Department of Mathematics, University of Udine

Embeddability of graphs and Weihrauch degrees

Vittorio Cipriani (University of Udine)
joint work with Arno Pauly (Swansea University)

Computability and Complexity in Analysis 2022
Glenside, PA, USA (online)
May 23, 2022

Some basic definitions

All the graphs $G = (V, E)$ we consider here are *countable* and *undirected* (no multiple edges and self-loops).

Some basic definitions

All the graphs $G = (V, E)$ we consider here are *countable* and *undirected* (no multiple edges and self-loops).

Definition

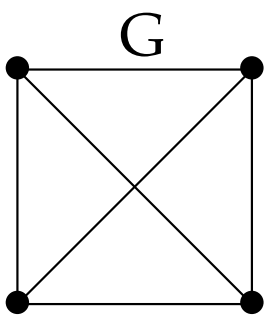
An undirected graph $G_0 = (V_0, E_0)$ is a *subgraph* of $G = (V, E)$ if and only if $V_0 \subseteq V$ and $E_0 \subseteq E$. We say it is an *induced subgraph* if furthermore $E_0 = E \cap (V_0 \times V_0)$.

Some basic definitions

All the graphs $G = (V, E)$ we consider here are *countable* and *undirected* (no multiple edges and self-loops).

Definition

An undirected graph $G_0 = (V_0, E_0)$ is a *subgraph* of $G = (V, E)$ if and only if $V_0 \subseteq V$ and $E_0 \subseteq E$. We say it is an *induced subgraph* if furthermore $E_0 = E \cap (V_0 \times V_0)$.

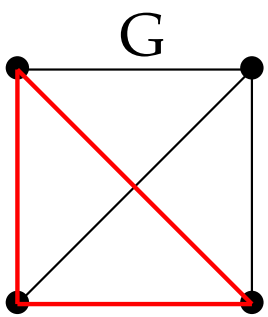


Some basic definitions

All the graphs $G = (V, E)$ we consider here are *countable* and *undirected* (no multiple edges and self-loops).

Definition

An undirected graph $G_0 = (V_0, E_0)$ is a *subgraph* of $G = (V, E)$ if and only if $V_0 \subseteq V$ and $E_0 \subseteq E$. We say it is an *induced subgraph* if furthermore $E_0 = E \cap (V_0 \times V_0)$.



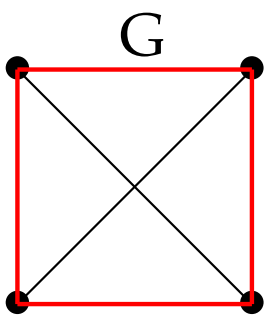
G_0 is an (induced) subgraph of G.

Some basic definitions

All the graphs $G = (V, E)$ we consider here are *countable* and *undirected* (no multiple edges and self-loops).

Definition

An undirected graph $G_0 = (V_0, E_0)$ is a *subgraph* of $G = (V, E)$ if and only if $V_0 \subseteq V$ and $E_0 \subseteq E$. We say it is an *induced subgraph* if furthermore $E_0 = E \cap (V_0 \times V_0)$.



G_0 is a subgraph of G, but not an induced one.

Some basic definitions

All the graphs $G = (V, E)$ we consider here are *countable* and *undirected* (no multiple edges and self-loops).

Definition

An undirected graph $G_0 = (V_0, E_0)$ is a *subgraph* of $G = (V, E)$ if and only if $V_0 \subseteq V$ and $E_0 \subseteq E$. We say it is an *induced subgraph* if furthermore $E_0 = E \cap (V_0 \times V_0)$.

In this talk, using the framework of *Weihrauch reducibility* we study the uniform computational content of the (induced) subgraph problem. The firsts to consider these problems in this context were BeMent, Hirst and Wallace in [BHW21].

Represented space

In general, to study computability on a space X, we "transfer" notions of computability in $\mathbb{N}^{\mathbb{N}}$ to X encoding each element of X with some $p \in \mathbb{N}^{\mathbb{N}}$.

