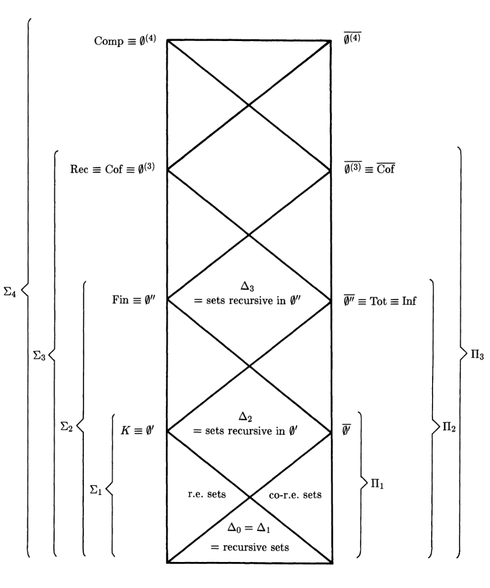
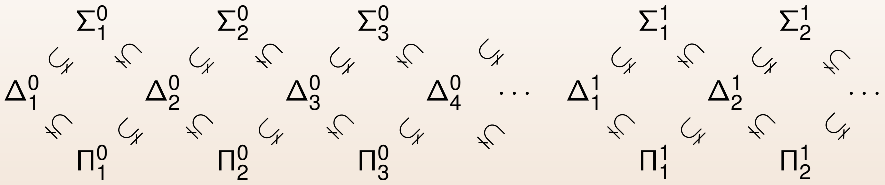
Computability and classification problems, 2.


Comp ≡ $\emptyset^{(4)}$
$\overline{\emptyset^{(4)}}$
Rec ≡ Cof ≡ $\emptyset^{(3)}$
$\overline{\emptyset^{(3)}} \equiv \overline{\text{Cof}}$
Fin ≡ $\emptyset''$
Δ_3
= sets recursive in $\emptyset''$
$\overline{\emptyset''}$ ≡ Tot ≡ Inf
$K \equiv \emptyset'$
Δ_2
= sets recursive in $\emptyset'$
$\overline{\emptyset'}$
r.e. sets
co-r.e. sets
$\Delta_0 = \Delta_1$
= recursive sets
Σ_4
Σ_3
Σ_2
Σ_1
Π_3
Π_2
Π_1

Set-quantifiers



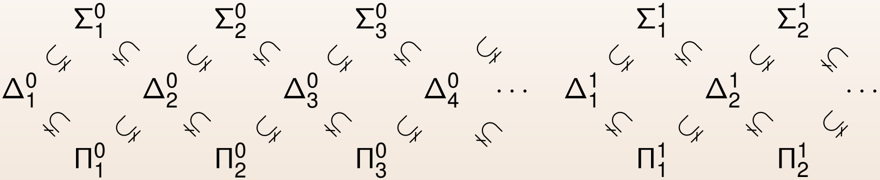
- Σ^1_1-sets are those that can be expressed using one existential functional quantifier:

$$x \in X \iff (\exists f : \mathbb{N} \to \mathbb{N})\,(\forall n)\,R(f, x, n),$$

  where R is a computable relation that uses f as an oracle.
- Π^1_1-sets are the complements of the Σ^1_1-sets:

$$x \in X \iff (\forall f : \mathbb{N} \to \mathbb{N})\,(\exists n)\,S(f, x, n).$$

Set-quantifiers



- Σ^1_1-sets are those that can be expressed using one existential functional quantifier:

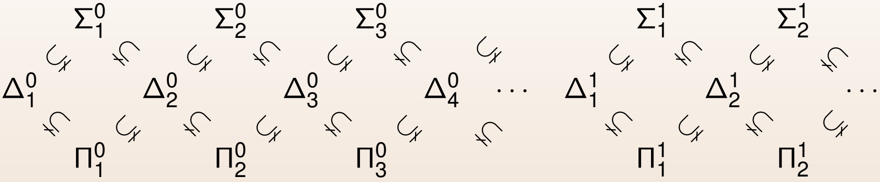
$$x \in X \iff (\exists f : \mathbb{N} \to \mathbb{N})\,(\forall n)\,R(f, x, n),$$

where R is a computable relation that uses f as an oracle.

- Π^1_1-sets are the complements of the Σ^1_1-sets:

$$x \in X \iff (\forall f : \mathbb{N} \to \mathbb{N})\,(\exists n)\,S(f, x, n).$$

Set-quantifiers



- Σ^1_1-sets are those that can be expressed using one existential functional quantifier:

$$x \in X \iff (\exists f : \mathbb{N} \to \mathbb{N})\,(\forall n)\, R(f, x, n),$$

  where R is a computable relation that uses f as an oracle.
- Π^1_1-sets are the complements of the Σ^1_1-sets:

$$x \in X \iff (\forall f : \mathbb{N} \to \mathbb{N})\,(\exists n)\, S(f, x, n).$$

Definition

A set $S \in \Sigma^n_m$ is complete in its class Σ^n_m if for every $X \in \Sigma^n_m$,

$$X \leq_1 S,$$

that is, there is a 1-1 total computable function f such that

$$x \in X \iff f(x) \in S.$$

(1-1 can be omitted).

This is similar to NP-completeness in complexity theory.

The crucial difference is that we know that the hierarchy does not collapse.

Definition

A set $S \in \Sigma_m^n$ is complete in its class Σ_m^n if for every $X \in \Sigma_m^n$,

$$X \leq_1 S,$$

that is, there is a 1-1 total computable function f such that

$$x \in X \iff f(x) \in S.$$

(1-1 can be omitted).

This is similar to NP-completeness in complexity theory.

The crucial difference is that we know that the hierarchy does not collapse.

We know that $0^{(n)}$ is Σ^0_n-complete. What are the most natural Σ^1_1-complete and Π^1_1-complete sets?

To define these sets we need to define the notion of a computable algebraic structure.

Definition

A computable presentation of a countably infinite algebraic structure A is an algebraic structure B such that:

- the set of elements of B is a computable subset of $\mathbb{N}$,
- the relations and operations of B are computable.

We know that $0^{(n)}$ is Σ^0_n-complete. What are the most natural Σ^1_1-complete and Π^1_1-complete sets?

To define these sets we need to define the notion of a computable algebraic structure.

Definition

A computable presentation of a countably infinite algebraic structure A is an algebraic structure B such that:

- the set of elements of B is a computable subset of $\mathbb{N}$,
- the relations and operations of B are computable.

We know that $0^{(n)}$ is Σ_n^0-complete. What are the most natural Σ_1^1-complete and Π_1^1-complete sets?

To define these sets we need to define the notion of a computable algebraic structure.

