Local Problems on Trees from the Perspectives of Distributed Algorithms, Finitary Factors, and Descriptive Combinatorics

Sebastian Brandt
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Yi-Jun Chang
National University of Singapore, Singapore

Jan Grebík
University of Warwick, Coventry, UK

Christoph Grunau
ETH Zürich, Switzerland

Václav Rozhoň
ETH Zürich, Switzerland

Zoltán Vidnyánszky
California Institute of Technology, Pasadena, CA, USA

Abstract

We study connections between three different fields: distributed local algorithms, finitary factors of iid processes, and descriptive combinatorics. We focus on two central questions: Can we apply techniques from one of the areas to obtain results in another? Can we show that complexity classes coming from different areas contain precisely the same problems? We give an affirmative answer to both questions in the context of local problems on regular trees:

1. We extend the Borel determinacy technique of Marks [Marks – J. Am. Math. Soc. 2016] coming from descriptive combinatorics and adapt it to the area of distributed computing, thereby obtaining a more generally applicable lower bound technique in descriptive combinatorics and an entirely new lower bound technique for distributed algorithms. Using our new technique, we prove deterministic distributed $\Omega(\log n)$-round lower bounds for problems from a natural class of homomorphism problems. Interestingly, these lower bounds seem beyond the current reach of the powerful round elimination technique [Brandt – PODC 2019] responsible for all substantial locality lower bounds of the last years. Our key technical ingredient is a novel *ID graph* technique that we expect to be of independent interest; in fact, it has already played an important role in a new lower bound for the Lovász local lemma in the Local Computation Algorithms model from sequential computing [Brandt, Grunau, Rozhoň – PODC 2021].
2. We prove that a local problem admits a Baire measurable coloring if and only if it admits a local algorithm with local complexity $O(\log n)$, extending the classification of Baire measurable colorings of Bernshteyn [Bernshteyn – personal communication]. A key ingredient of the proof is a new and simple characterization of local problems that can be solved in $O(\log n)$ rounds. We complement this result by showing separations between complexity classes from distributed computing, finitary factors, and descriptive combinatorics. Most notably, the class of problems that allow a distributed algorithm with sublogarithmic randomized local complexity is incomparable with the class of problems with a Borel solution.

We hope that our treatment will help to view all three perspectives as part of a common theory of locality, in which we follow the insightful paper of [Bernshteyn – arXiv 2004.04905].

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Distributed Algorithms, Descriptive Combinatorics

Digital Object Identifier 10.4230/LIPIcs.ITCS.2022.29

13th Innovations in Theoretical Computer Science Conference (ITCS 2022).
Editor: Mark Braverman; Article No. 29; pp. 29:1–29:26

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Related Version *Full Version*: https://arxiv.org/abs/2106.02066

Funding *Yi-Jun Chang*: Supported by Dr. Max Rössler, by the Walter Haefner Foundation, and by the ETH Zürich Foundation.
Jan Grebík: Supported by Leverhulme Research Project Grant RPG-2018-424.
Christoph Grunau: Supported by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 853109).
Václav Rozhoň: Supported by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 853109).
Zoltán Vidnyánszky: Partially supported by the National Research, Development and Innovation Office – NKFIH, grants no. 113047, no. 129211 and FWF Grant M2779.

Acknowledgements We would like to thank Anton Bernshteyn, Endre Csóka, Mohsen Ghaffari, Jan Hladký, Steve Jackson, Alexander Kechris, Edward Krohne, Oleg Pikhurko, Brandon Seward, Jukka Suomela, and Yufan Zheng for insightful discussions. Yi-Jun Chang would like to thank Yufan Zheng for a discussion about homomorphism LCL problems, in particular for suggesting potentially hard instances.

1 Introduction

In this work, we study local problems on regular trees from three different perspectives.

First, we consider the perspective of distributed algorithms. In distributed computing, the studied setup is a network of computers where each computer can only communicate with its neighbors. Roughly speaking, the question of interest in this area is which problems can be solved with only a few rounds of communication in the underlying network.

Second, we consider the perspective of (finitary) factors of iid processes. In probability, random processes model systems that appear to vary in a random manner. These include Bernoulli processes, Random walks etc. A particular, well-studied, example is the Ising model.

Third, we investigate the perspective of descriptive combinatorics. The goal of this area is to understand which constructions on infinite graphs can be performed without using the so-called axiom of choice.

Although many of the questions of interest asked in these three areas are quite similar to each other, no systematic connections were known until an insightful paper of Bernshteyn [19] who showed that results from distributed computing can automatically imply results in descriptive combinatorics. In this work, we show that the connections between the three areas run much deeper than previously known, both in terms of techniques and in terms of complexity classes. In fact, our work suggests that it is quite useful to consider all three perspectives as part of a common theory, and we will attempt to present our results accordingly. We refer the reader to Figure 1 for a partial overview of the rich connections between the three perspectives, some of which are proven in this paper.

In this work, we focus on the case where the graph under consideration is a regular tree. Despite its simplistic appearance, regular trees play an important role in each of the three areas, as we will substantiate at the end of this section. To already provide an example, in the area of distributed algorithms, the majority of known locality lower bounds is achieved on regular trees. Moreover, when regarding lower bounds, the property that they already apply on regular trees actually *strengthens* the result – a fact that is quite relevant for our work as our main contribution regarding the transfer of techniques between the areas is a new lower bound technique in the area of distributed computation that is an adaptation and generalization of a technique from descriptive combinatorics. Regarding our results about the relations between complexity classes from the three areas, we note that such connections are also studied in the context of paths and grids in other recent papers [55, 56].

For the rest of this section, we give a high-level overview of the three areas that we study. The purpose of these overviews is to provide the reader with a comprehensive picture of the studied settings that can also serve as a starting point for delving deeper into selected topics in those areas – in order to follow our paper, it is not necessary to obtain a detailed understanding of the results and connections presented in the overviews. Necessary technical details will be provided in Section 3. We present our contributions in Section 2.

Distributed Computing

The definition of the LOCAL model of distributed computing by Linial [75] was motivated by the desire to understand distributed algorithms in huge networks. As an example, consider a huge network of wifi routers. Let us think of two routers as connected by an edge if they are close enough to exchange messages. It is desirable that such close-by routers communicate with user devices on different channels to avoid interference. In graph-theoretic language, we want to properly color the underlying network. Even if we are allowed a color palette with $\Delta + 1$ colors where Δ denotes the maximum degree of the graph (which would admit a simple greedy algorithm in a sequential setting), the problem remains highly interesting in the distributed setting, as, ideally, each vertex decides on its output color after only a few rounds of communication with its neighbors, which does not allow for a simple greedy solution.

The LOCAL model of distributed computing formalizes this setup: we have a large network, where each vertex knows the network's size, n, and perhaps some other parameters like the maximum degree Δ. In the case of randomized algorithms, each vertex has access to a private random bit string, while in the case of deterministic algorithms, each vertex is equipped with a unique identifier from a range polynomial in the size n of the network. In one round, each vertex can exchange any message with its neighbors and can perform an arbitrary computation. The goal is to find a solution to a given problem in as few communication rounds as possible. As the allowed message size is unbounded, a t-round LOCAL algorithm can be equivalently described as a function that maps t-hop neighborhoods to outputs – the output of a vertex is then simply the output its t-hop neighborhood is mapped to by this function. An algorithm is correct if and only if the collection of outputs at all vertices constitutes a correct solution to the problem.

There is a rich theory of distributed algorithms and the local complexity of many problems is understood. The case of trees is a highlight of the theory: it is known that any local problem (a class of natural problems we will define later) belongs to one of only very few complexity classes. More precisely, for any local problem, its randomized local complexity is either $O(1), \Theta(\log^* n), \Theta(\log \log n), \Theta(\log n)$, or $\Theta(n^{1/k})$ for some $k \in \mathbb{N}$. Moreover, the deterministic complexity is always the same as the randomized one, except for the case $\Theta(\log \log n)$, for which the corresponding deterministic complexity is $\Theta(\log n)$ (see Figure 1).

(Finitary) Factors of iid Processes and Uniform Algorithms

In recent years, *factors of iid (fiid) processes* on trees attracted a lot of attention in combinatorics, probability, ergodic theory and statistical physics [1, 16, 3, 5, 2, 4, 6, 7, 23, 22, 36, 42, 49, 50, 57, 58, 61, 63, 64, 65, 67, 73, 88, 89, 76, 83, 90, 92, 93]. Intuitively, factors of iid processes are randomized algorithms on, e.g., infinite Δ-regular trees, where each vertex outputs a solution to a problem after it explores random strings on vertices of the whole tree. As an example, consider the *perfect matching* problem. An easy parity argument shows that perfect matching cannot be solved by any local randomized algorithm on finite

