TRIVIALITY OF ASYNCHRONOUS CONSENSUS ALLOWING ONE FAULTY PROCESS

KYLE MILLER

ABSTRACT. This is the argument of [FLP85] but reduced. We allow processes to perform arbitrary computations over their histories, an unlimited amount of information in each message, and for the arbiter to be a computable function over the entire run of an ongoing asynchronous protocol.

The space of runs modulo serialization irrelevance, given the topology induced by computable functions, is connected, and therefore any such arbiter function must be constant. It follows that asynchronous consensus protocols allowing at least one faulty process are constant.

1. INTRODUCTION

Instead of modeling a collection of state machines with unbounded memory along with a message delivery system, we simplify the system to an omniscient *arbiter* observing with perfect accuracy the history of messages between processes in an asynchronous system. With the assumption that an arbiter must be able to decide the outcome of the asynchronous system in finite time, if some unknown process is allowed to fail without notice, even when the arbiter is given the ability to perfectly model the processes, the arbiter must stubbornly be the constant function.

2. ARBITERS

Let M be a fixed set of *messages*, $[n] = \{0, 1, 2, \ldots, n-1\}$ a set of *process names* with $n \geqslant 2$, $\pi : M \to [n]$ a *recipient projector*, and $M_0 \subset M$ a finite set of *initial messages*. Let $\varepsilon : \mathscr{P}(M) \to \mathscr{P}(M)$ be the *message processor*, which takes a history of received messages for a particular process and outputs a nonempty finite set of messages. For $i \in \mathbb{N}$, the function $h_i : M^{\mathbb{N}} \to \mathscr{P}(M)$ defined by $h_i(r) = \{r_j : j \leqslant i \text{ and } \pi(r_j) = \pi(r_i)\}$ gives the history of so-far-received messages for the recipient of message i.

A *run* $r \in M^{\mathbb{N}}$ allowing a *faulty* process $\varphi \in [n]$ is a sequence (of *delivered messages*) with the following properties:

Causality: For all $i \in \mathbb{N}$, $\sum_{j=0}^{i}[r_i = r_j] \leqslant [r_i \in M_0] + \sum_{j=0}^{i-1}[r \in \varepsilon(h_j(r))]$.[1]

Semireliability: For all $i \in \mathbb{N}$, if $m \in M_0 \cup \bigcup_{j=0}^{i-1} \varepsilon(h_j(r))$ and $\pi(m) \neq \varphi$, then there is some $k \geqslant i$ such that $r_k = m$.

For $\varphi \in [n]$, let $R(\varphi)$ be the set of runs allowing a faulty φ, and let R be the set of all runs. If s is a finite sequence of messages, let $E(s) \subset R$ be the subset of runs with s as a prefix. We say s *extends to a run* if $E(s)$ is nonempty.

Lemma 1. *If s is a finite sequence of messages satisfying causality over its domain, then $E(s) \cap R(\varphi)$ is nonempty for all $\varphi \in [n]$.*

Lemma 2. *If r is a run and i is an index such that $\pi(r_i) \neq \pi(r_{i+1})$, then the sequence r' obtained from r by swapping entries i and $i+1$ is also a run.*

An *arbiter* is a function $c : R \to \{0, 1\}$ with the following properties:

Serialization irrelevance: If $r, r' \in R$ are runs that are equal at all indices except at i and $i+1$, where $r_i = r'_{i+1}$ and $r_{i+1} = r'_i$, and where $\pi(r_i) \neq \pi(r_{i+1})$, then $c(r) = c(r')$.

Date: Oct 22, 2017.

[1] This uses the Iverson bracket: $[\text{true}] = 1$ and $[\text{false}] = 0$.

Continuity/computability: With R as a subspace of the product topology $M^{\mathbb{N}}$, where M is given the discrete topology, c is a continuous function. In other words, for every run r, there is some prefix ρ of r such that $c(E(\rho)) = \{c(r)\}$.

Lemma 3. *Let c be an arbiter, $m \in M$, and s be a finite sequence of messages such that $c(E(s)) = \{0,1\}$ and sm extends to a run. Then there is a finite sequence s' extending s such that $c(E(s'm)) = \{0,1\}$.*

Proof. Let Σ be the nonempty set of all finite sequences of messages σ such that $s\sigma m$ extends to a run. If any $\sigma \in \Sigma$ has $c(E(s\sigma m)) = \{0,1\}$, then $s' = s\sigma$ satisfies the conclusion. Hence, suppose to the contrary that $|c(E(s\sigma m))| = 1$ for all $\sigma \in \Sigma$.