Definition

A computable presentation of a countably infinite algebraic structure A is an algebraic structure B such that:

- the set of elements of B is a computable subset of $\mathbb{N}$,
- the relations and operations of B are computable.

Example

Examples of computably poresentable algebraic structures:

- $(\mathbb{N}, +, \times, 0, 1)$ – the semiring of natural numbers.
- The field of algebraic numbers $\overline{Q}$.
- Any polynomial ring over a countable field you can find in a textbook.
- Any countable linear order you can think of.
- Every finitely presented group in which the word problem is decidable.

Example

Examples of computably poresentable algebraic structures:

- $(\mathbb{N}, +, \times, 0, 1)$ – the semiring of natural numbers.
- The field of algebraic numbers $\overline{Q}$.
- Any polynomial ring over a countable field you can find in a textbook.
- Any countable linear order you can think of.
- Every finitely presented group in which the word problem is decidable.

Example

Examples of computably poresentable algebraic structures:

- $(\mathbb{N}, +, \times, 0, 1)$ – the semiring of natural numbers.
- The field of algebraic numbers $\overline{Q}$.
- Any polynomial ring over a countable field you can find in a textbook.
- Any countable linear order you can think of.
- Every finitely presented group in which the word problem is decidable.

Example

Examples of computably poresentable algebraic structures:

- $(\mathbb{N}, +, \times, 0, 1)$ – the semiring of natural numbers.
- The field of algebraic numbers $\overline{Q}$.
- Any polynomial ring over a countable field you can find in a textbook.
- Any countable linear order you can think of.
- Every finitely presented group in which the word problem is decidable.

Example

Examples of computably poresentable algebraic structures:

- $(\mathbb{N}, +, \times, 0, 1)$ – the semiring of natural numbers.
- The field of algebraic numbers $\overline{Q}$.
- Any polynomial ring over a countable field you can find in a textbook.
- Any countable linear order you can think of.
- Every finitely presented group in which the word problem is decidable.

Example

Examples of computably poresentable algebraic structures:

- $(\mathbb{N}, +, \times, 0, 1)$ – the semiring of natural numbers.
- The field of algebraic numbers $\overline{Q}$.
- Any polynomial ring over a countable field you can find in a textbook.
- Any countable linear order you can think of.
- Every finitely presented group in which the word problem is decidable.

Recall that a (strict) linear order is a binary structure $(X, <)$ that satisfies:

1. $\forall x, y, z \qquad x < y$ and $y < z \implies x < z$;
2. for all $x \neq y$, either $x < y$ or $y < x$;
3. $x \not< x$.

A linear order $(X, <)$ is well-ordered or well-founded if it does not contain infinite ascending chains:

$$\ldots < x_n < \ldots < x_3 < x_2 < x_1 < x_0.$$

Recall that a (strict) linear order is a binary structure $(X, <)$ that satisfies:

1. $\forall x, y, z \qquad x < y$ and $y < z \implies x < z$;
2. for all $x \neq y$, either $x < y$ or $y < x$;
3. $x \nless x$.

A linear order $(X, <)$ is well-ordered or well-founded if it does not contain infinite ascending chains:

$$\ldots < x_n < \ldots < x_3 < x_2 < x_1 < x_0.$$

Using the Universal Turing Machine, we can produce a list of all partially computable structures:

$$R_0, R_1, R_2, \ldots.$$

Theorem (Kleene, Spector)

The index set of well-orders

$$\{e : R_e \text{ is well-ordered}\}$$

is Π^1_1 complete.

It also follows that

$$\{e : R_e \text{ is linearly ordered but not well-ordered}\}$$

is Σ^1_1-complete.

Using the Universal Turing Machine, we can produce a list of all partially computable structures:

$$R_0, R_1, R_2, \ldots.$$

Theorem (Kleene, Spector)

The index set of well-orders

$$\{e : R_e \text{ is well-ordered}\}$$

is Π^1_1 complete.

It also follows that

$$\{e : R_e \text{ is linearly ordered but not well-ordered}\}$$

is Σ^1_1-complete.

Proof.

1. R_e is well-ordered iff R_e does not have an ascending chain – this is a Π^1_1 condition (why?).
2. Fix $X \in \Pi^1_1$, so $x \in X \iff (\forall f : \mathbb{N} \to \mathbb{N})\,(\exists n)\,R(f, x, n)$.
3. For a given x, (computably) enumerate the tree T_x of all finite strings σ of natural numbers such that $(\exists n < length(\sigma))\,R(\sigma, x, n)$.
4. T_x has no infinite branches if, and only if, $x \in X$.
5. Define the Kleene–Brouwer order $KB(T_x)$ on strings in T_x: $t <_{KB} s$ when there is an n such that either:
    1. t is a prefix of s; or otherwise
    2. $t(n) < s(n)$, where n is the largest such that the n-prefixes of s and t are equal.
6. $KB(T_x)$ is a computably presented linear order.
7. T_x has no infinite branches if, and only if, $KB(T_x)$ is a well-ordering.

□

Proof.

1. R_e is well-ordered iff R_e does not have an ascending chain – this is a Π^1_1 condition (why?).
2. Fix $X \in \Pi^1_1$, so $x \in X \iff (\forall f : \mathbb{N} \to \mathbb{N})\, (\exists n)\, R(f, x, n)$.
3. For a given x, (computably) enumerate the tree T_x of all finite strings σ of natural numbers such that $(\exists n < length(\sigma))\, R(\sigma, x, n)$.
4. T_x has no infinite branches if, and only if, $x \in X$.
5. Define the Kleene–Brouwer order $KB(T_x)$ on strings in T_x: $t <_{KB} s$ when there is an n such that either:
   1. t is a prefix of s; or otherwise
   2. $t(n) < s(n)$, where n is the largest such that the n-prefixes of s and t are equal.
6. $KB(T_x)$ is a computably presented linear order.
7. T_x has no infinite branches if, and only if, $KB(T_x)$ is a well-ordering.

Proof.

1. R_e is well-ordered iff R_e does not have an ascending chain – this is a Π^1_1 condition (why?).
2. Fix $X \in \Pi^1_1$, so $x \in X \iff (\forall f : \mathbb{N} \to \mathbb{N})\,(\exists n)\,R(f, x, n)$.
3. For a given x, (computably) enumerate the tree T_x of all finite strings σ of natural numbers such that $(\exists n < length(\sigma))\,R(\sigma, x, n)$.
4. T_x has no infinite branches if, and only if, $x \in X$.
5. Define the Kleene–Brouwer order $KB(T_x)$ on strings in T_x: $t <_{KB} s$ when there is an n such that either:
    1. t is a prefix of s; or otherwise
    2. $t(n) < s(n)$, where n is the largest such that the n-prefixes of s and t are equal.
6. $KB(T_x)$ is a computably presented linear order.
7. T_x has no infinite branches if, and only if, $KB(T_x)$ is a well-ordering.

