Punctual Structures

Ellen Hammatt

Victoria University of Wellington, Te Herenga Waka

Online Logic Seminar, 15 February 2024

Plan for Today

Part 1. Introduction and Motivation

Computable Structure Theory

Definition

*A **computable presentation** of a structure $\mathcal{A}$ is a coding of $\mathcal{A}$ with universe $\mathbb{N}$ and all functions and relations computable on $\mathbb{N}$.*

We want to consider presentations up to the 'correct notion of sameness'.

Definition

Two computable presentations $\mathcal{A}, \mathcal{B}$ are computably isomorphic if there is a computable function $f : \mathcal{A} \to \mathcal{B}$ which is an isomorphism.

Computable Structure Theory

Definition

A ***computable presentation*** *of a structure $\mathcal{A}$ is a coding of $\mathcal{A}$ with universe $\mathbb{N}$ and all functions and relations computable on $\mathbb{N}$.*

We want to consider presentations up to the ‘correct notion of sameness’.

Definition

Two computable presentations $\mathcal{A}, \mathcal{B}$ are computably isomorphic if there is a computable function $f : \mathcal{A} \to \mathcal{B}$ which is an isomorphism.

Computable Structure Theory

Definition

A ***computable presentation*** *of a structure* $\mathcal{A}$ *is a coding of* $\mathcal{A}$ *with universe* $\mathbb{N}$ *and all functions and relations computable on* $\mathbb{N}$.

We want to consider presentations up to the 'correct notion of sameness'.

Definition

Two computable presentations $\mathcal{A}, \mathcal{B}$ *are computably isomorphic if there is a computable function* $f : \mathcal{A} \to \mathcal{B}$ *which is an isomorphism.*

An example

We begin with the following example.

Example

Any two computable dense countable linear orders without end points are computably isomorphic.

An example

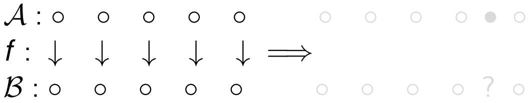
We begin with the following example.

Example

Any two computable dense countable linear orders without end points are computably isomorphic.

An Example

A typical stage in the proof:



Wait for an element to appear in the respective interval of $\mathcal{B}$:

But how long do we have to wait? This is an unbounded search.

An Example

A typical stage in the proof:

$$\begin{array}{ll} \mathcal{A}: & \circ \quad \circ \quad \circ \quad \circ \quad \circ \qquad\qquad \circ \quad \circ \quad \circ \quad \circ \; \bullet \; \circ \\ f: & \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \;\; \Longrightarrow \\ \mathcal{B}: & \circ \quad \circ \quad \circ \quad \circ \quad \circ \qquad\qquad \circ \quad \circ \quad \circ \quad \circ \; ? \; \circ \end{array}$$

Wait for an element to appear in the respective interval of $\mathcal{B}$:

$$\begin{array}{ll} & \circ \quad \circ \quad \circ \quad \circ \quad \bullet \quad \circ \\ f: & \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \Downarrow \quad \downarrow \\ & \circ \quad \circ \quad \circ \quad \circ \quad \bullet \quad \circ \end{array}$$

But how long do we have to wait? This is an unbounded search.

An Example

A typical stage in the proof:

$$\begin{array}{lccccccccccccc}
\mathcal{A}: & \circ & \circ & \circ & \circ & \circ & & \circ & \circ & \circ & \circ & \bullet & \circ \\
f: & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \Longrightarrow & & & & & & \\
\mathcal{B}: & \circ & \circ & \circ & \circ & \circ & & \circ & \circ & \circ & \circ & ? & \circ
\end{array}$$

Wait for an element to appear in the respective interval of $\mathcal{B}$:

$$\begin{array}{lcccccc}
 & \circ & \circ & \circ & \circ & \bullet & \circ \\
f: & \downarrow & \downarrow & \downarrow & \downarrow & \Downarrow & \downarrow \\
 & \circ & \circ & \circ & \circ & \bullet & \circ
\end{array}$$

But how long do we have to wait? This is an unbounded search.

An Example

A typical stage in the proof:

$$
\begin{array}{lccccccccccccc}
\mathcal{A}: & \circ & \circ & \circ & \circ & \circ & & \circ & \circ & \circ & \circ & \bullet & \circ \\
f: & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \Longrightarrow & & & & & & \\
\mathcal{B}: & \circ & \circ & \circ & \circ & \circ & & \circ & \circ & \circ & \circ & ? & \circ
\end{array}
$$

Wait for an element to appear in the respective interval of $\mathcal{B}$:

$$
\begin{array}{lcccccc}
 & \circ & \circ & \circ & \circ & \bullet & \circ \\
f: & \downarrow & \downarrow & \downarrow & \downarrow & \Downarrow & \downarrow \\
 & \circ & \circ & \circ & \circ & \bullet & \circ
\end{array}
$$

But how long do we have to wait? This is an unbounded search.

Primitive Recursion

What will happen if we forbid unbounded search?

In other words,

What is the **primitive recursive content** of mathematics?

Primitive Recursion

What will happen if we forbid unbounded search?

In other words,

What is the **primitive recursive content** of mathematics?

History

1. Mal'cev defined primitive recursive algebraic structures in the 1960s.
2. Goodstein's book (1961) 'Recursive analysis' is focused on primitive recursive processes in elementary real analysis.
3. There has been some work in group theory in the 1970s (word problem and such).

These initial attempts were essentially forgotten.

History

1. Mal'cev defined primitive recursive algebraic structures in the 1960s.
2. Goodstein's book (1961) 'Recursive analysis' is focused on primitive recursive processes in elementary real analysis.
3. There has been some work in group theory in the 1970s (word problem and such).

These initial attempts were essentially forgotten.

History

In the 1990s:

- Automatic algebra (Nerode, Khoussainov, Braun, Strüngmann and others). But unfortunately automatic algorithms are very rare: $(\mathbb{Q}, +)$ is not automatic (Tsankov, 2011).
- Online combinatorics (Kierstead, Trotter, Downey, Askes and many others). In contrast, “online” combinatorics relies on very crude models of computation
- **Polynomial time** algebra (Nerode, Remmel, Cenzer, Grigorieff, more recently Alaev and Selivanov and others). Can be notation dependent.