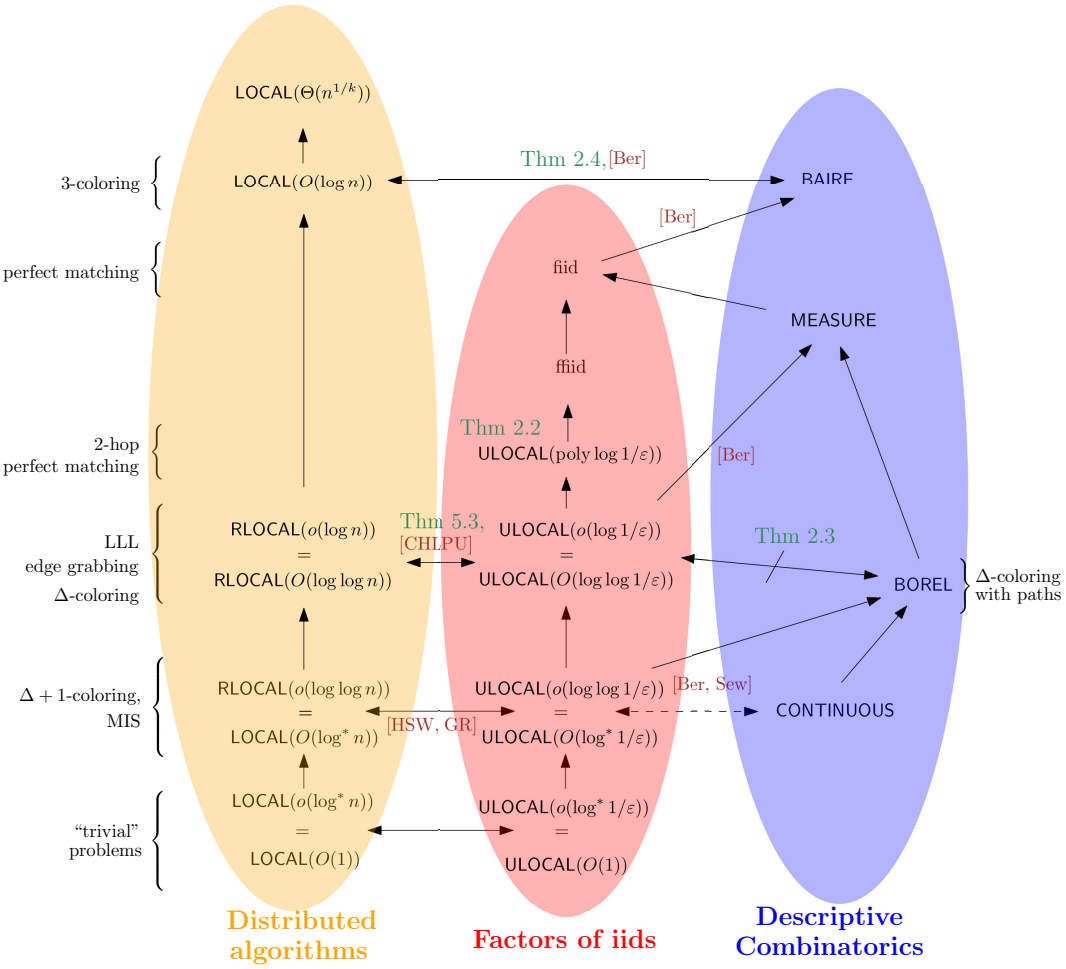

Figure 1 Complexity classes on regular trees considered in the three areas of distributed computing, factors of iid processes/uniform algorithms, and descriptive combinatorics. The left part shows complexity classes of distributed computing. We use the shorthand LOCAL if it does not matter whether we talk about the deterministic or randomized complexity. These two notions differ only for the class of problems of randomized local complexity $O(\log \log n)$, which have deterministic complexity $O(\log n)$.
The uniform complexity classes of sublogarithmic complexity are in correspondence to appropriate classes in the randomized local complexity model, as proven in the full version of the paper. On the other hand, the class fiid is very similar to the class MEASURE from descriptive combinatorics. The equivalence of the class CONTINUOUS and $\mathsf{LOCAL}(O(\log^* n)) = \mathsf{ULOCAL}(O(\log^* 1/\varepsilon))$ is marked with a dashed arrow as it was proven in case the tree is generated by a group action (think of the tree being equipped with an additional Δ-edge coloring). The inclusion $\mathsf{LOCAL}(O(\log^* n)) \subseteq \mathsf{BOREL}$ however clearly holds also in our setting. The class BOREL is incomparable with $\mathsf{RLOCAL}(O(\log \log n))$, as proven in the full version of the paper.

trees. However, if we allow a small fraction of vertices not to be matched, then, by a result of Nguyen and Onak [86] (see also [46]), there is a constant-round randomized algorithm that produces such a matching on high-girth graphs (where the constant depends on the fraction of unmatched vertices that we allow). This result can also be deduced from a result of Lyons and Nazarov [77], who showed that perfect matching can be described as a factor of iid process on an infinite Δ-regular tree. The high-level idea behind this connection

is that high-girth graphs approximate the infinite Δ-regular tree and constant-round local algorithms approximate factors of iid processes. This correspondence is formalized in the notion of *Benjamini-Schramm* or *local-global* convergence [17, 59]. We note that getting only "approximate" solutions, that is, solutions where a small fraction of vertices does not have to satisfy the constraints of a given problem, is intrinsic in this correspondence. Regardless, there are many techniques, such as entropy inequality [2] or correlation decay [4], and particular results such as the aforementioned perfect matching problem [77] that provide lower and upper bounds, respectively, in our setting as well. We refer the reader to [76, 6] for a comprehensive summary of the field.

In this paper, we mostly consider a stronger condition than fiid, namely so-called *finitary* factors of iid (ffiid) processes that are studied in the same context as fiid [62, 66, 91]. Perhaps surprisingly, the notion of ffiid is identical to the notion of so-called uniform distributed randomized algorithms [72, 56] that we now describe. We define a uniform local algorithm as a randomized local algorithm that does not know the size of the graph n – this enables us to run such an algorithm on infinite graphs, where there is no n. More precisely, we require that each vertex eventually outputs a solution that is compatible with the output in its neighborhood, but the time until the vertex finishes is a potentially unbounded random variable. As in the case of classical randomized algorithms, we can now measure the *uniform complexity* of a uniform local algorithm (known as the tail decay of ffiid [66]). The uniform complexity of an algorithm is defined as the function $t(\varepsilon)$ such that the probability that the algorithm run on a specific vertex needs to see outside its $t(\varepsilon)$-hop neighborhood is at most ε. As in the case of classical local complexity, there is a whole hierarchy of possible uniform complexities (see Figure 1).

We remark that uniform distributed local algorithms can be regarded as Las Vegas algorithms. The output will always be correct; there is however no fixed guarantee at what point all vertices have computed their final output. On the other hand, a randomized distributed local algorithm can be viewed as a Monte Carlo algorithm as it needs to produce an output after a fixed number of rounds, though the produced output might be incorrect.

Descriptive Combinatorics

The Banach-Tarski paradox states that a three-dimensional ball of unit volume can be decomposed into finitely many pieces that can be moved by isometries (distance preserving transformations such as rotations and translations) to form two three-dimensional balls each of them with unit volume(!). The graph theoretic problem lurking behind this paradox is the following: fix finitely many isometries of $\mathbb{R}^3$ and then consider a graph where x and y are connected if there is an isometry that sends x to y. Then our task becomes to find a perfect matching in the appropriate subgraph of this graph – namely, the bipartite subgraph where one partition contains points of the first ball and the other contains points of the other two balls. Banach and Tarski have shown that, with a suitably chosen set of isometries, the axiom of choice implies the existence of such a matching. In contrast, since isometries preserve the Lebesgue measure, the pieces in the decomposition cannot be Lebesgue measurable. Surprisingly, Dougherty and Foreman [44] proved that the pieces in the Banach-Tarski paradox can have the *Baire property*. The Baire property is a topological analogue of being Lebesgue measurable; a subset of $\mathbb{R}^3$ is said to have the Baire property if its difference from some open set is topologically negligible.

Recently, results similar to the Banach-Tarski paradox that lie on the border of combinatorics, logic, group theory, and ergodic theory led to an emergence of a new field often called *descriptive* or *measurable combinatorics*. The field focuses on the connection between the

discrete and continuous and is largely concerned with the investigation of graph-theoretic concepts. The usual setup in descriptive combinatorics is that we have a graph with uncountably many connected components, each being a countable graph of bounded degree. For example, in case of the Banach-Tarski paradox, the vertices of the underlying graph are the points of the three balls, edges correspond to isometries, and the degree of each vertex is bounded by the number of chosen isometries. Some of the most beautiful results related to the field include [74, 81, 54, 44, 78, 48, 71, 80, 41, 37, 43, 19], see [70, 87] for recent surveys.

Importantly, in many of these results, including the Banach-Tarski paradox, graphs where each component is an infinite Δ-regular tree appear naturally. Oftentimes, questions considered in descriptive combinatorics lead to constructing a solution to a local problem in the underlying uncountable graph (in the case of Banach-Tarski, the local problem is perfect matching). The construction needs to be such that the solution of the problem has some additional regularity properties. For example in the case of Banach-Tarski, a solution is possible when the regularity condition is the Baire property, but not if it is Lebesgue measurability. In fact, together with Borel measurability these are the most prominent regularity conditions studied in descriptive combinatorics. The corresponding complexity classes of local problems that always admit a solution with the respective regularity property are $\mathsf{BOREL}, \mathsf{MEASURE}, \mathsf{BAIRE}$ (See Figure 1). In this paper, we moreover consider the setting where each connected component of the underlying graph is a Δ-regular tree.

The connection between distributed computing and descriptive combinatorics arises from the fact that in descriptive combinatorics we care about constructions that do not use the axiom of choice. In the distributed language, the axiom of choice corresponds to leader election, that is, the constructions in descriptive combinatorics do not allow picking exactly one point in every component. To get some intuition about the power of the complexity class BOREL, we note that Borel constructions allow us to alternate countably many times the following two operations. First, any local algorithm with constant local complexity can be run. Second, we have an oracle that provides a maximal independent set (MIS) on any graph that can be constructed locally from the information computed so far [71]. Note that from the speedup result of [33] we get that every local problem with local complexity $O(\log^* n)$ can be solved by constant local constructions and *one* call to such an MIS oracle. This implies the inclusion $\mathsf{LOCAL}(O(\log^* n)) \subseteq \mathsf{BOREL}$ in Figure 1 proven in the insightful paper of Bernshteyn [19]. The relationship of the class $\mathsf{MEASURE}$ (and of the class fiid from the discussion of factors) to the class BOREL is analogous to the relationship of randomized distributed algorithms to deterministic distributed algorithms.

Local Problems on Regular Trees

After introducing the three areas of interest in this work, we conclude the section by briefly discussing the kinds of problems we focus on, which are local problems on regular trees. More precisely, we study *locally checkable labeling (LCL)* problems, which are a class of problems, where the correctness of the solution can be checked locally. Examples include classical problems from combinatorics such as proper vertex coloring, proper edge coloring, perfect matching, and maximal independent set. One main goal of this paper is to understand possible complexity classes of LCLs without inputs on infinite Δ-regular trees and their finite analogues (in the finite case, think of Δ as a constant). We refer the reader to Section 3 for a precise definition of a finite Δ-regular tree.

The motivation for studying regular trees in this work stems from different sources: (a) infinite Δ-regular trees are studied in the area of ergodic theory [23, 22], random processes [2, 4, 77] and descriptive combinatorics [80, 38], (b) many lower bounds in distributed

computing are proven in regular trees [8, 9, 27, 24, 30, 32, 53], and (c) connecting and comparing the techniques of the three areas in this simple setting reveals already deep connections, see Section 2.

2 Our Contributions

We believe that our main contribution is presenting all three perspectives as part of a common theory. Our technical contribution is split into three main parts.

2.1 Generalization of Marks' Technique

In the full version of the paper we extend the Borel determinacy technique of Marks [80], which was used to prove the nonexistence of Borel Δ-colorings and perfect matchings, to a broader class of problems, and adapt the extended technique to the distributed setting, thereby obtaining a simple method for proving distributed lower bounds. This method is the first lower bound technique for distributed computing using ideas coming from descriptive combinatorics (see [20] for a distributed computing upper bound motivated by descriptive combinatorics). Moreover, we show how to use the developed techniques to obtain both BOREL and LOCAL lower bounds for local problems from a natural class, called homomorphism problems. Our key technical ingredient for obtaining the mentioned techniques and results is a novel technique based on the notion of an *ID graph*. We note that a very similar concept to the ID graph was independently discovered by [47].

Marks' technique

In the following we give an introduction to Marks' technique by going through a variant of his proof [80, 79] that shows that Δ-coloring has deterministic local complexity $\Omega(\log n)$. The proof already works in the case where the considered regular tree comes with an input Δ-edge coloring. In this case, the output color of a given vertex u can be interpreted as that u "grabs" the incident edge of that color. The problem then reduces to the *edge grabbing* problem where every vertex is required to grab an incident edge such that no two vertices grab the same edge.

We first show the lower bound in the case that vertices do not have unique identifiers but instead are properly colored with $L > \Delta$ colors. Suppose there is an algorithm $\mathcal{A}$ solving the edge grabbing problem with local complexity $t(n) = o(\log n)$, and consider a tree rooted at vertex u of depth $t(n)$; such a tree has less than n vertices, for large enough n. Assume that u has input color $\sigma \in [L]$, and consider, for some fixed edge color α, the edge e that is incident to u and has color α. Two players, Alice and Bob, are playing the following game. In the i-th round, Alice colors the vertices at distance i from u in the subtree reachable via edge e with colors from $[L]$. Then, Bob colors all other vertices at distance i from u with colors from $[L]$. Consider the output of u when executing $\mathcal{A}$ on the obtained colored tree. Bob wins the game if u grabs the edge e, and Alice wins otherwise.