Let τ_0, τ_1 be finite sequences of messages such that $c(E(s\tau_i)) = \{i\}$. If m is in τ_i, then the truncation of τ_i to a prefix ending with m has $c(E(s\tau_i')) = \{i\}$, too, and if m is not in τ_i, then $c(E(s\tau_i m)) = \{i\}$. Hence, for each $i \in \{0,1\}$, there is some $\sigma_i \in \Sigma$ such that $c(E(s\sigma_i m)) = \{i\}$. Suppose for all $\sigma \in \Sigma$ and all messages $e \neq m$ such that $s\sigma em$ extends to a run that $c(E(s\sigma m)) = c(E(s\sigma em))$. Then since each prefix of σ is in Σ, by induction

$$c(E(s\sigma_1 m)) = c(E(sm)) = c(E(s\sigma_2 m)),$$

a contradiction.

Let $\sigma \in \Sigma$ and $e \neq m$ be a message such that $s\sigma$ and $s\sigma e$ both extend to runs and that $c(E(s\sigma m)) \neq c(E(s\sigma em))$. If it were the case that $\pi(e) \neq \pi(m)$, then we would have

$$c(E(s\sigma em)) = c(E(s\sigma me)) = c(E(s\sigma m)),$$

hence assume $\pi(e) = \pi(m)$. Let $r \in E(s\sigma)$ be a run where none of the messages after $s\sigma$ are for $\pi(m)$, and by continuity of c let ρ be a sequence such that $s\sigma\rho$ is a prefix of r and $|c(E(s\sigma\rho))| = 1$. Then

$$c(E(s\sigma em)) = c(E(s\sigma em\rho)) = c(E(s\sigma\rho em)) = c(E(s\sigma\rho)) = c(E(s\sigma\rho m)) = c(E(s\sigma m\rho)) = c(E(s\sigma m)),$$

a contradiction. □

Theorem 4. *Arbiters are constant functions.*

Proof. Let s_0 be an empty sequence of messages and let $A_0 = \emptyset$. By recursive definition, assume s_i is a finite sequence of messages such that $c(E(s_i)) = \{0,1\}$ and assume A_i is some finite set of messages that can extend s_i to a run. If A_i is empty, let $s_{i+1} = s_i$ and let A_{i+1} be the set of messages a such that $s_i a$ extends to a run. If A_i is not empty, then choose $m \in A_i$, let $s_{i+1} = s_i \sigma m$ be a sequence from applying Lemma 3 to s_i and m, and let $A_{i+1} = A_i - \{m\} - \sigma$.

The limit r of $(s_i)_{i \in \mathbb{N}}$ is a causal sequence of messages that by construction has semireliability, hence it is a run. This is a contradiction because $c(r)$ depends only on some prefix of r, but $c(E(s_i)) = \{0,1\}$ for all i. □

3. Consensus protocols

Let $\iota : [n] \times \{0,1\} \to M$ be an *initialization function* satisfying $\pi(\iota(k,i)) = k$ for all $k \in [n]$ and $i \in \{0,1\}$. A *consensus protocol* is a family of arbiters c_x indexed by $x \in \{0,1\}^{[n]}$, which are identical except that the initial message set for c_x is $\{\iota(k, x_k) : k \in [n]\}$. Let R_x denote the set of runs for c_x.

Lemma 5. *Let $x \in \{0,1\}^{[n]}$, and let $r \in R_x$ be such that there is some $\varphi \in [n]$ such that $\pi(r_i) \neq \varphi$ for all $i \geqslant 0$. If $x' \in \{0,1\}^{[n]}$ is equal to x except at index k, then $r \in R_{x'}$ and $c_x(r) = c_{x'}(r)$.*

Lemma 6. *There is some constant $a \in \{0,1\}$ such that $c_x(r) = a$ for all $x \in \{0,1\}^{[n]}$ and $r \in R_x$.*

Proof. Consider $x, x' \in \{0,1\}^{[n]}$ that are equal except at index i. Let $r \in R_x$ be a run such that process i receives no messages, hence $c_x(R_x) = c_{x'}(R_{x'})$. Since every pair of initialization vectors differs by a finite sequence of single-entry changes, $c(R_x)$ is the same one-element set for all $x \in \{0,1\}^{[n]}$. □

4. Extensions

If we do not require runs to be semireliable, R becomes a closed subspace so Theorem 4 is easier to prove, hence arbiters must still be constant functions in this case.

Requiring runs to be reliable simplifies to the case of a single process, and arbitration can be non-trivial. For example, if $M = M_0 = \{m_1, m_2\}$ and ε sends everything to $\{m_1, m_2\}$, then $s(r) = [r_0 = m_1]$ is a surjective arbiter.

5. Conclusion

The asynchronous consensus problem reduces to the case of an arbiter observing a run of messages. For a non-trivial arbiter to exist, there must be some condition on acceptible runs so that the space of such runs forms a disconnected topological space.

References

[FLP85] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. volume 32 of JACM, pages 374–382, 1985.