Descriptive Combinatorics and Distributed Algorithms

Anton Bernshteyn

In this article we shall explore a fascinating area called *descriptive combinatorics* and its recently discovered connections to *distributed algorithms*—a fundamental part of computer science that is becoming increasingly important in the modern era of decentralized computation. The interdisciplinary nature of these connections means that there is very little common background shared by the researchers who are interested in them. With this in mind, this article was written under the assumption that the reader would have close to no background in either descriptive set theory *or* computer science. The reader will judge to what degree this endeavor was successful.

The article comprises two parts. In the first part we give a brief introduction to some of the central notions and problems of descriptive combinatorics. The second part is devoted to a survey of some of the results concerning the interactions between descriptive combinatorics and distributed algorithms, as well as a few open problems.

Anton Bernshteyn is a professor of mathematics at the Georgia Institute of Technology. His email address is bahtoh@gatech.edu.

This work is partially supported by the NSF grant DMS-2045412.

Communicated by Notices *Associate Editor Emilie Purvine.*

DOI: https://doi.org/10.1090/noti2539

1. A Brief Introduction to Descriptive Combinatorics

1.1. Basic notions of descriptive set theory.

1.1.1. *Countable computation and Borel sets.* Descriptive combinatorics is an area that emerged quite recently (about two decades ago) as the result of a symbiosis between *combinatorics* (especially graph theory) and *descriptive set theory*. For excellent surveys of this subject, see [KM20] by Kechris and Marks and [Pik21] by Pikhurko. In addition to combinatorics and descriptive set theory, descriptive combinatorics has close connections to numerous other branches of mathematics, such as ergodic theory, topological dynamics, probability theory, and model theory, to name a few.

To motivate the questions studied in descriptive combinatorics, it will be beneficial to first briefly discuss descriptive set theory more abstractly.

One way (out of many) of thinking about descriptive set theory is that it provides a versatile framework for gauging the inherent difficulty of mathematical problems

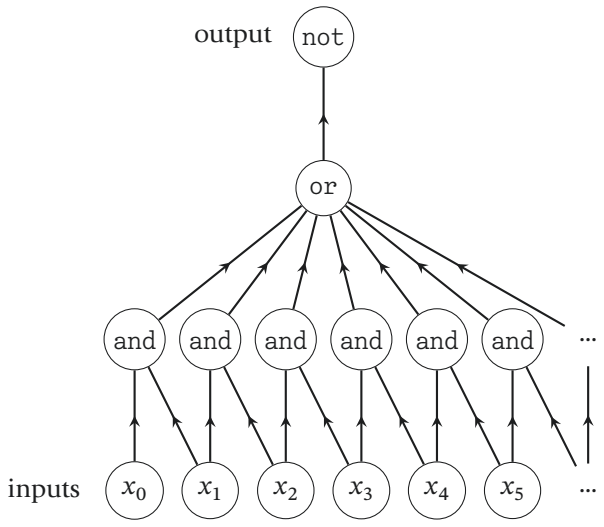

Figure 1. A countable circuit that outputs 1 if the input bit string does not contain two consecutive 1s.

pertaining to countable structures, in a manner analogous to how computational complexity theory gauges the inherent difficulty of finite problems. To pursue this analogy further, we can naturally present the basic concepts of descriptive set theory using a particular model of computation, namely *Boolean circuits*. An example of an *infinite* Boolean circuit is shown in Fig. 1. It is a network consisting of nodes, also called *gates*, joined by directed edges, or *wires*. The bottom layer comprises the *input nodes*. Each input node receives a bit value—0 or 1. The values then propagate through the network along the wires, with every gate computing a particular Boolean function of the values that feed into it: `not`, `and`, or `or`. Finally, one or more nodes are designated as the *outputs*. Thus, a Boolean circuit can be used to compute a function $\{0,1\}^{\mathrm{In}} \to \{0,1\}^{\mathrm{Out}}$, where In and Out are the sets of the input and the output nodes respectively. For example, the circuit in Fig. 1 computes the function $\{0,1\}^{\mathbb{N}} \to \{0,1\}$ that equals 1 if and only if the input string does not contain two consecutive 1s.

There is one technical issue that we must make explicit here. Not every directed network can be used to perform well-defined computation. For instance, if a network involves a directed cycle of nodes, such as $v_0 \leftarrow v_1 \leftarrow v_2 \leftarrow \cdots \leftarrow v_0$, then the computational process wouldn't even be able to start. More generally, a Boolean circuit must be *well-founded*, meaning that it must not contain an infinite descending sequence of nodes such as $v_0 \leftarrow v_1 \leftarrow v_2 \leftarrow v_3 \leftarrow \cdots$. It is not hard to show that well-foundedness is the only necessary requirement: any well-founded network of gates implements a well-defined function.

An important part of computational complexity theory is *circuit complexity*, which studies how large a (finite) circuit needs to be to compute a given function $f : \{0,1\}^n \to \{0,1\}^m$. In descriptive set theory we are similarly interested in functions that can be computed by *countable* circuits:

Definition 1.1 (Borel subsets of $\{0,1\}^{\mathbb{N}}$). A subset $A \subseteq \{0,1\}^{\mathbb{N}}$ is *Borel* if its characteristic function can be computed by a countable Boolean circuit. We let $\mathfrak{B}(\{0,1\}^{\mathbb{N}})$ denote the family of all Borel subsets of $\{0,1\}^{\mathbb{N}}$.

While Definition 1.1 applies to subsets of $\{0,1\}^{\mathbb{N}}$, it can naturally be extended to sets of other types, as long as their members can be somehow encoded by infinite bit strings. For example, a countable graph G with vertex set $V = \{v_0, v_1, v_2, \ldots\}$ can be represented by its *adjacency matrix* M_G, i.e., a countable table of 0s and 1s whose entry in row i and column j is 1 if and only if v_i and v_j are adjacent in G. We can then call a set A of countable graphs *Borel* if there is a countable Boolean circuit that decides, given the matrix M_G, whether G is in A or not. For instance, the following sets of countable graphs are Borel:

$$\{G \ : \ G \text{ is connected}\};$$
$$\{G \ : \ G \text{ is bipartite}\};$$
$$\{G \ : \ G \text{ is 3-colorable}\}.$$

(The last one may be a bit surprising, since deciding whether a finite graph is 3-colorable is believed to be a computationally difficult problem. The difficulty miraculously disappears in the countable world, however.) On the other hand, the following set can be shown to *not* be Borel:

$$\{G \ : \ G \text{ has an infinite clique}\}.$$

In fact, this set is *complete analytic*, which is a notion somewhat analogous to NP-completeness from computational complexity, with the added benefit that in descriptive set theory it is a theorem (due to Suslin from 1917) that complete analytic sets cannot be Borel. In general, *most* subsets of $\{0,1\}^{\mathbb{N}}$ are not Borel: $\{0,1\}^{\mathbb{N}}$ has $2^{2^{\aleph_0}}$ subsets but only $2^{\aleph_0}$ of them are Borel (this result follows by enumerating all countable Boolean circuits).

1.1.2. *Borel sets in Polish spaces.* A useful (and more classical) perspective on Borel sets is provided by topological considerations. We can define the distance between two infinite bit strings $x, y \in \{0,1\}^{\mathbb{N}}$ by the formula

$$\mathrm{dist}(x,y) := \sum_{n=0}^{\infty} \frac{1}{2^{n+1}} |x_n - y_n|.$$

(This can be thought of as a weighted version of the Hamming distance.) The coefficients $1/2^{n+1}$ are chosen so that the series $\sum_{n=0}^{\infty} 1/2^{n+1}$ converges (and hence the metric dist is bounded, in this case by 1), but otherwise they are arbitrary. In particular, using a different convergent series with positive terms would yield a different metric, but the resulting topology on the space $\{0,1\}^{\mathbb{N}}$ would be the same—namely, it is the product topology arising from viewing $\{0,1\}^{\mathbb{N}}$ as the product of countably many copies of the discrete space $\{0,1\}$. Actually, it is not hard to explicitly describe this topology without referring to the metric:

a set $A \subseteq \{0,1\}^{\mathbb{N}}$ is open if and only if for every sequence $x = (x_0, x_1, x_2, \ldots) \in A$, the membership of x in A can be confirmed by looking at only finitely many entries of x. More formally, A is open if for each $x \in A$, there is some $n \in \mathbb{N}$ such that every infinite bit string whose first n entries are $(x_0, x_1, \ldots, x_{n-1})$ belongs to A.