Note that either Alice or Bob has a winning strategy since the game is finite. Given the color σ of u, if, for each edge color α, Alice has a winning strategy in the game corresponding to the pair (σ, α), then we can create Δ copies of Alice and let them play their strategy on each subtree of u, telling them that the colors chosen by the other Alices are what Bob played. The result is a coloring of the input tree such that u, by definition, does not pick any edge, contradicting the fact that $\mathcal{A}$ provides a valid solution! So for every σ there is at least one α such that Bob has a winning strategy for the game corresponding to (σ, α). By

the pigeonhole principle, there are two colors σ_1, σ_2, such that Bob has a winning strategy for both pairs (σ_1, α) and (σ_2, α). But now we can imagine a tree rooted in an edge between vertices u_1, u_2 that are colored with colors σ_1, σ_2. We can now take two copies of Bob, one playing at u_1 and the other playing at u_2 and let them battle it out, telling each copy that the other color from $\{\sigma_1, \sigma_2\}$ and whatever the other copy plays are the moves of Alice. The resulting coloring has the property that both u_1 and u_2, when executing $\mathcal{A}$ on the obtained colored tree, grab the edge between them, a contradiction that finishes the proof!

The ID graph

The downside of the proof is that it does not work in the model with unique identifiers (where the players' moves consist in assigning identifiers instead of colors), since gluing copies of the same player could result in an identifier assignment where the identifiers are not unique. One possible remedy is to conduct the whole proof in the context of Borel graphs as was done by Marks. This proves an even stronger statement, namely that Δ-coloring is not in the class BOREL, but requires additional ad-hoc tricks and a pretty heavy set theoretic tool – Martin's celebrated Borel determinacy theorem [82] stating that even for infinite two-player games one of the players has to have a winning strategy if the payoff set is Borel. The ID graph enables us to adapt the proof (and its generalization that we develop in the full version of the paper) to the distributed setting, where the fact that one of the players has a winning strategy is obvious. Moreover, we use an infinite version of the ID graph to generalize Marks' technique also in the Borel setting, obtaining a number of new results in that context [26].

Here is how it works: The ID graph is a specific graph whose vertices are the possible unique input identifiers (for input graphs of size n), that is, numbers from $[n^{O(1)}]$. Its edges are colored with colors from $[\Delta]$ and its girth is $\Omega(\log n)$. When we define the game between Alice and Bob, we require them to label vertices with identifiers in such a way that whenever a new vertex is labeled with identifier i, and its already labeled neighbor has identifier j, then ij is an edge in the ID graph. Moreover, the color of edge ij in the ID graph is required to be the same as the color of the edge between the vertices labeled i and j in the tree where the game is played. It is straightforward to check that these conditions enforce that even if we let several copies of the same player play, the resulting tree is labeled with unique identifiers. Hence, the same argument as above now finally proves that the deterministic local complexity of Δ-coloring is $\Omega(\log n)$.

We note that our ID graph technique is of independent interest and may have applications in many different contexts. To give an example from distributed computing, consider the proof of the deterministic $\Omega(\log n)$-round lower bound for Δ-coloring developed by the distributed community [27, 33], which is based on the celebrated round elimination technique. Even though the result is deterministic, the proof is quite technical due to the fact that it relies on examining randomized algorithms, for reasons similar to the reasons why Marks' proof does not apply directly to the setting with unique identifiers. Fortunately, it can be again streamlined with the use of the ID graph technique. Moreover, the ID graph technique has already led to a new lower bound for the Lovász local lemma [28] in the area of Local Computation Algorithms (which is part of the realm of sequential computation), thereby giving further evidence for the versatility of the technique.

Marks vs. Round Elimination

It is quite insightful to compare Marks' technique (and our generalization of it) with the powerful round elimination technique [24], which has been responsible for all locality lower bounds of the last years [27, 9, 24, 15, 8, 30, 11, 32, 10]. While, on the surface, Marks' approach developed for the Borel world may seem quite different from the round elimination technique, there are actually striking similarities between the two methods. On a high level, in the round elimination technique, the following argument is used to prove lower bounds in the LOCAL model: If a T-round algorithm exists for a problem Π_0 of interest, then there exists a $(T-1)$-round algorithm for some problem Π_1 that can be obtained from Π_0 in a mechanical manner. By applying this step iteratively, we obtain a problem Π_t that can be solved in 0 rounds; by showing that there is no 0-algorithm for Π_t (which is easy to do if Π_t is known), a $(T+1)$-round lower bound for Π_0 is obtained.

The interesting part regarding the relation to Marks' technique is how the $(T-i-1)$-round algorithms $\mathcal{A}'$ are obtained from the $(T-i)$-round algorithms $\mathcal{A}$ in the round elimination framework: in order to execute $\mathcal{A}'$, each vertex v, being aware of its $(T-i-1)$-hop neighborhood, essentially asks whether, for all possible extensions of its view by one hop along a chosen incident edge, there exists some extension of its view by one hop along all other incident edges such that $\mathcal{A}$, executed on the obtained $(T-i)$-hop neighborhood, returns a certain output at v, and then bases its output on the obtained answer. It turns out that the vertex sets corresponding to these two extensions correspond precisely to two moves of the two players in the game(s) played in Marks' approach: more precisely, in round $T-i$ of a game corresponding to the considered vertex v and the chosen incident edge, the move of Alice consists in labeling the vertices corresponding to the first extension, and the move of Bob consists in labeling the vertices corresponding to the second extension.

However, despite the similarities, the two techniques (at least in their current forms) have their own strengths and weaknesses and are interestingly different in that there are local problems that we know how to obtain good lower bounds for with one technique but not the other, and vice versa. Finding provable connections between the two techniques is an exciting research direction that we expect to lead to a better understanding of the possibilities and limitations of both techniques.

In the full version of the paper we use our generalized and adapted version of Marks' technique to prove new lower bounds for so-called homomorphism problems. Homomorphism problems are a class of local problems that generalizes coloring problems – each vertex is to be colored with some color and there are constraints on which colors are allowed to be adjacent. The constraints can be viewed as a graph – in the case of coloring this graph is a clique. In general, whenever the underlying graph[1] of the homomorphism problem is Δ-colorable, its deterministic local complexity is $\Omega(\log n)$, because solving the problem would imply that we can solve Δ-coloring too (in the same runtime). It seems plausible that homomorphism problems of this kind are the only hard, i.e., $\Omega(\log n)$, homomorphism problems. However, our generalization of Marks' technique asserts that this is not true.

▶ **Theorem 1.** *There are homomorphism problems whose deterministic local complexity on trees of degree $\leq \Delta$ is $\Omega(\log n)$ such that the chromatic number of the underlying graph is $2\Delta - 2$.*

[1] Note that the maximum degree of the underlying graph is potentially very different from Δ, the maximum degree of the input tree.

It is not known how to prove the same lower bounds using round elimination[2]; in fact, as far as we know, these problems are the only known examples of problems on Δ-regular trees for which a lower bound is known to hold but currently not achievable by round elimination. Proving the same lower bounds via round elimination is an exciting open problem.

2.2 Separation of Various Complexity Classes

Uniform Complexity Landscape

We investigate the connection between randomized and uniform distributed local algorithms, where uniform algorithms are equivalent to the studied notion of finitary factors of iid. First, it is simple to observe that local problems with uniform complexity $t(\varepsilon)$ have randomized complexity $t(1/n^{O(1)})$ – by definition, every vertex knows its local output after that many rounds with probability $1 - 1/n^{O(1)}$. The result thus follows by a union bound over the n vertices of the input graph.

On the other hand, we observe that on Δ-regular trees the implication also goes in the opposite direction in the following sense. Every problem that has a randomized complexity of $t(n) = o(\log n)$ has a uniform complexity of $O(t(1/\varepsilon))$.

One could naively assume that this equivalence also holds for higher complexities, but this is not the case. Consider for example the 3-coloring problem. It is well-known in the distributed community that 3-coloring a tree can be solved deterministically in $O(\log n)$ rounds using the rake-and-compress decomposition [34, 84]. On the other hand, there is no uniform algorithm for 3-coloring a tree. If there were such a uniform algorithm, we could run it on any graph with large enough girth and color 99% of its vertices with three colors. This in turn would imply that the high-girth graph has an independent set of size at least $0.99 \cdot n/3$. This is a contradiction with the fact that there exist high-girth graphs with a much smaller independence number [21].

Interestingly, the characterization of Bernshteyn [18] implies that *any* uniform distributed algorithm can be "sped up" to a deterministic local $O(\log n)$ complexity, as we prove in Theorem 13.

We show that there are local problems that can be solved by a uniform algorithm but only with a complexity of $\Omega(\log 1/\varepsilon)$. Namely, the problem of constructing a 2-hop perfect matching on infinite Δ-regular trees for $\Delta \geq 3$ has a uniform local complexity between $\Omega(\log 1/\varepsilon)$ and $O(\operatorname{poly}\log 1/\varepsilon)$. Formally, this proves the following theorem.

▶ **Theorem 2.** $\mathsf{ULOCAL}(O(\log\log 1/\varepsilon)) \subsetneq \mathsf{ULOCAL}(O(\operatorname{poly}\log 1/\varepsilon))$.

The uniform algorithm for this problem is based on a so-called one-ended forest decomposition introduced in [38] in the descriptive combinatorics context. In a one-ended forest decomposition, each vertex selects exactly one of its neighbors as its parent by orienting the corresponding edge outwards. This defines a decomposition of the vertices into infinite trees. We refer to such a decomposition as a one-ended forest decomposition if the subtree rooted at each vertex only contains finitely many vertices. Having computed such a decomposition, 2-hop perfect matching can be solved inductively starting from the leaf vertices of each tree.

We leave the understanding of the uniform complexity landscape in the regime $\Omega(\log 1/\varepsilon)$ as an exciting open problem. In particular, does there exist a function $g(\varepsilon)$ such that each local problem that can be solved by a uniform algorithm has a uniform complexity of $O(g(\varepsilon))$?

[2] Indeed, the descriptions of the problems have comparably large numbers of labels and do not behave like so-called "fixed points" (i.e., nicely) under round elimination, which suggests that it is hard to find a round elimination proof with the currently known approaches.

Relationship of Distributed Classes with Descriptive Combinatorics

Bernshteyn recently proved that $\mathsf{LOCAL}(O(\log^* n)) \subseteq \mathsf{BOREL}$ [19]. That is, each local problem with a deterministic LOCAL complexity of $O(\log^* n)$ also admits a Borel-measurable solution. A natural question to ask is whether the converse also holds. Indeed, it is known that $\mathsf{LOCAL}(O(\log^* n)) = \mathsf{BOREL}$ on paths with no additional input [55]. We show that on regular trees the situation is different. On one hand, a characterization of Bernshteyn [18] implies that $\mathsf{BOREL} \subseteq \mathsf{BAIRE} \subseteq \mathsf{LOCAL}(O(\log n))$. On the other hand, we show that this result cannot be strengthened by proving the following result.

▶ **Theorem 3.** $\mathsf{BOREL} \not\subseteq \mathsf{RLOCAL}(o(\log n))$.

That is, there is a local problem that admits a Borel-measurable solution but cannot be solved by a (randomized) LOCAL algorithm running in a sublogarithmic number of rounds.