Represented space

In general, to study computability on a space X, we "transfer" notions of computability in $\mathbb{N}^{\mathbb{N}}$ to X encoding each element of X with some $p \in \mathbb{N}^{\mathbb{N}}$.

Definition

A represented space $\mathbf{X}$ is a pair $(X, \delta_{\mathbf{X}})$ where $\delta_{\mathbf{X}} :\subseteq \mathbb{N}^{\mathbb{N}} \to X$.

If $\delta_{\mathbf{X}}(p) = x$ for $p \in \mathbb{N}^{\mathbb{N}}$, we say that p is a *name* for $x \in X$, ($\subseteq$ indicates that $\delta_{\mathbf{X}}$ is partial).

Represented space

In general, to study computability on a space X, we "transfer" notions of computability in $\mathbb{N}^{\mathbb{N}}$ to X encoding each element of X with some $p \in \mathbb{N}^{\mathbb{N}}$.

Definition

A represented space $\mathbf{X}$ is a pair $(X, \delta_{\mathbf{X}})$ where $\delta_{\mathbf{X}} :\subseteq \mathbb{N}^{\mathbb{N}} \to X$.

If $\delta_{\mathbf{X}}(p) = x$ for $p \in \mathbb{N}^{\mathbb{N}}$, we say that p is a *name* for $x \in X$, ($\subseteq$ indicates that $\delta_{\mathbf{X}}$ is partial).

Notation: **Gr** is the represented space of countable undirected graphs, i.e. p is a name for a graph G iff

- $p(\langle i, i\rangle) = 1$ iff $v_i \in V(G)$ (i.e, v_i is a vertex of G) and
- for $i \neq j$, $p(\langle i, j\rangle) = 1$ iff $v_i, v_j \in V(G)$ and $(v_i, v_j) \in E(G)$ (i.e. (v_i, v_j) is an edge of G).

Weihrauch reducibility

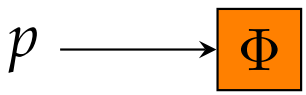
We denote a *multi-valued function* between represented spaces $\mathbf{X}$ and $\mathbf{Y}$ by $f :\subseteq \mathbf{X} \rightrightarrows \mathbf{Y}$. ($\rightrightarrows$ denotes that $x \in \mathbf{X}$ may be mapped to different elements of $\mathbf{Y}$).

Weihrauch reducibility

We denote a *multi-valued function* between represented spaces $\mathbf{X}$ and $\mathbf{Y}$ by $f :\subseteq \mathbf{X} \rightrightarrows \mathbf{Y}$. ($\rightrightarrows$ denotes that $x \in \mathbf{X}$ may be mapped to different elements of $\mathbf{Y}$).
Let f, g be (partial multivalued) functions.

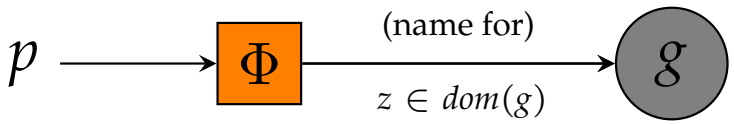
We denote a *multi-valued function* between represented spaces $\mathbf{X}$ and $\mathbf{Y}$ by $f :\subseteq \mathbf{X} \rightrightarrows \mathbf{Y}$. ($\rightrightarrows$ denotes that $x \in \mathbf{X}$ may be mapped to different elements of $\mathbf{Y}$).
Let f, g be (partial multivalued) functions.
f is *Weihrauch reducible* to g ($f \leq_{\mathrm{W}} g$) if there are computable $\Phi, \Psi :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ s.t.:

- Given a name p for $x \in \mathrm{dom}(f)$, $\Phi(p)$ is a name for $z \in \mathrm{dom}(g)$;
- Given a name q for $w \in g(z)$, $\Psi(p, q)$ is a name for $y \in f(x)$;