Now, it turns out that Borel subsets of $\{0,1\}^{\mathbb{N}}$ can be described in purely topological terms and without any reference to Boolean circuits as follows: $\mathfrak{B}(\{0,1\}^{\mathbb{N}})$ is the smallest family of subsets of $\{0,1\}^{\mathbb{N}}$ that includes all open sets and is closed under countable Boolean operations (i.e., complements, countable unions, and countable intersections). And this characterization can be taken as the definition of Borel subsets of any topological space:

Definition 1.2 (Borel sets in topological spaces). Let X be a topological space. We let $\mathfrak{B}(X)$ be the smallest family of subsets of X that includes all open sets and is closed under countable Boolean operations. A subset $A \subseteq X$ is *Borel* if it is a member of $\mathfrak{B}(X)$.

Although Definition 1.2 makes sense for an arbitrary topological space X, descriptive set theory is mostly concerned with so-called Polish spaces. Formally, a topological space is *Polish* if it is second-countable (i.e., it has a countable basis) and completely metrizable (i.e., the topology is generated by a complete metric). A particularly compelling reason for focusing on Polish spaces is that the points of a Polish space can be encoded by infinite bit strings in a "well-behaved" way. This statement is made precise by the following remarkable theorem:

Theorem 1.3 (Borel Isomorphism Theorem). *If X is an uncountable Polish space, then there is a bijection*

$$\mathsf{code} : X \to \{0,1\}^{\mathbb{N}}$$

such that a set $A \subseteq X$ is Borel in X if and only if the set $\{\mathsf{code}(x) : x \in A\}$ is Borel in $\{0,1\}^{\mathbb{N}}$.

In other words, the computational definition of Borel subsets of $\{0,1\}^{\mathbb{N}}$ given in Definition 1.1 can also be used to identify, via an appropriate coding, Borel sets in any uncountable Polish space. This is a powerful observation because many natural examples of topological spaces are Polish, for instance $\mathbb{R}$ and, more generally, $\mathbb{R}^n$ for $n \in \mathbb{N}$, the infinite-dimensional space $\mathbb{R}^{\mathbb{N}}$, the Baire space $\mathbb{N}^{\mathbb{N}}$, the unit intervals $[0,1]$, $(0,1)$, and $[0,1)$, the unit circle $\mathbb{S}^1$ and, more generally, any second-countable topological manifold, all compact metric spaces, all separable Banach spaces, etc. Furthermore, it is possible to parameterize various classes of mathematical structures by points in suitably defined Polish spaces. We have already seen how to use adjacency matrices to form a Polish space of countable graphs. In a similar vein, one can assemble, e.g., a Polish space of countable groups. There are also natural Polish spaces of continuous functions, measure-preserving transformations, separable Banach spaces, etc. It is even possible to define a Polish space of all Polish spaces![1]

1.1.3. *Other classes of sets in descriptive set theory.* In addition to Borel sets, there are various other classes of sets that play an important role in descriptive set theory. In this section we briefly introduce three such classes.

Let us temporarily return to the space $\{0,1\}^{\mathbb{N}}$ of infinite bit strings. Recall that a set $A \subseteq \{0,1\}^{\mathbb{N}}$ is Borel if the membership in A can be decided by a countable Boolean circuit. Sometimes it makes sense to inquire whether a set is not just Borel but *open*. As mentioned in §1.1.2, a set $A \subseteq \{0,1\}^{\mathbb{N}}$ is open if for every sequence $x \in A$, the membership of x in A can be confirmed by looking at only finitely many bits of x. Of course, we can also investigate open sets in any other Polish space.

On the other hand, sometimes a set we are working with may not be Borel and yet be "almost" Borel in a certain suitable sense. For instance, a set $A \subseteq \{0,1\}^{\mathbb{N}}$ is *measurable* if there is a countable Boolean circuit that correctly decides the membership in A for a *random* input point $x \in \{0,1\}^{\mathbb{N}}$. More precisely, A is measurable if there is a countable Boolean circuit C with the following property. Let $\mathbf{1}_A$ denote the characteristic function of A. Sample a sequence $x = (x_0, x_1, x_2, \ldots) \in \{0,1\}^{\mathbb{N}}$ randomly by making each x_i be 0 or 1 independently with probability 1/2. Then $C(x) = \mathbf{1}_A(x)$ with probability 1; in other words, C may fail to correctly determine x's membership in A, but the probability of failure is 0. We can similarly study measurable sets in any Polish space X, provided that it is equipped with a Borel probability measure μ.[2] In many applications, sets of measure 0 may be safely ignored, and thus measurability is often a "good enough" substitute for Borelness.

Finally, we also often consider the so-called Baire-measurable sets. The notion of Baire-measurability is analogous to measurability, but it is typically easier to work with. Furthermore, it is defined purely topologically, without reference to a measure or any other additional structure. The role of sets of measure 0 is played by the so-called *meager* sets, i.e., countable unions of nowhere dense sets. We think of meager sets as "topologically negligible." A set A is *Baire-measurable* if there is a Borel set A' such that

[1] *For the interested reader, we sketch the construction of the Polish space of Polish spaces. First, it turns out that every Polish space is homeomorphic to a closed subset of $\mathbb{R}^{\mathbb{N}}$. Now, let $(U_n)_{n\in\mathbb{N}}$ be a countable basis for the topology of $\mathbb{R}^{\mathbb{N}}$. To each closed set $X \subseteq \mathbb{R}^{\mathbb{N}}$, we assign a code $\mathsf{code}_X \in \{0,1\}^{\mathbb{N}}$ as follows: $\mathsf{code}_X(n) = 1$ if and only if $X \cap U_n \neq \emptyset$. Let $\mathfrak{X} \subseteq \{0,1\}^{\mathbb{N}}$ be the set of all codes of closed subsets of $\mathbb{R}^{\mathbb{N}}$. One can show that $\mathfrak{X}$ is Polish in the relative topology inherited from $\{0,1\}^{\mathbb{N}}$. Thus, all Polish spaces are parameterized by the points of the Polish space $\mathfrak{X}$.*

[2] *We should point out that the class of measurable sets varies with the choice of the measure μ. For simplicity, we shall assume throughout this article that whenever we speak of measurable sets, it is with respect to some implicitly fixed probability measure.*

the symmetric difference $A \vartriangle A'$ is meager—informally, A and A' are "topologically almost equal." The area studying meager and Baire-measurable sets is called *Baire category theory* (owing to the older term "sets of first category" for meager sets). For a detailed comparison of measure and Baire category, see the excellent book [Oxt80] by Oxtoby.

1.2. **Borel graphs and their combinatorics.** As mentioned earlier, descriptive combinatorics investigates classical combinatorial problems—such as graph coloring, matching, etc.—from the perspective of descriptive set theory. The central notion here is that of a Borel graph:

Definition 1.4 (Borel graphs). A *Borel graph* is a graph G whose vertex set $V(G)$ is a Polish space and whose edge set $E(G)$ is a Borel subset of $V(G) \times V(G)$.

In this section, we describe some examples of Borel graphs and highlight a few of their properties that are of interest in descriptive combinatorics.

1.2.1. *The hypercube.* For our first example, let H be the graph with vertex set $\{0,1\}^{\mathbb{N}}$ in which two bit strings $x = (x_0, x_1, x_2, \ldots)$, $y = (y_0, y_1, y_2, \ldots) \in \{0,1\}^{\mathbb{N}}$ are adjacent if and only if they differ in exactly one coordinate, i.e., if there is a unique index $n \in \mathbb{N}$ such that $x_n \neq y_n$. We call H the *infinite-dimensional hypercube* (it can be viewed as the inverse limit of finite-dimensional hypercubes, the first few of which are shown in Fig. 2). The graph H is Borel, since it is easy to construct a countable Boolean circuit which, given two infinite bit strings x and y, determines if they are adjacent in H.