History

In the 1990s:

- Automatic algebra (Nerode, Khoussainov, Braun, Strüngmann and others). But unfortunately automatic algorithms are very rare: $(\mathbb{Q}, +)$ is not automatic (Tsankov, 2011).
- Online combinatorics (Kierstead, Trotter, Downey, Askes and many others). In contrast, "online" combinatorics relies on very crude models of computation
- **Polynomial time** algebra (Nerode, Remmel, Cenzer, Grigorieff, more recently Alaev and Selivanov and others). Can be notation dependent.

History

In the 1990s:

- Automatic algebra (Nerode, Khoussainov, Braun, Strüngmann and others). But unfortunately automatic algorithms are very rare: $(\mathbb{Q}, +)$ is not automatic (Tsankov, 2011).

- Online combinatorics (Kierstead, Trotter, Downey, Askes and many others). In contrast, “online” combinatorics relies on very crude models of computation

- **Polynomial time** algebra (Nerode, Remmel, Cenzer, Grigorieff, more recently Alaev and Selivanov and others). Can be notation dependent.

History

In the 1990s:

- Automatic algebra (Nerode, Khoussainov, Braun, Strüngmann and others). But unfortunately automatic algorithms are very rare: $(\mathbb{Q}, +)$ is not automatic (Tsankov, 2011).
- Online combinatorics (Kierstead, Trotter, Downey, Askes and many others). In contrast, “online” combinatorics relies on very crude models of computation
- **Polynomial time** algebra (Nerode, Remmel, Cenzer, Grigorieff, more recently Alaev and Selivanov and others). Can be notation dependent.

Polynomial Time Proofs

The majority of proofs about polynomial time structures are focused on eliminating unbounded search.

Sometimes the resulting primitive recursive algorithm is already polynomial time “for free”.

To show that there is no polynomial time presentation, it is often easiest to prove there is no primitive recursive one.

Polynomial Time Proofs

The majority of proofs about polynomial time structures are focused on eliminating unbounded search.

Sometimes the resulting primitive recursive algorithm is already polynomial time “for free”.

To show that there is no polynomial time presentation, it is often easiest to prove there is no primitive recursive one.

Polynomial Time Proofs

The majority of proofs about polynomial time structures are focused on eliminating unbounded search.

Sometimes the resulting primitive recursive algorithm is already polynomial time “for free”.

To show that there is no polynomial time presentation, it is often easiest to prove there is no primitive recursive one.

Primitive Recursion

Primitive recursion is much less notation-dependent than polynomial time (more robust).

Primitive recursion refines the crude approach in online combinatorics.

Primitive recursion has a version of the Church-Turing thesis.

Primitive recursive algorithms occur naturally in model theory (Henkin, Tarski's quantifier elimination, etc.).

Primitive Recursion

Primitive recursion is much less notation-dependent than polynomial time (more robust).

Primitive recursion refines the crude approach in online combinatorics.

Primitive recursion has a version of the Church-Turing thesis.

Primitive recursive algorithms occur naturally in model theory (Henkin, Tarski's quantifier elimination, etc.).

Primitive Recursion

Primitive recursion is much less notation-dependent than polynomial time (more robust).

Primitive recursion refines the crude approach in online combinatorics.

Primitive recursion has a version of the Church-Turing thesis.

Primitive recursive algorithms occur naturally in model theory (Henkin, Tarski's quantifier elimination, etc.).

Primitive Recursion

Primitive recursion is much less notation-dependent than polynomial time (more robust).

Primitive recursion refines the crude approach in online combinatorics.

Primitive recursion has a version of the Church-Turing thesis.

Primitive recursive algorithms occur naturally in model theory (Henkin, Tarski's quantifier elimination, etc.).

The Main Definition

Definition (Kalimullin, Melnikov, Ng 2017)

An algebraic structure $\mathcal{A}$ is **punctual** if:

- the domain of $\mathcal{A}$ is $\mathbb{N}$,
- the operations and relations of $\mathcal{A}$ are primitive recursive.

Note that this restriction is not in Mal'cev's definition (delays were allowed in the domain).

The Main Definition

Definition (Kalimullin, Melnikov, Ng 2017)

An algebraic structure $\mathcal{A}$ is **punctual** if:

- the domain of $\mathcal{A}$ is $\mathbb{N}$,
- the operations and relations of $\mathcal{A}$ are primitive recursive.

Note that this restriction is not in Mal'cev's definition (delays were allowed in the domain).

Delays in the Domain

Mal'cev allowed a primitive recursive domain.

While the domain is decided quickly, elements in the domain could grow in an unbounded way.

This feature adds a delay into the domain.

It is important that we ensure the domain is all of $\mathbb{N}$.

Delays in the Domain

Mal'cev allowed a primitive recursive domain.

While the domain is decided quickly, elements in the domain could grow in an unbounded way.

This feature adds a delay into the domain.

It is important that we ensure the domain is all of $\mathbb{N}$.

Delays in the Domain

Mal'cev allowed a primitive recursive domain.

While the domain is decided quickly, elements in the domain could grow in an unbounded way.

This feature adds a delay into the domain.

It is important that we ensure the domain is all of $\mathbb{N}$.

Delays in the Domain

Mal'cev allowed a primitive recursive domain.

While the domain is decided quickly, elements in the domain could grow in an unbounded way.

This feature adds a delay into the domain.

It is important that we ensure the domain is all of $\mathbb{N}$.

Existence of Punctual Presentations

Theorem

In each of the following classes, every computable structure has a punctual presentation:

1. Linear orders [Grigorieff, 1990].
2. Torsion-free abelian groups [Kalimullin, Melnikov, Ng 2017].
3. Boolean algebras [Kalimullin, Melnikov, Ng 2017].
4. Abelian p-groups [Kalimullin, Melnikov, Ng 2017].

In most of these cases we get a polynomial time copy almost for free.

