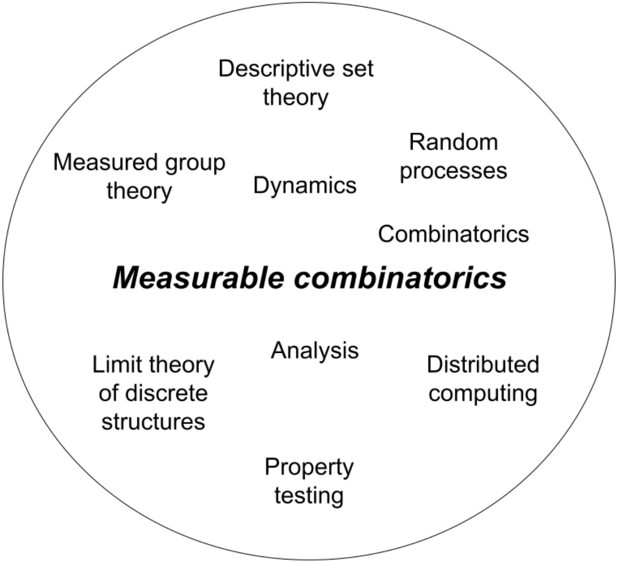
Measurable Combinatorics

Jan Grebík

Masaryk University

ADGA 2024 @ DISC, 28.10.2024

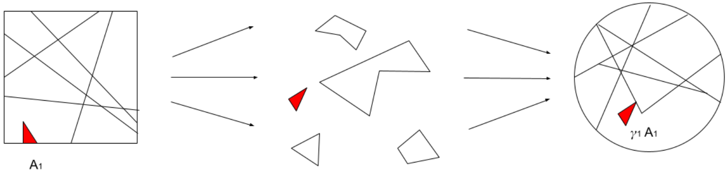


Rosetta stone are *complexity classes* of **Locally Checkable Labeling (LCL) problems**.

- some of the areas can be seen as a model for solving LCL problems,
- *Circle-squaring problem* of Tarski as a motivation for Measurable combinatorics and an illustrative example of the connections

Tarski's circle-squaring problem (1925):

Let $\mathbb{D}$ be a disc of unit area and $\mathbb{S}$ be a square of unit area in $\mathbb{R}^2$. Are $\mathbb{D}$ and $\mathbb{S}$ equidecomposable using rigid motions of $\mathbb{R}^2$?



$$\mathbb{S} = A_1 \sqcup \cdots \sqcup A_k \text{ and } \mathbb{D} = (\gamma_1 \cdot A_1) \sqcup \cdots \sqcup (\gamma_k \cdot A_k),$$

where $\gamma_1, \ldots, \gamma_k$ is a tuple of rigid motions of $\mathbb{R}^2$.

Couple of immediate questions:

What is the motivation behind this problem?

Couple of immediate questions:

What is the motivation behind this problem?

What is the connection with distributed computing?

Couple of immediate questions:

What is the motivation behind this problem?

What is the connection with distributed computing?

What is the connection with combinatorics?

Original motivation from foundations of measure theory:

- Lebesgue measure (1901) not defined on all subsets of $\mathbb{R}^d$, Vitali sets (1905),
- is there a finitely additive invariant measure defined on all subsets of $\mathbb{R}^d$,
- Banach-Tarski paradox (1924), amenability,

Original motivation from foundations of measure theory:

- Lebesgue measure (1901) not defined on all subsets of $\mathbb{R}^d$, Vitali sets (1905),
- is there a finitely additive invariant measure defined on all subsets of $\mathbb{R}^d$,
- Banach-Tarski paradox (1924), amenability,

- if Tarski's problem has *positive answer*, then $\mu(\mathbb{D}) = \mu(\mathbb{S})$ for every finitely additive invariant measure μ defined on *all* subsets of $\mathbb{R}^2$.