Note that the vertex set of H has cardinality $2^{\aleph_0}$. This is typical for graphs studied in descriptive combinatorics, since their vertex sets usually are uncountable Polish spaces.

Next, we observe that, somewhat surprisingly and in contrast to finite-dimensional Boolean cubes, the graph H is disconnected. To see this, recall that the *connected component* of a vertex x in a graph G is the smallest set $[x]_G \subseteq V(G)$ that contains x and such that there are no edges joining $[x]_G$ to $V(G) \setminus [x]_G$. Equivalently, $[x]_G$ is the set of all vertices reachable from x in G by a finite path. A graph is connected if it has a single connected component. Now take any vertex $x \in \{0,1\}^{\mathbb{N}}$ in H. A moment's thought reveals that the connected component $[x]_H$ is the set of all sequences $y \in \{0,1\}^{\mathbb{N}}$ that differ from x in only finitely many coordinates. For instance, if $x = (0,0,0,\ldots)$, then $[x]_H$ is the set of all bit strings with finitely many 1s. It follows that every connected component of H is countable, which implies that H has $2^{\aleph_0}$-many components. This is also a typical feature of graphs studied in descriptive combinatorics.

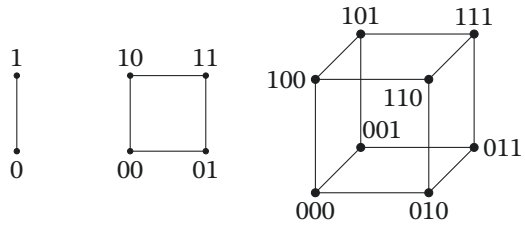

Figure 2. 1-, 2-, and 3-dimensional hypercubes. In contrast to the infinite-dimensional hypercube, these graphs are connected.

One of the most important parameters studied in graph theory is the chromatic number of a graph:

Definition 1.5 (Chromatic numbers). Let G be a graph. A subset $I \subseteq V(G)$ is *G-independent* if no edge of G joins two vertices in I. The *chromatic number* $\chi(G)$ of G is the smallest cardinal κ such that $V(G)$ can be partitioned into κ-many G-independent sets.

Let us compute the chromatic number of H:

Proposition 1.6. *We have $\chi(H) = 2$.*

Proof. Clearly, $\chi(H) \geq 2$. To prove that $\chi(H) \leq 2$, we need to partition $\{0,1\}^{\mathbb{N}}$ into two H-independent sets. To this end, choose an arbitrary representative from every connected component of H. For $x \in \{0,1\}^{\mathbb{N}}$, let $\mathrm{rep}(x)$ be the representative from $[x]_H$. Since x and $\mathrm{rep}(x)$ belong to the same component of H, they differ in finitely many coordinates. Thus, we can let $\delta(x)$ be the number of coordinates where x and $\mathrm{rep}(x)$ differ and define

$$A := \{x \in \{0,1\}^{\mathbb{N}} : \delta(x) \text{ is even}\},$$
$$B := \{x \in \{0,1\}^{\mathbb{N}} : \delta(x) \text{ is odd}\}.$$

If x and y are neighbors in H, then $\mathrm{rep}(x) = \mathrm{rep}(y)$, and hence $|\delta(x) - \delta(y)| = 1$. Therefore, the sets A and B are H-independent and partition $\{0,1\}^{\mathbb{N}}$, as desired. □

Since we know that $\chi(H) = 2$, it makes sense to ask the following question:

Can the set $\{0,1\}^{\mathbb{N}}$ be partitioned into two Borel H-independent sets?

This is a special case of the following general problem:

Definition 1.7 (Borel chromatic numbers). Let G be a Borel graph. The *Borel chromatic number* $\chi_{\mathsf{B}}(G)$ of G is the smallest $\kappa \in \{0, 1, 2, \ldots, \aleph_0, 2^{\aleph_0}\}$ such that $V(G)$ can be partitioned into κ-many Borel G-independent sets.[3] The *measurable* and *Baire-measurable chromatic numbers* $\chi_{\mathsf{M}}(G)$ and $\chi_{\mathsf{BM}}(G)$ are defined analogously.

It turns out that, although the chromatic number of H is 2, its Borel chromatic number is uncountable!

[3] *Note that, by definition, if $\chi_{\mathsf{B}}(G)$ is uncountable, then $\chi_{\mathsf{B}}(G) = 2^{\aleph_0}$. This convention is motivated by the fact that every uncountable Polish space has cardinality $2^{\aleph_0}$. However, it may still be meaningful to ask whether $V(G)$ can be partitioned into κ-many Borel G-independent sets for some $\aleph_0 < \kappa < 2^{\aleph_0}$. This issue will not play a major role in this article, since we shall be mostly interested in the case when $\chi_{\mathsf{B}}(G)$ is finite.*

Proposition 1.8. *The set $\{0,1\}^{\mathbb{N}}$ cannot be partitioned into countably many Borel H-independent sets. In other words, $\chi_{\mathsf{B}}(H) = 2^{\aleph_0}$.*

Actually, we even have $\chi_{\mathsf{M}}(H) = \chi_{\mathsf{BM}}(H) = 2^{\aleph_0}$ (and, of course, both $\chi_{\mathsf{M}}(H)$ and $\chi_{\mathsf{BM}}(H)$ are lower bounds for $\chi_{\mathsf{B}}(H)$). To prove this, one shows that every measurable H-independent set must have measure 0, while every Baire-measurable H-independent set must be meager. Since countable unions of measure-0/meager sets are still measure-0/meager, it is impossible to cover the entire space $\{0,1\}^{\mathbb{N}}$ by countably many measurable/Baire-measurable H-independent sets.

One interpretation of Proposition 1.8 is that there is no "explicit" partition of the vertex set of H into two—or even countably many—H-independent sets. What makes the construction in the proof of Proposition 1.6 "inexplicit" is the very first step, namely choosing a single vertex in each connected component of H. This step implicitly invokes the so-called *Axiom of Choice*:

Axiom 1.9 (Choice). *If $\mathcal{F}$ is a family of nonempty sets, then there is a way to pick one element from each set in $\mathcal{F}$. Formally, there is a function* choice *that assigns to each $A \in \mathcal{F}$ an element* $\mathsf{choice}(A) \in A$.

The Axiom of Choice is part of the axiom system ZFC (*Zermelo–Fraenkel set theory with the Axiom of Choice*) and plays a crucial role in the foundations of mathematics. While most other axioms of ZFC assert the existence of concrete sets, such as the powerset $\mathcal{P}(A)$ of a given set A, the Axiom of Choice postulates the existence of a choice function without actually describing *how* the choice is to be made. In this sense, the Axiom of Choice is nonconstructive—so much so that in the past it was viewed with deep suspicion by many mathematicians. After all, how can we assert that something exists without being able to provide even a single example? Although by now the controversy around the Axiom of Choice has largely died down and most mathematicians use it without reservation, it is still true that the Axiom of Choice—and hence our proof of Proposition 1.6 based on it—is inherently nonconstructive. From this perspective, Proposition 1.8 says that it is impossible to come up with an alternative, constructive argument—at least if by "constructive" we mean "Borel."[4]

1.2.2. *Translation and rotation.* In the remainder of this article we shall be concerned with Borel graphs *G of bounded degree*, meaning that there is a natural number $d \in \mathbb{N}$ such that every vertex of G has at most d neighbors. A typical example is the graph G_{tr} with vertex set $\mathbb{R}$ in which vertices x and y are adjacent if and only if $|x-y| = 1$. In other words, each $x \in \mathbb{R}$ has precisely two neighbors in G_{tr}: $x-1$ and $x+1$. It follows that every connected component of G_{tr} is a bi-infinite path *i.e., a path infinite in both directions*. In particular, just like the hypercube graph H from §1.2.1, G_{tr} has $2^{\aleph_0}$-many countable connected components. (This fact can also be observed directly, since the vertices in the half-open interval $[0,1)$ belong to distinct components.) Clearly, $\chi(G_{\mathrm{tr}}) = 2$. In contrast to the graph H from §1.2.1 however, we can explicitly describe a partition of G_{tr} into two independent sets A and B: for every integer $n \in \mathbb{Z}$, put the points in the interval $[n, n+1)$ in A if n is even and in B if n is odd (see Fig. 3). This yields

$$A = \bigcup_{k\in\mathbb{Z}} [2k, 2k+1) \quad \text{and} \quad B = \bigcup_{k\in\mathbb{Z}} [2k-1, 2k).$$

These sets are Borel, so we in fact have $\chi_{\mathsf{B}}(G_{\mathrm{tr}}) = 2$.