Proof.

1. R_e is well-ordered iff R_e does not have an ascending chain – this is a Π^1_1 condition (why?).
2. Fix $X \in \Pi^1_1$, so $x \in X \iff (\forall f : \mathbb{N} \to \mathbb{N})\,(\exists n)\, R(f, x, n)$.
3. For a given x, (computably) enumerate the tree T_x of all finite strings σ of natural numbers such that $(\exists n < length(\sigma))\, R(\sigma, x, n)$.
4. T_x has no infinite branches if, and only if, $x \in X$.
5. Define the Kleene–Brouwer order $KB(T_x)$ on strings in T_x: $t <_{KB} s$ when there is an n such that either:
    1. t is a prefix of s; or otherwise
    2. $t(n) < s(n)$, where n is the largest such that the n-prefixes of s and t are equal.
6. $KB(T_x)$ is a computably presented linear order.
7. T_x has no infinite branches if, and only if, $KB(T_x)$ is a well-ordering.

Proof.

1. R_e is well-ordered iff R_e does not have an ascending chain – this is a Π^1_1 condition (why?).
2. Fix $X \in \Pi^1_1$, so $x \in X \iff (\forall f : \mathbb{N} \to \mathbb{N}) (\exists n) R(f, x, n)$.
3. For a given x, (computably) enumerate the tree T_x of all finite strings σ of natural numbers such that $(\exists n < length(\sigma))\, R(\sigma, x, n)$.
4. T_x has no infinite branches if, and only if, $x \in X$.
5. Define the Kleene–Brouwer order $KB(T_x)$ on strings in T_x: $t <_{KB} s$ when there is an n such that either:
    1. t is a prefix of s; or otherwise
    2. $t(n) < s(n)$, where n is the largest such that the n-prefixes of s and t are equal.
6. $KB(T_x)$ is a computably presented linear order.
7. T_x has no infinite branches if, and only if, $KB(T_x)$ is a well-ordering.

Proof.

1. R_e is well-ordered iff R_e does not have an ascending chain – this is a Π^1_1 condition (why?).
2. Fix $X \in \Pi^1_1$, so $x \in X \iff (\forall f : \mathbb{N} \to \mathbb{N})(\exists n) R(f, x, n)$.
3. For a given x, (computably) enumerate the tree T_x of all finite strings σ of natural numbers such that $(\exists n < length(\sigma)) R(\sigma, x, n)$.
4. T_x has no infinite branches if, and only if, $x \in X$.
5. Define the Kleene–Brouwer order $KB(T_x)$ on strings in T_x: $t <_{KB} s$ when there is an n such that either:
    1. t is a prefix of s; or otherwise
    2. $t(n) < s(n)$, where n is the largest such that the n-prefixes of s and t are equal.
6. $KB(T_x)$ is a computably presented linear order.
7. T_x has no infinite branches if, and only if, $KB(T_x)$ is a well-ordering.

Proof.

1. R_e is well-ordered iff R_e does not have an ascending chain – this is a Π^1_1 condition (why?).
2. Fix $X \in \Pi^1_1$, so $x \in X \iff (\forall f : \mathbb{N} \to \mathbb{N})\,(\exists n)\,R(f, x, n)$.
3. For a given x, (computably) enumerate the tree T_x of all finite strings σ of natural numbers such that $(\exists n < length(\sigma))\,R(\sigma, x, n)$.
4. T_x has no infinite branches if, and only if, $x \in X$.
5. Define the Kleene–Brouwer order $KB(T_x)$ on strings in T_x: $t <_{KB} s$ when there is an n such that either:
   1. t is a prefix of s; or otherwise
   2. $t(n) < s(n)$, where n is the largest such that the n-prefixes of s and t are equal.
6. $KB(T_x)$ is a computably presented linear order.
7. T_x has no infinite branches if, and only if, $KB(T_x)$ is a well-ordering.

One cool property of the Kleene-Spector theorem is that it can be relativized to any oracle.

Fix an arbitrary oracle Y.

Definition

A set X is $\Pi^1_1(Y)$ if:

$$x \in X \iff (\forall f : \mathbb{N} \to \mathbb{N})\,(\exists n)\, S^Y(f, x, n)$$

where S is a predicate that is computable relative to Y.

One cool property of the Kleene-Spector theorem is that it can be relativized to any oracle.

Fix an arbitrary oracle Y.

Definition

A set X is $\Pi^1_1(Y)$ if:

$$x \in X \iff (\forall f : \mathbb{N} \to \mathbb{N})\,(\exists n)\, S^Y(f, x, n)$$

where S is a predicate that is computable relative to Y.

Produce a list of all structures partially computable relative to Y:

$$R_0^Y, R_1^Y, R_2^Y, \ldots.$$

Theorem (Kleene-Spector relativised)

The index set of well-orders

$$\{e : R_e^Y \text{ is well-ordered}\}$$

is $\Pi_1^1(Y)$-complete.

Proof:

1. All steps relativise to Y (just substitute 'computable' with 'computable relative to Y' throughout).
2. Furthermore, $T_x^Y \to KB(T_x^Y)$ is a computable transformation (even though T^Y and $KB(T_x^Y)$ are not necessarily computable).

Produce a list of all structures partially computable relative to Y:

$$R_0^Y, R_1^Y, R_2^Y, \ldots.$$

Theorem (Kleene-Spector relativised)

The index set of well-orders

$$\{e : R_e^Y \text{ is well-ordered}\}$$

is $\Pi_1^1(Y)$-complete.

Proof:

1. All steps relativise to Y (just substitute 'computable' with 'computable relative to Y' throughout).
2. Furthermore, $T_x^Y \to KB(T_x^Y)$ is a computable transformation (even though T^Y and $KB(T_x^Y)$ are not necessarily computable).

Produce a list of all structures partially computable relative to Y:

$$R_0^Y, R_1^Y, R_2^Y, \ldots.$$

Theorem (Kleene-Spector relativised)

The index set of well-orders

$$\{e : R_e^Y \text{ is well-ordered}\}$$

is $\Pi_1^1(Y)$-complete.

Proof:

1. All steps relativise to Y (just substitute 'computable' with 'computable relative to Y' throughout).
2. Furthermore, $T_x^Y \to KB(T_x^Y)$ is a computable transformation (even though T^Y and $KB(T_x^Y)$ are not necessarily computable).

Produce a list of all structures partially computable relative to Y:

$$R_0^Y, R_1^Y, R_2^Y, \ldots.$$

Theorem (Kleene-Spector relativised)

The index set of well-orders

$$\{e : R_e^Y \text{ is well-ordered}\}$$

is $\Pi_1^1(Y)$-complete.

