On the Asynchronous Computability Theorem

Rachid Guerraoui Petr Kouznetsov Bastian Pochon

Distributed Programming Laboratory

EPFL

Characterization $T1$ of wait-free computable decision tasks [Herlihy and Shavit, 1993, 1994]

- representing decision tasks through simplicial complexes and simplicial maps
- a task is solvable iff the corresponding complexes satisfy a topological property $T1$
- long-standing impossibility results: set agreement [BG93, SZ93] and $(n, 2n)$-renaming

[Borowsky and Gafni, 1997]: a "stronger" geometric property $T2$

- a "simpler" algorithmic proof of $T2$
- the equivalence of the properties: $T1 = T2$ (through proving a geometric result with a distributed algorithm)

Why "stronger" and "simpler" ?
What is the distributed algorithm?

Decision task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ [HS93]

- $\mathcal{I}$ — chromatic *input complex* (a set of possible input simplexes)
- $\mathcal{O}$ — chromatic *output complex* (a set of possible output simplexes)
- $\Delta \subseteq \mathcal{I} \times \mathcal{O}$ — *task specification* (a color-preserving map that carries every input simplex to a set of output simplexes)

Wait-free protocols

Every process starts with an *input* value, executes a number of writes and reads of the shared memory, and finishes with an *output* value (applies a *decision map* to its final state).

A protocol solves a task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ if it satisfies the task specification: for any input simplex $S \in \mathcal{I}$, any resulting output simplex $O \in \Delta(S)$.

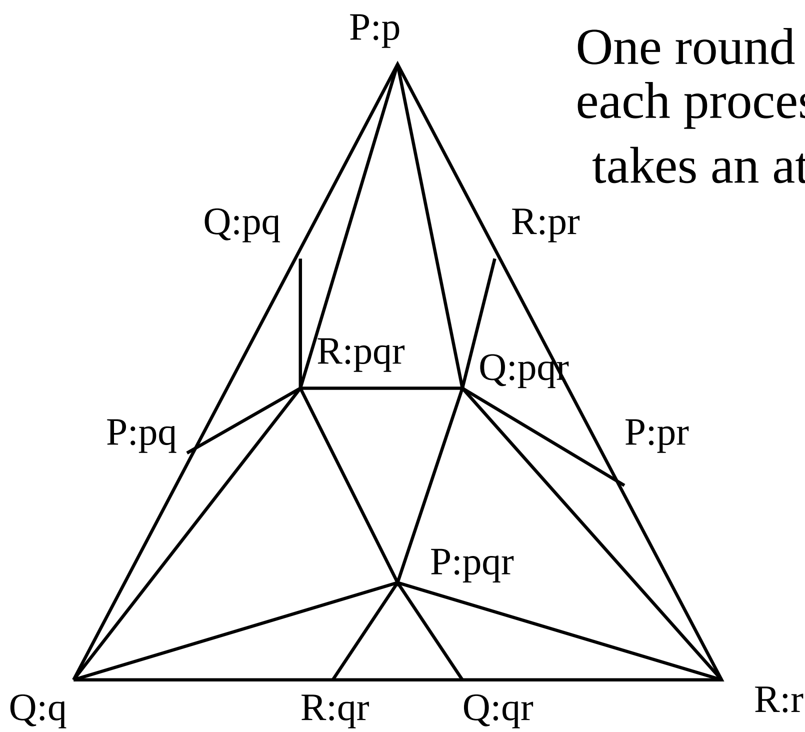
A protocol *wait-free* if every process finishes in a bounded number of its own steps, no matter how other processes behave.

Herlihy and Shavit's criterion [HS94]

Theorem 1. *A decision task* $(\mathcal{I}, \mathcal{O}, \Delta)$ *is wait-free solvable using read-write memory if and only if (*$T1$*) there exists a chromatic subdivision* σ *of* $\mathcal{I}$ *and a color-preserving simplicial map* $\mu : \sigma(\mathcal{I}) \rightarrow \mathcal{O}$ *such that for each simplex* $S \in \sigma(\mathcal{I})$*,* $\mu(S) \in \Delta(carrier(S, \mathcal{I}))$*.*

Immediate snapshot [BG93]

Standard chromatic subdivision (SDS) $\chi(S)$:



One round of IS execution:
each process writes and immediately
takes an atomic snapshot

Iterated immediate snapshot model

Processes proceed in rounds.

