The complexity of finding supergraphs

Vittorio Cipriani
J.w.w. Arno Pauly (Swansea University)

Computability in Europe 2023 - Batumi - 2023

25 July 2023

In this talk, we consider several variations of the following problem.

Fix a countable graph G.

- Is G a(n induced) supergraph of an input graph H? (decision problems).
- If yes, can we find a copy of H in G? (search problems).

The challenge is to classify the *Weihrauch degree* of such problems, and to do so we use tools coming from *effective descriptive set theory*.

In this talk, we consider several variations of the following problem.

Fix a countable graph G.

- Is G a(n induced) supergraph of an input graph H? (decision problems).
- If yes, can we find a copy of H in G? (search problems).

The challenge is to classify the *Weihrauch degree* of such problems, and to do so we use tools coming from *effective descriptive set theory*.

The analysis of such problems was put forth by BeMent, Hirst, and Wallace ("Reverse mathematics and Weihrauch analysis motivated by finite complexity theory", Computability, 2021).

In this talk, we consider several variations of the following problem.

Fix a countable graph G.

- Is G a(n induced) supergraph of an input graph H? (decision problems).
- If yes, can we find a copy of H in G? (search problems).

The challenge is to classify the *Weihrauch degree* of such problems, and to do so we use tools coming from *effective descriptive set theory*.

The analysis of such problems was put forth by BeMent, Hirst, and Wallace ("Reverse mathematics and Weihrauch analysis motivated by finite complexity theory", Computability, 2021).

We report some initial results here, and in particular, solve one of their open questions.

In this talk, we consider several variations of the following problem.

Fix a countable graph G.

- Is G a(n induced) supergraph of an input graph H? (decision problems).
- If yes, can we find a copy of H in G? (search problems).

The challenge is to classify the *Weihrauch degree* of such problems, and to do so we use tools coming from *effective descriptive set theory*.

The analysis of such problems was put forth by BeMent, Hirst, and Wallace ("Reverse mathematics and Weihrauch analysis motivated by finite complexity theory", Computability, 2021).

We report some initial results here, and in particular, solve one of their open questions.

In this talk, we always assume G to have a computable copy.

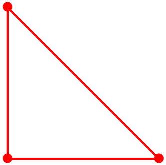
Graphs

The graphs $G = (V, E)$ we consider are countable, undirected, and without self-loops: that is, $V \subseteq \mathbb{N}$ and E satisfies anti-reflexivity and symmetry.

Definition

Given two graphs G and H we say that:

- G is a *supergraph* of H if $V(G) \supseteq V(H)$ and $E(G) \supseteq E(H)$;
- G is an *induced supergraph* of H if G is a supergraph of H and $E(G) = E(H) \cap (V(G) \times V(G))$.

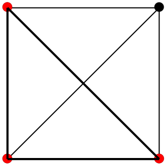


The graphs $G = (V, E)$ we consider are countable, undirected, and without self-loops: that is, $V \subseteq \mathbb{N}$ and E satisfies anti-reflexivity and symmetry.

Definition

Given two graphs G and H we say that:

- G is a *supergraph* of H if $V(G) \supseteq V(H)$ and $E(G) \supseteq E(H)$;
- G is an *induced supergraph* of H if G is a supergraph of H and $E(G) = E(H) \cap (V(G) \times V(G))$.



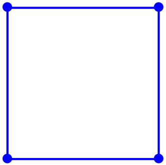
G_0 is an (induced) supergraph of G.

The graphs $G = (V, E)$ we consider are countable, undirected, and without self-loops: that is, $V \subseteq \mathbb{N}$ and E satisfies anti-reflexivity and symmetry.

Definition

Given two graphs G and H we say that:

- G is a *supergraph* of H if $V(G) \supseteq V(H)$ and $E(G) \supseteq E(H)$;
- G is an *induced supergraph* of H if G is a supergraph of H and $E(G) = E(H) \cap (V(G) \times V(G))$.

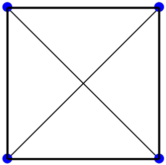


The graphs $G = (V, E)$ we consider are countable, undirected, and without self-loops: that is, $V \subseteq \mathbb{N}$ and E satisfies anti-reflexivity and symmetry.

Definition

Given two graphs G and H we say that:

- G is a *supergraph* of H if $V(G) \supseteq V(H)$ and $E(G) \supseteq E(H)$;
- G is an *induced supergraph* of H if G is a supergraph of H and $E(G) = E(H) \cap (V(G) \times V(G))$.



G_1 is a supergraph of G, but not an induced one.

Weihrauch reducibility and Effective Wadge reducibility

Computable analysis generalizes computability for functions on $\mathbb{N}$ to functions on $\mathbb{N}^{\mathbb{N}}$ (Baire space) and to represented spaces in general.

Definition

A *represented space* $\mathbf{X}$ is a pair (X, δ_X) where X is a set and $\delta_X :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ is a (possibly partial) surjective function called *representation map*. We say that $p \in \mathbb{N}^{\mathbb{N}}$ is a *name* for x if $\delta_X(p) = x$.

