An Introduction to Borel Reducibility for Countable Structures

Matthew Harrison-Trainor

University of Illinois Chicago

CiE 2024 Tutorial

Invariants

Let X be some class of mathematical objects with a notion of equivalence E, such as:

- Groups up to isomorphism
- Topological spaces up to homeomorphism
- Linear orders up to bi-embeddability

One of the major goals up mathematics is to classify these objects, often by giving complete invariants.

To each object $x \in X$, we want to assign an invariant $f(x)$. The invariants should tell us when two objects are equivalent:

$$x \, E \, y \quad \Longleftrightarrow \quad f(x) = f(y).$$

Example

Vector spaces V are classified by the invariant $\dim(V)$.

Invariants are not always numbers; they can also be groups or some other kind of structure.

If the invariants are groups, then instead of checking whether the invariants are equal, we should check whether they are isomorphic.

In general, we also have an equivalence relation on the space of invariants.

Example

Consider the rank one torsion-free abelian group, i.e., subgroups of the additive group $\mathbb{Q}$.

Baer (1937) gave invariants as follows.

Given $a \in G$, the p-height $h_p(a)$ of a is the highest power n of p dividing a:

$$\exists x \; p^n x = a.$$

The p-height may be ∞.

The height sequence of a is

$$h(a) = (h_2(a), h_3(a), h_5(a), \ldots).$$

The height sequence of a is

$$h(a) = (h_2(a), h_3(a), h_5(a), \ldots).$$

Two height sequences are equivalent if they are ∞ in the same places and differ in at most finitely many other places.

Fact: All elements of G have equivalent height sequences.

Call this equivalence class the type of G. It is a complete invariant.

Consider the groups

- $\mathbb{Z}$, picking 1, has type $(0, 0, 0, 0, 0, \ldots)$.
- $\mathbb{Z}$, picking 12, has type $(2, 1, 0, 0, 0, 0, 0, \ldots)$.
- $\mathbb{Z}[\frac{1}{2}]$, picking 1, has type $(1, 0, 0, 0, 0, \ldots)$.

These are all isomorphic.

Consider the groups

- $\mathbb{Z}[1/p : p \text{ prime}]$, picking 1, has type $(1, 1, 1, 1, \ldots)$.
- $\mathbb{Z}[1/2^n : n \in \mathbb{N}]$, picking 1, has type $(\infty, 0, 0, 0, 0, 0, \ldots)$.

These are not isomorphic to each other or to $\mathbb{Z}$.

Given (X, E), a complete set of invariants for (X, E) is an equivalence relation (Y, F) and a function f such that

$$x \, E \, y \iff f(x) \, F \, f(y).$$

For invariants to be useful, two things should be true:

- It is easy to compute the invariants $f(x)$.
- It is easy to tell when two invariants $f(x)$ and $f(y)$ are equivalent.

We will think of "easy to compute" as meaning Borel.

Why Borel and not computable?

- Some invariants which we think of as easy, such as the dimension of a vector space, are Borel but not computable.
- If we show that there are no good Borel invariants, this is stronger than showing that there are no good computable invariants.
- There is a reasonable analogy

$$\text{polynomial time on finite structures} \equiv \text{Borel on countable structures.}$$

What does Borel mean?

(You will not lose much in the rest of the tutorial if you think of a Borel function as a reasonably constructive function.)

Definition

A Polish space is a separable completely metrizable topological space.

Example

$\mathbb{N} = \omega$.

Example

$2^\omega = \{0, 1\}^\mathbb{N}$, the infinite binary strings.

Example

$\omega^\omega = \{0, 1, 2, \ldots\}^\mathbb{N}$, the infinite strings of natural numbers.

Let $\mathcal{L}$ be a signature / language.

Definition

$Mod(\mathcal{L})$ is the Polish space of countable $\mathcal{L}$-structures.

Formally:

- We consider $\mathcal{L}$-structures with domain $\mathbb{N}$.
- The basic clopen sets are determined by atomic and negated atomic formulas, e.g.,

$$[R(n_1, \ldots, n_\ell)] = \{\mathcal{A} : \mathcal{A} \vDash R(n_1, \ldots, n_\ell)\}$$

for $n_1, \ldots, n_\ell \in \mathbb{N}$.