Equidecompositions $\leftrightsquigarrow$ Perfect matchings

Recall that our *goal* is to get

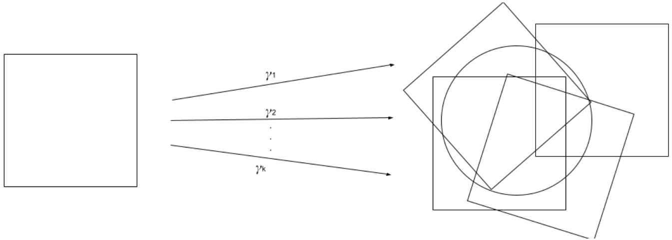
$$\mathbb{S} = A_1 \sqcup \cdots \sqcup A_k \text{ and } \mathbb{D} = (\gamma_1 \cdot A_1) \sqcup \cdots \sqcup (\gamma_k \cdot A_k).$$

Equidecompositions $\leftrightsquigarrow$ Perfect matchings

Recall that our *goal* is to get

$$\mathbb{S} = A_1 \sqcup \cdots \sqcup A_k \text{ and } \mathbb{D} = (\gamma_1 \cdot A_1) \sqcup \cdots \sqcup (\gamma_k \cdot A_k).$$

Suppose that someone smart gives us a tuple of rigid motions $(\gamma_1, \ldots, \gamma_k)$ with the promise that they work for the circle-squaring problem.

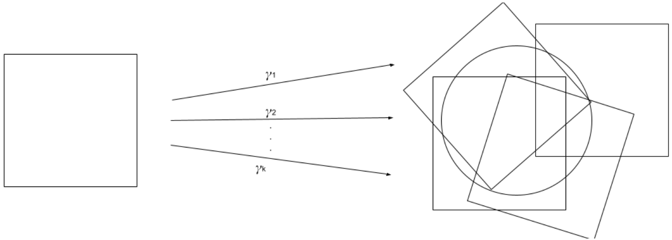


Equidecompositions $\leftrightsquigarrow$ Perfect matchings

Recall that our *goal* is to get

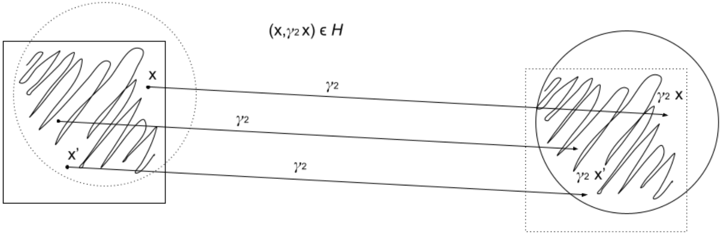
$$\mathbb{S} = A_1 \sqcup \cdots \sqcup A_k \text{ and } \mathbb{D} = (\gamma_1 \cdot A_1) \sqcup \cdots \sqcup (\gamma_k \cdot A_k).$$

Suppose that someone smart gives us a tuple of rigid motions $(\gamma_1, \ldots, \gamma_k)$ with the promise that they work for the circle-squaring problem.



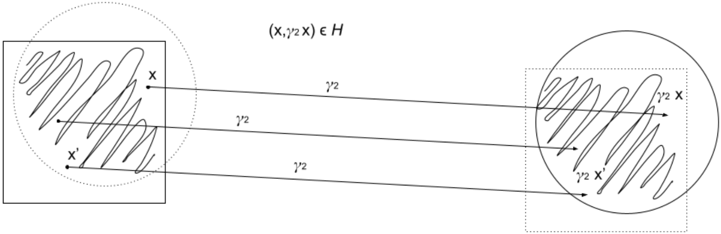
How to find the decompositions? How to interpret this promise?

Equidecompositions $\leftrightsquigarrow$ Perfect matchings

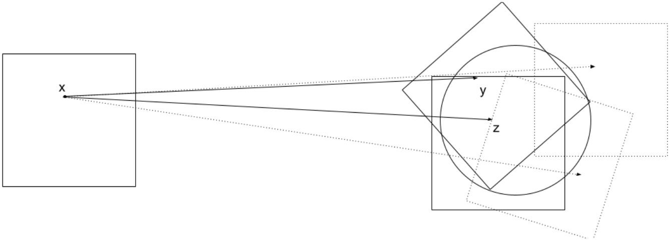


Define a bipartite graph $\mathcal{H}$ with vertex set $\mathbb{D} \sqcup \mathbb{S}$ such that $(x, y) \in \mathbb{D} \times \mathbb{S}$ is an edge if there is $1 \leq i \leq k$ such that $\gamma_i \cdot x = y$.