Computable analysis generalizes computability for functions on $\mathbb{N}$ to functions on $\mathbb{N}^{\mathbb{N}}$ (Baire space) and to represented spaces in general.

Definition

A *represented space* $\mathbf{X}$ is a pair (X, δ_X) where X is a set and $\delta_X :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ is a (possibly partial) surjective function called *representation map*. We say that $p \in \mathbb{N}^{\mathbb{N}}$ is a *name* for x if $\delta_X(p) = x$.

In this talk:

- **Gr** is the represented space of graphs, where a name for a graph is given by its characteristic function;
- **EGr** is the represented space of graphs, where a name for a graph is given by an enumeration of its vertices and edges.

f

Input: a name for an f-instance x.
Output: a name for (an element of) $f(x)$.

---

for a single input there may be multiple outputs!

UNIVERSITÀ DEGLI STUDI DI UDINE

f

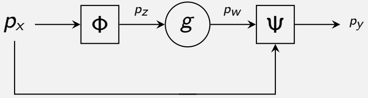
Input: a name for an f-instance x.
Output: a name for (an element of) $f(x)$.

for a single input there may be multiple outputs!

Definition

A problem f is *Weihrauch reducible* to g ($f \leq_W g$), if there are computable maps $\Phi, \Psi :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ s.t.

- for every name p_x for some input x of f, $\Phi(p_x) = p_z$, where p_z is a name for some input z for g and,
- for every name p_w for a solution w of $g(z)$, $\Psi(p_x \oplus p_w) = p_y$ where p_y is a name for a solution y of $f(x)$.



f

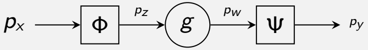
Input: a name for an f-instance x.
Output: a name for (an element of) $f(x)$.

for a single input there may be multiple outputs!

Definition

A problem f is *Weihrauch reducible* to g ($f \leq_{\mathrm{W}} g$), if there are computable maps $\Phi, \Psi :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ s.t.

- for every name p_x for some input x of f, $\Phi(p_x) = p_z$, where p_z is a name for some input z for g and,
- for every name p_w for a solution w of $g(z)$, $\Psi(p_x \oplus p_w) = p_y$ where p_y is a name for a solution y of $f(x)$.



In case Ψ has no access to the original input of f (i.e. $\Psi(p_w) = p_y$), we say that the reduction is *strong* ($f \leq_{\mathrm{sW}} g$).

Wadge reducibility gives a notion of complexity between sets of topological spaces. Here we study its effective counterpart.

Definition

Let $A, B \subseteq \mathbb{N}^{\mathbb{N}}$. We say that B *effectively Wadge reduces* to A if there exists a computable function f such that $x \in B \iff f(x) \in A$.
For a (non-ambiguous) class Γ, we say that A is *Γ-complete* if $A \in \Gamma$ and, for every $B \in \Gamma$, B effectively Wadge reduces to A.

Wadge reducibility gives a notion of complexity between sets of topological spaces. Here we study its effective counterpart.

Definition

Let $A, B \subseteq \mathbb{N}^{\mathbb{N}}$. We say that B *effectively Wadge reduces* to A if there exists a computable function f such that $x \in B \iff f(x) \in A$.
For a (non-ambiguous) class Γ, we say that A is *Γ-complete* if $A \in \Gamma$ and, for every $B \in \Gamma$, B effectively Wadge reduces to A.

Notation:

- $G \supseteq_{\mathsf{is}} H :\iff (\exists G' \cong G)(G'$ is an induced supergraph of $H)$;
- $G \supseteq_{\mathsf{s}} H :\iff (\exists G' \cong G)(G'$ is a supergraph of $H)$.

Wadge reducibility gives a notion of complexity between sets of topological spaces. Here we study its effective counterpart.

Definition

Let $A, B \subseteq \mathbb{N}^{\mathbb{N}}$. We say that B *effectively Wadge reduces* to A if there exists a computable function f such that $x \in B \iff f(x) \in A$.
For a (non-ambiguous) class Γ, we say that A is *Γ-complete* if $A \in \Gamma$ and, for every $B \in \Gamma$, B effectively Wadge reduces to A.

Notation:

- $G \supseteq_{\mathbf{is}} H :\iff (\exists G' \cong G)(G'$ is an induced supergraph of $H)$;
- $G \supseteq_{\mathbf{s}} H :\iff (\exists G' \cong G)(G'$ is a supergraph of $H)$.

For a fixed countable graph G, we consider sets of (names of) graphs of the form

$$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathbf{(i)s}} H\} := \{p \in \operatorname{dom}(\delta_{(E)Gr}) : G \supseteq_{\mathbf{(i)s}} \delta_{(E)Gr}(p)\},$$

i.e. the set of graphs H such that G is a(n induced) supergraph H.

Wadge reducibility gives a notion of complexity between sets of topological spaces. Here we study its effective counterpart.