Continuous functions on $Mod(\mathcal{L})$ are analogous to computable functions. (In fact every computable function is continuous but not vice versa.)

E.g., consider a function $f\colon Mod(\mathcal{L}) \to 2^\omega$.

Then f is continuous if for every finite string σ, there is a (possibly infinite) sequence of atomic or negated atomic formulas $\varphi_i(n_1, \ldots, n_{\ell_i})$ such that

$$f(\mathcal{A}) \text{ extends } \sigma \iff \text{for some } i,\ \mathcal{A} \vDash \varphi_i(n_1, \ldots, n_{\ell_i}).$$

That is, the function f, when it reads $\mathcal{A}$ as input and sees that $\mathcal{A}$ satisfies some φ_i, has output beginning with σ.

Definition

Let X be a Polish space. The Borel sets are the smallest subsets of X containing all open sets and closed under countable unions, countable intersections, and relative complements.

The Borel sets are those that can be built using quantification over elements of X but not over higher-order objects like subsets of X.

Example

In 2^ω, the set of binary strings with infinitely many 0's and infinitely many 1's is Borel.

Definition

A function $f\colon X \to Y$ is Borel if for every open set $U \subseteq Y$, $f^{-1}(U)$ is Borel.

Any computable function is continuous, and every continuous function is Borel.

You will not lose much in the rest of the tutorial if you think of a Borel function as a reasonably constructive functions.

Definition

Let (X, E) and (Y, F) be Polish spaces with equivalence relations. E is Borel reducible to F, written $E \leq_B F$, if there is a Borel function $g: X \to Y$ such that

$$xEy \iff g(x)\ F\ g(y).$$

This means that F is at least as complicated as E:

- If we know how to tell if elements of Y are F-equivalent, then we can tell if elements of X are E-equivalent.
- Any Borel invariants for F would also be Borel invariants for E.

Isomorphism is **not** a Borel equivalence relation. It is an analytic equivalence relation:

$$\mathcal{A} \cong \mathcal{B} \iff \exists f : f \text{ is an isomorphism } \mathcal{A} \to \mathcal{B}.$$

Definition

Let X be a Polish space. A set is analytic if it is the projection of a Borel set.

In descriptive set theory, there is an entire field of studying equivalence relations up to Borel reducibility, particularly when the equivalence relations are Borel.

In this tutorial, we are going to be concerned with the specific problem of classifying isomorphism among countable structures of particular kinds.

Borel reducibility for countable structures

In a seminal paper *A Borel reducibility theory for classes of countable structures* (1989), H. Friedman and Stanley introduced the Borel reduction between classes of structures.

Definition

Suppose $K \subseteq Mod(\mathcal{L})$ and $K' \subseteq Mod(\mathcal{L}')$ are closed under isomorphism. We say that K is Borel embeddable in K' if $(K, \cong)$ is Borel reducible to $(K', \cong)$, i.e., there is a Borel function $\Phi: K \to K'$ such that for $\mathcal{A}, \mathcal{B} \in K$,

$$\mathcal{A} \cong \mathcal{B} \Longleftrightarrow \Phi(\mathcal{A}) \cong \Phi(\mathcal{B}).$$

We write $K \leq_B K'$.

Any invariants that could be used to classify K' could be used to classify K. K' is at least as hard to classify as K.

Example

The class of $\mathbb{Q}$-vector spaces is Borel reducible to the class of subgroups of $\mathbb{Q}$, but not vice versa.

Let $p_0, p_1, p_2, \ldots$ list the primes. We get a Borel reduction

$$V \quad \longmapsto \quad \mathbb{Z}\left[\frac{1}{p_{\dim(V)}^{n}} : n \in \mathbb{N}\right].$$

It is a little harder to show that there is no Borel reduction in the other direction.

There are classes of structures which are as complicated as possible.

Definition

A class of structures $\mathcal{C}$ is Borel complete if for every other class $\mathcal{D}$ and language $\mathcal{L}$, $\mathcal{D} \leq_B \mathcal{C}$.

In practice, the Borel reduction is very often a computable reduction.

Theorem

The class of graphs is Borel complete.

Let's see why.