Every process consequently takes immediate snapshots of $M_0, M_1, \ldots$.

The resulting K-round protocol complex corresponds to the recursive SDS $\chi^K(S)$.

[BG93]: the (iterated) IS snapshot model can be implemented in the read write memory model.

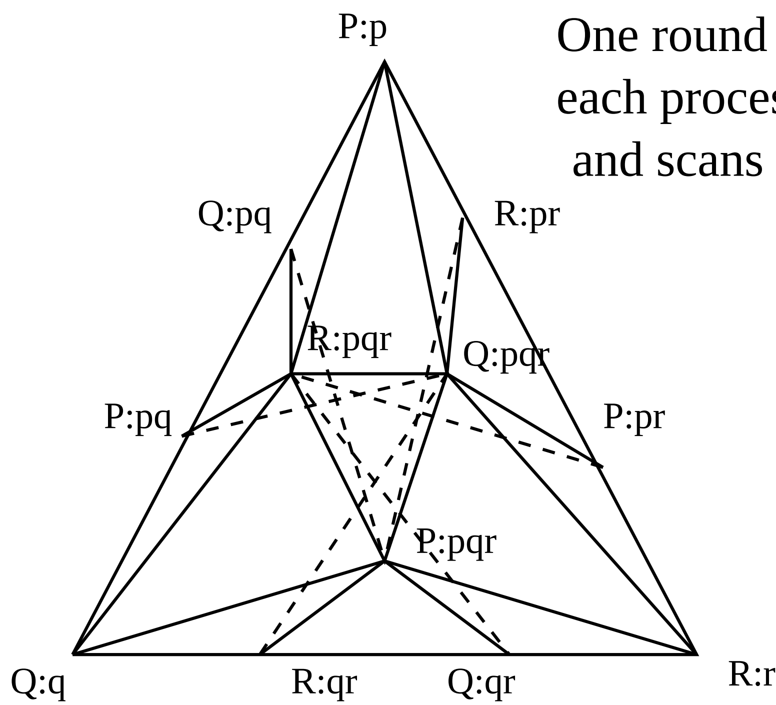
[HS94]: necessity part

Assume a task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ has a wait-free solution using read-write memory.

The aim is to find σ and μ such that $(T1)$ for each simplex $S \in \sigma(I)$, $\mu(S) \in \Delta(carrier(S, \mathcal{I}))$.

Let $\mathcal{P}(\mathcal{I})$ be the corresponding *protocol complex* and δ be the decision map.

[HS94]: *protocol complex* $\mathcal{P}(\mathcal{I})$



One round protocol complex:
each process writes
and scans the memory

$\mathcal{P}(\mathcal{I})$ is ***not*** a chromatic subdivision of an input complex.

[HS94]: locating a *span*

But every protocol complex $\mathcal{P}(\mathcal{I})$ has a span: a chromatic subdivision $\sigma(\mathcal{I})$ and a color and carrier preserving map ϕ from $\sigma(\mathcal{I})$ to $\mathcal{P}(\mathcal{I})$.

σ and a composition of ϕ and δ gives the result.

[HS94]: sufficiency part

Assume now that, for a given task $T = (\mathcal{I}, \mathcal{O}, \Delta)$, there is a subdivision $\sigma(\mathcal{I})$ and a map μ satisfying $T1$.

The aim is to find a protocol that solves T.

Chromatic simplex agreement task, $CSA(\sigma)$

The task $CSA(\sigma)$ has an simplex $S^n \in \mathcal{I}$ as an input complex and $\sigma(S^n)$ as an output complex. Every process starts with a vertex of S^n of its color and finishes with a vertex of $\sigma(S^n)$ of its color, so that all decided vertexes constitute a simplex of $\sigma(S^n)$.