Definition

Let $A, B \subseteq \mathbb{N}^{\mathbb{N}}$. We say that B *effectively Wadge reduces* to A if there exists a computable function f such that $x \in B \iff f(x) \in A$.
For a (non-ambiguous) class Γ, we say that A is *Γ-complete* if $A \in \Gamma$ and, for every $B \in \Gamma$, B effectively Wadge reduces to A.

Notation:

- $G \supseteq_{\mathsf{is}} H :\iff (\exists G' \cong G)(G'$ is an induced supergraph of $H)$;
- $G \supseteq_{\mathsf{s}} H :\iff (\exists G' \cong G)(G'$ is a supergraph of $H)$.

For a fixed countable graph G, we consider sets of (names of) graphs of the form

$$\{H \in \mathbf{(E)Gr} : G \supseteq_{\mathsf{(i)s}} H\} := \{p \in \mathrm{dom}(\delta_{(E)Gr}) : G \supseteq_{\mathsf{(i)s}} \delta_{(E)Gr}(p)\},$$

i.e. the set of graphs H such that G is a(n induced) supergraph H.
N.B. all sets above are Σ^1_1.

Decision problems

The following problems were introduced in [BHW21]. For a fixed graph G:

IS^G- (induced subgraph)

Input: $H \in \mathbf{Gr}$.
Output: 1 if $G \supseteq_{\mathbf{is}} H$, 0 otherwise.

If $G \in \mathbf{EGr}$, the corresponding problem is denoted by $e\mathsf{IS}_G$.

S^G - (subgraph)

Input: $H \in \mathbf{Gr}$.
Output: 1 if $G \supseteq_{\mathbf{s}} H$, 0 otherwise.

If $G \in \mathbf{EGr}$, the corresponding problem is denoted by $e\mathsf{S}_G$.

$G \in \mathbf{Gr} \rightarrow$ characteristic function, $G \in \mathbf{EGr} \rightarrow$ enumeration.

LPO

Input: $p \in 2^{\mathbb{N}}$.
Output: 1 if $\exists i(p(i) = 1)$, 0 otherwise.

LPO

Input: $p \in 2^{\mathbb{N}}$.
Output: 1 if $\exists i(p(i) = 1)$, 0 otherwise.

LPO can be also rephrased as the problem deciding a Σ^0_1 (or equivalently, Π^0_1) question relative to the input.

UNIVERSITÀ DEGLI STUDI DI UDINE

LPO

Input: $p \in 2^{\mathbb{N}}$.
Output: 1 if $\exists i(p(i) = 1)$, 0 otherwise.

LPO can be also rephrased as the problem deciding a Σ^0_1 (or equivalently, Π^0_1) question relative to the input.
Similarly, $\mathsf{LPO}^{(n)}$ is the problem deciding a Σ^0_{n+1} (Π^0_{n+1}) question relative to the input.

UNIVERSITÀ DEGLI STUDI DI UDINE

LPO

Input: $p \in 2^{\mathbb{N}}$.
Output: 1 if $\exists i(p(i) = 1)$, 0 otherwise.

LPO can be also rephrased as the problem deciding a Σ^0_1 (or equivalently, Π^0_1) question relative to the input.
Similarly, $\mathsf{LPO}^{(n)}$ is the problem deciding a Σ^0_{n+1} (Π^0_{n+1}) question relative to the input.

WF

Input: a tree $T \subseteq \mathbb{N}^{<\mathbb{N}}$.
Output: 1 if T is *well-founded* i.e., it has no infinite path, 0 otherwise.

UNIVERSITÀ DEGLI STUDI DI UDINE

LPO

Input: $p \in 2^{\mathbb{N}}$.
Output: 1 if $\exists i(p(i) = 1)$, 0 otherwise.

LPO can be also rephrased as the problem deciding a Σ^0_1 (or equivalently, Π^0_1) question relative to the input.
Similarly, $\mathsf{LPO}^{(n)}$ is the problem deciding a Σ^0_{n+1} (Π^0_{n+1}) question relative to the input.

WF

Input: a tree $T \subseteq \mathbb{N}^{<\mathbb{N}}$.
Output: 1 if T is *well-founded* i.e., it has no infinite path, 0 otherwise.

WF can be also rephrased as the problem deciding a Σ^1_1 (Π^1_1) question relative to the input.

LPO

Input: $p \in 2^{\mathbb{N}}$.
Output: 1 if $\exists i(p(i) = 1)$, 0 otherwise.

LPO can be also rephrased as the problem deciding a Σ^0_1 (or equivalently, Π^0_1) question relative to the input.
Similarly, $\mathsf{LPO}^{(n)}$ is the problem deciding a Σ^0_{n+1} (Π^0_{n+1}) question relative to the input.

WF

Input: a tree $T \subseteq \mathbb{N}^{<\mathbb{N}}$.
Output: 1 if T is *well-founded* i.e., it has no infinite path, 0 otherwise.

WF can be also rephrased as the problem deciding a Σ^1_1 (Π^1_1) question relative to the input.
N.B. $\{H \in \mathbf{Gr} : G \supseteq_{\mathsf{is}} H\}$ is $\Sigma^1_1 \implies \mathsf{IS}^G \leq_{\mathrm{sW}} \mathsf{WF}$ (similarly for the other sets/problems).