Theorem

The class of graphs is Borel complete.

Fix a language $\mathcal{L}$, say, for simplicity, $\mathcal{L}$ has a binary function symbol f.

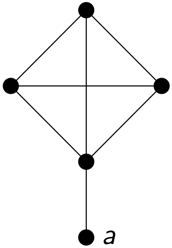
For each $\mathcal{L}$-structure $\mathcal{A}$, we must produce a graph $\Phi(\mathcal{A})$ such that

$$\mathcal{A} \cong \mathcal{B} \Longleftrightarrow \Phi(\mathcal{A}) \cong \Phi(\mathcal{B}).$$

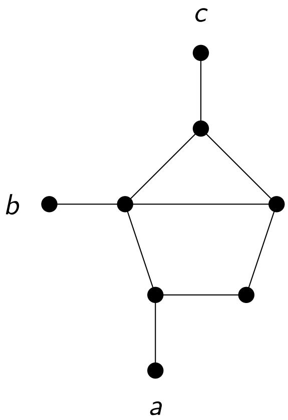
The structure $\mathcal{A}$ will be somehow embedded into the graph $\Phi(\mathcal{A})$.

We will have a bunch of vertices of the graph $\Phi(\mathcal{A})$ representing the elements of $\mathcal{A}$. Pick a "widget" that we will use to identify these vertices to distinguish them from all of the other vertices.

For example, we can identify a vertex a as representing an element of a if it has a 4-clique attached to it:



Given $a, b, c \in \mathcal{A}$, we represent that $f(a, b) = c$ by attaching a widget to the vertices representing a, b, c, e.g.:



Remember that each of these three vertices also has a 4-clique attached to them representing that they are elements of $\mathcal{A}$.

It is important that none of these widgets embeds in any other, so that we do not confuse part of one widget for part of another.

Given a copy of $\Phi(\mathcal{A})$, we can recover a copy of $\mathcal{A}$:

- First, search for the vertices which have a 4-clique attached to them. These vertices represent to domain of the copy of $\mathcal{A}$.
- Now we must determine f. For every two vertices a, b as determined above, look for a vertex c (also with a 4-clique) and a widget representing that $f(a, b) = c$.

If we choose the widgets properly, this will yield an isomorphic copy of $\mathcal{A}$.

If there are more functions or relations, we need to include more widgets. For a relation R, we also need to add a widget to represent the negation.

Theorem

The class of graphs is Borel complete.

Now to show that another class is Borel complete, you just need to reduce graphs (or your favourite Borel-complete class) to it.

Fact

There are analytic equivalence relations which are not Borel reducible to isomorphism of graphs:

- *Bi-embeddability of graphs (Louveau and Rosendal).*
- *Homeomorphism of compact topological spaces (Zielinski).*

Scott's isomorphism theorem says that for each particular countable structure $\mathcal{A}$, the set

$$\{\mathcal{B} \mid \mathcal{A} \cong \mathcal{B}\}$$

of countable structures isomorphic to $\mathcal{A}$ is Borel.

This means that every $\cong$-equivalence class is Borel. This is not true for general analytic equivalence relations.

What other classes of structures are Borel complete?

- partial orders
- rings
- integral domains
- 2-step nilpotent groups

These are all proved by codings graphs into the given class, possibly indirectly.

Theorem

Partial orders are Borel complete.

Let $G = (V, E)$ be a graph.

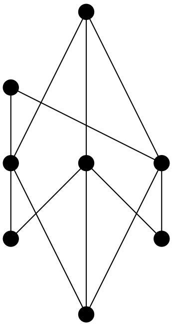
Define a partial order $(P, \leq) = P(G)$ with domain:

$$\{a, b\} \cup V \cup \{x_{\{u,v\}} : u, v \in V\}.$$

Put

- a is greater than every v and b is less than every v;
- the v are incomparable;
- $x_{\{u,v\}} < u, v$ if $(u, v) \in E$, and $x_{\{u,v\}} > u, v$ if $(u, v) \notin E$.

