one of the Ω's and a consensus-object can be built to agree on the simulated steps of such a process.

Since we can compute automatically the disagreement power of an adversary (using our structural predicate), we can thus automatically derive results for an "exotic" adversary using known results about a more orthodox ("uniform") adversary with the same disagreement power.

Indirectly, our partitioning contributes to the idea that a very small subset of results and ad-hoc proofs in distributed computing should suffice to derive all others. In particular, if indeed needed to reason about set agreement for the "wait-free" adversary ($n-1$-set agreement), topology is not needed for all the other ones. Results concerning other k-failures ("uniform") adversaries can be deduced by [3, 4], whereas results for all other ("non-uniform") $(2^{2^n}-(n+1))$ adversaries can be deduced from our characterization.

4 Concluding Remarks

This paper presents a novel way to precisely characterize *adversaries* : the notion of *disagreement power*, i.e., the biggest integer k for which an adversary can prevent processes from agreeing on k values. This notion partitions the set of all adversaries into n distinct *equivalence* classes, one for every disagreement power. Any two adversaries with the same disagreement power solve exactly the same set of (colorless) tasks. We believe that our result could be extended to colored tasks but this is subject to future work.

At the heart of our partitioning lies our simulation between adversaries. Interestingly, the simulation works also if we assume the existence of stronger objects than registers or even non-deterministic object types. Furthermore, the simulation (as well as our implementation of k-set agreement with a given adversary) remains correct even if the adversary is known only eventually, i.e., not necessarily from the beginning.

Références

[1] B. V. Ashwinkumar, A. Patra, A. Choudhary, K. Srinathan, and C. P. Rangan. On tradeoff between network connectivity, phase complexity and communication complexity of reliable communication tolerating mixed adversary. In *PODC*, pages 115–124, 2008.

[2] E. Borowsky and E. Gafni. Generalized FLP impossibility result for t-resilient asynchronous computations. In *STOC*, pages 91–100, 1993.

[3] E. Borowsky, E. Gafni, N. A. Lynch, and S. Rajsbaum. The BG distributed simulation algorithm. *Distributed Computing*, 14(3) :127–146, 2001.

[4] T. D. Chandra, V. Hadzilacos, P. Jayanti, and S. Toueg. Generalized irreducibility of consensus and the equivalence of t-resilient and wait-free implementations of consensus. *SIAM J. Comput.*, 34(2) :333–357, 2004.

[5] T. D. Chandra, V. Hadzilacos, and S. Toueg. The weakest failure detector for solving consensus. *J. ACM*, 43(4) :685–722, 1996.

[6] S. Chaudhuri. Agreement is harder than consensus : set consensus problems in totally asynchronous systems. In *PODC*, pages 311–324, 1990.

[7] C. Delporte-Gallet, H. Fauconnier, R. Guerraoui, and A. Tielmann. The disagreement power of an adversary. In I. Keidar, editor, *DISC*, volume 5805 of *Lecture Notes in Computer Science*, pages 8–21. Springer, 2009.

[8] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2) :374–382, 1985.

[9] M. Fitzi and U. M. Maurer. Efficient byzantine agreement secure against general adversaries. In *DISC*, pages 134–148, 1998.

[10] M. Herlihy and S. Rajsbaum. The decidability of distributed decision tasks (extended abstract). In *STOC*, pages 589–598, 1997.

[11] M. Herlihy and N. Shavit. The topological structure of asynchronous computability. *J. ACM*, 46(6) :858–923, 1999.

[12] F. P. Junqueira and K. Marzullo. Designing algorithms for dependent process failures. In *Future Directions in Distributed Computing*, pages 24–28, 2003.

[13] M. E. Saks and F. Zaharoglou. Wait-free k-set agreement is impossible : The topology of public knowledge. *SIAM J. Comput.*, 29(5) :1449–1483, 2000.

[14] P. Zielinski. Anti-Omega : the weakest failure detector for set agreement. In *PODC*, pages 55–64, 2008.