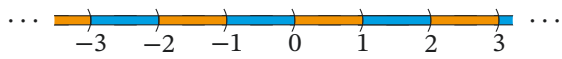

Figure 3. A partition of $\mathbb{R}$ into two Borel G_{tr}-independent sets (indicated by the colors).

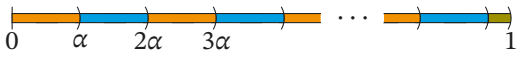

Figure 4. A partition of $[0,1)$ into three Borel G_{rot}-independent sets (indicated by the colors).

Next we fix a number $\alpha \in (0,1) \setminus \mathbb{Q}$ and define a graph G_{rot} as follows. The vertex set of G_{rot} is the half-open unit interval $[0,1)$. Two vertices $x, y \in [0,1)$ are adjacent in G_{rot} if and only if $y - x = \pm\alpha \pmod 1$. Again, every vertex x has precisely two neighbors, namely $x+\alpha \pmod 1$ and $x-\alpha \pmod 1$. (Another way of thinking about this graph is by "wrapping" the unit interval around a circle, which turns adding α modulo 1 into rotation by the angle $2\pi\alpha$.) Since α is irrational, it is not hard to see that G_{rot} has $2^{\aleph_0}$-many connected components, each of which is a bi-infinite path. In other words, the graphs G_{rot} and G_{tr} are isomorphic. In particular, $\chi(G_{\mathrm{rot}}) = 2$. But can we describe a bipartition of G_{rot} explicitly? An attempt to adapt the construction used for G_{tr} fails, although it does yield a partition of G_{rot} into three Borel independent sets (see Fig. 4). In fact, it turns out that *no explicit bipartition of G_{rot} exists*. This statement is made precise by the following proposition:

Proposition 1.10. *We have $\chi_{\mathsf{B}}(G_{\mathrm{rot}}) = 3$. Furthermore, $\chi_{\mathsf{M}}(G_{\mathrm{rot}}) = \chi_{\mathsf{BM}}(G_{\mathrm{rot}}) = 3$ as well.*

In other words, G_{tr} and G_{rot} are two graphs that are combinatorially isomorphic, but the topological structures on their vertex sets affect their Borel, measurable, and Baire-measurable chromatic numbers in different ways.

[4] *This is not the only way in which the Axiom of Choice affects chromatic numbers of graphs. For instance, the following remarkable fact was established by Galvin and Komjáth: the statement that the chromatic number is well-defined for every graph G is* equivalent *to the Axiom of Choice.*

1.2.3. *Locally checkable labeling problems.* In the above examples, we have been interested in chromatic numbers of graphs as well as in their Borel, measurable, etc. analogs. Naturally, there are many other combinatorial concepts we may want to study. Many of them fall into the general framework of *locally checkable labelling (LCL) problems* (also known as *local coloring problems*). In an LCL problem Π, we are asked to assign labels from a finite set Λ to the vertices of a graph G in such a way that certain constraints are satisfied. The constraints must be "local" in the sense that they can be checked by looking at the labels of each vertex and its neighbors. For a formal (and somewhat more general) definition, see, e.g., [Ber20, §2.A.1]. For our purposes, it will suffice to give a few examples that illustrate what sort of problems fall into this category.

The prototypical example is the *k-coloring problem*: assign the labels $\Lambda = \{1, \dots, k\}$, referred to as "colors," to the vertices of G so that the label of each vertex is distinct from the labels of its neighbors. Note that this is a "local" constraint in the sense described above: it only involves comparing the label of a vertex to those of its neighbors. By definition, the set of all vertices receiving a particular label in a k-coloring must be a G-independent set, and hence G admits a k-coloring if and only if $\chi(G) \leq k$.

Another well-studied problem is the *Maximal Independent Set (MIS) problem*: find an independent set $I \subseteq V(G)$ that is maximal under inclusion. This can be interpreted as an LCL problem as follows. We can encode a set $I \subseteq V(G)$ by its characteristic function, which assigns the labels $\Lambda = \{0, 1\}$ to the vertices of G. It is easy to see that I is an MIS if and only if this labelling satisfies the following "local" constraint:

> *If a vertex is labeled* 1*, then all its neighbors are labeled* 0*, while if a vertex is labeled* 0*, then at least one of its neighbors is labeled* 1*.*

As our last example, consider the *perfect matching problem*: find a set of edges $M \subseteq E(G)$ such that every vertex is incident to exactly one edge in M. If xy is an edge in M, then we say that the vertices x and y are *matched* to each other. Thus, a perfect matching splits $V(G)$ into pairs of matched vertices. To place the perfect matching problem into the LCL framework, let us assume that G is a graph of bounded degree; more precisely, suppose that every vertex of G has at most d neighbors, for some $d \in \mathbb{N}$. For each $x \in V(G)$, we fix an arbitrary ordering of the neighbors of x. (If G is a Borel graph, it is possible to fix such an ordering "in a Borel way," in the sense that for each i, the set $\{(x, y) \in V(G) \times V(G) \,:\, y \text{ is the } i\text{-th neighbor of } x\}$ is Borel.) Then we can identify a perfect matching with a labeling of $V(G)$ using the labels $\Lambda = \{1, \dots, d\}$ such that a vertex is given the label i when it is matched to its i-th neighbor. Such labeling encodes a perfect matching if and only if the following constraint holds:

> *Let x be a vertex labeled i and let y be the i-th neighbor of x. If y's label is j, then x is the j-th neighbor of y.*

In other words, if x is matched to y, then y must be matched to x. This constraint is again "local," which makes the perfect matching problem an LCL problem (at least on bounded degree graphs).

In descriptive combinatorics, we are interested in Borel solutions to LCL problems:

Definition 1.11 (Borel solutions to LCL problems)**.** Let Π be an LCL problem with label set Λ and let G be a Borel graph. A labeling of the vertices of G by the labels from Λ is a *Borel solution* to Π if it fulfills all the constraints of the problem Π and, additionally, for each label $\lambda \in \Lambda$, the set of all vertices labeled λ is Borel.

Measurable and *Baire-measurable solutions* to Π are defined analogously.

By applying Definition 1.11 to specific LCL problems, we obtain, as special cases, Borel k-colorings, Borel maximal independent sets, and Borel perfect matchings. In general, a lot of research in descriptive combinatorics can be seen as addressing the following comprehensive question:

Question 1.12 (Descriptive complexity of LCL problems)**.** Given a class $\mathbb{G}$ of Borel graphs, which LCL problems Π admit Borel/measurable/etc. solutions on the graphs in $\mathbb{G}$?

As stated, this question may seem way too general to say anything meaningful about. However, rather surprisingly, some recent work has led to significant progress toward a complete answer! The key to this new work is the discovery of an intimate connection between the answer to Question 1.12 and the complexity of solving the problem Π on finite graphs using a distributed algorithm. The second part of this paper is dedicated to a survey of this connection.

2. Connections to Distributed Computing

2.1. **The LOCAL model of distributed computation.** The area of distributed computing studies computational processes performed by decentralized groups of independent agents. There are several different models of distributed computation that emphasize various aspects of it, such as fault-tolerance, communication bandwidth, etc. For our purposes, the relevant model is called LOCAL and was formally introduced by Linial in 1992 [Lin92] (although some related results had already been established somewhat earlier). For a comprehensive introduction to this model, see the book [BE13] by Barenboim and Elkin. We now briefly describe the key elements of this model.