Equidecompositions $\leftrightsquigarrow$ Perfect matchings



Define a bipartite graph $\mathcal{H}$ with vertex set $\mathbb{D} \sqcup \mathbb{S}$ such that $(x, y) \in \mathbb{D} \times \mathbb{S}$ is an edge if there is $1 \leq i \leq k$ such that $\gamma_i \cdot x = y$.



Equidecompositions $\leftrightsquigarrow$ Perfect matchings

Observation. Finding equidecomposition using the tuple $(\gamma_1, \ldots, \gamma_k)$ is equivalent with finding a *perfect matching* in $\mathcal{H}$.

Equidecompositions $\leftrightsquigarrow$ Perfect matchings

Observation. Finding equidecomposition using the tuple $(\gamma_1, \ldots, \gamma_k)$ is equivalent with finding a *perfect matching* in $\mathcal{H}$.

If M is a perfect matching define

$$A_i = \{x \in \mathbb{D} : (x, \gamma_i \cdot x) \in M\}.$$

Equidecompositions $\leftrightsquigarrow$ Perfect matchings

Observation. Finding equidecomposition using the tuple $(\gamma_1, \ldots, \gamma_k)$ is equivalent with finding a *perfect matching* in $\mathcal{H}$.

If M is a perfect matching define

$$A_i = \{x \in \mathbb{D} : (x, \gamma_i \cdot x) \in M\}.$$

If $\mathbb{D} = A_1 \sqcup \cdots \sqcup A_k$ is the decomposition, define

$$M = \bigcup_{i=1}^{m} \{(x, \gamma_i \cdot x) \in \mathcal{H} : x \in A_i\}.$$

Trick of Laczkovich

In general, it is hard to control properties of the bipartite graph $\mathcal{H}$ for a given tuple $(\gamma_1, \ldots, \gamma_k)$, in this case, Hall's condition.

Theorem (Laczkovich 1990). $\mathbb{D}$ *and* $\mathbb{S}$ *are equidecomposable using translations only.*

The number of pieces in the proof is around 10^{40}, but people believe it should be around 20.

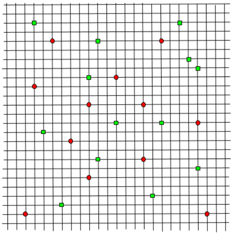
Trick of Laczkovich

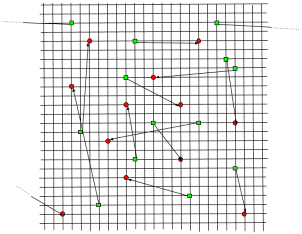
In general, it is hard to control properties of the bipartite graph $\mathcal{H}$ for a given tuple $(\gamma_1, \ldots, \gamma_k)$, in this case, Hall's condition.

Theorem (Laczkovich 1990). $\mathbb{D}$ *and* $\mathbb{S}$ *are equidecomposable using translations only.*

The number of pieces in the proof is around 10^{40}, but people believe it should be around 20.

Laczkovich modifies the problem to a special instance of a Locally Checkable Labeling (LCL) problem (with inputs) in $\mathbb{Z}^2$.





Constructible solutions

Algorithmic solutions of the equivalent problem in $\mathbb{Z}^2 \Rightarrow$ *constructible pieces* in the equidecomposition.

Theorem (Grabowski–Máthé–Pikhurko 2017). *$\mathbb{D}$ and $\mathbb{S}$ are equidecomposable with* <u>Lebesgue measurable</u> *pieces using translations only.*

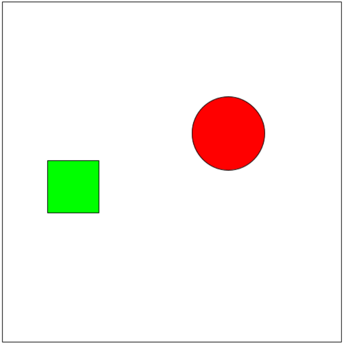
Theorem (Marks–Unger 2017). *$\mathbb{D}$ and $\mathbb{S}$ are equidecomposable with* <u>Borel measurable</u> *pieces using translations only.*

Theorem (Máthé–Noel–Pikhurko 2022). *$\mathbb{D}$ and $\mathbb{S}$ are equidecomposable with* <u>Jordan measurable</u> *pieces using translations only.*

The number of pieces in the proof is around 10^{200}, but people believe it should be around 20.