Proof:

1. All steps relativise to Y (just substitute 'computable' with 'computable relative to Y' throughout).
2. Furthermore, $T_x^Y \to KB(T_x^Y)$ is a computable transformation (even though T^Y and $KB(T_x^Y)$ are not necessarily computable).

3 It follows from the partially relativized s-m-n Theorem (something that we skipped) that the index of $KB(T_x^Y)$ in the enumeration

$$R_0^Y, R_1^Y, R_2^Y, \ldots.$$

is computable from x, and not merely computable relative to Y.

4 This is because we transform the programs that use the oracles rather than actually use any information about the oracle.

5 This gives a computable reduction h for any $S \in \Pi_1^1(Y)$:

$$x \in S \iff h(e) \in \{e : R_e^Y \text{ is a well-order}\}.$$

$\square$

3 It follows from the partially relativized s-m-n Theorem (something that we skipped) that the index of $KB(T_x^Y)$ in the enumeration

$$R_0^Y, R_1^Y, R_2^Y, \ldots.$$

is computable from x, and not merely computable relative to Y.

4 This is because we transform the programs that use the oracles rather than actually use any information about the oracle.

5 This gives a computable reduction h for any $S \in \Pi_1^1(Y)$:

$$x \in S \iff h(e) \in \{e : R_e^Y \text{ is a well-order}\}.$$

$\square$

3 It follows from the partially relativized s-m-n Theorem (something that we skipped) that the index of $KB(T^Y_x)$ in the enumeration

$$R^Y_0, R^Y_1, R^Y_2, \dots.$$

is computable from x, and not merely computable relative to Y.

4 This is because we transform the programs that use the oracles rather than actually use any information about the oracle.

5 This gives a computable reduction h for any $S \in \Pi^1_1(Y)$:

$$x \in S \iff h(e) \in \{e : R^Y_e \text{ is a well-order}\}.$$

□

3. It follows from the partially relativized s-m-n Theorem (something that we skipped) that the index of $KB(T_x^Y)$ in the enumeration

$$R_0^Y, R_1^Y, R_2^Y, \ldots.$$

is computable from x, and not merely computable relative to Y.

4. This is because we transform the programs that use the oracles rather than actually use any information about the oracle.

5. This gives a computable reduction h for any $S \in \Pi_1^1(Y)$:

$$x \in S \iff h(e) \in \{e : R_e^Y \text{ is a well-order}\}.$$

□

3 It follows from the partially relativized s-m-n Theorem (something that we skipped) that the index of $KB(T_x^Y)$ in the enumeration

$$R_0^Y, R_1^Y, R_2^Y, \ldots.$$

is computable from x, and not merely computable relative to Y.

4 This is because we transform the programs that use the oracles rather than actually use any information about the oracle.

5 This gives a computable reduction h for any $S \in \Pi_1^1(Y)$:

$$x \in S \iff h(e) \in \{e : R_e^Y \text{ is a well-order}\}.$$

□

3 It follows from the partially relativized s-m-n Theorem (something that we skipped) that the index of $KB(T_x^Y)$ in the enumeration

$$R_0^Y, R_1^Y, R_2^Y, \ldots.$$

is computable from x, and not merely computable relative to Y.

4 This is because we transform the programs that use the oracles rather than actually use any information about the oracle.

5 This gives a computable reduction h for any $S \in \Pi_1^1(Y)$:

$$x \in S \iff h(e) \in \{e : R_e^Y \text{ is a well-order}\}.$$

$\square$

1. Any Π^1_1 relation $\forall f \exists n S(f, Y; n, e)$ with a set parameter Y is computably equivalent to the statement of the form

$$R_e^Y \text{ is well ordered.}$$

2. It follows that the property of being well-founded is an intrinsically second-order property.
3. There is no "local" way to test whether an arbitrary countable linear order is well-founded.
4. This works for arbitrary countable linear orders, not for just computable linear orders. (Every linear order is computable relative to some oracle.)

There is no property characterising well-orders among all linear orders which would be simpler than the definition.

1. Any Π^1_1 relation $\forall f \exists n S(f, Y; n, e)$ with a set parameter Y is computably equivalent to the statement of the form

$$R^Y_e \text{ is well ordered.}$$

2. It follows that the property of being well-founded is an intrinsically second-order property.
3. There is no “local” way to test whether an arbitrary countable linear order is well-founded.
4. This works for arbitrary countable linear orders, not for just computable linear orders. (Every linear order is computable relative to some oracle.)

There is no property characterising well-orders among all linear orders which would be simpler than the definition.

1. Any Π^1_1 relation $\forall f \exists n S(f, Y; n, e)$ with a set parameter Y is computably equivalent to the statement of the form

$$R_e^Y \text{ is well ordered.}$$

2. It follows that the property of being well-founded is an intrinsically second-order property.
3. There is no “local” way to test whether an arbitrary countable linear order is well-founded.
4. This works for arbitrary countable linear orders, not for just computable linear orders. (Every linear order is computable relative to some oracle.)

There is no property characterising well-orders among all linear orders which would be simpler than the definition.

1. Any Π^1_1 relation $\forall f \exists n S(f, Y; n, e)$ with a set parameter Y is computably equivalent to the statement of the form

$$R_e^Y \text{ is well ordered.}$$

2. It follows that the property of being well-founded is an intrinsically second-order property.
3. There is no “local” way to test whether an arbitrary countable linear order is well-founded.
4. This works for arbitrary countable linear orders, not for just computable linear orders. (Every linear order is computable relative to some oracle.)

There is no property characterising well-orders among all linear orders which would be simpler than the definition.

1. Any Π^1_1 relation $\forall f \exists n S(f, Y; n, e)$ with a set parameter Y is computably equivalent to the statement of the form

$$R^Y_e \text{ is well ordered.}$$

2. It follows that the property of being well-founded is an intrinsically second-order property.
3. There is no "local" way to test whether an arbitrary countable linear order is well-founded.
4. This works for arbitrary countable linear orders, not for just computable linear orders. (Every linear order is computable relative to some oracle.)

There is no property characterising well-orders among all linear orders which would be simpler than the definition.

There are however plenty of examples when a property is actually simpler than its definition suggests.

Example

Recall that the *free abelian group* A_α of rank $\alpha \in \mathbb{N} \cup \{\omega\}$ is

$$\bigoplus_{i \leq \alpha} \mathbb{Z}.$$

A countable group G is free iff, for some α, $G \cong A_\alpha$:

$$(\exists \alpha)(\exists f)(f \text{ is an isomorphism } G \to A_\alpha).$$

The definition is Σ^1_1.

There are however plenty of examples when a property is actually simpler than its definition suggests.

