Lowness in recursive model theory

Johanna Franklin

University of Connecticut

May 21, 2013

Lowness

A set X is *low* for a relativizable class $\mathcal{C}$ if $\mathcal{C}^X = \mathcal{C}$.

Examples:

- Classical degree theory
- Randomness
  - Martin-Löf randomness: K-trivial
  - recursive randomness: recursive
  - Schnorr randomness: recursively traceable
- Computational learning theory
  - EX-learning: low and 1-generic

Lowness and recursive structure theory

A set is *low for isomorphism* if, whenever it can compute an isomorphism between two recursively presented structures, there is a recursive isomorphism between them.

Prima facie

A degree $\mathbf{d}$ is a *degree of categoricity* if there is a recursive structure $\mathcal{A}$ such that $\mathbf{d}$ can compute an isomorphism between any two recursive copies of $\mathcal{A}$ and $\mathbf{d}$ is the least degree with this property.

Theorem (Fokina, Kalimullin, R. Miller)

Any degree d.r.e. in and above $0^{(n)}$ is a degree of categoricity.

Corollary

No degree that computes $0'$ is low for isomorphism.

A useful fact

Theorem (Hirschfeldt, Khoussainov, Shore, Slinko)

Arbitrary countable structures $\mathcal{A}$ and $\mathcal{B}$ in a recursive language can be coded into countable directed graphs $G(\mathcal{A})$ and $G(\mathcal{B})$ such that

1. $\mathcal{A} \cong \mathcal{B}$ *if and only if* $G(\mathcal{A}) \cong G(\mathcal{B})$.
2. $\mathcal{A}$ *is recursive if and only if* $G(\mathcal{A})$ *is.*
3. *If* $\mathcal{A}$ *and* $\mathcal{B}$ *are recursive, then for every degree* $\mathbf{d}$,

$$\mathcal{A} \cong_{\mathbf{d}} \mathcal{B} \ \text{ if and only if } \ G(\mathcal{A}) \cong_{\mathbf{d}} G(\mathcal{B}).$$

Δ^0_2 degrees

Theorem

No nonrecursive Δ^0_2 degree is low for isomorphism.

Δ^0_2 degrees

Theorem

No nonrecursive Δ^0_2 degree is low for isomorphism.

Let D be a nonrecursive Δ^0_2 set. We build two structures isomorphic in D but not 0:

$$\Longrightarrow$$

Questions

- How does the class of these degrees compare to other lowness classes?
- Do these degrees form an ideal?
- How computationally weak are these degrees?
- How large is the class of these degrees?

Forcing works

Being low for isomorphism can be forced: we can force functions to

- converge/diverge,
- be partial/total, and
- be surjective/not surjective.

This is enough to show that given a certain level of genericity, we can force a recursive isomorphism to exist.

Interesting subclasses

Theorem

Every 2-generic degree is low for isomorphism.

Proof.

Cohen forcing and a back-and-forth construction. $\square$

Interesting subclasses

Theorem

Every 2-generic degree is low for isomorphism.

Proof.

Cohen forcing and a back-and-forth construction. □

Theorem

Every degree that is 3-generic for Mathias forcing is low for isomorphism.

Proof.

Mathias forcing, Martin's characterization of high degrees, and a back-and-forth construction. □

Interesting subclasses

Theorem
Every 2-generic degree is low for isomorphism.

Proof.
Cohen forcing and a back-and-forth construction. □

Theorem
Every degree that is 3-generic for Mathias forcing is low for isomorphism.

Proof.
Mathias forcing, Martin's characterization of high degrees, and a back-and-forth construction. □

Corollary
The degrees that are low for isomorphism do not form an ideal.

Interesting properties

We can also use forcing with perfect trees to create degrees that are low for isomorphism that have the standard properties one can get in this way.