Existence of Punctual Presentations

Theorem

In each of the following classes, every computable structure has a punctual presentation:

1. Linear orders [Grigorieff, 1990].
2. Torsion-free abelian groups [Kalimullin, Melnikov, Ng 2017].
3. Boolean algebras [Kalimullin, Melnikov, Ng 2017].
4. Abelian p-groups [Kalimullin, Melnikov, Ng 2017].

In most of these cases we get a polynomial time copy almost for free.

No Punctual Presentations

Theorem

In each of the following classes, there exists a computable structure that does not admit a punctual presentation

1. Torsion abelian groups [Cenzer and Remmel, ∼2000].
2. Archimedean ordered abelian groups [Kalimullin, Melnikov, Ng 2017].
3. Undirected graphs [Kalimullin, Melnikov, Ng 2017].

In each case the result gives the simplest construction of a computable structure without polynomial time copy.

No Punctual Presentations

Theorem

In each of the following classes, there exists a computable structure that does not admit a punctual presentation

1. Torsion abelian groups [Cenzer and Remmel, $\sim$2000].
2. Archimedean ordered abelian groups [Kalimullin, Melnikov, Ng 2017].
3. Undirected graphs [Kalimullin, Melnikov, Ng 2017].

In each case the result gives the simplest construction of a computable structure without polynomial time copy.

Part 2. Categoricity and Dimension

Punctual Categoricity

Recall that in the computable case we look at presentations up to computable isomorphism.

What do we do in the punctual case?

The inverse of a primitive recursive function is not necessarily primitive recursive.

Definition

$f : \mathbb{N} \to \mathbb{N}$ is **punctual** *if both f and f^{-1} are primitive recursive.*

Definition (Kalimullin, Melnikov, Ng 2017)

A punctual A is **punctually categorical** if it has a unique punctual presentation up to punctual isomorphism.

Punctual Categoricity

Recall that in the computable case we look at presentations up to computable isomorphism.

What do we do in the punctual case?

The inverse of a primitive recursive function is not necessarily primitive recursive.

Definition

$f : \mathbb{N} \to \mathbb{N}$ is **punctual** *if both f and f^{-1} are primitive recursive.*

Definition (Kalimullin, Melnikov, Ng 2017)

A punctual A is **punctually categorical** if it has a unique punctual presentation up to punctual isomorphism.

Punctual Categoricity

Recall that in the computable case we look at presentations up to computable isomorphism.

What do we do in the punctual case?

The inverse of a primitive recursive function is not necessarily primitive recursive.

Definition

$f : \mathbb{N} \to \mathbb{N}$ is **punctual** *if both f and f^{-1} are primitive recursive.*

Definition (Kalimullin, Melnikov, Ng 2017)

A punctual A is **punctually categorical** if it has a unique punctual presentation up to punctual isomorphism.

Punctual Categoricity

Recall that in the computable case we look at presentations up to computable isomorphism.

What do we do in the punctual case?

The inverse of a primitive recursive function is not necessarily primitive recursive.

Definition

$f : \mathbb{N} \to \mathbb{N}$ *is* **punctual** *if both* f *and* f^{-1} *are primitive recursive.*

Definition (Kalimullin, Melnikov, Ng 2017)

A punctual A is **punctually categorical** if it has a unique punctual presentation up to punctual isomorphism.

Punctual Categoricity

Recall that in the computable case we look at presentations up to computable isomorphism.

What do we do in the punctual case?

The inverse of a primitive recursive function is not necessarily primitive recursive.

Definition

$f : \mathbb{N} \to \mathbb{N}$ *is* **punctual** *if both f and f^{-1} are primitive recursive.*

Definition (Kalimullin, Melnikov, Ng 2017)

A punctual A is **punctually categorical** if it has a unique punctual presentation up to punctual isomorphism.

Punctual Categoricity

Theorem (Kalimullin, Melnikov, Ng 2017)

1. A linear order is punctually categorical iff it is finite.
2. A Boolean algebra is punctually categorical iff it is finite.
3. An abelian p-group is punctually-categorical iff it has the form $F \oplus \mathbb{V}$, where $p\mathbb{V} = \mathbf{0}$ and F is finite.
4. A torsion-free abelian group is punctually categorical iff it is the trivial group $\mathbf{0}$.

This resembles:

Theorem [Khoussainov and Nerode 1994] A structure is automatically categorical iff it is finite.

Punctual Categoricity

Theorem (Kalimullin, Melnikov, Ng 2017)

1. A linear order is punctually categorical iff it is finite.
2. A Boolean algebra is punctually categorical iff it is finite.
3. An abelian p-group is punctually-categorical iff it has the form $F \oplus \mathbb{V}$, where $p\mathbb{V} = \mathbf{0}$ and F is finite.
4. A torsion-free abelian group is punctually categorical iff it is the trivial group $\mathbf{0}$.

This resembles:

Theorem [Khoussainov and Nerode 1994] A structure is automatically categorical iff it is finite.

Non-Categoricity

All examples of punctually categorical structures on the previous slide were computably categorical.

Question

Is every punctually categorical structure computably categorical?

Theorem (Kalimullin, Melnikov, Ng 2017)

There exists a **punctually categorical** structure which is

not computably categorical.

The techniques used in this proof are novel and the structure is constructed by hand.

Repeating patterns are coded into the structure.

Non-Categoricity

All examples of punctually categorical structures on the previous slide were computably categorical.

Question

Is every punctually categorical structure computably categorical?

Theorem (Kalimullin, Melnikov, Ng 2017)

There exists a **punctually categorical** structure which is

not computably categorical.

The techniques used in this proof are novel and the structure is constructed by hand.

Repeating patterns are coded into the structure.

Non-Categoricity

All examples of punctually categorical structures on the previous slide were computably categorical.

Question

Is every punctually categorical structure computably categorical?

Theorem (Kalimullin, Melnikov, Ng 2017)

There exists a **punctually categorical** structure which is

not computably categorical.

The techniques used in this proof are novel and the structure is constructed by hand.

Repeating patterns are coded into the structure.

Non-Categoricity

All examples of punctually categorical structures on the previous slide were computably categorical.

Question

Is every punctually categorical structure computably categorical?

Theorem (Kalimullin, Melnikov, Ng 2017)

There exists a **punctually categorical** structure which is

not computably categorical.