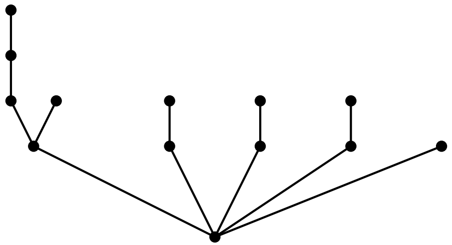
transforms into



Trees and linear orders

Some classes of structures require a slightly different strategy. The classical examples are linear orders and trees.

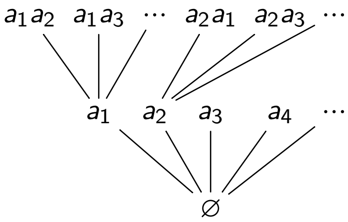
We'll start with coloured trees. For us, a tree has a distinguished root node, and there is a binary relation "x is a child of y":



Definition

The tree of tuples of a structure $\mathcal{A}$, $\mathcal{T}(\mathcal{A})$, is a labeled tree consisting of all the tuples from $\mathcal{A}$ ordered by extension where each tuple $\bar{a}$ is labeled by a number coding its finite atomic diagram $D_{\mathcal{A}}(\bar{a})$.

Let $\mathcal{A}$ have elements $a_1, a_2, a_3, \ldots$. Then $\mathcal{T}(\mathcal{A})$ looks like:



We also have the labels on the vertices which specify the relations holding of the corresponding tuple.

It is clear that

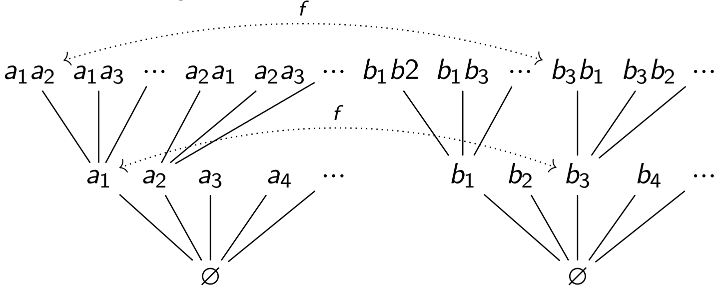
$$\mathcal{A} \cong \mathcal{B} \Longrightarrow \mathcal{T}(\mathcal{A}) \cong \mathcal{T}(\mathcal{B}).$$

We also need the reverse,

$$\mathcal{A} \cong \mathcal{B} \Longleftarrow \mathcal{T}(\mathcal{A}) \cong \mathcal{T}(\mathcal{B}).$$

This follows from a back-and-forth argument.

Let $\mathcal{A}$ consist of elements $a_1, a_2, a_3, \ldots$ and let $\mathcal{B}$ consist of elements $b_1, b_2, b_3, \ldots$. Suppose that $\mathcal{T}(\mathcal{A})$ and $\mathcal{T}(\mathcal{B})$ are isomorphic via an isomorphism f. We argue that $\mathcal{A}$ and $\mathcal{B}$ are isomorphic.



$a_1 \mapsto ? a_1$ $\qquad \mapsto b_3$

$? \mapsto b_1 a_2$ $\qquad \mapsto b_1$

We have shown that

$$\mathcal{A} \cong \mathcal{B} \Longleftrightarrow \mathcal{T}(\mathcal{A}) \cong \mathcal{T}(\mathcal{B}).$$

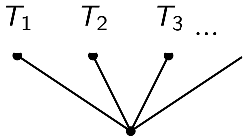
Theorem

Labeled trees are Borel complete.

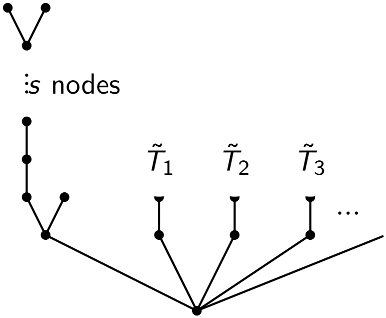
Now we need to code the labels into a tree.

We will define a way to recursively replace a labeled subtree T by an unlabeled tree $\tilde{T}$.

Given T with root node labeled s and labeled subtrees $T_1, T_2, \ldots$:



Define $\tilde{T}$ as follows:



We can construct this tree from the labeled tree, and recover a copy of the labeled tree from a copy of this tree.

Theorem

Trees are Borel complete.