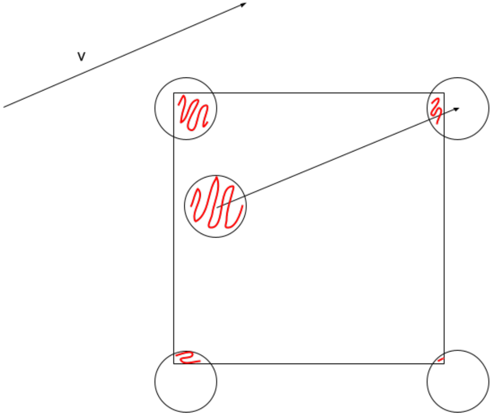
Trick of Laczkovich

Let $[0, \alpha) \times [0, \alpha)$ be a large enough torus that contain disjoint fixed copies of $\mathbb{D}$ and $\mathbb{S}$.



Trick of Laczkovich

Translation of $[0, \alpha) \times [0, \alpha)$ by a vector $v \in \mathbb{R}^2$ works as in the SNAKE game.

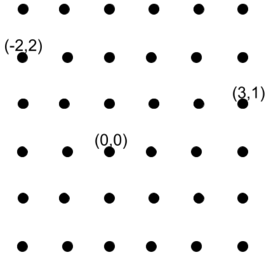


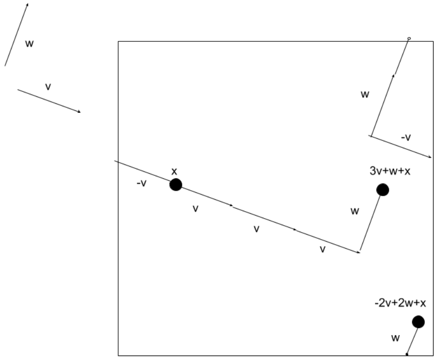
Localizing the problem

Fix two random translations v and w, and consider any $x \in [0, \alpha) \times [0, \alpha)$.

There is a correspondence between $\mathbb{Z}^2$ and the orbit

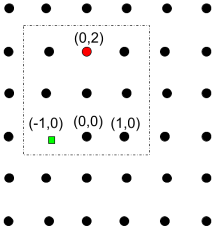
$$\{n \cdot v + m \cdot w + x : (n, m) \in \mathbb{Z}^2\}.$$

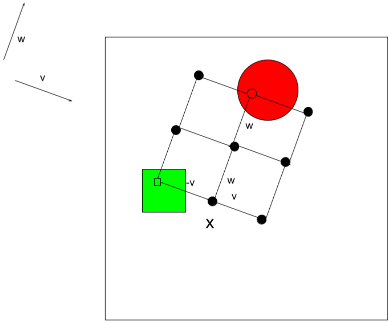




Localizing the problem

Moreover, we can mark when we hit $\mathbb{D}$ and $\mathbb{S}$.

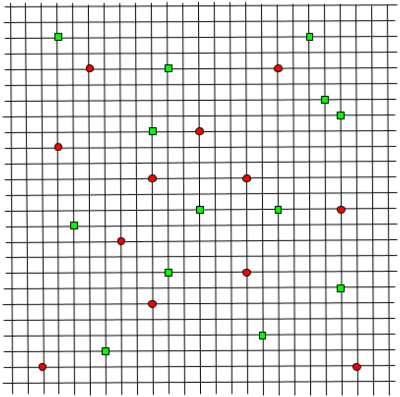




$\rightsquigarrow \mathbb{Z}^2$ *with input labels.*

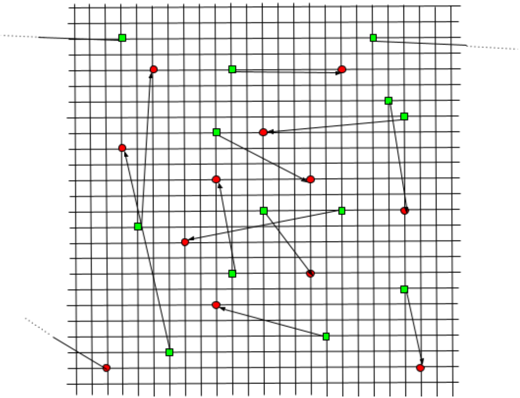
Localizing the problem

Theorem (Laczkovich). For every x, the inputs of $\mathbb{D}$ and $\mathbb{S}$ are equidisitrbuted in $\mathbb{Z}^2$.