The LOCAL model operates on an n-vertex graph G. (Note that, in contrast to the examples discussed in §1.2, the graph G here is finite.) We think of G as representing a decentralized communication network where each vertex

plays the role of a processor and edges represent communication links. The computation proceeds in *rounds*. During each round, the vertices first perform some local computations and then synchronously broadcast messages to all their neighbors. After a number of rounds, every vertex must generate its own part of the output of the algorithm. For example, if the goal of the algorithm is to solve some LCL problem, then each vertex must eventually decide on its own label. The efficiency of such an algorithm is measured by the number of communication rounds required to produce the output. In particular, there are no restrictions on the complexity of the local computations that the vertices perform during each round (we imagine that the computational resources available to the vertices are infinite) or on the length of the messages that the vertices send to each other.

An important feature of the LOCAL model is that every vertex of G is executing *the same* algorithm. Therefore, to make this model nontrivial, there must be a way to distinguish the vertices from each other. There are two standard symmetry-breaking approaches, leading to the distinction between deterministic and randomized LOCAL algorithms:

- In the *deterministic* version of the LOCAL model, each vertex $x \in V(G)$ is assigned, as part of its input, an identifier $\mathrm{Id}(x)$, which is a string of $\Theta(\log n)$ bits. It is guaranteed that the identifiers assigned to different vertices are distinct. Subject to this guarantee, the algorithm must always output a correct solution to the problem, regardless of the specific assignment of the identifiers.
- In the *randomized* version of the LOCAL model, each vertex may generate an arbitrarily long finite sequence of independent uniformly distributed random bits. The algorithm may fail to produce a correct solution to the problem, but the probability of failure must not exceed $1/n$.

We remark that the randomized version of the model is more computationally powerful than the deterministic one. This is because a deterministic LOCAL algorithm can be simulated by a randomized one: each vertex can simply generate a random sequence of $\lceil 3 \log_2 n \rceil$ bits and use it as its identifier—the probability that two identifiers generated in this way coincide is easily seen to be less than $1/n$.

If x and y are two vertices whose graph distance in G (i.e., the length of the shortest xy-path) is T, then no information from y can reach x in fewer than T communication rounds (this explains the name "LOCAL" given to the model). Conversely, in T rounds every vertex can collect all the data present at the vertices at distance at most T from it. Thus, a T-round LOCAL algorithm may be construed simply as a function that, given the structure of the radius-T ball around x (including all the additional information, such as the assignment of the identifiers or random bit sequences to its vertices), decides on x's output. In other words, the complexity of a LOCAL algorithm measures how far in the graph an individual vertex should be allowed to "see" in order to be able to produce its own output. Naturally, if every vertex is allowed to "see" the entire graph, then the model trivializes. Thus, we are interested in algorithms that only allow each vertex to access a relatively small part of the graph.

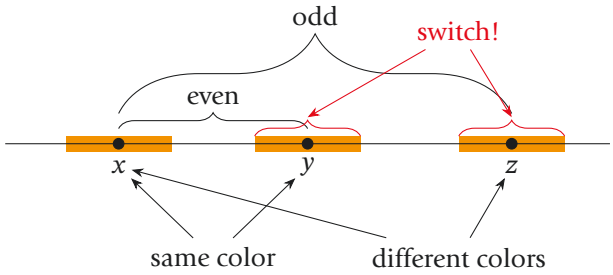

Figure 5. An illustration for the proof of Proposition 2.1. The three marked intervals are of length $2T = o(n)$.

As an illustration, let us consider a simple example. Suppose that G is an n-vertex path. Of course, $\chi(G) = 2$. How fast can a 2-coloring of G be computed by a LOCAL algorithm? Obviously, $O(n)$ rounds suffice (because in that many rounds each vertex will have access to the entire graph). On the other hand, it is fairly easy to show that $o(n)$ rounds are not enough:

Proposition 2.1. *There is no $o(n)$-round* LOCAL *algorithm (either deterministic or randomized) that finds a 2-coloring of an n-vertex path.*

Proof. We give an argument in the deterministic case (leaving the randomized case as a nice exercise to the reader). Suppose that there is a T-round deterministic LOCAL algorithm $\mathcal{A}$ for 2-coloring an n-vertex path, where $T \ll n$. We can pick three vertices x, y, z such that the distance between x and y is even, while the distance between x and z is odd, and, furthermore, all the pairwise distances between x, y, and z exceed $2T$ (see Fig. 5). Fix any assignment of identifiers and let $c(x)$, $c(y)$, $c(z)$ be the colors given by the algorithm $\mathcal{A}$ to x, y, and z respectively. In a 2-coloring of a path, the two colors alternate, so we must have $c(x) = c(y) \neq c(z)$. Without loss of generality, say $c(x) = c(y) = 1$ and $c(z) = 2$. Now let us exchange the assignments of the identifiers in the interval of length $2T$ centered around y with those in the interval centered around z and let $c'(x)$, $c'(y)$, $c'(z)$ be the colors given to x, y, and z by the algorithm $\mathcal{A}$ with this new assignment of identifiers. Since the output of $\mathcal{A}$ at a vertex is determined by the identifiers in the interval of length $2T$ centered around it, we have $c'(x) = c'(z) = 1$ and $c'(y) = 2$. This is a contradiction since the colors of x and y must be the same, while the colors of x and z must be different. □

2.2. **A sample of results.** The first hint of the connection between distributed algorithms and descriptive combinatorics can be obtained by comparing some known results in the two areas. For example, consider the following problem:

> *Fix a natural number $d \in \mathbb{N}$. What is the minimum $k = k(d)$ such that a k-coloring of a graph of maximum degree d can be found efficiently?*

Here the *maximum degree* of a graph is the maximum number of neighbors of a vertex. If we interpret the word "efficiently" to mean "with a fast LOCAL algorithm," then we have the following result:

Theorem 2.2 (Goldberg–Plotkin–Shannon [BE13, §3]). *There exists a deterministic* LOCAL *algorithm that finds a $(d+1)$-coloring of an n-vertex graph of maximum degree d in $O(\log^* n)$ rounds.*[5]

In the statement of Theorem 2.2, $\log^* n$ is the *iterated logarithm* of n, i.e., the number of times the logarithm function must be iteratively applied to n before the result becomes at most 1. This is an extremely slow-growing function. In particular—and most importantly for us—it is asymptotically of order $o(\log n)$. In a graph of maximum degree d, a vertex can "see" at most d^T other vertices in T rounds of the LOCAL model. Therefore, if $T = o(\log n)$, a vertex will definitely not have access to the entire graph. In this sense, $o(\log n)$ rounds is a natural threshold for "truly local" algorithms.

Can we reduce the number of colors to d? An obvious obstacle is that G may contain a clique on $d+1$ vertices, all of which would have to receive different colors. On the other hand, the so-called Brooks's theorem in graph theory asserts that if $d \geq 3$ and G is a graph of maximum degree d without a $(d+1)$-clique, then $\chi(G) \leq d$, i.e., a d-coloring of G exists. Nevertheless, it turns out that no efficient deterministic LOCAL algorithm can find such a coloring, even if G has no cycles:

Theorem 2.3 (Chang–Kopelowitz–Pettie [CKP19]). *There is no deterministic* LOCAL *algorithm that finds a d-coloring of an n-vertex acyclic graph of maximum degree d in $o(\log n)$ rounds.*

In contrast to Theorem 2.3, it is possible to reduce the number of colors using a randomized algorithm:

Theorem 2.4 (Ghaffari–Hirvonen–Kuhn–Maus [Gha+18]). *There exists a randomized* LOCAL *algorithm that finds a d-coloring of an n-vertex graph G of maximum degree d in $O((\log\log n)^2)$ rounds, provided that $d \geq 3$ and G has no $(d+1)$-cliques.*

[5] *Here and in the sequel we treat d as a constant, meaning that the implied factors in the $O(\cdot)$ notation may depend on d.*

Now let us turn to the descriptive combinatorics side of the picture. How many colors do we need for a Borel coloring of a Borel graph of maximum degree d? The first result parallels Theorem 2.2:

Theorem 2.5 (Kechris–Solecki–Todorcevic [KST99]). *If G is a Borel graph of maximum degree d, then $\chi_{\mathsf{B}}(G) \leq d+1$.*

On the other hand, reducing the number of colors to d is impossible even for graphs with no cycles:

Theorem 2.6 (Marks [Mar16]). *For any $d \in \mathbb{N}$, there is an acyclic Borel graph G of maximum degree d such that $\chi_{\mathsf{B}}(G) > d$.*

However, if we only want a measurable or Baire-measurable coloring, then d colors suffice:

Theorem 2.7 (Conley–Marks–Tucker-Drob [CMTD16]). *If G is a Borel graph of maximum degree d, then both $\chi_{\mathsf{M}}(G)$ and $\chi_{\mathsf{BM}}(G)$ are at most d, provided that $d \geq 3$ and G has no $(d+1)$-cliques.*

Theorems 2.2–2.4 and 2.5–2.7 clearly mirror each other. It seems that deterministic LOCAL algorithms somehow correspond to Borel constructions, while randomized algorithms—to measurable and Baire-measurable ones. It turns out that, indeed, this is not a coincidence and one can prove some general connections of this form.

2.3. **Efficient distributed algorithms yield descriptive results.** In [Ber20], the following general result is established:

Theorem 2.8 (Bernshteyn [Ber20]). *Let $\mathbb{G}$ be a class of finite graphs closed under adding isolated vertices and let G be a Borel graph of bounded degree all of whose finite induced subgraphs are in $\mathbb{G}$. Fix an LCL problem Π.*

(a) *If Π can be solved on n-vertex graphs from $\mathbb{G}$ by an $o(\log n)$-round deterministic* LOCAL *algorithm, then G admits a Borel solution to Π.*

(b) *If Π can be solved on n-vertex graphs from $\mathbb{G}$ by an $o(\log n)$-round randomized* LOCAL *algorithm, then G admits both a measurable and a Baire-measurable solution to Π.*

In short, efficient deterministic algorithms yield Borel solutions, while efficient randomized algorithms yield measurable and Baire-measurable solutions. The following implications are special cases of Theorem 2.8:

$$\text{Theorem 2.2} \implies \text{Theorem 2.5},$$
$$\text{Theorem 2.4} \implies \text{Theorem 2.7},$$
$$\text{Theorem 2.6} \implies \text{Theorem 2.3}.$$

In addition to its theoretical significance, Theorem 2.8 has a number of applications to specific problems. For instance, if we already know an efficient LOCAL algorithm for some LCL problem, we can then use it to derive corresponding results in descriptive combinatorics. Several

instances of this are given in [Ber20]. For example, Theorem 2.8, together with known distributed algorithms due to Chung, Pettie, and Su, is used in [Ber20, §3.B] to obtain asymptotically optimal upper bounds on the measurable chromatic number of Borel graphs without short cycles. In the opposite direction, impossibility results in descriptive combinatorics yield lower bounds on the running time of distributed algorithms. In a recent paper [Bra+21], Brandt et al. used this idea to derive new lower bounds for distributed algorithms via a version of the so-called determinacy method that was originally developed by Marks to prove Theorem 2.6.

Let us now say a few words about the proof of Theorem 2.8. The proof of part (a) is actually not too difficult:

Proof of Theorem 2.8(a) (sketch). Suppose that Π is solved on n-vertex graphs from $\mathbb{G}$ by an $o(\log n)$-round deterministic LOCAL algorithm $\mathcal{A}$. The idea is to simulate the execution of the algorithm $\mathcal{A}$ on the given Borel graph G by "pretending" that G is finite.

Let d be the maximum degree of G (which is finite by assumption). Take a very large natural number n and let $T \ll \log n$ be the running time of $\mathcal{A}$ on n-vertex graphs. Let S be the set of all bit strings of length $\lceil \log_2 n \rceil$.

Claim. There is a Borel labeling $\mathrm{Id} : V(G) \to S$ such that $\mathrm{Id}(x) \neq \mathrm{Id}(y)$ whenever $x \neq y$ and the graph distance between x and y is at most $2T + 2$.

Proof of the claim. Define an auxiliary graph G' with the same vertex set as G where distinct vertices x and y are adjacent if and only if their graph distance in G is at most $2T + 2$. It is not hard to see that the graph G' is Borel. Since $|S| \geq n$, we just need to find a Borel n-coloring of G'. The maximum degree of G' is at most d^{2T+2}, which is less than n since $2T + 2 \ll \log n$. Therefore, by Theorem 2.5, G' has a Borel n-coloring, as desired. ⊣

We treat the bit string $\mathrm{Id}(x)$ as a substitute for an identifier of x. By construction, although there may be other vertices with the same identifier, they are far from x in G. In particular, if x explores its radius-T ball in G, it will only see distinct identifiers. Since the algorithm $\mathcal{A}$ applied on n-vertex graphs doesn't allow a vertex to see outside its radius-T ball, we may run $\mathcal{A}$ on G for T rounds (as if G had n vertices) using the function Id in place of the identifier assignment. The output labeling will then be a solution to Π, and, since Id is a Borel assignment, one can show that the output will also be Borel. □

Arguments similar to the above proof sketch are common in distributed computing theory. For example, an analogous approach was used by Chang, Kopelowitz, and Pettie in [CKP19] to prove that no LCL problem can have deterministic LOCAL complexity in the interval between $\omega(\log^* n)$ and $o(\log n)$.

The proof of Theorem 2.8(b) is quite a bit more involved, and we won't venture to explain it here. Let us just mention one interesting feature of it. The key tool used to prove Theorem 2.8(b) is the measurable version of the Local Lemma, which is also established in [Ber20]. The Local Lemma is a well-known powerful tool in probabilistic combinatorics that is often used to show that a given LCL problem has a solution. The measurable version of this lemma additionally guarantees that the solution is measurable. And here we encounter another point of interaction between descriptive combinatorics and distributed computing: the proof of the measurable Local Lemma in [Ber20] crucially relies on the efficient randomized LOCAL algorithm for the Local Lemma developed by Fischer and Ghaffari [FG17].

2.4. Perfect harmony between the two worlds: the case of continuous solutions.

Theorem 2.8 allows one to turn efficient distributed algorithms into results in descriptive combinatorics. It is natural to ask if some sort of converse to Theorem 2.8 also holds, i.e., if we can obtain efficient algorithms from descriptive results. While in general this question remains widely open, there is one salient case in which an exact correspondence between the descriptive and the distributed worlds has been established. This case concerns continuous solutions to LCL problems.

Definition 2.9 (Continuous solutions to LCL problems). Let Π be an LCL problem with label set Λ and let G be a Borel graph. A labeling of the vertices of G by the labels from Λ is a *continuous solution* to Π if it fulfills all the constraints of the problem Π and, additionally, for each label $\lambda \in \Lambda$, the set of all vertices labeled λ is open.

Note that the set of all vertices receiving a given label in a continuous solution must be both open and closed—*clopen*, in short. Hence, it is only interesting to study continuous solutions to LCL problems on graphs G such that the space $V(G)$ has many clopen subsets. Specifically, we focus on *zero-dimensional* spaces, i.e., spaces whose topology is generated by clopen sets. Even though familiar spaces such as $\mathbb{R}$ do not have any nontrivial clopen sets, zero-dimensional spaces are rather common. For example, the space $\{0,1\}^{\mathbb{N}}$ is zero-dimensional. Another example of a zero-dimensional Polish space is $\mathbb{R} \setminus \mathbb{Q}$. Also, given any Polish space X, it is possible to refine the topology on X to make it zero-dimensional without changing the Borel sets.

The main result we want to describe in this section says, roughly, that an LCL problem can be solved continuously if and only if it can be solved via an efficient deterministic distributed algorithm. To make this statement precise, we need to specify a particular graph or a class of graphs on which we attempt to solve the problem. While it is possible to make the statement somewhat more general, it will be convenient to consider classes of graphs that look

very symmetric, such as regular trees or multi-dimensional grids. The following general definition includes these examples as special cases:

Definition 2.10 (Γ-graphs)**.** Let Γ be a group generated by a fixed finite symmetric[6] set $S \subseteq \Gamma$.