Theorem

Linear orders are Borel complete.

To do this, we need the construction of a shuffle sum.

Choose a partition $\mathbb{Q} = \bigsqcup_n Q_n$ where each part is dense.

Given $q \in \mathbb{Q}$ let $n(q)$ be the part that q is in.

Given a sequence of linear orders $\mathcal{L}_n$, the shuffle sum is

$$\sum_{q \in \mathbb{Q}} \mathcal{L}_{n(q)}.$$

The isomorphism type does not depend on the partition.

We will make use of the fact that labeled trees are Borel complete. Given a labeled tree T, we form a linear order $\mathcal{L}(T)$.

We may assume that the label of each node encodes what level of the tree that node is on.

Define $\mathcal{L}(T)$ recursively: $\mathcal{L}(T)$ is the shuffle sum of

$$\mathbb{Q} + \ell(\sigma) + 2 + \mathbb{Q} + \mathcal{L}(T_\sigma)$$

for all children σ of the root node, where

- T_σ is the subtree of T below σ and
- $\ell(\sigma)$ is the (integer code for) the label of σ.

What, formally, is the difference between the former examples (graphs, rings, partial orders) and the latter examples (trees, linear orders)?

For graphs, the structure can be found inside the graph. Certain vertices of the graph, which can be identified, represent elements of the structure. Certain aspects of the graph allow us to recover the relations on the structure.

For trees (and linear orders), the structure cannot be found inside the tree in the same way.

We will call classes of structures with the stronger kind of reduction universal.

What do we formally mean by this?

One approach, from Hirschfeldt, Khoussainov, Shore, and Slinko is that the former examples allow one to transfer examples from one class to another:

Theorem (Hirschfeldt, Khoussainov, Shore, and Slinko)

Given a structure with some property from the list below, there is a graph, partial order, integral domain, and 2-step nilpotent group with that property.

The list of properties is:

- *degree spectra of nontrivial structures,*
- *effective dimensions,*
- *expansion by constants, and*
- *degree spectra of relations.*

Linear orders cannot be added to this list.

Another way of differentiating the constructions is that for graphs, partial orders, integral domains, and 2-step nilpotent groups,

- the Borel reductions are functorial, and
- there is a functorial inverse to the Borel reduction.

E.g., given the encoding of a structure into a graph, we can recover the original structure from the graph in a functorial way.

This is equivalent to saying that there is an "infinitary bi-interpretation" between the two structures. (Harrison-Trainor, Melnikov, Miller, Montalbán)

In particular, they have the same automorphism group.

Moreover, often the functorial Borel reduction and its inverse are both computable. In this case, we say that the class is computably universal.

This is equivalent to saying that there is a “computable bi-interpretation” between the two structures.

Theorem (Harrison-Trainor, Montalbán)

For each computable ordinal α, there is a structure $\mathcal{A}$ with no Δ^0_α-computable copies but for which $\mathcal{T}(\mathcal{A})$ has a computable copy.

In particular, there are no $\mathcal{L}_{\omega_1\omega}$ formulas that uniformly (in the choice of language) interpret the structure $\mathcal{A}$ in the tree of tuples $\mathcal{T}(\mathcal{A})$. (Also proved by Knight, A. Soskova, and Vatev.)

Theorem (Friedman, Stanley)

Fields are Borel complete and are universal.

In their argument, the recovery of the graph from the field is not computable. For a long time it was not known that this could be done computably, until 2017.

Theorem (Miller, Poonen, Schoutens, Shlapentokh)

Fields are computably universal.

Theorem (Miller, Poonen, Schoutens, Shlapentokh)

Fields are computably universal.

The idea to reduce a graph G to a field is:

- take the pure transcendental extension $\mathbb{Q}(t_1, t_2, \ldots)$;
- use $t_1, t_2, \ldots$ to represent the vertices of G, and distinguish those elements by adding solutions to a certain equation involving t_i;
- for each t_i, t_j, add a solution to a certain equation involving t_i and t_j if they are adjacent in G, and add a solution to another equation if they are not.

Choosing the equations carefully, this encodes the graph in the field.

That's it for today!

Next time:

- Comparisons with the graph isomorphism problem in complexity theory
- Torsion-free abelian groups