Example

Recall that the *free abelian group* A_α of rank $\alpha \in \mathbb{N} \cup \{\omega\}$ is

$$\bigoplus_{i \leq \alpha} \mathbb{Z}.$$

A countable group G is free iff, for some α, $G \cong A_\alpha$:

$$(\exists \alpha)(\exists f)(f \text{ is an isomorphism } G \to A_\alpha).$$

The definition is Σ^1_1.

There are however plenty of examples when a property is actually simpler than its definition suggests.

Example

Recall that the *free abelian group* A_α of rank $\alpha \in \mathbb{N} \cup \{\omega\}$ is

$$\bigoplus_{i \leq \alpha} \mathbb{Z}.$$

A countable group G is free iff, for some α, $G \cong A_\alpha$:

$$(\exists \alpha)(\exists f)(f \text{ is an isomorphism } G \to A_\alpha).$$

The definition is Σ^1_1.

But we understand these groups really quite well, so there should be a better "local" way to test whether G is free abelian.

Fact

The index set of free abelian groups is arithmetical (Π^0_2).

Proof idea.

- To get the upper bound of Π^0_2, try to build a free basis of G.
- Use Pontryagin's criterion:

$$a_1, \ldots, a_n \text{ are freely independent } \iff$$

$$(\forall m, m_i \in \mathbb{Z})\,[(\exists x \in G)(mx = \sum_i m_i a_i) \implies \&_i m | m_i].$$

- Attempt to build a free basis.
- If we never get stuck, then G has to be free.
- Note this is relativizable to any oracle.

$\square$

But we understand these groups really quite well, so there should be a better "local" way to test whether G is free abelian.

Fact

The index set of free abelian groups is arithmetical (Π^0_2).

Proof idea.

- To get the upper bound of Π^0_2, try to build a free basis of G.
- Use Pontryagin's criterion:

$$a_1, \dots, a_n \text{ are freely independent} \iff$$

$$(\forall m, m_i \in \mathbb{Z})\,[(\exists x \in G)(mx = \sum_i m_i a_i) \implies \&_i m | m_i].$$

- Attempt to build a free basis.
- If we never get stuck, then G has to be free.
- Note this is relativizable to any oracle.

□

But we understand these groups really quite well, so there should be a better "local" way to test whether G is free abelian.

Fact

The index set of free abelian groups is arithmetical (Π^0_2).

Proof idea.

- To get the upper bound of Π^0_2, try to build a free basis of G.
- Use Pontryagin's criterion:

$$a_1, \dots, a_n \text{ are freely independent} \iff$$

$$(\forall m, m_i \in \mathbb{Z}) [(\exists x \in G)(mx = \sum_i m_i a_i) \implies \&_i m | m_i].$$

- Attempt to build a free basis.
- If we never get stuck, then G has to be free.
- Note this is relativizable to any oracle.

□

But we understand these groups really quite well, so there should be a better "local" way to test whether G is free abelian.

Fact

The index set of free abelian groups is arithmetical (Π^0_2).

Proof idea.

- To get the upper bound of Π^0_2, try to build a free basis of G.
- Use Pontryagin's criterion:

$$a_1, \ldots, a_n \text{ are freely independent} \iff$$

$$(\forall m, m_i \in \mathbb{Z})\,[(\exists x \in G)(mx = \sum_i m_i a_i) \implies \&_i m | m_i].$$

- Attempt to build a free basis.
- If we never get stuck, then G has to be free.
- Note this is relativizable to any oracle.

□

But we understand these groups really quite well, so there should be a better "local" way to test whether G is free abelian.

Fact

The index set of free abelian groups is arithmetical (Π^0_2).

Proof idea.

- To get the upper bound of Π^0_2, try to build a free basis of G.
- Use Pontryagin's criterion:

$$a_1, \ldots, a_n \text{ are freely independent } \iff$$

$$(\forall m, m_i \in \mathbb{Z})\,[(\exists x \in G)(mx = \sum_i m_i a_i) \implies \&_i m | m_i].$$

- Attempt to build a free basis.
- If we never get stuck, then G has to be free.
- Note this is relativizable to any oracle.

□

But we understand these groups really quite well, so there should be a better "local" way to test whether G is free abelian.

Fact

The index set of free abelian groups is arithmetical (Π^0_2).

Proof idea.

- To get the upper bound of Π^0_2, try to build a free basis of G.
- Use Pontryagin's criterion:

$$a_1, \dots, a_n \text{ are freely independent } \iff$$

$$(\forall m, m_i \in \mathbb{Z})\,[(\exists x \in G)(mx = \sum_i m_i a_i) \implies \&_i m | m_i].$$

- Attempt to build a free basis.
- If we never get stuck, then G has to be free.
- Note this is relativizable to any oracle.

□

But we understand these groups really quite well, so there should be a better “local” way to test whether G is free abelian.

Fact

The index set of free abelian groups is arithmetical (Π^0_2).

Proof idea.

- To get the upper bound of Π^0_2, try to build a free basis of G.
- Use Pontryagin’s criterion:

$$a_1, \ldots, a_n \text{ are freely independent } \iff$$

$$(\forall m, m_i \in \mathbb{Z})\,[(\exists x \in G)(mx = \sum_i m_i a_i) \implies \&_i m | m_i].$$

- Attempt to build a free basis.
- If we never get stuck, then G has to be free.
- Note this is relativizable to any oracle.

□

Part 2: Applications to classification problems

One of the central problems of abelian group theory:

Problem

Determine whether a given abelian group G splits into the direct sum of its proper subgroups:

$$G \cong A \oplus B.$$

When you read Fuchs you realise that a *local* property would be most desirable.

Example

An additive abelian group $(D, +)$ is divisible if, for every $k \in \mathbb{N}$ and each $a \in A$,

$$(\exists b \in A)\ kb = b + b + \ldots (n\ times) \ldots + b = a.$$

For example, the group of the rationals $(\mathbb{Q}, +)$ is like that. Note that this is a "local" property. So, if G is not divisible and D is a divisible subgroup of G, then $G = D \oplus X$ (both non-trivial).

One of the central problems of abelian group theory:

Problem

Determine whether a given abelian group G splits into the direct sum of its proper subgroups:

$$G \cong A \oplus B.$$

When you read Fuchs you realise that a *local* property would be most desirable.

Example

An additive abelian group $(D, +)$ is divisible if, for every $k \in \mathbb{N}$ and each $a \in A$,

$$(\exists b \in A)\ kb = b + b + \ldots (n\ times) \ldots + b = a.$$

For example, the group of the rationals $(\mathbb{Q}, +)$ is like that. Note that this is a "local" property. So, if G is not divisible and D is a divisible subgroup of G, then $G = D \oplus X$ (both non-trivial).

One of the central problems of abelian group theory:

Problem

Determine whether a given abelian group G splits into the direct sum of its proper subgroups:

$$G \cong A \oplus B.$$

When you read Fuchs you realise that a *local* property would be most desirable.