The *Cayley graph* Cay(Γ) of Γ corresponding to S is the graph with vertex set Γ that includes, for every $\gamma \in \Gamma$ and $\sigma \in S$, an edge from γ to $\sigma\gamma$. To be precise, we view Cay(Γ) as an edge-labeled graph, where the label of the edge $(\gamma, \sigma\gamma)$ is σ, but the reader may safely ignore this technicality.

A Γ*-graph* is a graph (with edge labels) in which every connected component is isomorphic to Cay(Γ).

For instance, the Cayley graph of the additive group $\mathbb{Z}$ with generating set $S = \{1, -1\}$ is a bi-infinite path. Similarly, the Cayley graph of $\mathbb{Z}^d$ with respect to an appropriately chosen generating set is an infinite d-dimensional square grid. On the other hand, the Cayley graph of the free group $\mathbb{F}_d$ of rank d is an infinite d-regular tree.

Given any group Γ with a finite symmetric generating set S, there is a canonical way to define a particular Γ-graph $\mathbf{S}_\Gamma$ on a zero-dimensional vertex set, called the *shift graph* of Γ. The shift graph $\mathbf{S}_\Gamma$ is an extremely important example that plays a central role not only in descriptive combinatorics, but also in such areas as ergodic theory and topological dynamics. Before giving the general definition, it will be instructive to consider the special case $\Gamma = \mathbb{Z}$, with generating set $S = \{1, -1\}$. For a subset $A \subseteq \mathbb{Z}$ and an element $n \in \mathbb{Z}$, we write

$$A + n \coloneqq \{m + n \;:\; m \in A\}.$$

The set $A + n$ can naturally be seen as a "shift" of A by n (hence the term "shift graph"). We say that a set $A \subseteq \mathbb{Z}$ is *aperiodic* if $A + n \neq A$ for every non-zero $n \in \mathbb{Z}$. This is equivalent to saying that the characteristic function $\mathbf{1}_A$ of A is not periodic, i.e., there is no integer $n \neq 0$ such that $\mathbf{1}_A(m) = \mathbf{1}_A(m + n)$ for all $m \in \mathbb{Z}$. The shift graph $\mathbf{S}_\mathbb{Z}$ is the graph whose vertices are the aperiodic subsets of $\mathbb{Z}$ and in which every aperiodic set A has two neighbors: $A + 1$ and $A - 1$. Thus, the connected component of A in $\mathbf{S}_\mathbb{Z}$ is the following infinite path:

$$\cdots \text{—} (A-2) \text{—} (A-1) \text{—} (A) \text{—} (A+1) \text{—} (A+2) \text{—} \cdots$$

(Since A is aperiodic, all the vertices on this path are distinct.) In particular, every component of $\mathbf{S}_\mathbb{Z}$ is isomorphic to the Cayley graph of $\mathbb{Z}$, so it is indeed a $\mathbb{Z}$-graph.

Now for the general construction. Let Γ be a group with a finite symmetric generating set S. For a set $A \subseteq \Gamma$ and an element $\gamma \in \Gamma$, let

$$\gamma A \coloneqq \{\gamma\alpha \;:\; \alpha \in A\}.$$

A subset $A \subseteq \Gamma$ is *aperiodic* if $\gamma A \neq A$ for all non-identity elements $\gamma \in \Gamma$. The vertices of $\mathbf{S}_\Gamma$ are the aperiodic subsets $A \subseteq \Gamma$. For every $A \in V(\mathbf{S}_\Gamma)$ and $\sigma \in S$, $\mathbf{S}_\Gamma$ includes an edge from A to σA labeled σ. By construction, for each $A \in V(\mathbf{S}_\Gamma)$, the mapping $\gamma \mapsto \gamma A$ establishes an isomorphism between Cay(Γ) and the connected component of A in $\mathbf{S}_\Gamma$ (the fact that A is aperiodic guarantees that this mapping is injective). Thus, $\mathbf{S}_\Gamma$ is a Γ-graph. Furthermore, by identifying each subset of Γ with its characteristic function, we can view $V(\mathbf{S}_\Gamma)$ as a subset of $\{0,1\}^\Gamma$, which makes $\mathbf{S}_\Gamma$ a Borel graph on a zero-dimensional Polish space.

We now have the following result, obtained independently by Bernshteyn [Ber21] and Seward (unpublished):

Theorem 2.11 (Bernshteyn [Ber21]/Seward)**.** *Let Γ be a group generated by a finite symmetric set $S \subseteq \Gamma$. For every LCL problem Π, the following statements are equivalent:*

(i) $\mathbf{S}_\Gamma$ *admits a continuous solution to* Π.
(ii) *There is an $o(\log n)$-round deterministic* LOCAL *algorithm that solves* Π *on n-vertex subgraphs of the Cayley graph* Cay(Γ).

The implication (ii) $\Longrightarrow$ (i) of Theorem 2.11 is proved in a manner analogous to the proof of Theorem 2.8(a) sketched in §2.3. The implication (i) $\Longrightarrow$ (ii) is significantly more difficult and involves a careful analysis of the structure of continuous functions on $V(\mathbf{S}_\Gamma)$. Interestingly, this analysis again relies, among other things, on tools from computer science such as the so-called method of conditional probabilities. It also builds on earlier work of Gao–Jackson–Seward [GJS16], Seward–Tucker-Drob [STD16], and Elek [Ele18].

Theorem 2.11 explains the analogies between known results in continuous combinatorics and distributed computing. For example, in the case $\Gamma = \mathbb{Z}^d$, the continuous combinatorics of the shift graph have been studied in detail by Gao et al. [Gao+18]. Similarly, Brandt et al. [Bra+17] investigated the LCL problems that can be solved on d-dimensional square grids by efficient distributed algorithms. The results obtained by these two groups of researchers—independently and using different methods—perfectly parallel each other, exactly as Theorem 2.11 predicts.

2.5. Baire-measurable solutions to LCL problems on trees.

The last result we would like to mention is a very recent theorem that establishes a perfect equivalence between yet another pair of facets of the worlds of descriptive combinatorics and distributed computing:

Theorem 2.12 (Brandt et al. [Bra+21])**.** *Fix $d \in \mathbb{N}$. For every LCL problem Π, the following statements are equivalent:*

(i) *Every Borel graph G in which every component is an infinite d-regular tree admits a Baire-measurable solution to* Π.
(ii) *There is an $O(\log n)$-round deterministic* LOCAL *algorithm that solves* Π *on n-vertex trees of maximum degree d.*

[6] *A subset $S \subseteq \Gamma$ is* symmetric *if for all $\gamma \in S$, $\gamma^{-1} \in S$ as well.*

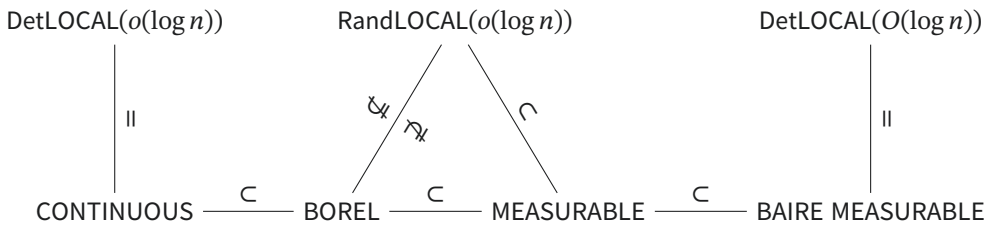

Figure 6. Known equalities and inequalities between classes of LCL problems on d-regular trees. Here DetLOCAL($T(n)$) and RandLOCAL($T(n)$) denote the classes of LCL problems solvable by a $T(n)$-round determinisitic (respectively, randomized) LOCAL algorithm. The symbol "$\subset$" indicates strict inclusion.