Localizing the problem

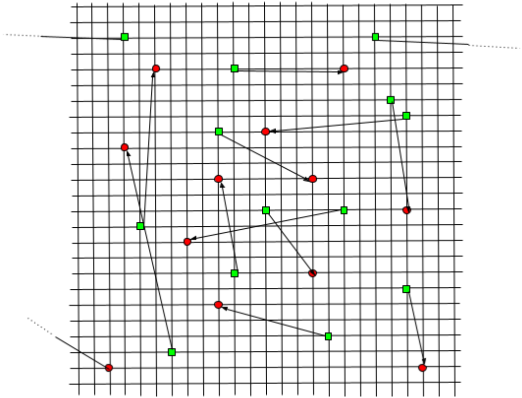
To solve Tarski's problem it is enough to find a pairing between these inputs of uniformly bounded distance in the graph, say $m \in \mathbb{N}$.



Then taking all the elements of the form $\gamma = av + bw$, where $|a|, |b| \leq m$ is the desired "smart" tuple of rigid motions.

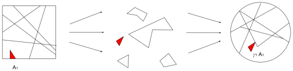
Localizing the problem

Producing a pairing between these inputs by an efficient *local algorithm* impacts the definability/measurability of the final pieces.

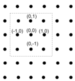


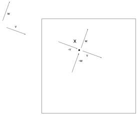
Tarski's problem

- problem in measure theory and geometry,

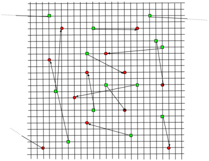


- dynamical interpretation using an action of $\mathbb{Z}^2$ on a torus





- localization to LCL problem on orbits



- local algorithms solving this problem $\rightsquigarrow$ measurable solutions.

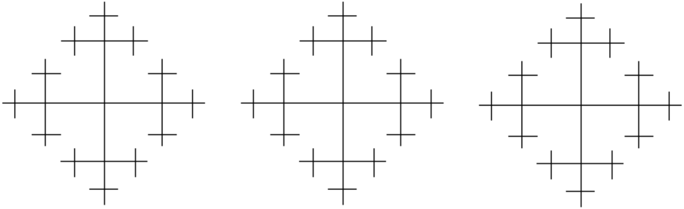
Measurable combinatorics

Systematic study of combinatorial questions on graphs that come, and are motivated by problems, from dynamics using *localization* and *local algorithms/rules*.

Measurable combinatorics

Systematic study of combinatorial questions on graphs that come, and are motivated by problems, from dynamics using *localization* and *local algorithms/rules*.

For example, measurable actions of free group on two generators, $\mathbb{F}_2 \curvearrowright (X, \mu)$



correspond to combinatorics on regular trees.

Connections

Theorem (Bernshteyn 2023). *Efficient local algorithms give automatically the existence of measurable solutions.*

Connections

Theorem (Bernshteyn 2023). *Efficient local algorithms give automatically the existence of measurable solutions.*

Theorem *Measurable construction can be used in the design of fast distributed algorithms.*

Connections

Theorem (Bernshteyn 2023). *Efficient local algorithms give automatically the existence of measurable solutions.*

Theorem *Measurable construction can be used in the design of fast distributed algorithms.*

- (Bernshteyn) Vizing's edge coloring based on augmenting chain technique of G.–Pikhurko developed for measurable edge colorings,
- (Brandt–Chang–Grunau-G.–Rozhoň–Vidnyánszky) lower bound method in the LOCAL model based on Marks' games method developed for proving non-existence of Borel measurable vertex colorings.