Example

An additive abelian group $(D, +)$ is divisible if, for every $k \in \mathbb{N}$ and each $a \in A$,

$$(\exists b \in A)\ kb = b + b + \dots (n\ times) \dots + b = a.$$

For example, the group of the rationals $(\mathbb{Q}, +)$ is like that. Note that this is a "local" property. So, if G is not divisible and D is a divisible subgroup of G, then $G = D \oplus X$ (both non-trivial).

In 2012 Kudinov and I asked whether the property of being directly indecomposable could be Π^1_1-complete.

Theorem (Riggs 2017)

The index set of directly indecomposable groups Π^1_1-complete.

Proof.

Design a computable transformation which, given a computable tree T, produces a torsion-free abelian group with the property:

$$T \text{ has an infinite branch} \iff G(T) \text{ non-trivially directly splits.}$$

This was inspired by an earlier result of Downey and Montalban,
- who were using an earlier work of Hjorth,
- who was building on indecomposability techniques developed by Fuchs,
- who was based on Pontryagin's well-known work,
- Pontryagin was a blind genius mathematician!

□

In 2012 Kudinov and I asked whether the property of being directly indecomposable could be Π^1_1-complete.

Theorem (Riggs 2017)

The index set of directly indecomposable groups Π^1_1-complete.

Proof.

Design a computable transformation which, given a computable tree T, produces a torsion-free abelian group with the property:

$$T \text{ has an infinite branch} \iff G(T) \text{ non-trivially directly splits.}$$

This was inspired by an earlier result of Downey and Montalban,
- who were using an earlier work of Hjorth,
- who was building on indecomposability techniques developed by Fuchs,
- who was based on Pontryagin's well-known work,
- Pontryagin was a blind genius mathematician!

□

In 2012 Kudinov and I asked whether the property of being directly indecomposable could be Π^1_1-complete.

Theorem (Riggs 2017)

The index set of directly indecomposable groups Π^1_1-complete.

Proof.

Design a computable transformation which, given a computable tree T, produces a torsion-free abelian group with the property:

$$T \text{ has an infinite branch} \iff G(T) \text{ non-trivially directly splits.}$$

This was inspired by an earlier result of Downey and Montalban,
- who were using an earlier work of Hjorth,
- who was building on indecomposability techniques developed by Fuchs,
- who was based on Pontryagin's well-known work,
- Pontryagin was a blind genius mathematician!

□

In 2012 Kudinov and I asked whether the property of being directly indecomposable could be Π^1_1-complete.

Theorem (Riggs 2017)

The index set of directly indecomposable groups Π^1_1-complete.

Proof.

Design a computable transformation which, given a computable tree T, produces a torsion-free abelian group with the property:

$$T \text{ has an infinite branch} \iff G(T) \text{ non-trivially directly splits.}$$

This was inspired by an earlier result of Downey and Montalban,
- who were using an earlier work of Hjorth,
- who was building on indecomposability techniques developed by Fuchs,
- who was based on Pontryagin's well-known work,
- Pontryagin was a blind genius mathematician!

□

In 2012 Kudinov and I asked whether the property of being directly indecomposable could be Π^1_1-complete.

Theorem (Riggs 2017)

The index set of directly indecomposable groups Π^1_1-complete.

Proof.

Design a computable transformation which, given a computable tree T, produces a torsion-free abelian group with the property:

$$T \text{ has an infinite branch} \iff G(T) \text{ non-trivially directly splits.}$$

This was inspired by an earlier result of Downey and Montalban,
- who were using an earlier work of Hjorth,
- who was building on indecomposability techniques developed by Fuchs,
- who was based on Pontryagin's well-known work,
- Pontryagin was a blind genius mathematician!

□

In 2012 Kudinov and I asked whether the property of being directly indecomposable could be Π^1_1-complete.

Theorem (Riggs 2017)

The index set of directly indecomposable groups Π^1_1-complete.

Proof.

Design a computable transformation which, given a computable tree T, produces a torsion-free abelian group with the property:

$$T \text{ has an infinite branch} \iff G(T) \text{ non-trivially directly splits.}$$

This was inspired by an earlier result of Downey and Montalban,
- who were using an earlier work of Hjorth,
- who was building on indecomposability techniques developed by Fuchs,
- who was based on Pontryagin's well-known work,
- Pontryagin was a blind genius mathematician!

□

In 2012 Kudinov and I asked whether the property of being directly indecomposable could be Π^1_1-complete.

Theorem (Riggs 2017)

The index set of directly indecomposable groups Π^1_1-complete.

Proof.

Design a computable transformation which, given a computable tree T, produces a torsion-free abelian group with the property:

$$T \text{ has an infinite branch} \iff G(T) \text{ non-trivially directly splits.}$$

This was inspired by an earlier result of Downey and Montalban,
- who were using an earlier work of Hjorth,
- who was building on indecomposability techniques developed by Fuchs,
- who was based on Pontryagin's well-known work,
- Pontryagin was a blind genius mathematician!

□

1. As before, the result of Riggs is fully relativizable.
2. It follows that the property of (in)decomposability is intrinsically global/second-order.
3. So it follows that there is no local characterisation of being decomposable.
4. If you are an abelian group theorist trying to find such a characterisation STOP NOW.
5. In their defence (after 50+ years of trying) abelian group theorists did realise that one should not hope for any reasonable characterisation of decomposability.

1. As before, the result of Riggs is fully relativizable.
2. It follows that the property of (in)decomposability is intrinsically global/second-order.
3. So it follows that there is no local characterisation of being decomposable.
4. If you are an abelian group theorist trying to find such a characterisation STOP NOW.
5. In their defence (after 50+ years of trying) abelian group theorists did realise that one should not hope for any reasonable characterisation of decomposability.

1. As before, the result of Riggs is fully relativizable.
2. It follows that the property of (in)decomposability is intrinsically global/second-order.
3. So it follows that there is no local characterisation of being decomposable.
4. If you are an abelian group theorist trying to find such a characterisation STOP NOW.
5. In their defence (after 50+ years of trying) abelian group theorists did realise that one should not hope for any reasonable characterisation of decomposability.

1. As before, the result of Riggs is fully relativizable.
2. It follows that the property of (in)decomposability is intrinsically global/second-order.
3. So it follows that there is no local characterisation of being decomposable.
4. If you are an abelian group theorist trying to find such a characterisation STOP NOW.
5. In their defence (after 50+ years of trying) abelian group theorists did realise that one should not hope for any reasonable characterisation of decomposability.

1. As before, the result of Riggs is fully relativizable.
2. It follows that the property of (in)decomposability is intrinsically global/second-order.
3. So it follows that there is no local characterisation of being decomposable.
4. If you are an abelian group theorist trying to find such a characterisation STOP NOW.
5. In their defence (after 50+ years of trying) abelian group theorists did realise that one should not hope for any reasonable characterisation of decomposability.