Lowness for isomorphism and "simple" degrees

Class	Low for isomorphism	Not low for isomorphism
Δ^0_2	none	all

Lowness for isomorphism and “simple” degrees

Class	Low for isomorphism	Not low for isomorphism
Δ^0_2	none	all
Δ^0_3	$\checkmark$ (2-gen.)	$\checkmark$ (Δ^0_2)

Lowness for isomorphism and “simple” degrees

Class	Low for isomorphism	Not low for isomorphism
Δ^0_2	none	all
Δ^0_3	✓(2-gen.)	✓(Δ^0_2)
hyperimmune	✓(2-gen.)	✓(Δ^0_2)

Lowness for isomorphism and “simple” degrees

Class	Low for isomorphism	Not low for isomorphism
Δ^0_2	none	all
Δ^0_3	✓(2-gen.)	✓(Δ^0_2)
hyperimmune	✓(2-gen.)	✓(Δ^0_2)
hyperimmune free	✓(perfect trees)	✓(separating sets)

Lowness for isomorphism and “simple” degrees

Class	Low for isomorphism	Not low for isomorphism
Δ_2^0	none	all
Δ_3^0	✓ (2-gen.)	✓ (Δ_2^0)
hyperimmune	✓ (2-gen.)	✓ (Δ_2^0)
hyperimmune free	✓ (perfect trees)	✓ (separating sets)
minimal	✓ (perfect trees)	✓ (Δ_2^0)

Lowness for isomorphism and the jump

Theorem

If $\mathbf{d}$ *is low for isomorphism, then for every* n*, a degree* $\mathbf{c}$ *can be found such that* $\mathbf{d} <_T \mathbf{c}$*,* $\mathbf{c}$ *is low for isomorphism, and* $\mathbf{0^{(n)}} <_T \mathbf{c}'$*.*

Proof.

Let $D \in \mathbf{d}$ and define a sequence of sets $D = X_0 <_T X_1 <_T \dots$ such that

- X_n is low for isomorphism and
- $X_n'' \leq_T X_{n+1}'$.

Given such an X_n, choose Y_n to be 3-X_n-generic for X_n-recursive Mathias forcing. $X_n \oplus Y_n$ will be low for isomorphism, and we can use Martin's Theorem to get that $X_n'' \leq_T (X_n \oplus Y_n)'$. $\square$

Generalized low and high

Theorem

There are both GL_1 and GH_1 degrees that are low for isomorphism.

Proof.

There are GL_1 2-generics for Cohen forcing and GH_1 3-generics for Mathias forcing (Cholak et al.). $\square$

How many degrees are low for isomorphism?

Theorem

The class of degrees that are low for isomorphism is comeager.

Theorem

The class of degrees that are low for isomorphism has measure 0.

Proof for measure

We just need to show that the class of degrees that are not low for isomorphism has positive measure.

We will build two recursive graphs $\mathcal{G}_0$ and $\mathcal{G}_1$ and a recursive tree $T \subseteq \{0,1\}^{<\omega}$ such that

1. $\mathcal{G}_0$ and $\mathcal{G}_1$ are not recursively isomorphic,
2. if $X \in [T]$, X computes an isomorphism from $\mathcal{G}_0$ to $\mathcal{G}_1$, and
3. $\mu([T]) \geq \frac{1}{2}$.

Proof for measure

We just need to show that the class of degrees that are not low for isomorphism has positive measure.

We will build two recursive graphs $\mathcal{G}_0$ and $\mathcal{G}_1$ and a recursive tree $T \subseteq \{0,1\}^{<\omega}$ such that

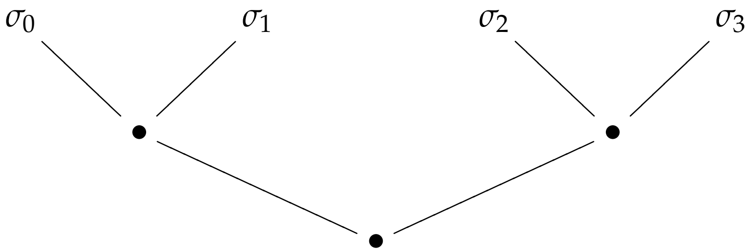
1. $\mathcal{G}_0$ and $\mathcal{G}_1$ are not recursively isomorphic,
2. if $X \in [T]$, X computes an isomorphism from $\mathcal{G}_0$ to $\mathcal{G}_1$, and
3. $\mu([T]) \geq \frac{1}{2}$.