Let us sketch a weaker separation, namely that $\mathsf{BOREL} \setminus \mathsf{LOCAL}(O(\log^* n)) \neq \emptyset$. Consider a version of Δ-coloring where a subset of vertices can be left uncolored. However, the subgraph induced by the uncolored vertices needs to be a collection of doubly-infinite paths (in finite trees, this means each path needs to end in a leaf vertex). The nonexistence of a fast distributed algorithm for this problem essentially follows from the celebrated $\Omega(\log n)$ deterministic lower bound for Δ-coloring of [27]. On the other hand, the problem allows a Borel solution. First, sequentially compute $\Delta - 2$ maximal independent sets, each time coloring all vertices in the MIS with the same color, followed by removing all the colored vertices from the graph. In that way, a total of $\Delta - 2$ colors are used. Moreover, each uncolored vertex has at most 2 uncolored neighbors. This implies that the set of uncolored vertices forms a disjoint union of finite paths, one ended infinite paths and doubly infinite paths. The first two classes can be colored inductively with two additional colors, starting at one endpoint of each path in a Borel way (namely it can be done by making use of the countably many MISes in larger and larger powers of the input graph). Hence, in the end only doubly infinite paths are left uncolored, as desired.

To show the stronger separation between the classes BOREL and $\mathsf{RLOCAL}(o(\log n))$ we use a variation of the 2-hop perfect matching problem. In this variation, some of the vertices can be left unmatched, but similar as in the variant of the Δ-coloring problem described above, the graph induced by all the unmatched vertices needs to satisfy some additional constraints.

We conclude the paragraph by noting that the separation between the classes BOREL and $\mathsf{LOCAL}(O(\log^* n))$ is not as simple as it may look in the following sense. This is because problems typically studied in the LOCAL model with a LOCAL complexity of $\omega(\log^* n)$ like Δ-coloring and perfect matching also do not admit a Borel-measurable solution due to the technique of Marks [80] that we discussed in Section 2.1.

2.3 $\mathsf{LOCAL}(O(\log n)) = \mathsf{BAIRE}$

We already discussed that one of complexity classes studied in descriptive combinatorics is the class BAIRE. Recently, Bernshteyn proved [18] that all local problems that are in the complexity class BAIRE, MEASURE or fiid have to satisfy a simple combinatorial condition which we call being ℓ-full. On the other hand, all ℓ-full problems allow a BAIRE solution [18]. This implies a complete combinatorial characterization of the class BAIRE. We defer the formal definition of ℓ-fullness to Section 4 as it requires a formal definition of a local problem. Informally speaking, in the context of vertex labeling problems, a problem is ℓ-full if we can choose a subset S of the labels with the following property. Whenever we label two

endpoints of a path of at least ℓ vertices with two labels from S, we can extend the labeling with labels from S to the whole path such that the overall labeling is valid. For example, proper 3-coloring is 3-full with $S = \{1, 2, 3\}$ because for any path of three vertices such that its both endpoints are colored arbitrarily, we can color the middle vertex so that the overall coloring is proper. On the other hand, proper 2-coloring is not ℓ-full for any ℓ.

We complement this result as follows. First, we prove that any ℓ-full problem has local complexity $O(\log n)$, thus proving that all complexity classes considered in the areas of factors of iids and descriptive combinatorics from Figure 1 are contained in $\mathsf{LOCAL}(O(\log n))$. In particular, this implies that the existence of *any* uniform algorithm implies a local distributed algorithm for the same problem of local complexity $O(\log n)$. We obtain this result via the well-known rake-and-compress decomposition [84].

On the other hand, we prove that any problem in the class $\mathsf{LOCAL}(O(\log n))$ satisfies the ℓ-full condition. The proof combines a machinery developed by Chang and Pettie [34] with additional nontrivial ideas. In this proof we construct recursively a sequence of sets of rooted, layered, and partially labeled trees, where the partial labeling is computed by simulating any given $O(\log n)$-round distributed algorithm, and then the set S meeting the ℓ-full condition is constructed by considering all possible extensions of the partial labeling to complete correct labeling of these trees.

This result implies the following equality:

▶ **Theorem 4.** $\mathsf{LOCAL}(O(\log n)) = \mathsf{BAIRE}$.

This equality is surprising in that the definitions of the two classes do not seem to have much in common at first glance! Moreover, the proof of the equality relies on nontrivial results in both distributed algorithms (the technique of Chang and Pettie [34]) and descriptive combinatorics (the fact that a hierarchical decomposition, so-called toast, can be constructed in BAIRE, [40], see Proposition 11).

The combinatorial characterization of the local complexity class $\mathsf{LOCAL}(O(\log n))$ on Δ-regular trees is interesting from the perspective of distributed computing alone. This result can be seen as a part of a large research program aiming at classification of possible local complexities on various graph classes [13, 27, 33, 34, 31, 35, 12, 8, 14]. That is, we wish not only to understand possible complexity classes (see the left part of Figure 1 for possible local complexity classes on regular trees), but also to find combinatorial characterizations of problems in those classes that allow us to efficiently decide for a given problem which class it belongs to. Unfortunately, even for grids with input labels, it is *undecidable* whether a given local problem can be solved in $O(1)$ rounds [85, 29], since local problems on grids can be used to simulate a Turing machine. This undecidability result does not apply to paths and trees, hence for these graph classes it is still hopeful that we can find simple and useful characterizations for different classes of distributed problems.

In particular, on paths it is decidable what classes a given local problem belongs to, for all classes coming from the three areas considered here, and this holds even if we allow inputs [35, 55]. The situation becomes much more complicated when we consider trees. Recently, classification results on trees were obtained for so-called binary-labeling problems [8]. More recently, a complete classification was obtained in the case of *rooted regular trees* [12]. Although their algorithm takes exponential time in the worst case, the authors provided a practical implementation fast enough to classify many typical problems of interest.

Much less is known for general, *unoriented* trees, with an arbitrary number of labels. In general, deciding the optimal distributed complexity for a local problem on bounded-degree trees is $\mathsf{EXPTIME}$-hard [31], such a hardness result does not rule out the possibility for

having a simple and polynomial-time characterization for the case of *regular trees*, where there is no input and the constraints are placed only on degree-Δ vertices. Indeed, it was stated in [12] as an open question to find such a characterization. Our characterization of $\mathsf{LOCAL}(O(\log n)) = \mathsf{BAIRE}$ by ℓ-full problems makes progress in better understanding the distributed complexity classes on trees and towards answering this open question.

Roadmap

All results mentioned in Section 1 are proven in the full version of the paper [25], see also [26]. In this short version we only focus on the result $\mathsf{BAIRE} = \mathsf{LOCAL}(O(\log n))$ and, due to the space constraints, provide only a sketch of the proof in Section 4.

3 Preliminaries

In this section, we explain the setup we work with and the main definitions and results that are needed for the one particular result that we discuss in Section 4; all the other definitions are in the full version of the paper. The class of graphs that we consider in this work are either infinite Δ-regular trees, or their finite analogue that we define formally in Section 3.1.

We sometimes explicitly assume $\Delta > 2$. The case $\Delta = 2$, that is, studying paths, behaves differently and seems much easier to understand [55]. Unless stated otherwise, we do not consider any additional structure on the graphs, but sometimes it is natural to work with trees with an input Δ-edge-coloring.

3.1 Local Problems on Δ-regular trees

The problems we study in this work are locally checkable labeling (LCL) problems, which, roughly speaking, are problems that can be described via local constraints that have to be satisfied in a suitable neighborhood of each vertex. In the context of distributed algorithms, these problems were introduced in the seminal work by Naor and Stockmeyer [85], and have been studied extensively since. In the modern formulation introduced in [24], instead of labeling vertices or edges, LCL problems are described by labeling half-edges, i.e., pairs of a vertex and an incident edge. This formulation is very general in that it not only captures vertex and edge labeling problems, but also others such as orientation problems, or combinations of all of these types. Before we can provide this general definition of an LCL, we need to introduce some definitions. We start by formalizing the notion of a half-edge.

▶ **Definition 5** (Half-edge). *A* half-edge *is a pair (v, e) where v is a vertex, and e an edge incident to v. We say that a half-edge (v, e) is* incident *to a vertex w if $w = v$, we say that a vertex w is* contained *in a half edge (v, e) if $w = v$, and we say that (v, e)* belongs *to an edge e' if $e' = e$. We denote the set of all half-edges of a graph G by $H(G)$. A* half-edge labeling *is a function $c\colon H(G) \to \Sigma$ that assigns to each half-edge an element from some label set Σ.*

In order to speak about finite Δ-regular trees, we need to consider a slightly modified definition of a graph. We think of each vertex to be contained in Δ-many half-edges, however, not every half edge belongs to an actual edge of the graph. That is half-edges are pairs (v, e), but e is formally not a pair of vertices. Sometimes we refer to these half-edges as *virtual* half-edges. We include a formal definition to avoid confusions.

▶ **Definition 6** (Δ-regular trees). *A tree T, finite or infinite is a* Δ-regular tree *if either it is infinite and $T = T_\Delta$, where T_Δ is the unique infinite Δ-regular tree, that is each vertex has exactly Δ-many neighbors, or it is finite of maximum degree Δ and each vertex $v \in T$ of degree $d \leq \Delta$ is contained in $(\Delta - d)$-many virtual half-edges.*

Formally, we can view T as a triplet $(V(T), E(T), H(T))$, where $(V(T), E(T))$ is a tree of maximum degree Δ and $H(T)$ consists of real half-edges, that is pairs (v, e), where $v \in V(T)$, $e \in E(T)$ and e is incident to v, together with some virtual edges, in the case when T is finite, such that each vertex $v \in V(T)$ is contained in exactly Δ-many half-edges (real or virtual).

As we are considering trees in this work, each LCL problem can be described in a specific form that provides two lists, one describing all label combinations that are allowed on the half-edges incident to a vertex, and the other describing all label combinations that are allowed on the two half-edges belonging to an edge.[3] We arrive at the following definition for LCLs on Δ-regular trees.[4]

▶ **Definition 7** (LCLs on Δ-regular trees)**.** *A* locally checkable labeling problem*, or* LCL *for short, is a triple $\Pi = (\Sigma, \mathcal{V}, \mathcal{E})$, where Σ is a finite set of labels, $\mathcal{V}$ is a subset of unordered cardinality-Δ multisets[5] of labels from Σ , and $\mathcal{E}$ is a subset of unordered cardinality-2 multisets of labels from Σ.*

We call $\mathcal{V}$ and $\mathcal{E}$ the vertex constraint *and* edge constraint *of Π, respectively. Moreover, we call each multiset contained in $\mathcal{V}$ a* vertex configuration *of Π, and each multiset contained in $\mathcal{E}$ an* edge configuration *of Π.*

Let T be a Δ-regular tree and $c : H(T) \to \Sigma$ a half-edge labeling of T with labels from Σ. We say that c is a Π-coloring*, or, equivalently, a correct solution for Π, if, for each vertex v of T, the multiset of labels assigned to the half-edges incident to v is contained in $\mathcal{V}$, and, for each edge e of T, the cardinality-2 multiset of labels assigned to the half-edges belonging to e is an element of $\mathcal{E}$.*

An equivalent way to define our setting would be to consider Δ-regular trees as commonly defined, that is, there are vertices of degree Δ and vertices of degree 1, i.e., leaves. In the corresponding definition of LCL one would consider leaves as unconstrained w.r.t. the vertex constraint, i.e., in the above definition of a correct solution the condition "for each vertex v" is replaced by "for each non-leaf vertex v". Equivalently, we could allow arbitrary trees of maximum degree Δ as input graphs, but, for vertices of degree $< \Delta$, we require the multiset of labels assigned to the half-edges to be extendable to some cardinality-Δ multiset in $\mathcal{V}$. When it helps the exposition of our ideas and is clear from the context, we may make use of these different but equivalent perspectives.