The techniques used in this proof are novel and the structure is constructed by hand.

Repeating patterns are coded into the structure.

Computable Dimension

In 1980 Goncharov proved that there is a structure having exactly two computable presentations, up to computable isomorphism.

We call the number of computable presentations of a structure $\mathcal{A}$ up to computable isomorphism, the computable dimension of $\mathcal{A}$.

Computable Dimension

In 1980 Goncharov proved that there is a structure having exactly two computable presentations, up to computable isomorphism.

We call the number of computable presentations of a structure $\mathcal{A}$ up to computable isomorphism, the computable dimension of $\mathcal{A}$.

Examples of Computable Dimension 2

In each of the following classes, there is a structure with computable dimension 2:

1. *two-step nilpotent groups [Goncharov 1981]*
2. *fields [Miller, Poonen, Schoutens, Shlapentokh 2018]*
3. *and many other classes [Hirschfeldt, Khoussainov, Shore, Slinko 2002]*

In all of these cases the structures must be specifically constucted and are complex.

Punctual Dimension

We call the number of punctual presentations of a structure $\mathcal{A}$ up to punctual isomorphism, the punctual dimension of $\mathcal{A}$.

Theorem (Melnikov, Ng 2020)

There is a structure of punctual dimension 2.

This proof is non-standard, it does not resemble the techniques as in the proofs for finite computable dimension.

Techniques are based on those from the structure that is punctually categorical but not computably categorical.

Punctual Dimension

We call the number of punctual presentations of a structure $\mathcal{A}$ up to punctual isomorphism, the punctual dimension of $\mathcal{A}$.

Theorem (Melnikov, Ng 2020)

There is a structure of punctual dimension 2.

This proof is non-standard, it does not resemble the techniques as in the proofs for finite computable dimension.

Techniques are based on those from the structure that is punctually categorical but not computably categorical.

Punctual Dimension

We call the number of punctual presentations of a structure $\mathcal{A}$ up to punctual isomorphism, the punctual dimension of $\mathcal{A}$.

Theorem (Melnikov, Ng 2020)

There is a structure of punctual dimension 2.

This proof is non-standard, it does not resemble the techniques as in the proofs for finite computable dimension.

Techniques are based on those from the structure that is punctually categorical but not computably categorical.

Punctual Dimension

We call the number of punctual presentations of a structure $\mathcal{A}$ up to punctual isomorphism, the punctual dimension of $\mathcal{A}$.

Theorem (Melnikov, Ng 2020)

There is a structure of punctual dimension 2.

This proof is non-standard, it does not resemble the techniques as in the proofs for finite computable dimension.

Techniques are based on those from the structure that is punctually categorical but not computably categorical.

Punctual Dimension

It is folklore that there exists structures of computable dimension n for any $n \in \mathbb{N}$.

This is done by using disjoint unions of a structure of computable dimension 2.

What about the punctual case?

Theorem (H.)

For all punctual structures $\mathcal{A}$ and $\mathcal{B}$, the disjoint union of $\mathcal{A}$ and $\mathcal{B}$ has punctual dimension 1 or ∞.

We are provably justified to construct structures of punctual dimension n by hand.

Theorem (H.)

There exists a structure of punctual dimension n for any $n \in \mathbb{N}$

Punctual Dimension

It is folklore that there exists structures of computable dimension n for any $n \in \mathbb{N}$.

This is done by using disjoint unions of a structure of computable dimension 2.

What about the punctual case?

Theorem (H.)

For all punctual structures $\mathcal{A}$ and $\mathcal{B}$, the disjoint union of $\mathcal{A}$ and $\mathcal{B}$ has punctual dimension 1 or ∞.

We are provably justified to construct structures of punctual dimension n by hand.

Theorem (H.)

There exists a structure of punctual dimension n for any $n \in \mathbb{N}$

Punctual Dimension

It is folklore that there exists structures of computable dimension n for any $n \in \mathbb{N}$.

This is done by using disjoint unions of a structure of computable dimension 2.

What about the punctual case?

Theorem (H.)

For all punctual structures $\mathcal{A}$ and $\mathcal{B}$, the disjoint union of $\mathcal{A}$ and $\mathcal{B}$ has punctual dimension 1 or ∞.

We are provably justified to construct structures of punctual dimension n by hand.

Theorem (H.)

There exists a structure of punctual dimension n for any $n \in \mathbb{N}$

Punctual Dimension

It is folklore that there exists structures of computable dimension n for any $n \in \mathbb{N}$.

This is done by using disjoint unions of a structure of computable dimension 2.

What about the punctual case?

Theorem (H.)

For all punctual structures $\mathcal{A}$ and $\mathcal{B}$, the disjoint union of $\mathcal{A}$ and $\mathcal{B}$ has punctual dimension 1 or ∞.

We are provably justified to construct structures of punctual dimension n by hand.

Theorem (H.)

There exists a structure of punctual dimension n for any $n \in \mathbb{N}$

Part 3. The Punctual Degrees

The Punctual Degrees

Notice that primitive recursive isomorphism induces a natural order on the collection of presentations of a structure.

Let $PR(\mathcal{A})$ be the collection of all punctual presentations of a countably infinite structure $\mathcal{A}$.

Definition (Kalimullin, Melnikov, Ng 2017)

For $\mathcal{A}_1, \mathcal{A}_2 \in PR(\mathcal{A})$, write $\mathcal{A}_1 \leq_{pr} \mathcal{A}_2$ if there exists a primitive recursive isomorphism from $\mathcal{A}_1$ onto $\mathcal{A}_2$.

Note that this is a completely new idea that is not present in the computable case.

The punctual degrees of $\mathcal{A}$ is denoted as $\mathbf{PR}(\mathcal{A}) = PR(\mathcal{A})/\cong_{pr}$

The Punctual Degrees

Notice that primitive recursive isomorphism induces a natural order on the collection of presentations of a structure.

Let $PR(\mathcal{A})$ be the collection of all punctual presentations of a countably infinite structure $\mathcal{A}$.