Given a countable graph G, does $(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} \mathsf{WF}$ or $(e)\mathsf{IS}^G <_{\mathrm{sW}} \mathsf{WF}$ (similarly for $(e)\mathsf{S}^G$)?

Given a countable graph G, does $(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} \mathsf{WF}$ or $(e)\mathsf{IS}^G <_{\mathrm{sW}} \mathsf{WF}$ (similarly for $(e)\mathsf{S}^G$)?
In [BHW21], the authors also studied the "opposite" problem, namely (always fixing a countable graph G)

Given in input a graph H, answer whether H contains an (induced) subgraph isomorphic to G.

Given a countable graph G, does $(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} \mathsf{WF}$ or $(e)\mathsf{IS}^G <_{\mathrm{sW}} \mathsf{WF}$ (similarly for $(e)\mathsf{S}^G$)?
In [BHW21], the authors also studied the "opposite" problem, namely (always fixing a countable graph G)

Given in input a graph H, answer whether H contains an (induced) subgraph isomorphic to G.

In [CP22], we solved one of their open questions showing that,

- for the induced subgraph case, if the input graph is in **Gr**, these problems are either equivalent to LPO (if G is finite) or to WF (if G is infinite);
- for the subgraph case, we can find different graphs whose corresponding decision problem is equivalent to $\mathsf{LPO}^{(n)}$ for every n and to WF.

A REMARK

Given a countable graph G, does $(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} \mathsf{WF}$ or $(e)\mathsf{IS}^G <_{\mathrm{sW}} \mathsf{WF}$ (similarly for $(e)\mathsf{S}^G$)?
In [BHW21], the authors also studied the "opposite" problem, namely (always fixing a countable graph G)

Given in input a graph H, answer whether H contains an (induced) subgraph isomorphic to G.

In [CP22], we solved one of their open questions showing that,

- for the induced subgraph case, if the input graph is in **Gr**, these problems are either equivalent to LPO (if G is finite) or to WF (if G is infinite);
- for the subgraph case, we can find different graphs whose corresponding decision problem is equivalent to $\mathsf{LPO}^{(n)}$ for every n and to WF.

Observation: it is easy to find graphs for which the corresponding problem reaches WF (i.e., the infinite ray R).

We will show that for the supergraph problem "it's difficult being difficult".

Summary

	Wadge	Weihrauch
G finite	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{is}} H\}$ is Π^0_1-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} \mathsf{LPO}$
G finite	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{s}} H\}$ is Π^0_1-complete	$(e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}$

Summary

	Wadge	Weihrauch
G finite	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{is}} H\}$ is Π^0_1-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} \mathsf{LPO}$
G finite	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{s}} H\}$ is Π^0_1-complete	$(e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}$
K_ω	$\{H \in (\mathbf{E})\mathbf{Gr} : K_\omega \supseteq_{\mathsf{s}} H\}$ is computable	$(e)\mathsf{S}^{K_\omega} \equiv_{\mathrm{sW}} \mathsf{id}$
K_ω	$\{H \in \mathbf{Gr} : K_\omega \supseteq_{\mathsf{is}} H\}$ is Π^0_1-complete	$\mathsf{IS}^{K_\omega} \equiv_{\mathrm{sW}} \mathsf{LPO}$
K_ω	$\{H \in \mathbf{EGr} : K_\omega \supseteq_{\mathsf{is}} H\}$ is Π^0_2-complete	$e\mathsf{IS}^{K_\omega} \equiv_{\mathrm{sW}} \mathsf{LPO}'$

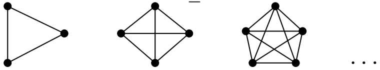
K_ω denotes the complete graph on $\mathbb{N}$.

Summary

	Wadge	Weihrauch
G finite	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{is}} H\}$ is Π^0_1-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} \mathsf{LPO}$
G finite	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{s}} H\}$ is Π^0_1-complete	$(e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}$
K_ω	$\{H \in (\mathbf{E})\mathbf{Gr} : K_\omega \supseteq_{\mathsf{s}} H\}$ is computable	$(e)\mathsf{S}^{K_\omega} \equiv_{\mathrm{sW}} \mathsf{id}$
K_ω	$\{H \in \mathbf{Gr} : K_\omega \supseteq_{\mathsf{is}} H\}$ is Π^0_1-complete	$\mathsf{IS}^{K_\omega} \equiv_{\mathrm{sW}} \mathsf{LPO}$
K_ω	$\{H \in \mathbf{EGr} : K_\omega \supseteq_{\mathsf{is}} H\}$ is Π^0_2-complete	$e\mathsf{IS}^{K_\omega} \equiv_{\mathrm{sW}} \mathsf{LPO}'$
$G = \bigotimes_{i\geq 1} R_i$	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{(i)s}} H\}$ is Π^0_3-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} (e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}''$
$G = \bigotimes_{i\geq 1} K_i$	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{(i)s}} H\}$is Π^0_3-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} (e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}''$