We illustrate the difference between our setting and the "standard setting" without virtual half-edges on the perfect matching problem. A standard definition of the perfect matching problem is that some edges are picked in such a way that each vertex is covered by exactly one edge. It is easy to see that there is no local algorithm to solve this problem on the class of finite trees (without virtual half-edges), by a simple parity argument. However, in our setting, every vertex needs to pick exactly one half-edge (real or virtual) in such a way that both endpoints of each edge are either picked or not picked. We remark that in our setting it is not difficult to see that (if $\Delta > 2$) this problem can be solved by a local deterministic algorithm of local complexity $O(\log(n))$.

---

[3] Every problem that can be described in the form given by Naor and Stockmeyer [85] can be equivalently described as an LCL problem in this list form, by simply requiring each output label on some half-edge h to encode all output labels in a suitably large (constant) neighborhood of h in the form given in [85].

[4] Note that the defined LCL problems do not allow the correctness of the output to depend on input labels.

[5] Recall that a multiset is a modification of the concept of sets, where repetition is allowed.

3.2 The LOCAL model

In this section, we define local algorithms and local complexity. Recall that when we talk about a distributed algorithm on Δ-regular trees, the algorithm has access to n, the size of the tree. Also recall that $B(v,t)$ is the t-hop neighborhood of a vertex v in an underlying graph G.

▶ **Definition 8** (Local algorithm). *A* distributed local algorithm $\mathcal{A}$ *of local complexity* $t(n)$ *is a function defined on all possible* $t(n)$*-hop neighborhoods of a vertex. Applying an algorithm* $\mathcal{A}$ *on an input graph* G *means that the function is applied to a* $t(n)$*-hop neighborhood* $B(u,t(n))$ *of each vertex* u *of* G*. The output of the function is a labeling of the half-edges around the given vertex. The algorithm also takes as input the size of the input graph* n*.*

▶ **Definition 9** (Local complexity). *We say that an LCL problem* Π *has a deterministic local complexity* $t(n)$ *if there is a local algorithm* $\mathcal{A}$ *of local complexity* $t(n)$ *such that when run on the input graph* G*, with each of its vertices having a unique identifier from* $[n^{O(1)}]$*,* $\mathcal{A}$ *always returns a valid solution to* Π*. We also say* $\Pi \in \mathsf{LOCAL}(O(t(n)))$*.*

Whenever we talk about *local* complexity on Δ-regular trees, we always tacitly think about the class of finite Δ-regular trees.

3.3 Descriptive combinatorics

Let us give a high-level overview of descriptive complexity classes and their connection to distributing computing for the readers more familiar with the latter. Formal definitions of the descriptive combinatorics classes can be found in the full version of the paper [25], here we only provide a description of the class BAIRE, as this class is used in the next section.

The complexity class that partially captures deterministic local complexity classes is called BOREL. First note that by a result of Kechris, Solecki and Todorčević [71] the *maximal independent set problem* (with any parameter $r \in \mathbb{N}$) is in this class for any bounded degree graph.[6] In particular, this yields that BOREL contains the class $\mathsf{LOCAL}(O(\log^* n))$ by the characterization of [33], see [19]. Moreover, as mentioned before, BOREL is closed under countably many iterations of the operations of finding maximal independent set (for some parameter that might grow) and of applying a constant local rule that takes into account what has been constructed previously.[7]

To get a better grasp of what this means, consider for example the proper vertex 2-coloring problem on half-lines. It is clear that no local algorithm can solve this problem. However, as it is possible to determine the starting vertex after countably many iterations of the maximal independent set operation, we conclude that this problem is in the class BOREL. The idea that BOREL can compute some unbounded, global, information will be implicitly used in all the constructions in the full version of the paper that separate BOREL from local classes.

The intuition behind the class MEASURE is that it relates in the same way to the class BOREL, as randomized local algorithms relate to deterministic ones. In particular, the operations that are allowed in the class MEASURE are the same as in the class BOREL but the solution of a given LCL can be incorrect on a measure zero set.

[6] That is, it is possible to find a Borel maximal independent set, i.e., a maximal independent set which is, moreover, a Borel subset of the vertex set.

[7] It is in fact an open problem, whether this captures fully the class BOREL. However, note that an affirmative answer to this question would yield that problems can be solved in an "effective" manner in the Borel context, which is known not to be the case in unbounded degree graphs [94].

The class BAIRE can be considered as a topological equivalent of the measure theoretic class MEASURE, that is, a solution can be incorrect on a topologically negligible set. The main difference between the classes MEASURE and BAIRE is that in the later there is a hierarchical decomposition that is called *toast*. (Note that this phenomenon is present in the case of MEASURE exactly on so-called amenable graphs. It is also tightly connected with the notion of hyperfiniteness [40, 52].) The independence of colorings on a tree together with this structure allows for a combinatorial characterization of the class BAIRE, which was proven by Bernshteyn [18], see also Section 4.

Next we formulate the precise definition. We refer the reader to [87, 70, 19, 69], or to [55, Introduction, Section 4.1] and [56, Section 7.1, 7.2] for intuitive examples and standard facts of descriptive set theory. In particular, we do not define here the notions standard Borel space, Polish topology, Baire property etc.

Let $\mathcal{G}$ be a Borel graph of bounded maximum degree on a standard Borel space X. In this paper we consider exclusively acyclic Δ-regular Borel graphs and we refer to them as Δ-*regular Borel forests*. It is easy to see that the set of half-edges (see Definition 5) is naturally a standard Borel space; we denote this set by $H(\mathcal{G})$. Thus, it makes sense to consider Borel labelings of $H(\mathcal{G})$. Moreover, if $\mathcal{G}$ is a Δ-regular Borel forest and $\Pi = (\Sigma, \mathcal{V}, \mathcal{E})$ is an LCL, we can also decide whether a coloring $f : \mathcal{H}(\mathcal{G}) \to \Sigma$ is a solution to Π as in Definition 7. Similarly, we say that the coloring f solves Π, e.g., on a τ-comeager set for some compatible Polish topology τ on X if there is a Borel set $C \subseteq X$ that is comeager, the vertex constraints are satisfied around every $x \in C$ and the edge constraints are satisfied for every $x, y \in C$ that form an edge in $\mathcal{G}$.

▶ **Definition 10** (The complexity class BAIRE)**.** *We say that* Π *is in the class* BAIRE *if for every acyclic* Δ*-regular Borel graph* $\mathcal{G}$ *on a standard Borel space* X *and every compatible Polish topology* τ *on* X*, there is a Borel function* $f : H(\mathcal{G}) \to \Sigma$ *that is a* Π*-coloring of* $\mathcal{G}$ *on a* τ*-comeager set.*

4 BAIRE = LOCAL($O(\log n)$)

In this section, we present a sketch of the proof of the fact that on Δ-regular trees the classes BAIRE and LOCAL($O(\log n)$) are the same. At first glance, this result looks rather counter-intuitive. This is because in finite Δ-regular trees every vertex can see a leaf of distance $O(\log n)$, while there are no leaves at all in an infinite Δ-regular tree. However, there is an intuitive reason why these classes are the same: in both setups there is a technique to decompose an input graph into a hierarchy of subsets. Furthermore, the existence of a solution that is defined inductively with respect to these decompositions can be characterized by the same combinatorial condition of Bernshteyn [18]. We start with a high-level overview of the decomposition techniques used in both contexts.

Rake and Compress

The hierarchical decomposition in the context of distributed computing is based on a variant of a decomposition algorithm of Miller and Reif [84]. Their original decomposition algorithm works as follows. Start with a tree T, and repeatedly apply the following two operations alternately: Rake (remove all degree-1 vertices) and Compress (remove all degree-2 vertices). Then $O(\log n)$ iterations suffice to remove all vertices in T [84]. To view it another way, this produces a decomposition of the vertex set V into $2L - 1$ layers

$$V = V_1^{\mathsf{R}} \cup V_1^{\mathsf{C}} \cup V_2^{\mathsf{R}} \cup V_2^{\mathsf{C}} \cup V_3^{\mathsf{R}} \cup V_3^{\mathsf{C}} \cup \cdots \cup V_L^{\mathsf{R}},$$

with $L = O(\log n)$, where V_i^{R} is the set of vertices removed during the i-th Rake operation and V_i^{C} is the set of vertices removed during the i-th Compress operation. We will use a variant [34] of this decomposition in the proof of Proposition 15.

Variants of this decomposition turned out to be useful in designing LOCAL algorithms [34, 32, 31]. In our context, we assume that the given LCL satisfies a certain combinatorial condition and then find a solution inductively, in the reversed order of the construction of the decomposition. Namely, in the Rake step we want to be able to existentially extend the inductive partial solution to all relative degree 1-vertices (each $v \in V_i^{\mathsf{R}}$ has degree at most 1 in the subgraph induced by $V_i^{\mathsf{R}} \cup \cdots \cup V_L^{\mathsf{R}}$) and in the Compress step we want to extend the inductive partial solution to paths with endpoints labeled from the induction (the vertices in V_i^{C} form degree-2 paths in the subgraph induced by $V_i^{\mathsf{C}} \cup \cdots \cup V_L^{\mathsf{R}}$).

TOAST

Finding a hierarchical decomposition in the context of descriptive combinatorics is tightly connected with the notion of *Borel hyperfiniteness*. Understanding which Borel graphs are Borel hyperfinite is a major theme in descriptive set theory [45, 51, 37]. It is known that grids, and generally polynomial growth graphs are hyperfinite, while, e.g., acyclic graphs are not in general hyperfinite [68]. A strengthening of hyperfiniteness that is of interest to us is called a *toast* [52, 40]. A q-toast, where $q \in \mathbb{N}$, of a graph G is a collection $\mathcal{D}$ of finite subsets of G with the property that (i) every pair of vertices is covered by an element of $\mathcal{D}$ and (ii) the boundaries of every $D \neq E \in \mathcal{D}$ are at least q apart. The idea to use a toast structure to solve LCLs appears in [40] and has many applications since then [52, 81]. This approach has been formalized in [56], where the authors introduce TOAST algorithms. Roughly speaking, an LCL Π admits a TOAST algorithm if there is $q \in \mathbb{N}$ and a partial extending function (the function is given a finite subset of a tree that is partially colored and outputs an extension of this coloring on the whole finite subset) that has the property that whenever it is applied inductively to a q-toast, then it produces a Π-coloring. An advantage of this approach is that once we know that a given Borel graph admits, e.g., a Borel toast structure and a given LCL Π admits a TOAST algorithm, then we may conclude that Π is in the class BOREL. Similarly for MEASURE, BAIRE or ULOCAL, we refer the reader to [56] for more details and results concerning grids.

In the case of trees there is no way of constructing a Borel toast in general, however, it is a result of Hjorth and Kechris [60] that every Borel graph is hyperfinite on a comeager set for every compatible Polish topology. A direct consequence of [78, Lemma 3.1] together with a standard construction of toast via Voronoi cells gives the following strengthening to toast. We include a sketch of the proof for completeness.

▶ **Proposition 11.** *Let $\mathcal{G}$ be a Borel graph on a Polish space (X, τ) with degree bounded by $\Delta \in \mathbb{N}$. Then for every $q > 0$ there is a Borel $\mathcal{G}$-invariant τ-comeager set C on which $\mathcal{G}$ admits a Borel q-toast.*

Therefore to understand LCLs in the class BAIRE we need to understand which LCLs on trees admit TOAST algorithms. It turns out that these notions are equivalent, again by using the combinatorial characterization of Bernshteyn [18] that we now discuss.