Here is another property which is also naturally global (not first-order):

Definition (Baer)

A group G is completely decomposable if it splits into the direct sum

$$G \cong \bigoplus_i H_i,$$

where each of the H_i is a subgroup of the additive group of the rationals $(\mathbb{Q}, +)$.

This does not look like a local property at all.

We need to ask if there exist subsets H_i of G s.t. $G \cong \bigoplus_i H_i$.

In a way, this looks even worse that just direct decomposability.

Here is another property which is also naturally global (not first-order):

Definition (Baer)

A group G is completely decomposable if it splits into the direct sum

$$G \cong \bigoplus_i H_i,$$

where each of the H_i is a subgroup of the additive group of the rationals $(\mathbb{Q}, +)$.

This does not look like a local property at all.

We need to ask if there exist subsets H_i of G s.t. $G \cong \bigoplus_i H_i$.

In a way, this looks even worse that just direct decomposability.

Here is another property which is also naturally global (not first-order):

Definition (Baer)

A group G is completely decomposable if it splits into the direct sum

$$G \cong \bigoplus_i H_i,$$

where each of the H_i is a subgroup of the additive group of the rationals $(\mathbb{Q}, +)$.

This does not look like a local property at all.

We need to ask if there exist subsets H_i of G s.t. $G \cong \bigoplus_i H_i$.

In a way, this looks even worse that just direct decomposability.

Here is another property which is also naturally global (not first-order):

Definition (Baer)

A group G is completely decomposable if it splits into the direct sum

$$G \cong \bigoplus_i H_i,$$

where each of the H_i is a subgroup of the additive group of the rationals $(\mathbb{Q}, +)$.

This does not look like a local property at all.

We need to ask if there exist subsets H_i of G s.t. $G \cong \bigoplus_i H_i$.

In a way, this looks even worse that just direct decomposability.

Theorem (Downey and Melnikov, 2014)

The index set of completely indecomposable groups Σ^0_7.

Proof idea.

1. Design a new independence property inspired by Nielsen transformations and Pontryagin's (abelian) freeness criterion.
2. Attempt to build a "decomposition basis" of a given G.
3. Simultaneously, use some specific combinatorics to arithmetically list all isomorphism types of computable completely decomposable groups:

$$C_0, C_1, \ldots.$$

4. Use the basis to show that the property $(\exists i) G \cong C_i$ is indeed arithmetical (Σ^0_7) and not merely Σ^1_1.

□

Theorem (Downey and Melnikov, 2014)

The index set of completely indecomposable groups Σ^0_7.

Proof idea.

1. Design a new independence property inspired by Nielsen transformations and Pontryagin's (abelian) freeness criterion.
2. Attempt to build a "decomposition basis" of a given G.
3. Simultaneously, use some specific combinatorics to arithmetically list all isomorphism types of computable completely decomposable groups:

$$C_0, C_1, \ldots .$$

4. Use the basis to show that the property $(\exists i) G \cong C_i$ is indeed arithmetical (Σ^0_7) and not merely Σ^1_1.

□

Theorem (Downey and Melnikov, 2014)

The index set of completely indecomposable groups Σ^0_7.

Proof idea.

1. Design a new independence property inspired by Nielsen transformations and Pontryagin's (abelian) freeness criterion.
2. Attempt to build a "decomposition basis" of a given G.
3. Simultaneously, use some specific combinatorics to arithmetically list all isomorphism types of computable completely decomposable groups:

$$C_0, C_1, \ldots.$$

4. Use the basis to show that the property $(\exists i) G \cong C_i$ is indeed arithmetical (Σ^0_7) and not merely Σ^1_1.

□

Theorem (Downey and Melnikov, 2014)

The index set of completely indecomposable groups Σ^0_7.

Proof idea.

1. Design a new independence property inspired by Nielsen transformations and Pontryagin's (abelian) freeness criterion.
2. Attempt to build a "decomposition basis" of a given G.
3. Simultaneously, use some specific combinatorics to arithmetically list all isomorphism types of computable completely decomposable groups:

$$C_0, C_1, \ldots.$$

4. Use the basis to show that the property $(\exists i) G \cong C_i$ is indeed arithmetical (Σ^0_7) and not merely Σ^1_1.

□

Theorem (Downey and Melnikov, 2014)

The index set of completely indecomposable groups Σ^0_7.

Proof idea.

1. Design a new independence property inspired by Nielsen transformations and Pontryagin's (abelian) freeness criterion.
2. Attempt to build a "decomposition basis" of a given G.
3. Simultaneously, use some specific combinatorics to arithmetically list all isomorphism types of computable completely decomposable groups:

$$C_0, C_1, \ldots.$$

4. Use the basis to show that the property $(\exists i) G \cong C_i$ is indeed arithmetical (Σ^0_7) and not merely Σ^1_1.

□

1. As before, the proof can be relativised to any oracle.
2. So complete decomposability is a local property.
3. There is a local independence property describing such groups similar to how Nielsen transformations (in a way) describe free groups.
4. For example, if $G \cong \bigoplus_i H_i$ where all the H_i are isomorphic, then there is a set of primes S so that independence looks as follows:

$$g_1, \ldots, g_k \text{ are S-independent } \iff$$

$$(\forall p \in S)(\forall m_1, \ldots, m_k \in \mathbb{Z})\, [p | \sum_i m_i g_i \implies \&_{i \leq k} m_i | k].$$

5. In the general case it is more complex but still tractable.

1. As before, the proof can be relativised to any oracle.
2. So complete decomposability is a local property.
3. There is a local independence property describing such groups similar to how Nielsen transformations (in a way) describe free groups.
4. For example, if $G \cong \bigoplus_i H_i$ where all the H_i are isomorphic, then there is a set of primes S so that independence looks as follows:

$$g_1, \ldots, g_k \text{ are S-independent} \iff$$

$$(\forall p \in S)(\forall m_1, \ldots, m_k \in \mathbb{Z})\,[p| \sum_i m_i g_i \implies \&_{i \leq k} m_i | k].$$

5. In the general case it is more complex but still tractable.

1. As before, the proof can be relativised to any oracle.
2. So complete decomposability is a local property.
3. There is a local independence property describing such groups similar to how Nielsen transformations (in a way) describe free groups.
4. For example, if $G \cong \bigoplus_i H_i$ where all the H_i are isomorphic, then there is a set of primes S so that independence looks as follows:

$$g_1, \ldots, g_k \text{ are S-independent } \iff$$

$$(\forall p \in S)(\forall m_1, \ldots, m_k \in \mathbb{Z})\, [p| \sum_i m_i g_i \implies \&_{i \leq k} m_i | k].$$