$\bigotimes_{i\geq 1} R_i$

$\bigotimes_{i\geq 1} K_i$

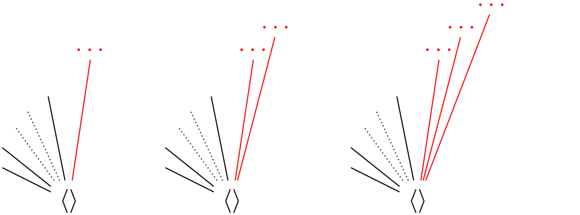


The results in red answer positively a question left open in [BHW21], namely:

Is there a computable graph G such that $\mathsf{LPO} <_{\mathrm{sW}} \mathsf{IS}^G$? Yes.

	Wadge	Weihrauch
G finite	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{is}} H\}$ is Π^0_1-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} \mathsf{LPO}$
G finite	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{s}} H\}$ is Π^0_1-complete	$(e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}$
K_ω	$\{H \in (\mathbf{E})\mathbf{Gr} : K_\omega \supseteq_{\mathsf{s}} H\}$ is computable	$(e)\mathsf{S}^{K_\omega} \equiv_{\mathrm{sW}} \mathsf{id}$
K_ω	$\{H \in \mathbf{Gr} : K_\omega \supseteq_{\mathsf{is}} H\}$ is Π^0_1-complete	$\mathsf{IS}^{K_\omega} \equiv_{\mathrm{sW}} \mathsf{LPO}$
K_ω	$\{H \in \mathbf{EGr} : K_\omega \supseteq_{\mathsf{is}} H\}$ is Π^0_2-complete	$e\mathsf{IS}^{K_\omega} \equiv_{\mathrm{sW}} \mathsf{LPO}'$
$G = \bigotimes_{i\geq 1} R_i$	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{(i)s}} H\}$ is Π^0_3-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} (e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}''$
$G = \bigotimes_{i\geq 1} K_i$	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{(i)s}} H\}$is Π^0_3-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} (e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}''$
$\mathcal{S}$	$\{H \in (\mathbf{E})\mathbf{Gr} : \mathcal{S} \supseteq_{\mathsf{(i)s}} H\}$ is Π^0_5-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} (e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}^{(4)}$.

$\mathcal{S}$ is the disconnected union of $(T_n)_{n\in\mathbb{N}}$, where every T_n is a tree having finite paths of any length (in black) and $n+1$-many paths of infinite length (in red).



The proof of the fact that this set is complete was suggested by an anonymous referee of [CP23].

UNIVERSITÀ DEGLI STUDI DI UDINE

	Wadge	Weihrauch
G finite	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{is}} H\}$ is Π^0_1-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} \mathsf{LPO}$
G finite	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{s}} H\}$ is Π^0_1-complete	$(e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}$
K_ω	$\{H \in (\mathbf{E})\mathbf{Gr} : K_\omega \supseteq_{\mathsf{s}} H\}$ is computable	$(e)\mathsf{S}^{K_\omega} \equiv_{\mathrm{sW}} \mathsf{id}$
K_ω	$\{H \in \mathbf{Gr} : K_\omega \supseteq_{\mathsf{is}} H\}$ is Π^0_1-complete	$\mathsf{IS}^{K_\omega} \equiv_{\mathrm{sW}} \mathsf{LPO}$
K_ω	$\{H \in \mathbf{EGr} : K_\omega \supseteq_{\mathsf{is}} H\}$ is Π^0_2-complete	$e\mathsf{IS}^{K_\omega} \equiv_{\mathrm{sW}} \mathsf{LPO}'$
$G = \bigotimes_{i \geq 1} R_i$	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{(i)s}} H\}$ is Π^0_3-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} (e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}''$
$G = \bigotimes_{i \geq 1} K_i$	$\{H \in (\mathbf{E})\mathbf{Gr} : G \supseteq_{\mathsf{(i)s}} H\}$is Π^0_3-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} (e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}''$
$\mathcal{S}$	$\{H \in (\mathbf{E})\mathbf{Gr} : \mathcal{S} \supseteq_{\mathsf{(i)s}} H\}$ is Π^0_5-complete	$(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} (e)\mathsf{S}^G \equiv_{\mathrm{sW}} \mathsf{LPO}^{(4)}$.

Maybe with other "strange" graphs, we could go beyond Π^0_5: but is there some G such that $(e)\mathsf{IS}^G \equiv_{\mathrm{sW}} \mathsf{WF}$?

Search problems

(preliminary results!)

The problems below lead to 8 versions, depending on whether the input/output is given via characteristic function or enumeration (4 for the induced supergraph case and 4 for the supergraph one).

The problems below lead to 8 versions, depending on whether the input/output is given via characteristic function or enumeration (4 for the induced supergraph case and 4 for the supergraph one).

$\mathsf{ISupCopy}_G$

Input: a graph H s.t. $G \supseteq_{\mathbf{is}} H$.
Output: H', where $H' \cong H$ and $G \supseteq_{\mathbf{is}} H'$.