Connections

Theorem (Bernshteyn 2023). *Efficient local algorithms give automatically the existence of measurable solutions.*

Theorem *Measurable construction can be used in the design of fast distributed algorithms.*

- (Bernshteyn) Vizing's edge coloring based on augmenting chain technique of G.–Pikhurko developed for measurable edge colorings,
- (Brandt–Chang–Grunau-G.–Rozhoň–Vidnyánszky) lower bound method in the LOCAL model based on Marks' games method developed for proving non-existence of Borel measurable vertex colorings.

Theorem *Sometimes the setups are completely equivalent.*

Connections

Theorem (Bernshteyn 2023). *Efficient local algorithms give automatically the existence of measurable solutions.*

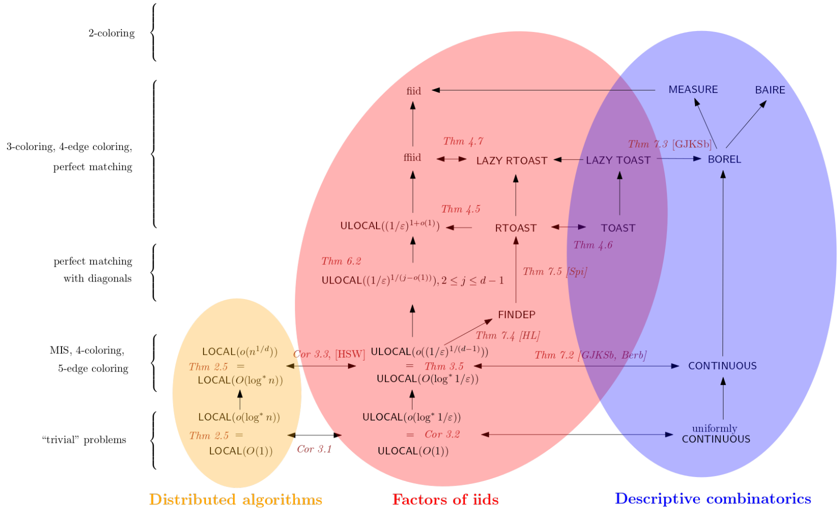
Theorem *Measurable construction can be used in the design of fast distributed algorithms.*

- (Bernshteyn) Vizing's edge coloring based on augmenting chain technique of G.–Pikhurko developed for measurable edge colorings,
- (Brandt–Chang–Grunau-G.–Rozhoň–Vidnyánszky) lower bound method in the LOCAL model based on Marks' games method developed for proving non-existence of Borel measurable vertex colorings.

Theorem *Sometimes the setups are completely equivalent.*

- (Bernshteyn, Brandt–Chang–Grunau-G.–Rozhoň–Vidnyánszky) $O(\log^*(n))$ regime in the deterministic LOCAL model and the existence of continuous colorings,
- (Brandt–Chang–Grunau-G.–Rozhoň–Vidnyánszky) $O(\log(n))$ regime in the deterministic LOCAL model for regular trees and the existence of Baire measurable colorings.

Rosetta stone on grid graphs (G.-Rozhoň)



Finitary factor of iid model of distributed computing (Holroyd–Schramm–Wilson)

A version of a randomized LOCAL model, where the algorithm does not have access to the size of the graph.

Number of rounds at each vertex is a random variable that is almost surely finite. Complexity measured by the tail decay of this variable.

Finitary factor of iid model of distributed computing (Holroyd–Schramm–Wilson)

A version of a randomized LOCAL model, where the algorithm does not have access to the size of the graph.

Number of rounds at each vertex is a random variable that is almost surely finite. Complexity measured by the tail decay of this variable.

The complexity hierarchy strictly extends the one from the randomized LOCAL model.

Usually run on an infinite transtive graph (e.g. grids, regular trees, Cayely graphs).

Finitary factor of iid model of distributed computing (Holroyd–Schramm–Wilson)

A version of a randomized LOCAL model, where the algorithm does not have access to the size of the graph.

Number of rounds at each vertex is a random variable that is almost surely finite. Complexity measured by the tail decay of this variable.

The complexity hierarchy strictly extends the one from the randomized LOCAL model.

Usually run on an infinite transtive graph (e.g. grids, regular trees, Cayely graphs).

<u>**Open problem:**</u> Study the complexity hierarchy of LCL problems in this setup.

Happy to explain more during the week.

Thank you!