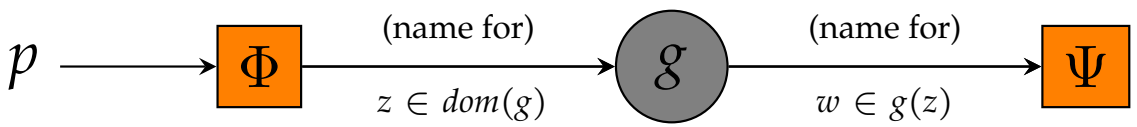
We denote a *multi-valued function* between represented spaces $\mathbf{X}$ and $\mathbf{Y}$ by $f :\subseteq \mathbf{X} \rightrightarrows \mathbf{Y}$. ($\rightrightarrows$ denotes that $x \in \mathbf{X}$ may be mapped to different elements of $\mathbf{Y}$).
Let f, g be (partial multivalued) functions.
f is *Weihrauch reducible* to g ($f \leq_W g$) if there are computable $\Phi, \Psi :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ s.t.:

- Given a name p for $x \in \mathrm{dom}(f)$, $\Phi(p)$ is a name for $z \in \mathrm{dom}(g)$;
- Given a name q for $w \in g(z)$, $\Psi(p, q)$ is a name for $y \in f(x)$;



We denote a *multi-valued function* between represented spaces $\mathbf{X}$ and $\mathbf{Y}$ by $f :\subseteq \mathbf{X} \rightrightarrows \mathbf{Y}$. ($\rightrightarrows$ denotes that $x \in \mathbf{X}$ may be mapped to different elements of $\mathbf{Y}$).
Let f, g be (partial multivalued) functions.
f is *Weihrauch reducible* to g ($f \leq_W g$) if there are computable $\Phi, \Psi :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ s.t.:

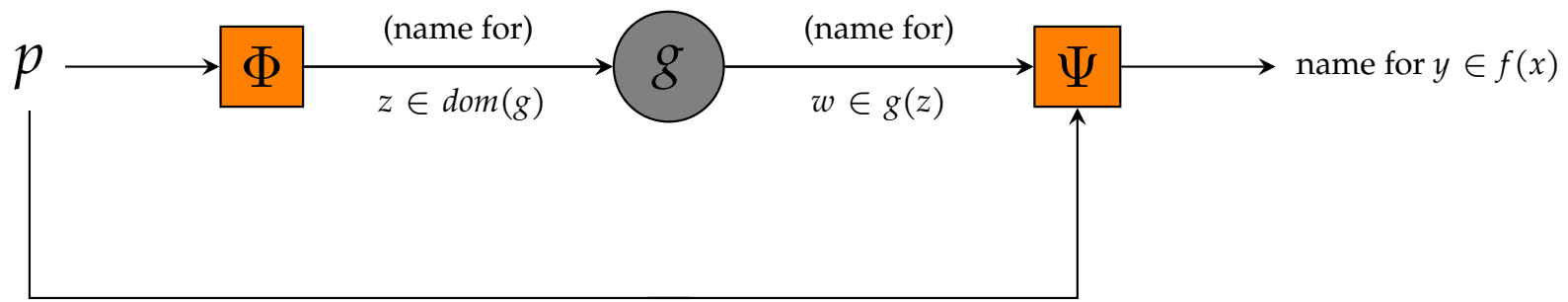
- Given a name p for $x \in \mathrm{dom}(f)$, $\Phi(p)$ is a name for $z \in \mathrm{dom}(g)$;
- Given a name q for $w \in g(z)$, $\Psi(p, q)$ is a name for $y \in f(x)$;