5. In the general case it is more complex but still tractable.

1. As before, the proof can be relativised to any oracle.
2. So complete decomposability is a local property.
3. There is a local independence property describing such groups similar to how Nielsen transformations (in a way) describe free groups.
4. For example, if $G \cong \bigoplus_i H_i$ where all the H_i are isomorphic, then there is a set of primes S so that independence looks as follows:

$$g_1, \dots, g_k \text{ are S-independent } \iff$$

$$(\forall p \in S)(\forall m_1, \dots, m_k \in \mathbb{Z})\, [p | \sum_i m_i g_i \implies \&_{i \leq k} m_i | k].$$

5. In the general case it is more complex but still tractable.

1. As before, the proof can be relativised to any oracle.
2. So complete decomposability is a local property.
3. There is a local independence property describing such groups similar to how Nielsen transformations (in a way) describe free groups.
4. For example, if $G \cong \bigoplus_i H_i$ where all the H_i are isomorphic, then there is a set of primes S so that independence looks as follows:

$$g_1, \ldots, g_k \text{ are S-independent} \iff$$

$$(\forall p \in S)(\forall m_1, \ldots, m_k \in \mathbb{Z})\,[p| \sum_i m_i g_i \implies \&_{i \leq k} m_i | k].$$

5. In the general case it is more complex but still tractable.

This approach works for separable structures as well – we omit the details.

Some results by various authors, including Spector, Knight, Goncharov, Downey, McNicholl, M., Turetsky, Nies, Solecki, McCoy, and many others:

Characterisation Problem	Complexity
Well-foundedness of a linear order	Π^1_1-comp
Atomicity of a Boolean algebra	Π^1_1-comp.
Direct decomposability of a group	Σ^1_1-comp.
Complete decomposability of a group	Σ^0_7
Freeness of a group	Π^0_4-comp.
Being a separable Lebesgue space	Π^0_3
Being a representation of $C[0,1]$	Π^0_5
Being a connected compact Polish space	Π^0_3(-comp.)
Being a locally compact Polish space	Π^1_1-comp
Being a compact Polish group	Π^0_3(-comp.)

§2.1 The isomorphism problem.

Let K be a class of countable structures.

Definition

The isomorphism problem for K is the set

$$\{2^x 3^y \ : \ M_x, M_y \in K \text{ and } M_x \cong M_y\}.$$

The index set $\{e : M_e \in K\}$ of K reflexes the complexity of the characterisation problem.

The isomorphism problem measures how hard it is to classify structures up to isomorphism.

More results by various authors:

Isomorphism problem	Complexity
countable torsion-free abelian groups	Σ^1_1-comp.
countable completely decomposable groups	Σ^0_7
countable Boolean algebras	Σ^1_1-comp.
countable linear orders	Σ^1_1-comp.
Separable L^p-spaces, $p \neq 0$	co-3-Σ^0_3-comp.
Connected Polish abelian groups	Σ^1_1-comp.

The general framework agrees with the thesis:

Mathematical structures that admit tractable classifications have both the characterisation problem and the isomorphism problem Σ^0_n for some n.

In this case we say that the class admits a local classification.

Unclassifiable structures have one of the two Π^1_1- or Σ^1_1-complete.

There are not many "natural" examples in-between (at the transfinite hyperarithmetical levels).

The general framework agrees with the thesis:

Mathematical structures that admit tractable classifications have both the characterisation problem and the isomorphism problem Σ^0_n for some n.

In this case we say that the class admits a local classification.

Unclassifiable structures have one of the two Π^1_1- or Σ^1_1-complete.

There are not many "natural" examples in-between (at the transfinite hyperarithmetical levels).

The general framework agrees with the thesis:

Mathematical structures that admit tractable classifications have both the characterisation problem and the isomorphism problem Σ^0_n for some n.

In this case we say that the class admits a local classification.

Unclassifiable structures have one of the two Π^1_1- or Σ^1_1-complete.

There are not many “natural” examples in-between (at the transfinite hyperarithmetical levels).

§2.2 An application.

Definition

A computable structure is automatic if the operations and relations are computed by a finite state automaton.

A finite automaton is a memoryless computational device (your laptop without the hard drive).

Approximately 20 years ago Khoussainov and Nerode asked for a characterisation of structures that admit an automatic presentation.

Sample results:

1. A finitely generated group is automatic iff it is virtually abelian (Oliver and Thomas 2005).
2. The automatic ordinals are exactly those below $\omega^{<\omega}$ (Delhomme 2004).
3. Similar characterisations exist for Boolean algebras and some other classes.

§2.2 An application.

Definition

A computable structure is automatic if the operations and relations are computed by a finite state automaton.

A finite automaton is a memoryless computational device (your laptop without the hard drive).

Approximately 20 years ago Khoussainov and Nerode asked for a characterisation of structures that admit an automatic presentation.

Sample results:

1. A finitely generated group is automatic iff it is virtually abelian (Oliver and Thomas 2005).
2. The automatic ordinals are exactly those below $\omega^{<\omega}$ (Delhomme 2004).
3. Similar characterisations exist for Boolean algebras and some other classes.

§2.2 An application.

Definition

A computable structure is automatic if the operations and relations are computed by a finite state automaton.

A finite automaton is a memoryless computational device (your laptop without the hard drive).

Approximately 20 years ago Khoussainov and Nerode asked for a characterisation of structures that admit an automatic presentation.

Sample results:

1. A finitely generated group is automatic iff it is virtually abelian (Oliver and Thomas 2005).
2. The automatic ordinals are exactly those below $\omega^{<\omega}$ (Delhomme 2004).
3. Similar characterisations exist for Boolean algebras and some other classes.

Some years later they asked the more specific question:

Question (Khoussainov and Nerode 2008)

What is the complexity of the index set

$$\{e : M_e \text{ is isomorphic to an automatic structure }\}$$

of automatically presentable structures?

Theorem (B.H.-T.K.M.N.)

The index set of automatic structures is Σ^1_1-complete.

Proof.

Hard. □

According to our framework, there is no characterization of automatic presentability.

Theorem (B.H.-T.K.**M.**N.)

The index set of automatic structures is Σ^1_1-complete.

Proof.

Hard. □

According to our framework, there is no characterization of automatic presentability.

Theorem (B.H.-T.K.M.N.)

The index set of automatic structures is Σ^1_1-complete.

Proof.

Hard. □

According to our framework, there is no characterization of automatic presentability.

References

For a detailed exposition of the background, see the books:

Ash and Knight. Computable Structures and the Hyperarithmetical Hierarchy.

Soare. Recursively Enumerable Sets and Degrees.

For lots of open questions and bib references, see our recent survey:

`https://www.massey.ac.nz/~amelniko/SmallSurvey1.pdf`