Definition (Kalimullin, Melnikov, Ng 2017)

For $\mathcal{A}_1, \mathcal{A}_2 \in PR(\mathcal{A})$, write $\mathcal{A}_1 \leq_{pr} \mathcal{A}_2$ if there exists a primitive recursive isomorphism from $\mathcal{A}_1$ onto $\mathcal{A}_2$.

Note that this is a completely new idea that is not present in the computable case.

The punctual degrees of $\mathcal{A}$ is denoted as $\mathbf{PR}(\mathcal{A}) = PR(\mathcal{A}) / \cong_{pr}$

The Punctual Degrees

Notice that primitive recursive isomorphism induces a natural order on the collection of presentations of a structure.

Let $PR(\mathcal{A})$ be the collection of all punctual presentations of a countably infinite structure $\mathcal{A}$.

Definition (Kalimullin, Melnikov, Ng 2017)

For $\mathcal{A}_1, \mathcal{A}_2 \in PR(\mathcal{A})$, write $\mathcal{A}_1 \leq_{pr} \mathcal{A}_2$ if there exists a primitive recursive isomorphism from $\mathcal{A}_1$ onto $\mathcal{A}_2$.

Note that this is a completely new idea that is not present in the computable case.

The punctual degrees of $\mathcal{A}$ is denoted as $\mathbf{PR}(\mathcal{A}) = PR(\mathcal{A})/\cong_{pr}$

The Punctual Degrees

Notice that primitive recursive isomorphism induces a natural order on the collection of presentations of a structure.

Let $PR(\mathcal{A})$ be the collection of all punctual presentations of a countably infinite structure $\mathcal{A}$.

Definition (Kalimullin, Melnikov, Ng 2017)

For $\mathcal{A}_1, \mathcal{A}_2 \in PR(\mathcal{A})$, write $\mathcal{A}_1 \leq_{pr} \mathcal{A}_2$ if there exists a primitive recursive isomorphism from $\mathcal{A}_1$ onto $\mathcal{A}_2$.

Note that this is a completely new idea that is not present in the computable case.

The punctual degrees of $\mathcal{A}$ is denoted as $\mathbf{PR}(\mathcal{A}) = PR(\mathcal{A})/\cong_{pr}$

Non-Isomorphic Punctual Degrees

Naturally we wish to investigate the structure of the punctual degrees.

Theorem (Melnikov, Ng 2018)

The punctual degrees of:
- the dense linear order η,
- the random graph $\mathcal{R}$, and
- the universal divisible abelian p-group $\mathcal{P}$

are **pairwise non-isomorphic**.

The punctual degrees are able to separate the subtle difference between these structures.

Non-Isomorphic Punctual Degrees

Naturally we wish to investigate the structure of the punctual degrees.

Theorem (Melnikov, Ng 2018)

The punctual degrees of:
- the dense linear order η,
- the random graph $\mathcal{R}$, and
- the universal divisible abelian p-group $\mathcal{P}$

are **pairwise non-isomorphic**.

The punctual degrees are able to separate the subtle difference between these structures.

Non-Isomorphic Punctual Degrees

Naturally we wish to investigate the structure of the punctual degrees.

Theorem (Melnikov, Ng 2018)

The punctual degrees of:
- the dense linear order η,
- the random graph $\mathcal{R}$, and
- the universal divisible abelian p-group $\mathcal{P}$

are **pairwise non-isomorphic**.

The punctual degrees are able to separate the subtle difference between these structures.

Density in the Punctual Degrees

We have the following results about the density of the punctual degrees of various structures:

- For a finitely generated structure M, $PR(M)$ is dense [Bazhenov, Kalimullin, Melnikov, Ng 2020]
- More examples of density including almost rigid structures and $(\mathbb{Z}, <)$ [Downey, Dorzhieva, H., Melnikov, Ng 2023]
- There exist structures where the punctual degrees are not dense [Greenberg, Harrison-Trainor, Melnikov, Turetsky 2020]

Density in the Punctual Degrees

We have the following results about the density of the punctual degrees of various structures:

- For a finitely generated structure *M*, *PR*(*M*) is dense [Bazhenov, Kalimullin, Melnikov, Ng 2020]
- More examples of density including almost rigid structures and $(\mathbb{Z}, <)$ [Downey, Dorzhieva, H., Melnikov, Ng 2023]
- There exist structures where the punctual degrees are not dense [Greenberg, Harrison-Trainor, Melnikov, Turetsky 2020]

Density in the Punctual Degrees

We have the following results about the density of the punctual degrees of various structures:

- For a finitely generated structure M, $PR(M)$ is dense [Bazhenov, Kalimullin, Melnikov, Ng 2020]
- More examples of density including almost rigid structures and $(\mathbb{Z}, <)$ [Downey, Dorzhieva, H., Melnikov, Ng 2023]
- There exist structures where the punctual degrees are not dense [Greenberg, Harrison-Trainor, Melnikov, Turetsky 2020]

The Punctual Degrees of $(\mathbb{Q}, <)$

Theorem (Koh, Melnikov, Ng 2024)

The punctual degrees of $(\mathbb{Q}, <)$ are not dense.

The rationals are not dense enough!

The proof is brutal. We wish to understand the structure of the punctual degrees of $(\mathbb{Q}, <)$ further.

The Punctual Degrees of $(\mathbb{Q}, <)$

Theorem (Koh, Melnikov, Ng 2024)

The punctual degrees of $(\mathbb{Q}, <)$ are not dense.

The rationals are not dense enough!

The proof is brutal. We wish to understand the structure of the punctual degrees of $(\mathbb{Q}, <)$ further.

The Punctual Degrees of $(\mathbb{Q}, <)$

Theorem (Koh, Melnikov, Ng 2024)

The punctual degrees of $(\mathbb{Q}, <)$ are not dense.

The rationals are not dense enough!

The proof is brutal. We wish to understand the structure of the punctual degrees of $(\mathbb{Q}, <)$ further.

The Punctual Degrees of $(\mathbb{Q}, <)$

We have been working on embedding the atomless Boolean algebra into the punctual degrees of $(\mathbb{Q}, <)$ (with Dorzhieva).