Combinatorial Condition – ℓ-full set

In both decompositions, described above, we need to extend a partial coloring along paths that have their endpoints colored from the inductive step. The precise formulation of the

combinatorial condition that captures this demand was extracted by Bernshteyn [18]. He proved that it characterizes the class BAIRE for Cayley graphs of virtually free groups. Note that this class contains, e.g., Δ-regular trees with a proper edge Δ-coloring.

▶ **Definition 12** (Combinatorial condition – an ℓ-full set)**.** *Let $\Pi = (\Sigma, \mathcal{V}, \mathcal{E})$ be an LCL and $\ell \geq 2$. A set $\mathcal{V}' \subseteq \mathcal{V}$ is* ℓ-full *whenever the following is satisfied. Take a path with at least ℓ vertices, and add half-edges to it so that each vertex has degree Δ. Take any $c_1, c_2 \in \mathcal{V}'$ and label arbitrarily the half-edges around the endpoints with c_1 and c_2, respectively. Then there is a way to label the half-edges around the remaining at least $\ell-2$ vertices with configurations from $\mathcal{V}'$ such that all the $\ell - 1$ edges on the path have valid edge configuration on them.*

Now we are ready to formulate the result that combines Bernshteyn's result [18] (equivalence between (1.) and (3.), and the moreover part) with the main results of this section.

▶ **Theorem 13.** *Let Π be an LCL on regular trees. Then the following are equivalent:*
1. $\Pi \in$ BAIRE*,*
2. Π *admits a* TOAST *algorithm,*
3. Π *admits an ℓ-full set,*
4. $\Pi \in \mathsf{LOCAL}(O(\log(n)))$.

Moreover, any of the equivalent conditions is necessary for $\Pi \in$ fiid.

Next we discuss the proof of Theorem 13. We refer the reader to Bernshteyn's paper [18] for full proofs in the case of BAIRE and fiid, here we only sketch the argument for completeness. We also note that instead of using the toast construction, he used a path decomposition of acyclic graphs of Conley, Marks and Unger [39].

4.1 Sufficiency

We first show that the combinatorial condition is sufficient for BAIRE and $\mathsf{LOCAL}(O(\log(n)))$. Namely, it follows from the next results together with Proposition 11 that (2.) implies all the other conditions in Theorem 13. As discussed above the main idea is to color inductively along the decompositions.

▶ **Proposition 14.** *Let $\Pi = (\Sigma, \mathcal{V}, \mathcal{E})$ be an LCL that admits an ℓ-full set $\mathcal{V}' \subseteq \mathcal{V}$ for some $\ell > 0$. Then Π admits a* TOAST *algorithm that produces a Π-coloring for every $(2\ell+2)$-toast $\mathcal{D}$.*

Proof sketch. Our aim is to build a partial extending function. Set $q := 2\ell + 2$. Let E be a piece in a q-toast $\mathcal{D}$ and suppose that $D_1, \ldots, D_k \in \mathcal{D}$ are subsets of E such that the boundaries are separated. Suppose, moreover, that we have defined inductively a coloring of half-edges of vertices in $D = \bigcup D_i$ using only vertex configurations from $\mathcal{V}'$ such that every edge configuration $\mathcal{E}$ is satisfied for every edge in D.

We handle each connected component of $E \setminus D$ separately. Let A be one of them. Let $u \in A$ be a boundary vertex of E. Such a vertex exists since every vertex in E has degree Δ. The distance of u and any D_i is at least $2\ell+2$ for every $i \in [k]$. We orient all the edges from A towards u. Moreover if $v_i \in A$ is a boundary vertex of some D_i we assign to v_i a path V_i of length ℓ towards u. Note that V_i and V_j have distance at least 1, in particular, they are disjoint for $i \neq j \in [k]$. Now, until you encounter some path V_i, color any half-edges of vertices in A inductively starting at u in such a way that the edge constraint $\mathcal{E}$ is satisfied on every edge and only vertex configurations from $\mathcal{V}'$ are used. Use the definition of ℓ-full set to find a coloring of any such V_i and continue in a similar manner until the whole A is colored. ◀

▶ **Proposition 15** (ℓ-full $\Rightarrow$ LOCAL($O(\log n)$)). *Let $\Pi = (\Sigma, \mathcal{V}, \mathcal{E})$ be an LCL with an ℓ-full set $\mathcal{V}' \subseteq \mathcal{V}$. Then Π can be solved in $O(\log n)$ rounds in* LOCAL.

Proof. The proof uses a variant of the rake-and-compress decomposition considered in [34].

The Decomposition

The decomposition is parameterized by an integer $\ell' \geq 1$, and it decomposes the vertices of T into $2L-1$ layers $V = V_1^{\mathsf{R}} \cup V_1^{\mathsf{C}} \cup V_2^{\mathsf{R}} \cup V_2^{\mathsf{C}} \cup V_3^{\mathsf{R}} \cup V_3^{\mathsf{C}} \cup \cdots \cup V_L^{\mathsf{R}}$ with $L = O(\log n)$. We write G_i^{C} to denote the subtree induced by the vertices $\left(\bigcup_{j=i+1}^{L} V_j^{\mathsf{R}}\right) \cup \left(\bigcup_{j=i}^{L-1} V_j^{\mathsf{C}}\right)$. Similarly, G_i^{R} is the subtree induced by the vertices $\left(\bigcup_{j=i}^{L} V_j^{\mathsf{R}}\right) \cup \left(\bigcup_{j=i}^{L-1} V_j^{\mathsf{C}}\right)$. The sets V_i^{R} and V_i^{C} are required to satisfy the following requirements.

- Each $v \in V_i^{\mathsf{R}}$ has degree at most 1 in the graph G_i^{R}.
- Each $v \in V_i^{\mathsf{C}}$ has degree exactly 2 in the graph G_i^{C}. Moreover, the V_i^{C}-vertices in G_i^{C} form paths with s vertices, with $\ell' \leq s \leq 2\ell'$.

For any given constant $\ell' \geq 1$, it was shown in [34] that such a decomposition of a tree T can be computed in $O(\log n)$ rounds. See Figure 2 for an example of such a decomposition with $\ell' = 4$.

The Algorithm

Given such a decomposition with $\ell' = \max\{1, \ell - 2\}$, Π can be solved in $O(\log n)$ rounds by labeling the vertices in this order: V_L^{R}, V_{L-1}^{C}, V_{L-1}^{R}, …, V_1^{R}, as follows. The algorithm only uses the vertex configurations in the ℓ-full set $\mathcal{V}'$.

Labeling V_i^{R}

Suppose all vertices in $V_L^{\mathsf{R}}, V_{L-1}^{\mathsf{C}}, V_{L-1}^{\mathsf{R}}, \ldots, V_i^{\mathsf{C}}$ have been labeled using $\mathcal{V}'$. Recall that each $v \in V_i^{\mathsf{R}}$ has degree at most one in the graph G_i^{R}. If $v \in V_i^{\mathsf{R}}$ has no neighbor in $V_L^{\mathsf{R}} \cup V_{L-1}^{\mathsf{C}} \cup V_{L-1}^{\mathsf{R}} \cup \cdots \cup V_i^{\mathsf{C}}$, then we can label the half-edges surrounding v by any $c \in \mathcal{V}'$. Otherwise, $v \in V_i^{\mathsf{R}}$ has exactly one neighbor u in $V_L^{\mathsf{R}} \cup V_{L-1}^{\mathsf{C}} \cup V_{L-1}^{\mathsf{R}} \cup \cdots \cup V_i^{\mathsf{C}}$. Suppose the vertex configuration of u is c, where the half-edge label on $\{u, v\}$ is $\mathtt{a} \in c$. A simple observation from the definition of ℓ-full sets is that for any $c \in \mathcal{V}'$ and any $\mathtt{a} \in c$, there exist $c' \in \mathcal{V}'$ and $\mathtt{a}' \in c'$ in such a way that $\{\mathtt{a}, \mathtt{a}'\} \in \mathcal{E}$. Hence we can label the half-edges surrounding v by $c' \in \mathcal{V}'$ where the half-edge label on $\{u, v\}$ is $\mathtt{a}' \in c'$.

Labeling V_i^{C}

Suppose all vertices in $V_L^{\mathsf{R}}, V_{L-1}^{\mathsf{C}}, V_{L-1}^{\mathsf{R}}, \ldots, V_{i+1}^{\mathsf{R}}$ have been labeled using $\mathcal{V}'$. Recall that the V_i^{C}-vertices in G_i^{C} form degree-2 paths $P = (v_1, v_2, \ldots, v_s)$, with $\ell' \leq s \leq 2\ell'$. Let $P' = (x, v_1, v_2, \ldots, v_s, y)$ be the path resulting from appending to P the neighbors of the two end-points of P in G_i^{C}. The two vertices x and y are in $V_L^{\mathsf{R}} \cup V_{L-1}^{\mathsf{C}} \cup V_{L-1}^{\mathsf{R}} \cup \cdots \cup V_{i+1}^{\mathsf{R}}$, so they have been assigned half-edge labels using $\mathcal{V}'$. Since P' contains at least $\ell' + 2 \geq \ell$ vertices, the definition of ℓ-full sets ensures that we can label $v_1, v_2, \ldots, v_s$ using vertex configurations in $\mathcal{V}'$ in such a way that the half-edge labels on $\{x, v_1\}, \{v_1, v_2\}, \ldots, \{v_s, y\}$ are all in $\mathcal{E}$. ◀

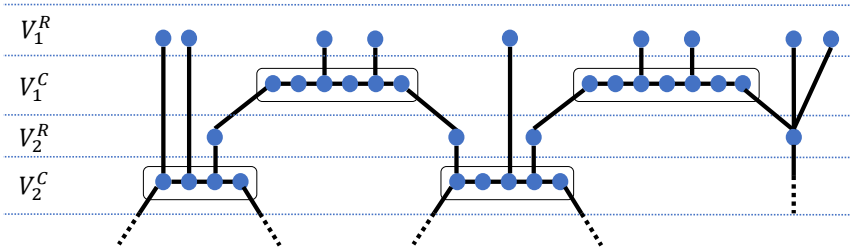

Figure 2 The variant of the rake-and-compress decomposition used in the proof of Proposition 15.

4.2 Necessity

We start by sketching that (2.) in Theorem 13 is necessary for BAIRE and fiid.

▶ **Theorem 16** (Bernshteyn [18]). *Let $\Pi = (\Sigma, \mathcal{V}, \mathcal{E})$ be an LCL and suppose that $\Pi \in$ BAIRE or $\Pi \in$ fiid. Then Π admits an ℓ-full set $\mathcal{V}' \subseteq \mathcal{V}$ for some $\ell > 0$.*

Proof Sketch. We start with BAIRE. Suppose that every Borel acyclic Δ-regular graph admits a Borel solution on a τ-comeager set for every compatible Polish topology τ. In particular, this holds for the Borel graph induced by the standard generators of the free product of Δ-copies of $\mathbb{Z}_2$ on the free part of the shift action on the alphabet $\{0,1\}$ endowed with the product topology. Let F be such a solution. Write $\mathcal{V}' \subseteq \mathcal{V}$ for the configurations of half-edge labels around vertices that F outputs on a non-meager set. Let C be a comeager set on which F is continuous. Then, every element of $\mathcal{V}'$ is encoded by some finite window in the shift on C, that is, for each element there are a $k \in \mathbb{N}$ and function $s : B(1,k) \to \{0,1\}$ such that F is constant on the set $N_s \cap C$ (where N_s is the basic open neighbourhood determined by s, and $B(1,k)$ is the k-neighbourhood of the identity in the Cayley graph of the group). Since $\mathcal{V}'$ is finite, we can take $t > 0$ to be the maximum of such k's. It follows by standard arguments that $\mathcal{V}'$ is ℓ-full for $\ell > 2t + 1$.