$\mathsf{SupCopy}_G$

Input: a graph H s.t. $G \supseteq_{\mathbf{s}} H$.
Output: H', where $H' \cong H$ and $G \supseteq_{\mathbf{s}} H'$.

The problems below lead to 8 versions, depending on whether the input/output is given via characteristic function or enumeration (4 for the induced supergraph case and 4 for the supergraph one).

$\mathsf{ISupCopy}_G$

Input: a graph H s.t. $G \supseteq_{\mathbf{is}} H$.
Output: H', where $H' \cong H$ and $G \supseteq_{\mathbf{is}} H'$.

$\mathsf{SupCopy}_G$

Input: a graph H s.t. $G \supseteq_{\mathbf{s}} H$.
Output: H', where $H' \cong H$ and $G \supseteq_{\mathbf{s}} H'$.

The first result we obtain is that there exists a graph whose lower bound is $\mathsf{C}_{\mathbb{N}}$.

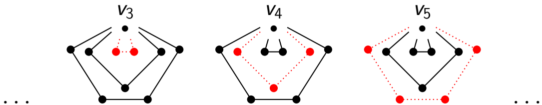
$\mathsf{C}_{\mathbb{N}}$

Input: an enumeration of the complement of a nonempty closed subset A of $\mathbb{N}$.
Output: some $p \in A$.

The graph H_{CN} has:

- as vertex set, for every $n \in \mathbb{N}$ a dedicated vertex v_n and
- for every $k \neq n$, a cycle of length k containing v_n. All the cycles are otherwise disjoint.

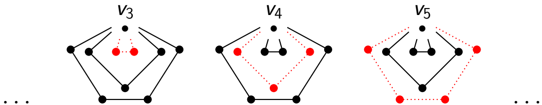
For example, H_{CN} on v_3, v_4 and v_5 looks like this (red vertices/edges are missing):



The graph H_{CN} has:

- as vertex set, for every $n \in \mathbb{N}$ a dedicated vertex v_n and
- for every $k \neq n$, a cycle of length k containing v_n. All the cycles are otherwise disjoint.

For example, H_{CN} on v_3, v_4 and v_5 looks like this (red vertices/edges are missing):



Theorem (C., Pauly)

$\mathsf{C}_{\mathbb{N}} \leq_{\mathrm{W}} \mathsf{SupCopy}_{H_{\mathrm{CN}}}$.

The graph R (the infinite ray)

We study the same problem for the more "natural" graph R.

$\mathsf{SupCopy}_\mathsf{R}$ can be rephrased as

- given in **input** either only finite line segments or finitely many line segments (possibly zero) plus a copy of R,
- **output** an "arrangement" of such line segments in R.

THE GRAPH R (THE INFINITE RAY)

We study the same problem for the more "natural" graph R.

$\mathsf{SupCopy}_{\mathsf{R}}$ can be rephrased as

- given in **input** either only finite line segments or finitely many line segments (possibly zero) plus a copy of R,
- **output** an "arrangement" of such line segments in R.

It is easy to notice that $\mathsf{SupCopy}_{\mathsf{R}} \equiv_{\mathrm{sW}} \mathsf{ISupCopy}_{\mathsf{R}}$.

The graph R (the infinite ray)

We study the same problem for the more “natural” graph R.

$\mathsf{SupCopy}_{\mathrm{R}}$ can be rephrased as

- given in **input** either only finite line segments or finitely many line segments (possibly zero) plus a copy of R,
- **output** an “arrangement” of such line segments in R.

It is easy to notice that $\mathsf{SupCopy}_{\mathrm{R}} \equiv_{\mathrm{sW}} \mathsf{ISupCopy}_{\mathrm{R}}$.

We explored only the version where the output is enumerated.

THE GRAPH R (THE INFINITE RAY)

We study the same problem for the more "natural" graph R.
$\mathsf{SupCopy}_{\mathsf{R}}$ can be rephrased as

- given in **input** either only finite line segments or finitely many line segments (possibly zero) plus a copy of R,
- **output** an "arrangement" of such line segments in R.

It is easy to notice that $\mathsf{SupCopy}_{\mathsf{R}} \equiv_{\mathrm{sW}} \mathsf{ISupCopy}_{\mathsf{R}}$.
We explored only the version where the output is enumerated.

lim

Input: a converging sequence $(p_n)_{n\in\mathbb{N}} \in (\mathbb{N}^{\mathbb{N}})^{\mathbb{N}}$.
Output: $\lim_{n\to\infty} p_n$.

The graph R (the infinite ray)

We study the same problem for the more "natural" graph R.
$\mathsf{SupCopy}_{\mathsf{R}}$ can be rephrased as

- given in **input** either only finite line segments or finitely many line segments (possibly zero) plus a copy of R,
- **output** an "arrangement" of such line segments in R.