This is a truly inspiring result, which highlights how deep the connections between descriptive combinatorics and distributed computing lie. What makes it different from Theorems 2.8 and 2.11 is that the complexity bound on the distributed algorithm here is $O(\log n)$, not $o(\log n)$. Note that in $O(\log n)$ rounds, a vertex of an n-vertex tree of maximum degree $d \geq 3$ is guaranteed to discover a vertex of degree less than d—and hence this algorithm cannot be directly used on a d-regular graph G. Unsurprisingly, the proof of the implication (ii) $\Longrightarrow$ (i) in Theorem 2.12 is considerably more difficult than the proof of Theorem 2.8(a). In fact, both implications in Theorem 2.12 are highly nontrivial. The high-level idea of the argument is to isolate a certain combinatorial property of LCL problems and show that this property is equivalent to each of (i), (ii) separately (the former equivalence was established by Bernshteyn (unpublished), the latter—by Brandt *et al.* [Bra+21]).

2.6. **Final thoughts and open problems.** The intimate interactions between descriptive combinatorics and distributed computing have only very recently been discovered, and most questions about them remain open. Perhaps the central open problem is this:

Question 2.13. Is there a version of Theorem 2.11 for Borel solutions to LCL problems? In other words, is there a precise algorithmic counterpart to the notion of a Borel solution (similar to how $o(\log n)$-round LOCAL algorithms are equivalent to continuous solutions)?

The same questions for measurable and (except for graphs with no cycles) Baire-measurable solutions are also open. Even on graphs as simple as d-regular trees, many open questions remain. The diagram in Fig. 6 summarizes our knowledge regarding the complexity of LCL problems on d-regular trees. In particular, we know that there are LCL problems that admit Borel solutions but cannot be solved by an $o(\log n)$-round randomized algorithm, but there are also LCL problems that can be solved by such an algorithm but do not admit Borel solutions. Therefore, a new algorithmic framework seems to be needed to capture the notion of a Borel solution precisely.

On classes of graphs other than trees, our knowledge is even scanter. For example, the following basic question is open:

Question 2.14. Fix an integer $d \geq 2$. Does there exist an LCL problem Π such that the shift graph $\mathbf{S}_{\mathbb{Z}^d}$ admits a measurable solution to Π but not a Borel one? What about Baire-measurable solutions?

Another area with close ties to distributed computing and descriptive combinatorics is the study of so-called *finitary factors of i.i.d. processes*. This is a class of particularly well-behaved random processes on networks of great interest in probability theory. Finitary factors of i.i.d. processes can be used to create a complexity hierarchy of LCL problems, and Grebík and Rozhoň recently discovered that in many cases this hierarchy parallels the one arising from considering descriptive and LOCAL complexity. For more details on this exciting connection, see the papers [GR21] by Grebík and Rozhoň and [Bra+21] by Brandt et al.

In spite of all the open problems mentioned above, it really does appear that the combination of tools from descriptive set theory and distributed computing is the key to attaining a deep understanding of the behavior of LCL problems under various "effectiveness" requirements. Exciting discoveries await, and they will surely enrich descriptive combinatorics and distributed computing alike.

ACKNOWLEDGMENT. I am very grateful to the anonymous referees for carefully reading this article and providing many helpful suggestions.

References

[BE13] L. Barenboim and M. Elkin, *Distributed graph coloring: Fundamentals and recent developments*, Synthesis Lectures on Distributed Computing Theory, vol. 11, Morgan & Claypool Publishers, Williston, VT, 2013. MR3184899

[Ber20] A. Bernshteyn, *Distributed algorithms, the Lovász local lemma, and descriptive combinatorics*, 2020, https://arxiv.org/abs/2004.04905.

[Ber21] A. Bernshteyn, *Probabilistic constructions in continuous combinatorics and a bridge to distributed algorithms*, 2021, https://arxiv.org/abs/2102.08797.

[Bra+21] S. Brandt, Y.-J. Chang, J. Grebík, C. Grunau, V. Rozhoň, and Z. Vidnyánszky, *Local problems on trees from the perspectives of distributed algorithms, finitary factors, and descriptive combinatorics*, 2021, https://arxiv.org/abs/2106.02066.

[Bra+17] S. Brandt, J. Hirvonen, J. H. Korhonen, T. Lempiäinen, P. R. J. Östergård, C. Purcell, J. Rybicki, J. Suomela, and P. Uznański, LCL *problems on grids*, ACM Symposium on Principles of Distributed Computing (PODC), 2017, 101–110. Full version: https://arxiv.org/abs/1702.05456.

[CKP19] Y.-J. Chang, T. Kopelowitz, and S. Pettie, *An exponential separation between randomized and deterministic complexity in the local model*, SIAM J. Comput. **48** (2019), no. 1, 122–143, DOI 10.1137/17M1117537. MR3904438

[CMTD16] C. T. Conley, A. S. Marks, and R. D. Tucker-Drob, *Brooks' theorem for measurable colorings*, Forum Math. Sigma **4** (2016), Paper No. e16, 23, DOI 10.1017/fms.2016.14. MR3514902

[Ele18] G. Elek, *Qualitative graph limit theory. Cantor dynamical systems and constant-time distributed algorithms*, 2018, https://arxiv.org/abs/1812.07511.

[FG17] M. Fischer and M. Ghaffari, *Sublogarithmic distributed algorithms for Lovász local lemma, and the complexity hierarchy*, 31 International Symposium on Distributed Computing, LIPIcs. Leibniz Int. Proc. Inform., vol. 91, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2017, pp. Art. No. 18, 16. MR3746281

[Gao+18] S. Gao, S. Jackson, E. Krohne, and B. Seward, *Continuous combinatorics of abelian group actions*, 2018, https://arxiv.org/abs/1803.03872.

[GJS16] S. Gao, S. Jackson, and B. Seward, *Group colorings and Bernoulli subflows*, Mem. Amer. Math. Soc. **241** (2016), no. 1141, vi+241, DOI 10.1090/memo/1141. MR3478758

[Gha+18] M. Ghaffari, J. Hirvonen, F. Kuhn, and Y. Maus, *Improved distributed Δ-coloring*, ACM Symposium on Principles of Distributed Computing (PODC), 2018, 427–436. Full version: https://arxiv.org/abs/1803.03248.

[GR21] J. Grebík and V. Rozhoň, *Local problems on grids from the perspective of distributed algorithms, finitary factors, and descriptive combinatorics*, 2021, https://arxiv.org/abs/2103.08394.

[KM20] A. S. Kechris and A. S. Marks, *Descriptive graph combinatorics*, 2020, http://www.math.caltech.edu/~kechris/papers/combinatorics20book.pdf.

[KST99] A. S. Kechris, S. Solecki, and S. Todorcevic, *Borel chromatic numbers*, Adv. Math. **141** (1999), no. 1, 1–44, DOI 10.1006/aima.1998.1771. MR1667145

[Lin92] N. Linial, *Locality in distributed graph algorithms*, SIAM J. Comput. **21** (1992), no. 1, 193–201, DOI 10.1137/0221015. MR1148825

[Mar16] A. S. Marks, *A determinacy approach to Borel combinatorics*, J. Amer. Math. Soc. **29** (2016), no. 2, 579–600, DOI 10.1090/jams/836. MR3454384

[Oxt80] J. C. Oxtoby, *Measure and category*, 2nd ed., Graduate Texts in Mathematics, vol. 2, Springer-Verlag, New York-Berlin, 1980. A survey of the analogies between topological and measure spaces. MR584443

[Pik21] O. Pikhurko, *Borel combinatorics of locally finite graphs*, Surveys in combinatorics 2021, London Math. Soc. Lecture Note Ser., vol. 470, Cambridge Univ. Press, Cambridge, 2021, pp. 267–319. MR4273433

[STD16] B. Seward and R. D. Tucker-Drob, *Borel structurability on the 2-shift of a countable group*, Ann. Pure Appl. Logic **167** (2016), no. 1, 1–21, DOI 10.1016/j.apal.2015.07.005. MR3413491

Anton Bernshteyn

Credits

Opener is courtesy of enot-poloskun via Getty.

Figures 1–5 and author photo are courtesy of Anton Bernshteyn.