Theorem (Dorzhieva, H. 2024)

There are $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$ in ***PR***$(\mathbb{Q}, <)$ *such that $\mathcal{C}$ and $\mathcal{D}$ are incomparable, $\mathcal{A} <_{pr} \mathcal{C}, \mathcal{D} <_{pr} \mathcal{B}$ and $\mathcal{A} = \inf(\mathcal{C}, \mathcal{D})$ and $\mathcal{B} = \sup(\mathcal{C}, \mathcal{D})$.*

The strategy to preserve the supremum and infimum required careful attention.

Conjecture

Any distributive lattice can be embedded into the punctual degrees of $(\mathbb{Q}, <)$

Recall that the atomless Boolean algebra is universal for distributive lattices.

The Punctual Degrees of $(\mathbb{Q}, <)$

We have been working on embedding the atomless Boolean algebra into the punctual degrees of $(\mathbb{Q}, <)$ (with Dorzhieva).

Theorem (Dorzhieva, H. 2024)

*There are $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$ in **PR**$(\mathbb{Q}, <)$ such that $\mathcal{C}$ and $\mathcal{D}$ are incomparable, $\mathcal{A} <_{pr} \mathcal{C}, \mathcal{D} <_{pr} \mathcal{B}$ and $\mathcal{A} = \inf(\mathcal{C}, \mathcal{D})$ and $\mathcal{B} = \sup(\mathcal{C}, \mathcal{D})$.*

The strategy to preserve the supremum and infimum required careful attention.

Conjecture

Any distributive lattice can be embedded into the punctual degrees of $(\mathbb{Q}, <)$

Recall that the atomless Boolean algebra is universal for distributive lattices.

The Punctual Degrees of $(\mathbb{Q}, <)$

We have been working on embedding the atomless Boolean algebra into the punctual degrees of $(\mathbb{Q}, <)$ (with Dorzhieva).

Theorem (Dorzhieva, H. 2024)

*There are $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$ in **PR**$(\mathbb{Q}, <)$ such that $\mathcal{C}$ and $\mathcal{D}$ are incomparable, $\mathcal{A} <_{pr} \mathcal{C}, \mathcal{D} <_{pr} \mathcal{B}$ and $\mathcal{A} = \inf(\mathcal{C}, \mathcal{D})$ and $\mathcal{B} = \sup(\mathcal{C}, \mathcal{D})$.*

The strategy to preserve the supremum and infimum required careful attention.

Conjecture

Any distributive lattice can be embedded into the punctual degrees of $(\mathbb{Q}, <)$

Recall that the atomless Boolean algebra is universal for distributive lattices.

The Punctual Degrees of $(\mathbb{Q}, <)$

We have been working on embedding the atomless Boolean algebra into the punctual degrees of $(\mathbb{Q}, <)$ (with Dorzhieva).

Theorem (Dorzhieva, H. 2024)

There are $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$ in $\textbf{PR}(\mathbb{Q}, <)$ such that $\mathcal{C}$ and $\mathcal{D}$ are incomparable, $\mathcal{A} <_{pr} \mathcal{C}, \mathcal{D} <_{pr} \mathcal{B}$ and $\mathcal{A} = \inf(\mathcal{C}, \mathcal{D})$ and $\mathcal{B} = \sup(\mathcal{C}, \mathcal{D})$.

The strategy to preserve the supremum and infimum required careful attention.

Conjecture

Any distributive lattice can be embedded into the punctual degrees of $(\mathbb{Q}, <)$

Recall that the atomless Boolean algebra is universal for distributive lattices.

The Punctual Degrees of $(\mathbb{Q}, <)$

We have been working on embedding the atomless Boolean algebra into the punctual degrees of $(\mathbb{Q}, <)$ (with Dorzhieva).

Theorem (Dorzhieva, H. 2024)

*There are $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$ in **PR**$(\mathbb{Q}, <)$ such that $\mathcal{C}$ and $\mathcal{D}$ are incomparable, $\mathcal{A} <_{pr} \mathcal{C}, \mathcal{D} <_{pr} \mathcal{B}$ and $\mathcal{A} = \inf(\mathcal{C}, \mathcal{D})$ and $\mathcal{B} = \sup(\mathcal{C}, \mathcal{D})$.*

The strategy to preserve the supremum and infimum required careful attention.

Conjecture

Any distributive lattice can be embedded into the punctual degrees of $(\mathbb{Q}, <)$

Recall that the atomless Boolean algebra is universal for distributive lattices.

The Punctual Degrees of Other Linear Orders

$\mathbb{Q}$ has been very difficult.

We have looked into the punctual degrees of other linear orders and we have the following result.

Theorem (Dorzhieva, H. 2024)

Let $\mathcal{L}$ be a linear order such that there is an infinite interval $\mathcal{L}_0$ in $\mathcal{L}$ such that for any $\varphi \in \mathrm{Aut}(\mathcal{L})$, $\varphi \restriction_{L_0} = \mathrm{id}_{L_0}$. The atomless Boolean algebra can be embedded into the punctual degrees of $\mathcal{L}$.

The Punctual Degrees of Other Linear Orders

$\mathbb{Q}$ has been very difficult.

We have looked into the punctual degrees of other linear orders and we have the following result.

Theorem (Dorzhieva, H. 2024)

Let $\mathcal{L}$ be a linear order such that there is an infinite interval $\mathcal{L}_0$ in $\mathcal{L}$ such that for any $\varphi \in \mathrm{Aut}(\mathcal{L})$, $\varphi \restriction_{L_0} = \mathrm{id}_{L_0}$. The atomless Boolean algebra can be embedded into the punctual degrees of $\mathcal{L}$.

The Punctual Degrees of Other Linear Orders

$\mathbb{Q}$ has been very difficult.

We have looked into the punctual degrees of other linear orders and we have the following result.

Theorem (Dorzhieva, H. 2024)

Let $\mathcal{L}$ be a linear order such that there is an infinite interval $\mathcal{L}_0$ in $\mathcal{L}$ such that for any $\varphi \in \mathrm{Aut}(\mathcal{L})$, $\varphi \restriction_{L_0} = \mathrm{id}_{L_0}$. The atomless Boolean algebra can be embedded into the punctual degrees of $\mathcal{L}$.