Solving $T = (\mathcal{I}, \mathcal{O}, \Delta)$ given σ and μ is equivalent to solving $CSA(\sigma)$!

[HS94]: sufficiency part (contd.)

The chromatic simplex agreement task is solved in the IIS model by using ε-*perturbation* of $\chi^K(S^n)$.

The fact that IIS is implementable in RW [BG93] implies the result.

Borowsky and Gafni's criterion[BG97]

Theorem 2. *A decision task* $(\mathcal{I}, \mathcal{O}, \Delta)$ *is wait-free solvable using read-write memory if and only if (*$T2$*) there exists an* iterated standard chromatic *subdivision* χ^K *of* $\mathcal{I}$ *and a color-preserving simplicial map* $\mu : \chi^K(\mathcal{I}) \rightarrow \mathcal{O}$ *such that for each simplex* $S \in \chi^K(I)$*,* $\mu(S) \in \Delta(carrier(S, \mathcal{I}))$*.*

[BG97]: IIS is equivalent to RW for decision tasks!

Any read-write protocol that employs a bounded number of reads and writes can be simulated in the IIS model.

By König's lemma, any read-write memory protocol that solves a task employs only a bounded number of reads and writes.

[BG97]: computability in the IIS model

A task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ is solvable in the IIS model iff for some sufficiently large K, there exists a color-preserving map μ that carries every simplex S of $\chi^K(\mathcal{I})$ to a simplex of $\Delta(carrier(S, \mathcal{I}))$.

Since IIS is equivalent to RW, we have $T2$!

$T1$ and $T2$ *must* be equivalent!

$T1$ requires *any* subdivision, while $T2$ requires *iterated SDS*.

$\Longrightarrow T2$ is at least as strong as $T1$.

- Any task that satisfies $T2$, satisfies $T1$.
- Any task that is unsolvable by $T1$ is unsolvable by $T2$.

$T1 \Rightarrow T2$?

In the IIS model, we must solve chromatic simplex agreement task over $\sigma(S^n)$, $CSA(\sigma)$.

The corollary to the simplicial approximation theorem (NB: can be derived algorithmically):

Lemma 3. *There exists K and a carrier preserving map ϕ from the iterated standard chromatic subdivision $\chi^K(S^n)$ to $\sigma(S^n)$.*

How to get the colors?

Subdivision preserves connectivity of the original simplex.

$\Rightarrow$ Every $k+1 \leq n+1$ vertexes of S^n imply an image of a subdivided k-simplex.

$\Rightarrow$ $k+1$ processes can solve $NCSA$ over this image.

The convergence algorithm: the general idea

Processes start from solving $NCSA(\sigma)$ on S^n.

In every round, a process advertises the decided vertex and scans the memory, then posts the seen vertexes, and scans the memory. If a vertex of its color is found in the intersection, the process decides on it.

In every round, at least one process decides. The rest continue with $NCSA$ on the link of the decided vertexes.

Conclusions

- $T1$ and $T2$ are equivalent.
- $T2$ is shown by simple algorithmic reductions.
- $T1$ is derived from $T2$ by proving a geometric result with a distributed algorithm.

Thank you!

References

[BG93] Elizabeth Borowsky and Eli Gafni. Generalized FLP impossibility result for t-resilient asynchronous computations. In *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC)*, pages 91–100, May 1993.

[BG97] Elizabeth Borowsky and Eli Gafni. A simple algorithmically reasoned characterization of wait-free computation. In *Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing (PODC97)*, August 1997.

[HS93] Maurice Herlihy and Nir Shavit. The asynchronous computability theorem for t-resilient tasks. In *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC)*, May 1993.

[HS94] Maurice Herlihy and Nir Shavit. A simple constructive computability theorem for wait-free computation. In *Proceedings of the 26th ACM Symposium on Theory of Computing (STOC)*, July 1994.

[SZ93] Michael Saks and Fotios Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. In *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC)*, pages 101–110, May 1993.