A similar argument works for the fiid, however, for the sake of brevity, we sketch a shorter argument that uses the fact that there must be a correlation decay for factors of iid's. Let $\Pi \in$ fiid. That is, there is an $\mathrm{Aut}(T)$-equivariant measurable function from iid's on T (without colored edges this time) into the space of Π-colorings. Let $\mathcal{V}'$ be the set of half-edge configurations around vertices that have non-zero probability to appear. Let $u, v \in T$ be vertices of distance $k_0 \in \mathbb{N}$. By [6] the correlation between the configurations around u and v tends to 0 as $k_0 \to \infty$. This means that if the distance is big enough, then all possible pairs of $\mathcal{V}'$ configurations need to appear. ◀

To finish the proof of Theorem 13 we need to demonstrate the following theorem. Note that $\mathsf{LOCAL}(n^{o(1)}) = \mathsf{LOCAL}(O(\log n))$ according to the $\omega(\log n) - n^{o(1)}$ complexity gap [34].

▶ **Theorem 17.** *Let $\Pi = (\Sigma, \mathcal{V}, \mathcal{E})$ be an LCL solvable in $\mathsf{LOCAL}(n^{o(1)})$ rounds. Then there exists an ℓ-full set $\mathcal{V}' \subseteq \mathcal{V}$ for some $\ell \geq 2$.*

We sketch the high-level idea of the proof. A natural attempt for showing $\mathsf{LOCAL}(n^{o(1)})$ $\Rightarrow$ ℓ-full is to simply take any $\mathsf{LOCAL}(n^{o(1)})$ algorithm $\mathcal{A}$ solving Π, and then take $\mathcal{V}'$ to be all vertex configurations that can possibly occur in an output of $\mathcal{A}$. It is not hard to see that this approach does not work in general, because the algorithm might use a special strategy to label vertices with degree smaller than Δ. Specifically, there might be some vertex configuration c used by $\mathcal{A}$ so that some $\mathtt{a} \in c$ will only be used to label *virtual* half-edges. It will be problematic to include c in $\mathcal{V}'$.

To cope with this issue, we do not deal with general bounded-degree trees. Instead, we construct recursively a sequence $(W_1^*, W_2^*, \ldots, W_L^*)$ of sets of rooted, layered, and *partially labeled* trees in a special manner. A tree T is included in W_i^* if it can be constructed by gluing a multiset of rooted trees in W_{i-1}^* and a new root vertex r in a certain fixed manner. A vertex is said to be in layer i if it is introduced during the i-th step of the construction, i.e., it is introduced as the root r during the construction of W_i^* from W_{i-1}^*. All remaining vertices are said to be in layer 0.

We show that each $T \in W_L^*$ admits a correct labeling that extends the given partial labeling, as these partial labelings are computed by a simulation of $\mathcal{A}$. Moreover, in these correct labelings, the variety of possible configurations of half-edge labels around vertices in different non-zero layers is the same for each layer. This includes vertices of non-zero layer whose half-edges are labeled by the given partial labeling. We simply pick $\mathcal{V}'$ to be the set of all configurations of half-edge labels around vertices that can appear in a non-zero layer in a correct labeling of a tree $T \in W_L^*$. Our construction ensures that each $c \in \mathcal{V}'$ appears as the labeling of some degree-Δ vertex in some tree that we consider.

The proof that $\mathcal{V}'$ is an ℓ-full set is based on finding paths using vertices of non-zero layers connecting two vertices with any two vertex configurations in $\mathcal{V}'$ in different lengths. These paths exist because the way rooted trees in W_{i-1}^* are glued together in the construction of W_i^* is sufficiently flexible. The reason that we need $\mathcal{A}$ to have complexity $\mathsf{LOCAL}(n^{o(1)})$ is that the construction of the trees can be parameterized by a number w so that all the trees have size polynomial in w and the vertices needed to be assigned labeling are at least distance w apart from each other. Since the number of rounds of $\mathcal{A}$ executed on trees of size $w^{O(1)}$ is much less than w, each labeling assignment can be calculated locally and independently. The construction of the trees as well as the analysis are based on a machinery developed in [34]. Specifically, we will consider the equivalence relation $\overset{\star}{\sim}$ defined in [34] and prove some of its properties, including a pumping lemma for bipolar trees. The exact definition of $\overset{\star}{\sim}$ in this paper is different from the one in [34] because the mathematical formalism describing LCL problems in this paper is different from the one in [34]. After that, we will consider a procedure for gluing trees parameterized by a labeling function f similar to the one used in [34]. We will apply this procedure iteratively to generate a set of trees. We will show that the desired ℓ-full set $\mathcal{V}' \subseteq \mathcal{V}$ can be constructed by considering the set of all possible correct labelings of these trees.

References

1 Omer Angel, Itai Benjamini, Ori Gurel-Gurevich, Tom Meyerovitch, and Ron Peled. Stationary map coloring. *Ann. Inst. Henri Poincaré Probab. Stat.*, 48(2):327–342, 2012.

2 Ágnes Backhausz, Balázs Gerencsér, and Viktor Harangi. Entropy inequalities for factors of IID. *Groups Geom. Dyn.*, 13(2):389–414, 2019. doi:10.4171/GGD/492.

3 Ágnes Backhausz and Balázs Szegedy. On large girth regular graphs and random processes on trees. *Random Structures Algorithms*, 53(3):389–416, 2018.

4 Ágnes Backhausz, Balázs Szegedy, and Bálint Virág. Ramanujan graphings and correlation decay in local algorithms. *Random Structures Algorithms*, 47(3):424–435, 2015. doi:10.1002/rsa.20562.

5 Ágnes Backhausz and Bálint Virag. Spectral measures of factor of i.i.d. processes on vertex-transitive graphs. *Ann. Inst. Henri Poincaré Probab. Stat.*, 53(4):2260–2278, 2017.

6 Ágnes Backhausz, Balázs Gerencsér, Viktor Harangi, and Máté Vizer. Correlation bounds for distant parts of factor of iid processes. *Combin. Probab. Comput.*, 27(1):1–20, 2018.

7 K. Ball. Poisson thinning by monotone factors. *Electron. Comm. Probab.*, 10:60–69, 2005.

8 Alkida Balliu, Sebastian Brandt, Yuval Efron, Juho Hirvonen, Yannic Maus, Dennis Olivetti, and Jukka Suomela. Classification of distributed binary labeling problems. In Hagit Attiya, editor, *34th International Symposium on Distributed Computing (DISC)*, volume 179 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.DISC.2020.17.

9 Alkida Balliu, Sebastian Brandt, Juho Hirvonen, Dennis Olivetti, Mikaël Rabie, and Jukka Suomela. Lower bounds for maximal matchings and maximal independent sets. In *Proceedings of the IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 481–497, 2019. doi:10.1109/FOCS.2019.00037.

10 Alkida Balliu, Sebastian Brandt, Fabian Kuhn, and Dennis Olivetti. Improved distributed lower bounds for mis and bounded (out-)degree dominating sets in trees. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing (PODC)*, 2021.

11 Alkida Balliu, Sebastian Brandt, and Dennis Olivetti. Distributed lower bounds for ruling sets. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 365–376, 2020. doi:10.1109/FOCS46700.2020.00042.

12 Alkida Balliu, Sebastian Brandt, Dennis Olivetti, Jan Studený, Jukka Suomela, and Aleksandr Tereshchenko. Locally checkable problems in rooted trees. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, 2021.

13 Alkida Balliu, Sebastian Brandt, Dennis Olivetti, and Jukka Suomela. Almost global problems in the local model. *Distributed Computing*, pages 1–23, 2020.

14 Alkida Balliu, Juho Hirvonen, Janne H Korhonen, Tuomo Lempiäinen, Dennis Olivetti, and Jukka Suomela. New classes of distributed time complexity. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1307–1318, 2018.

15 Alkida Balliu, Juho Hirvonen, Dennis Olivetti, and Jukka Suomela. Hardness of minimal symmetry breaking in distributed computing. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 369–378, 2019.

16 Ferenc Bencs, Aranka Hrušková, and László Márton Tóth. Factor of iid schreier decoration of transitive graphs. *arXiv:2101.12577*, 2021.

17 Itai Benjamini and Oded Schramm. Recurrence of distributional limits of finite planar graphs. In *Selected Works of Oded Schramm*, pages 533–545. Springer, 2011.

18 Anton Bernshteyn. Work in progress.

19 Anton Bernshteyn. Distributed algorithms, the Lovász Local Lemma, and descriptive combinatorics. *arXiv preprint arXiv:2004.04905*, 2020.

20 Anton Bernshteyn. A fast distributed algorithm for $(\Delta+1)$-edge-coloring. *preprint*, 2020.

21 Béla Bollobás. The independence ratio of regular graphs. *Proc. Amer. Math. Soc.*, 83(2):433–436, 1981. doi:10.2307/2043545.

22 Lewis Bowen. The ergodic theory of free group actions: entropy and the f -invariant. *Groups Geom. Dyn.*, 4(3):419–432, 2010.

23 Lewis Bowen. A measure-conjugacy invariant for free group actions. *Ann. of Math.*, 171(2):1387–1400, 2010.

24 Sebastian Brandt. An automatic speedup theorem for distributed problems. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 379–388, 2019. doi:10.1145/3293611.3331611.

25 Sebastian Brandt, Yi-Jun Chang, Jan Grebík, Christoph Grunau, Vaclav Rozhoň, and Zoltán Vidnyánszky. Local problems on trees from the perspectives of distributed algorithms, finitary factors, and descriptive combinatorics. *arXiv preprint arXiv:2106.02066*, 2021. A short version will be presented at the *13th Innovations in Theoretical Computer Science conference (ITCS 2022)*. arXiv:2106.02066.

26 Sebastian Brandt, Yi-Jun Chang, Jan Grebík, Christoph Grunau, Václav Rozhoň, and Zoltán Vidnyánszky. On homomorphism graphs, 2021. arXiv:2111.03683.

27 Sebastian Brandt, Orr Fischer, Juho Hirvonen, Barbara Keller, Tuomo Lempiäinen, Joel Rybicki, Jukka Suomela, and Jara Uitto. A lower bound for the distributed Lovász local lemma. In *Proc. 48th ACM Symp. on Theory of Computing (STOC)*, pages 479–488, 2016.

28 Sebastian Brandt, Christoph Grunau, and Václav Rozhoň. Randomized local computation complexity of the Lovasz local lemma. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, 2021.

29 Sebastian Brandt, Juho Hirvonen, Janne H. Korhonen, Tuomo Lempiäinen, Patric R.J. Östergård, Christopher Purcell, Joel Rybicki, Jukka Suomela, and Przemys law Uznański. LCL problems on grids. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pages 101–110, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3087801.3087833.

30 Sebastian Brandt and Dennis Olivetti. Truly tight-in-Δ bounds for bipartite maximal matching and variants. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, PODC '20, pages 69–78, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3382734.3405745.