To do this, we will

1. ensure that for each e, Φ_e is not an isomorphism from $\mathcal{G}_0$ to $\mathcal{G}_1$,
2. define a Turing functional Γ such that for all $X \in [T]$, Γ^X is an isomorphism from $\mathcal{G}_0$ to $\mathcal{G}_1$, and
3. remove only small amounts from T in the process.

Meeting the requirement for Φ_0

We start with the tree



If Φ_0 gives us part of an isomorphism, we will remove at most one of σ_0, σ_1, σ_2, and σ_3 from T.

Coding nodes for Φ_0

$$
\mathcal{G}_0: \quad
\begin{array}{c|c|c|c|c}
 & \sigma_0 \text{ nodes} & \sigma_1 \text{ nodes} & \sigma_2 \text{ nodes} & \sigma_3 \text{ nodes} \\
\hline
a & a_0^{\sigma_0} & a_0^{\sigma_1} & a_0^{\sigma_2} & a_0^{\sigma_3} \\
 & a_1^{\sigma_0} & a_1^{\sigma_1} & a_1^{\sigma_2} & a_1^{\sigma_3} \\
 & a_2^{\sigma_0} & a_2^{\sigma_1} & a_2^{\sigma_2} & a_2^{\sigma_3} \\
 & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

$$
\mathcal{G}_1: \quad
\begin{array}{c|c|c|c|c}
 & \sigma_0 \text{ nodes} & \sigma_1 \text{ nodes} & \sigma_2 \text{ nodes} & \sigma_3 \text{ nodes} \\
\hline
b_0 & c_0^{\sigma_0} & c_0^{\sigma_1} & c_0^{\sigma_2} & c_0^{\sigma_3} \\
b_1 & c_1^{\sigma_0} & c_1^{\sigma_1} & c_1^{\sigma_2} & c_1^{\sigma_3} \\
b_2 & c_2^{\sigma_0} & c_2^{\sigma_1} & c_2^{\sigma_2} & c_2^{\sigma_3} \\
b_3 & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

Γ^{σ_0}

$$
\mathcal{G}_0: \quad
\begin{array}{c|c|c|c|c}
 & \sigma_0 \text{ nodes} & \sigma_1 \text{ nodes} & \sigma_2 \text{ nodes} & \sigma_3 \text{ nodes} \\
\hline
a & a_0^{\sigma_0} & a_0^{\sigma_1} & a_0^{\sigma_2} & a_0^{\sigma_3} \\
 & a_1^{\sigma_0} & a_1^{\sigma_1} & a_1^{\sigma_2} & a_1^{\sigma_3} \\
 & a_2^{\sigma_0} & a_2^{\sigma_1} & a_2^{\sigma_2} & a_2^{\sigma_3} \\
 & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

$$
\mathcal{G}_1: \quad
\begin{array}{c|c|c|c|c}
 & \sigma_0 \text{ nodes} & \sigma_1 \text{ nodes} & \sigma_2 \text{ nodes} & \sigma_3 \text{ nodes} \\
\hline
b_0 & c_0^{\sigma_0} & c_0^{\sigma_1} & c_0^{\sigma_2} & c_0^{\sigma_3} \\
b_1 & c_1^{\sigma_0} & c_1^{\sigma_1} & c_1^{\sigma_2} & c_1^{\sigma_3} \\
b_2 & c_2^{\sigma_0} & c_2^{\sigma_1} & c_2^{\sigma_2} & c_2^{\sigma_3} \\
b_3 & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

Actions

Case 1: $\Phi_0(a) = c_k^{\sigma_i}$ for some $0 \leq i \leq 3, k \geq 0$.
For this value of i:

- Kill all paths in T through σ_i.
- Add "tails" as follows (for $i = 0$):

$$
\mathcal{G}_0: \quad
\begin{array}{c|c|c|c|c}
 & \sigma_0 \text{ nodes} & \sigma_1 \text{ nodes} & \sigma_2 \text{ nodes} & \sigma_3 \text{ nodes} \\
a & a_0^{\sigma_0} \lhd & a_0^{\sigma_1} & a_0^{\sigma_2} & a_0^{\sigma_3} \\
 & a_1^{\sigma_0} \lhd & a_1^{\sigma_1} & a_1^{\sigma_2} & a_1^{\sigma_3} \\
 & a_2^{\sigma_0} \lhd & a_2^{\sigma_1} & a_2^{\sigma_2} & a_2^{\sigma_3} \\
 & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

$$
\mathcal{G}_1: \quad
\begin{array}{c|c|c|c|c}
 & \sigma_0 \text{ nodes} & \sigma_1 \text{ nodes} & \sigma_2 \text{ nodes} & \sigma_3 \text{ nodes} \\
b_0 & c_0^{\sigma_0} \lhd & c_0^{\sigma_1} & c_0^{\sigma_2} & c_0^{\sigma_3} \\
b_1 & c_1^{\sigma_0} \lhd & c_1^{\sigma_1} & c_1^{\sigma_2} & c_1^{\sigma_3} \\
b_2 & c_2^{\sigma_0} \lhd & c_2^{\sigma_1} & c_2^{\sigma_2} & c_2^{\sigma_3} \\
b_3 & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

Actions

Case 2: $\Phi_0(a) = b_i$ for some $0 \leq i \leq 3$.
For this value of i:

- Kill all paths in T through σ_i.
- Add "tails" as follows (for $i = 1$):

$$\mathcal{G}_0: \quad \begin{array}{c|c|c|c|c} & \sigma_0 \text{ nodes} & \sigma_1 \text{ nodes} & \sigma_2 \text{ nodes} & \sigma_3 \text{ nodes} \\ \hline a & a_0^{\sigma_0} \lhd & a_0^{\sigma_1} & a_0^{\sigma_2} \lhd & a_0^{\sigma_3} \lhd \\ & a_1^{\sigma_0} & a_1^{\sigma_1} & a_1^{\sigma_2} & a_1^{\sigma_3} \\ & a_2^{\sigma_0} & a_2^{\sigma_1} & a_2^{\sigma_2} & a_2^{\sigma_3} \\ & \vdots & \vdots & \vdots & \vdots \end{array}$$

$$\mathcal{G}_1: \quad \begin{array}{c|c|c|c|c} & \sigma_0 \text{ nodes} & \sigma_1 \text{ nodes} & \sigma_2 \text{ nodes} & \sigma_3 \text{ nodes} \\ \hline b_0 & c_0^{\sigma_0} \lhd & c_0^{\sigma_1} & c_0^{\sigma_2} \lhd & c_0^{\sigma_3} \lhd \\ b_1 \lhd & c_1^{\sigma_0} & c_1^{\sigma_1} & c_1^{\sigma_2} & c_1^{\sigma_3} \\ b_2 & c_2^{\sigma_0} & c_2^{\sigma_1} & c_2^{\sigma_2} & c_2^{\sigma_3} \\ b_3 & \vdots & \vdots & \vdots & \vdots \end{array}$$

Lowness for classes of structures

Let $\mathcal{C}$ be a class of structures. A degree is *low for isomorphism for* $\mathcal{C}$ if, whenever it can compute an isomorphism between any two recursively presented structures in $\mathcal{C}$, there is a recursive isomorphism between them.

Theorem (Suggs)

If $\mathbf{d}$ *is* Δ_2^0 *but not recursive, then* $\mathbf{d}$ *is not low for isomorphism for linear orders of order type* ω.

Thank you!