Part 4: 1-Decidability

1-Decidability

A computable presentation of a structure is **1-decidable** if given an existential formula there is an algorithm to decide the truth of this formula in this presentation.

Definition

*A punctual presentation is **punctually 1-decidable** if for any existential formula $\exists\bar{x}\varphi(\bar{x},\bar{a})$, there is a primitive recursive algorithm that outputs $\bar{x}$ such that $\varphi(\bar{x},\bar{a})$ holds, and otherwise outputs -1.*

Notice the difference in this definition.

1-Decidability

A computable presentation of a structure is **1-decidable** if given an existential formula there is an algorithm to decide the truth of this formula in this presentation.

Definition

A punctual presentation is ***punctually 1-decidable*** *if for any existential formula* $\exists\bar{x}\varphi(\bar{x},\bar{a})$*, there is a primitive recursive algorithm that outputs* $\bar{x}$ *such that* $\varphi(\bar{x},\bar{a})$ *holds, and otherwise outputs* -1*.*

Notice the difference in this definition.

1-Decidability

A computable presentation of a structure is **1-decidable** if given an existential formula there is an algorithm to decide the truth of this formula in this presentation.

Definition

A punctual presentation is ***punctually 1-decidable*** *if for any existential formula* $\exists \bar{x} \varphi(\bar{x}, \bar{a})$*, there is a primitive recursive algorithm that outputs* $\bar{x}$ *such that* $\varphi(\bar{x}, \bar{a})$ *holds, and otherwise outputs* -1*.*

Notice the difference in this definition.

Boolean Algebras

$(B, \vee, \wedge, \neg, 0, 1)$

- finite and cofinite subsets of $\mathbb{N}$ *the 1-atom*
- interval algebra of $\mathbb{Q}$ (finite unions of left half-closed intervals with rational end points) *the atomless*

Definition

For a Boolean Algebra B. An element $x \in B$ is called an atom if there is no $y, z \in B$ such that $x = y \vee z$ and $y \wedge z = 0$.

Theorem (Alaev 2018)

A Boolean algebra B is punctually 1-decidable if B is punctual, the relation $\mathtt{Atom}(x)$ is primitive recursive and there is a primitive recursive function $w(x)$ which given x, outputs y, z such that $y \vee z = x$ if they exist.

Boolean Algebras

$(B, \vee, \wedge, \neg, 0, 1)$

- finite and cofinite subsets of $\mathbb{N}$ *the 1-atom*
- interval algebra of $\mathbb{Q}$ (finite unions of left half-closed intervals with rational end points) *the atomless*

Definition

For a Boolean Algebra B. An element $x \in B$ is called an atom if there is no $y, z \in B$ such that $x = y \vee z$ and $y \wedge z = 0$.

Theorem (Alaev 2018)

A Boolean algebra B is punctually 1-decidable if B is punctual, the relation $\mathtt{Atom}(x)$ is primitive recursive and there is a primitive recursive function $w(x)$ which given x, outputs y, z such that $y \vee z = x$ if they exist.

Boolean Algebras

$(B, \vee, \wedge, \neg, 0, 1)$

- finite and cofinite subsets of $\mathbb{N}$ *the 1-atom*
- interval algebra of $\mathbb{Q}$ (finite unions of left half-closed intervals with rational end points) *the atomless*

Definition

For a Boolean Algebra B. An element $x \in B$ is called an atom if there is no $y, z \in B$ such that $x = y \vee z$ and $y \wedge z = 0$.

Theorem (Alaev 2018)

A Boolean algebra B is punctually 1-decidable if B is punctual, the relation $\texttt{Atom}(x)$ is primitive recursive and there is a primitive recursive function $w(x)$ which given x, outputs y, z such that $y \vee z = x$ if they exist.

Boolean Algebras

$(B, \vee, \wedge, \neg, 0, 1)$

- finite and cofinite subsets of $\mathbb{N}$ *the 1-atom*
- interval algebra of $\mathbb{Q}$ (finite unions of left half-closed intervals with rational end points) *the atomless*

Definition

For a Boolean Algebra B. An element $x \in B$ is called an atom if there is no $y, z \in B$ such that $x = y \vee z$ and $y \wedge z = 0$.

Theorem (Alaev 2018)

A Boolean algebra B is punctually 1-decidable if B is punctual, the relation $\mathtt{Atom}(x)$ is primitive recursive and there is a primitive recursive function $w(x)$ which given x, outputs y, z such that $y \vee z = x$ if they exist.

Punctually 1-Decidable Boolean Algebras

Theorem (Alaev 2017, Downey 2021)

Any 1-decidable Boolean algebra is isomorphic to a punctually 1-decidable Boolean algebra.

The idea is to use the following theorem:

Theorem (Remmel-Vaught 1989)

Suppose a Boolean algebra B has infinitely many atoms. Let $\hat{B}$ be the Boolean algebra obtained by splitting each atom of B finitely many times. Then $\hat{B}$ is isomorphic to B.

Punctually 1-Decidable Boolean Algebras

Theorem (Alaev 2017, Downey 2021)

Any 1-decidable Boolean algebra is isomorphic to a punctually 1-decidable Boolean algebra.

The idea is to use the following theorem:

Theorem (Remmel-Vaught 1989)

Suppose a Boolean algebra B has infinitely many atoms. Let $\hat{B}$ be the Boolean algebra obtained by splitting each atom of B finitely many times. Then $\hat{B}$ is isomorphic to B.

Proof Sketch

- The opponent plays a 1-decidable Boolean algebra B and we build a punctually 1-decidable Boolean algebra A isomorphic to B.
- We build A by copying B but the opponent can wait unbounded lengths of time before declaring whether an element is an atom or not. We are building a punctual copy, **we cannot wait.**
- While we 'wait' we split all elements that are not yet declared to be atoms.
- If an element in B is eventually declared to be an atom. Then we stop splitting and declare all descendants of the copy of this element in A to be an atom.

By applying Remmel-Vaught we succeed.