It is easy to notice that $\mathsf{SupCopy}_{\mathsf{R}} \equiv_{\mathrm{sW}} \mathsf{ISupCopy}_{\mathsf{R}}$.
We explored only the version where the output is enumerated.

lim

Input: a converging sequence $(p_n)_{n\in\mathbb{N}} \in (\mathbb{N}^{\mathbb{N}})^{\mathbb{N}}$.
Output: $\lim_{n\to\infty} p_n$.

Theorem (C., Pauly)

$\mathsf{LPO} \le_{\mathrm{W}} \mathsf{SupCopy}_{\mathsf{R}} \le_{\mathrm{W}} \mathsf{C}_{\mathbb{N}} * \lim * \lim$.

$f * g$ means "apply g, do something computable and then apply f".

The graph R (the infinite ray)

We study the same problem for the more "natural" graph R.

$\mathsf{SupCopy}_{\mathsf{R}}$ can be rephrased as

- given in **input** either only finite line segments or finitely many line segments (possibly zero) plus a copy of R,
- **output** an "arrangement" of such line segments in R.

It is easy to notice that $\mathsf{SupCopy}_{\mathsf{R}} \equiv_{\mathrm{sW}} \mathsf{ISupCopy}_{\mathsf{R}}$.

We explored only the version where the output is enumerated.

lim

Input: a converging sequence $(p_n)_{n\in\mathbb{N}} \in (\mathbb{N}^{\mathbb{N}})^{\mathbb{N}}$.
Output: $\lim_{n\to\infty} p_n$.

Theorem (C., Pauly)

$\mathsf{LPO} \le_{\mathrm{W}} \mathsf{SupCopy}_{\mathsf{R}} \le_{\mathrm{W}} \mathsf{C}_{\mathbb{N}} * \lim * \lim$.

$f * g$ means "apply g, do something computable and then apply f".

The style of reasoning employed to in the study of $\mathsf{SupCopy}_{\mathsf{R}}$ is reminiscent of the study of the degrees of bi-embeddable categoricity of equivalence relations (as only the number and size of connected components matter).

Work in Progress

(Pt. 1)

Lower bounds for $\mathsf{SupCopy}_{\mathsf{R}}$

So far we know that $\mathsf{LPO} \le_{\mathrm{W}} \mathsf{SupCopy}_{\mathsf{R}} \le_{\mathrm{W}} \mathsf{C}_{\mathbb{N}} * \lim * \lim$.

Lower bounds for $\mathsf{SupCopy}_\mathsf{R}$

So far we know that $\mathsf{LPO} \leq_{\mathrm{W}} \mathsf{SupCopy}_\mathsf{R} \leq_{\mathrm{W}} \mathsf{C}_\mathbb{N} * \lim * \lim$. Consider the following problem.

Lower bounds for $\mathsf{SupCopy}_{\mathrm{R}}$

So far we know that $\mathsf{LPO} \le_{\mathrm{W}} \mathsf{SupCopy}_{\mathrm{R}} \le_{\mathrm{W}} \mathsf{C}_{\mathbb{N}} * \lim * \lim$. Consider the following problem.

Π^0_1-Bound

Input: a finite closed subset of $\mathbb{N}$.
Output: an upper bound for it.

Lower bounds for $\mathsf{SupCopy}_{\mathrm{R}}$

So far we know that $\mathsf{LPO} \le_{\mathrm{W}} \mathsf{SupCopy}_{\mathrm{R}} \le_{\mathrm{W}} \mathsf{C}_{\mathbb{N}} * \lim * \lim$. Consider the following problem.

Π^0_1-Bound

Input: a finite closed subset of $\mathbb{N}$.
Output: an upper bound for it.

Such a problem characterizes the power of $\mathsf{SupCopy}_{\mathrm{R}}$ for solving sufficiently uniform problems (*fractals*) with a computable point in their domain (*pointed*).

LOWER BOUNDS FOR $\mathsf{SupCopy}_{\mathrm{R}}$

So far we know that $\mathsf{LPO} \le_{\mathrm{W}} \mathsf{SupCopy}_{\mathrm{R}} \le_{\mathrm{W}} \mathsf{C}_{\mathbb{N}} * \lim * \lim$. Consider the following problem.

Π^0_1-Bound

Input: a finite closed subset of $\mathbb{N}$.
Output: an upper bound for it.

Such a problem characterizes the power of $\mathsf{SupCopy}_{\mathrm{R}}$ for solving sufficiently uniform problems (*fractals*) with a computable point in their domain (*pointed*).

Theorem (C., Pauly)

Let f be a pointed fractal. T.f.a.e.:

- $\mathsf{ACC}_{\mathbb{N}} \times f \le_{\mathrm{W}} \mathsf{SupCopy}_{\mathrm{R}}$;
- $f \le_{\mathrm{W}} \Pi^0_1\text{-}\mathsf{Bound}$.

$\mathsf{ACC}_{\mathbb{N}}$ is the restriction of $\mathsf{C}_{\mathbb{N}}$ to sets of the form $\{\mathbb{N}\}$ or $\{\mathbb{N} \setminus \{n\} : n \in \mathbb{N}\}$.
$f \times g$ means "perform f and g in parallel".