31 Yi-Jun Chang. The Complexity Landscape of Distributed Locally Checkable Problems on Trees. In Hagit Attiya, editor, *34th International Symposium on Distributed Computing (DISC 2020)*, volume 179 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.DISC.2020.18.

32 Yi-Jun Chang, Qizheng He, Wenzheng Li, Seth Pettie, and Jara Uitto. Distributed edge coloring and a special case of the constructive Lovász local lemma. *ACM Trans. Algorithms*, 16(1):8:1–8:51, 2020. doi:10.1145/3365004.

33 Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. An exponential separation between randomized and deterministic complexity in the local model. *SIAM Journal on Computing*, 48(1):122–143, 2019.

34 Yi-Jun Chang and Seth Pettie. A time hierarchy theorem for the local model. *SIAM Journal on Computing*, 48(1):33–69, 2019.

35 Yi-Jun Chang, Jan Studený, and Jukka Suomela. Distributed graph problems through an automata-theoretic lens. *arXiv:2002.07659*, 2020.

36 Sourav Chatterjee, Ron Peled, Yuval Peres, and Dan Romik. Gravitational allocation to poisson points. *Ann. of Math.*, 172(1):617–671, 2010.

37 Clinton T. Conley, Steve Jackson, Andrew S. Marks, Brandon Seward, and Robin Tucker-Drob. Borel asymptotic dimension and hyperfinite equivalence relations. *arXiv:2009.06721*, 2020.

38 Clinton T. Conley, Andrew S. Marks, and Robin D. Tucker-Drob. Brooks' theorem for measurable colorings. *Forum Math. Sigma*, e16(4):23pp, 2016.

39 Clinton T. Conley, Andrew S. Marks, and Spencer T. Unger. Measurable realizations of abstract systems of congruences. In *Forum of Mathematics, Sigma*, volume 8. Cambridge University Press, 2020.

40 Clinton T. Conley and Benjamin D. Miller. A bound on measurable chromatic numbers of locally finite Borel graphs. *Math. Res. Lett.*, 23(6):1633–1644, 2016. doi:10.4310/MRL.2016.v23.n6.a3.

41 Clinton T. Conley and Benjamin D. Miller. Measure reducibility of countable Borel equivalence relations. *Ann. of Math. (2)*, 185(2):347–402, 2017. doi:10.4007/annals.2017.185.2.1.

42 Endre Csóka, Balázs Gerencsér, Viktor Harangi, and Bálint Virág. Invariant gaussian processes and independent sets on regular graphs of large girth. *Random Structures & Algorithms*, 47(2):284–303, 2015.

43 Endre Csóka, Lukasz Grabowski, András Máthé, Oleg Pikhurko, and Konstantinos Tyros. Borel version of the Local Lemma. *arXiv:1605.04877*, 2017.

44 Randall Dougherty and Matthew Foreman. Banach-Tarski paradox using pieces with the property of Baire. *Proc. Nat. Acad. Sci. U.S.A.*, 89(22):10726–10728, 1992. doi:10.1073/pnas.89.22.10726.

45 Randall Dougherty, Steve Jackson, and Alexander S. Kechris. The structure of hyperfinite Borel equivalence relations. *Trans. Amer. Math. Soc.*, 341(1):193–225, 1994. doi:10.2307/2154620.

46 Gábor Elek and Gábor Lippner. Borel oracles. An analytical approach to constant-time algorithms. *Proc. Amer. Math. Soc.*, 138(8):2939–2947, 2010. doi:10.1090/S0002-9939-10-10291-3.

47 Klaus-Tycho Foerster, Janne H. Korhonen, Ami Paz, Joel Rybicki, Stefan Schmid, and Jukka Suomela. Work in progress.

48 Damien Gaboriau. Coût des relations d'équivalence et des groupes. *Invent. Math.*, 139(1):41–98, 2000. doi:10.1007/s002229900019.

49 Damien Gaboriau and Russell Lyons. A measurable-group-theoretic solution to von neumann's problem. *Invent. Math.*, 177(3):533–540, 2009.

50 David Gamarnik and Madhu Sudan. Limits of local algorithms over sparse random graphs. *Proceedings of the 5-th Innovations in Theoretical Computer Science conference, ACM Special Interest Group on Algorithms and Computation Theory*, 2014.

51 Su Gao and Steve Jackson. Countable abelian group actions and hyperfinite equivalence relations. *Inventiones Math.*, 201(1):309–383, 2015.

52 Su Gao, Steve Jackson, Edward Krohne, and Brandon Seward. Forcing constructions and countable borel equivalence relations. *arXiv:1503.07822*, 2015.

53 Mika Göös, Juho Hirvonen, and Jukka Suomela. Linear-in-Δ lower bounds in the LOCAL model. In *Proc. 33rd ACM Symp. on Principles of Distributed Computing (PODC)*, pages 86–95, 2014.

54 Lukasz Grabowski, András Máthé, and Oleg Pikhurko. Measurable circle squaring. *Ann. of Math. (2)*, 185(2):671–710, 2017. doi:10.4007/annals.2017.185.2.6.

55 Jan Grebík and Václav Rozhoň. Classification of local problems on paths from the perspective of descriptive combinatorics. *arXiv preprint arXiv:2103.14112*, 2021. arXiv:2103.14112.

56 Jan Grebík and Václav Rozhoň. Local problems on grids from the perspective of distributed algorithms, finitary factors, and descriptive combinatorics. *arXiv preprint arXiv:2103.08394*, 2021. arXiv:2103.08394.

57 Ori Gurel-Gurevich and Ron Peled. Poisson thickening. *Israel J. Math.*, 196(1):215–234, 2013.

58 Viktor. Harangi and Bálint Virág. Independence ratio and random eigenvectors in transitive graphs. *Ann. Probab.*, 43(5):2810–2840, 2015.

59 Hamed Hatami, László Lovász, and Balázs Szegedy. Limits of locally-globally convergent graph sequences. *Geom. Funct. Anal.*, 24(1):269–296, 2014. doi:10.1007/s00039-014-0258-7.

60 Greg Hjorth and Alexander S. Kechris. Borel equivalence relations and classifications of countable models. *Annals of pure and applied logic*, 82(3):221–272, 1996.

61 Alexander E. Holroyd. Geometric properties of poisson matchings. *Probab. Theory Related Fields*, 150(3–4):511–527, 2011.

62 Alexander E. Holroyd. One-dependent coloring by finitary factors. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 53(2):753–765, 2017. doi:10.1214/15-AIHP735.

63 Alexander E. Holroyd, Russel Lyons, and Terry Soo. Poisson splitting by factors. *Ann. Probab.*, 39(5):1938–1982, 2011.

64 Alexander E. Holroyd, Robin Pemantle, Y. Peres, and Oded Schramm. Poisson matching. *Ann. Inst. Henri Poincaré Probab. Stat.*, 45(1):266–287, 2009.

65 Alexander E. Holroyd and Yuval Peres. Trees and matchings from point processes. *Electron. Comm. Probab.*, 8:17–27, 2003.

66 Alexander E. Holroyd, Oded Schramm, and David B. Wilson. Finitary coloring. *Ann. Probab.*, 45(5):2867–2898, 2017.

67 Carlos Hoppen and Nicholas Wormald. Local algorithms, regular graphs of large girth, and random regular graphs. *Combinatorica*, 38(3):619–664, 2018.

68 Steve Jackson, Alexander S. Kechris, and Alain Louveau. Countable Borel equivalence relations. *J. Math. Logic*, 2(01):1–80, 2002.

69 Alexander S. Kechris. *Classical descriptive set theory*, volume 156 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995. doi:10.1007/978-1-4612-4190-4.

70 Alexander S. Kechris and Andrew S. Marks. Descriptive graph combinatorics. *http://www.math.caltech.edu/~kechris/papers/combinatorics20book.pdf*, 2020.

71 Alexander S. Kechris, S lawomir Solecki, and Stevo Todorčević. Borel chromatic numbers. *Adv. Math.*, 141(1):1–44, 1999.

72 Amos Korman, Jean-Sébastien Sereni, and Laurent Viennot. Toward more localized local algorithms: removing assumptions concerning global knowledge. *Distributed Computing*, 26(5-6):289–308, September 2012. doi:10.1007/s00446-012-0174-8.

73 Gábor Kun. Expanders have a spanning lipschitz subgraph with large girth. *preprint*, 2013.

74 Miklós Laczkovich. Equidecomposability and discrepancy; a solution of Tarski's circle-squaring problem. *J. Reine Angew. Math.*, 404:77–117, 1990. doi:10.1515/crll.1990.404.77.

75 Nati Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.

76 Russell Lyons. Factors of iid on trees. *Combin. Probab. Comput.*, 26(2):285–300, 2017.

77 Russell Lyons and Fedor Nazarov. Perfect matchings as iid factors on non-amenable groups. *European Journal of Combinatorics*, 32(7):1115–1125, 2011.

78 Andrew Marks and Spencer Unger. Baire measurable paradoxical decompositions via matchings. *Advances in Mathematics*, 289:397–410, 2016.

79 Andrew S. Marks. A short proof that an acyclic n-regular Borel graph may have borel chromatic number $n+1$.

80 Andrew S. Marks. A determinacy approach to Borel combinatorics. *J. Amer. Math. Soc.*, 29(2):579–600, 2016.

81 Andrew S. Marks and Spencer T. Unger. Borel circle squaring. *Ann. of Math. (2)*, 186(2):581–605, 2017. doi:10.4007/annals.2017.186.2.4.

82 Donald A. Martin. Borel determinacy. *Ann. of Math. (2)*, 102(2):363–371, 1975. doi:10.2307/1971035.

83 Péter Mester. A factor of i.i.d with uniform marginals and infinite clusters spanned by equal labels. *preprint*, 2011.

84 Gary L. Miller and John H. Reif. Parallel tree contraction–Part I: fundamentals. *Advances in Computing Research*, 5:47–72, 1989.

85 Moni Naor and Larry Stockmeyer. What can be computed locally? *SIAM Journal on Computing*, 24(6):1259–1277, 1995.

86 Huy N. Nguyen and Krzysztof Onak. Constant-time approximation algorithms via local improvements. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 327–336, 2008. doi:10.1109/FOCS.2008.81.

87 Oleg Pikhurko. Borel combinatorics of locally finite graphs. *arXiv preprint arXiv:2009.09113*, 2021. arXiv:2009.09113.

88 Mustazee Rahman. Factor of iid percolation on trees. *SIAM J. Discrete Math.*, 30(4):2217–2242, 2016.

89 Mustazee Rahman and Bálint Virág. Local algorithms for independent sets are half-optimal. *Ann. Probab.*, 45(3):1543–1577, 2017.

90 Christian Schmidt, Nils-Eric Guenther, and Lenka Zdeborová. Circular coloring of random graphs: statistical physics investigation. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(8):083303, 2016.

91 Yinon Spinka. Finitely dependent processes are finitary. *Ann. Probab.*, 48(4):2088–2117, 2020.

92 Ádám Timár. Tree and grid factors for general point processes. *Electron. Comm. Probab.*, 9(53–59), 2004.

93 Ádám Timár. Invariant colorings of random planar maps. *Ergodic Theory Dynam. Systems*, 31(2):549–562, 2011.

94 Stevo Todorčević and Zoltán Vidnyánszky. A complexity problem for Borel graphs. *Invent. Math.*, 226:225–249, 2021.