Proof Sketch

- The opponent plays a 1-decidable Boolean algebra B and we build a punctually 1-decidable Boolean algebra A isomorphic to B.
- We build A by copying B but the opponent can wait unbounded lengths of time before declaring whether an element is an atom or not. We are building a punctual copy, **we cannot wait.**
- While we 'wait' we split all elements that are not yet declared to be atoms.
- If an element in B is eventually declared to be an atom. Then we stop splitting and declare all descendants of the copy of this element in A to be an atom.

By applying Remmel-Vaught we succeed.

Proof Sketch

- The opponent plays a 1-decidable Boolean algebra B and we build a punctually 1-decidable Boolean algebra A isomorphic to B.
- We build A by copying B but the opponent can wait unbounded lengths of time before declaring whether an element is an atom or not. We are building a punctual copy, **we cannot wait.**
- While we 'wait' we split all elements that are not yet declared to be atoms.
- If an element in B is eventually declared to be an atom. Then we stop splitting and declare all descendants of the copy of this element in A to be an atom.

By applying Remmel-Vaught we succeed.

Proof Sketch

- The opponent plays a 1-decidable Boolean algebra B and we build a punctually 1-decidable Boolean algebra A isomorphic to B.
- We build A by copying B but the opponent can wait unbounded lengths of time before declaring whether an element is an atom or not. We are building a punctual copy, **we cannot wait.**
- While we 'wait' we split all elements that are not yet declared to be atoms.
- If an element in B is eventually declared to be an atom. Then we stop splitting and declare all descendants of the copy of this element in A to be an atom.

By applying Remmel-Vaught we succeed.

Proof Sketch

- The opponent plays a 1-decidable Boolean algebra B and we build a punctually 1-decidable Boolean algebra A isomorphic to B.
- We build A by copying B but the opponent can wait unbounded lengths of time before declaring whether an element is an atom or not. We are building a punctual copy, **we cannot wait.**
- While we 'wait' we split all elements that are not yet declared to be atoms.
- If an element in B is eventually declared to be an atom. Then we stop splitting and declare all descendants of the copy of this element in A to be an atom.

By applying Remmel-Vaught we succeed.

Not Computably Isomorphic

But the isomorphism described in the previous slide is not necessarily computable.

Theorem (Downey, H., Melnikov 2023)

There exists a 1-decidable Boolean algebra that is not computably isomorphic to any punctually 1-decidable Boolean algebra.

Not Computably Isomorphic

But the isomorphism described in the previous slide is not necessarily computable.

Theorem (Downey, H., Melnikov 2023)

There exists a 1-decidable Boolean algebra that is not computably isomorphic to any punctually 1-decidable Boolean algebra.

Characterisation

We have a complete characterisation.

Theorem (Downey, H., Melnikov 2023)

For a countable Boolean algebra $\mathcal{B}$, the following are equivalent:

1. *Every 1-decidable presentation $\mathcal{A}$ of $\mathcal{B}$ is computably isomorphic to some punctually 1-decidable $\mathcal{P} \cong \mathcal{B}$.*
2. *$\mathcal{B}$ splits into finitely many $\mathcal{C}_0, ..., \mathcal{C}_k$ such that each $\mathcal{C}_i$ is either atomless, an atom, or a 1-atom.*

Note that (2) is exactly the Boolean algebras which are computably categorical relative to the 1-decidable presentations.

Characterisation

We have a complete characterisation.

Theorem (Downey, H., Melnikov 2023)

For a countable Boolean algebra $\mathcal{B}$, the following are equivalent:

1. *Every 1-decidable presentation $\mathcal{A}$ of $\mathcal{B}$ is computably isomorphic to some punctually 1-decidable $\mathcal{P} \cong \mathcal{B}$.*
2. *$\mathcal{B}$ splits into finitely many $\mathcal{C}_0, ..., \mathcal{C}_k$ such that each $\mathcal{C}_i$ is either atomless, an atom, or a 1-atom.*

Note that (2) is exactly the Boolean algebras which are computably categorical relative to the 1-decidable presentations.

1-Decidable Categoricity

Recall the structure that is punctually categorical but not computably categorical.

What about the 1-decidable case?

Unfortunately we cannot use the same strategy as in the non 1-decidable proof.

Theorem (H., Melnikov, Ng 2024)

There is a structure that is punctually categorical relative to 1-decidable presentations but not computably categorical relative to 1-decidable presentations.

The construction uses a mix of priority and Marker's extension and a new labelling technique. These techniques are novel.

1-Decidable Categoricity

Recall the structure that is punctually categorical but not computably categorical.

What about the 1-decidable case?

Unfortunately we cannot use the same strategy as in the non 1-decidable proof.

Theorem (H., Melnikov, Ng 2024)

There is a structure that is punctually categorical relative to 1-decidable presentations but not computably categorical relative to 1-decidable presentations.

The construction uses a mix of priority and Marker's extension and a new labelling technique. These techniques are novel.

1-Decidable Categoricity

Recall the structure that is punctually categorical but not computably categorical.

What about the 1-decidable case?

Unfortunately we cannot use the same strategy as in the non 1-decidable proof.

Theorem (H., Melnikov, Ng 2024)

There is a structure that is punctually categorical relative to 1-decidable presentations but not computably categorical relative to 1-decidable presentations.

The construction uses a mix of priority and Marker's extension and a new labelling technique. These techniques are novel.

1-Decidable Categoricity

Recall the structure that is punctually categorical but not computably categorical.

What about the 1-decidable case?

Unfortunately we cannot use the same strategy as in the non 1-decidable proof.

Theorem (H., Melnikov, Ng 2024)

There is a structure that is punctually categorical relative to 1-decidable presentations but not computably categorical relative to 1-decidable presentations.

The construction uses a mix of priority and Marker's extension and a new labelling technique. These techniques are novel.

1-Decidable Categoricity

Recall the structure that is punctually categorical but not computably categorical.

What about the 1-decidable case?

Unfortunately we cannot use the same strategy as in the non 1-decidable proof.

Theorem (H., Melnikov, Ng 2024)

There is a structure that is punctually categorical relative to 1-decidable presentations but not computably categorical relative to 1-decidable presentations.

The construction uses a mix of priority and Marker's extension and a new labelling technique. These techniques are novel.

Thank you!