LOWER BOUNDS FOR $\mathsf{SupCopy}_{\mathsf{R}}$

So far we know that $\mathsf{LPO} \le_W \mathsf{SupCopy}_{\mathsf{R}} \le_W \mathsf{C}_{\mathbb{N}} * \lim * \lim$. Consider the following problem.

Π^0_1-Bound

Input: a finite closed subset of $\mathbb{N}$.
Output: an upper bound for it.

Such a problem characterizes the power of $\mathsf{SupCopy}_{\mathsf{R}}$ for solving sufficiently uniform problems (*fractals*) with a computable point in their domain (*pointed*).

Theorem (C., Pauly)

Let f be a pointed fractal. T.f.a.e.:

- $\mathsf{ACC}_{\mathbb{N}} \times f \le_W \mathsf{SupCopy}_{\mathsf{R}}$;
- $f \le_W \Pi^0_1$-Bound.

$\mathsf{ACC}_{\mathbb{N}}$ is the restriction of $\mathsf{C}_{\mathbb{N}}$ to sets of the form $\{\mathbb{N}\}$ or $\{\mathbb{N} \setminus \{n\} : n \in \mathbb{N}\}$.
$f \times g$ means "perform f and g in parallel".

As a corollary, we obtain that $\lim \not\le_W \mathsf{SupCopy}_{\mathsf{R}}$ and Π^0_2-$\mathsf{C}_{\mathbb{N}} \not\le_W \mathsf{SupCopy}_{\mathsf{R}}$.

Lower bounds for $\mathsf{SupCopy}_{\mathsf{R}}$

So far we know that $\mathsf{LPO} \le_{\mathrm{W}} \mathsf{SupCopy}_{\mathsf{R}} \le_{\mathrm{W}} \mathsf{C}_{\mathbb{N}} * \lim * \lim$. Consider the following problem.

Π^0_1-Bound

Input: a finite closed subset of $\mathbb{N}$.
Output: an upper bound for it.

Such a problem characterizes the power of $\mathsf{SupCopy}_{\mathsf{R}}$ for solving sufficiently uniform problems (*fractals*) with a computable point in their domain (*pointed*).

Theorem (C., Pauly)

Let f be a pointed fractal. T.f.a.e.:

- $\mathsf{ACC}_{\mathbb{N}} \times f \le_{\mathrm{W}} \mathsf{SupCopy}_{\mathsf{R}}$;
- $f \le_{\mathrm{W}} \Pi^0_1$-Bound.

$\mathsf{ACC}_{\mathbb{N}}$ is the restriction of $\mathsf{C}_{\mathbb{N}}$ to sets of the form $\{\mathbb{N}\}$ or $\{\mathbb{N} \setminus \{n\} : n \in \mathbb{N}\}$.
$f \times g$ means "perform f and g in parallel".

As a corollary, we obtain that $\lim \not\le_{\mathrm{W}} \mathsf{SupCopy}_{\mathsf{R}}$ and Π^0_2-$\mathsf{C}_{\mathbb{N}} \not\le_{\mathrm{W}} \mathsf{SupCopy}_{\mathsf{R}}$. It is still open whether such a problem is non-uniformly computable.

Work in Progress

(Pt. 2)

So far, we have obtained sets of (names of) graphs being Γ-complete for some Γ being a Σ or Π class.

Other sets of graphs

So far, we have obtained sets of (names of) graphs being Γ-complete for some Γ being a Σ or Π class.
We obtained two natural examples of graphs falling outside this schema.

So far, we have obtained sets of (names of) graphs being Γ-complete for some Γ being a Σ or Π class.
We obtained two natural examples of graphs falling outside this schema.
Here L denotes the bi-infinite line.

So far, we have obtained sets of (names of) graphs being Γ-complete for some Γ being a Σ or Π class.
We obtained two natural examples of graphs falling outside this schema.
Here L denotes the bi-infinite line.

Theorem (C., Marcone, Pauly)

$\{H \in (\mathbf{E})\mathbf{Gr} : \mathrm{R} \supseteq_{(\mathbf{i})\mathbf{s}} H\}$ and $\{H \in (\mathbf{E})\mathbf{Gr} : L \supseteq_{(\mathbf{i})\mathbf{s}} H\}$ are $\Sigma^0_3 \cup \Pi^0_3$-complete.

UNIVERSITÀ DEGLI STUDI DI UDINE

Zach BeMent, Jeffry Hirst, and Asuka Wallace.
Reverse mathematics and Weihrauch analysis motivated by finite complexity theory.
Computability, 10(4):343–354, 2021.

Vittorio Cipriani and Arno Pauly.
Embeddability of graphs and Weihrauch degrees.
Available at https://arxiv.org/abs/2305.00935, 2022.

Vittorio Cipriani and Arno Pauly.
The complexity of finding supergraphs.
In Gianluca Della Vedova, Besik Dundua, Steffen Lempp, and Florin Manea, editors, *Unity of Logic and Computation*, pages 178–189, Cham, 2023. Springer Nature Switzerland.

Thanks for Your attention!