Weihrauch reducibility

We denote a *multi-valued function* between represented spaces $\mathbf{X}$ and $\mathbf{Y}$ by $f :\subseteq \mathbf{X} \rightrightarrows \mathbf{Y}$. ($\rightrightarrows$ denotes that $x \in \mathbf{X}$ may be mapped to different elements of $\mathbf{Y}$).
Let f, g be (partial multivalued) functions.
f is *Weihrauch reducible* to g ($f \leq_{\mathrm{W}} g$) if there are computable $\Phi, \Psi :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ s.t.:

- Given a name p for $x \in \mathrm{dom}(f)$, $\Phi(p)$ is a name for $z \in \mathrm{dom}(g)$;
- Given a name q for $w \in g(z)$, $\Psi(p, q)$ is a name for $y \in f(x)$;



LPO and WF

If $f \leq_W g$ and viceversa we write $f \equiv_W g$. The $\equiv_W$-equivalence classes are called *Weihrauch degrees*.

LPO and WF

If $f \leq_W g$ and viceversa we write $f \equiv_W g$. The $\equiv_W$-equivalence classes are called *Weihrauch degrees*.
The following principles are important for the rest of the talk:

- LPO : given in input $p \in 2^{\mathbb{N}}$, $\mathsf{LPO}(p) = 1$ iff $p = 0^{\mathbb{N}}$ (answers a $\mathbf{\Sigma}^0_1$ or $\mathbf{\Pi}^0_1$ question).
- WF: Given in input a tree $T \subseteq \mathbb{N}^{<\mathbb{N}}$, $\mathsf{WF}(T) = 1$ if T is well-founded (i.e. has no infinite branches), 0 otherwise (answers a $\mathbf{\Pi}^1_1$ or $\mathbf{\Sigma}^1_1$ question).

The (induced) subgraph problem

(Induced) subgraph problem

Notation: Given two graphs G and H,

$G \subseteq_{\mathbf{is}} H := (\exists G' \subseteq H)(G' \cong G \wedge G'$ is a induced subgraph of $H)$,

similarly for $\subseteq_{\mathbf{s}}$ (subgraph).

(Induced) subgraph problem

Notation: Given two graphs G and H,

$$G \subseteq_{\mathbf{is}} H := (\exists G' \subseteq H)(G' \cong G \wedge G' \text{ is a induced subgraph of } H),$$

similarly for $\subseteq_{\mathbf{s}}$ (subgraph).
In [BHW21], the authors, using a different notation, defined the functions $\mathsf{IS}_G : \mathbf{Gr} \to 2$ and $\mathsf{S}_G : \mathbf{Gr} \to 2$ for some fixed graph G as

$$\mathsf{IS}_G(H) = 1 \iff G \subseteq_{\mathbf{is}} H \text{ and } \mathsf{S}_G(H) = 1 \iff G \subseteq_{\mathbf{s}} H$$

(Induced) subgraph problem

Notation: Given two graphs G and H,

$$G \subseteq_{\mathbf{is}} H := (\exists G' \subseteq H)(G' \cong G \wedge G' \text{ is a induced subgraph of } H),$$

similarly for $\subseteq_{\mathbf{s}}$ (subgraph).
In [BHW21], the authors, using a different notation, defined the functions $\mathsf{IS}_G : \mathbf{Gr} \to 2$ and $\mathsf{S}_G : \mathbf{Gr} \to 2$ for some fixed graph G as

$$\mathsf{IS}_G(H) = 1 \iff G \subseteq_{\mathbf{is}} H \text{ and } \mathsf{S}_G(H) = 1 \iff G \subseteq_{\mathbf{s}} H$$

Let R (ray) be s.t. $V(\mathsf{R}) = \mathbb{N}$ and $E(\mathsf{R}) = \{(i, i+1) : i \in \mathbb{N}\}$.

(Induced) subgraph problem

Notation: Given two graphs G and H,

$$G \subseteq_{\mathbf{is}} H := (\exists G' \subseteq H)(G' \cong G \wedge G' \text{ is a induced subgraph of } H),$$

similarly for $\subseteq_{\mathbf{s}}$ (subgraph).
In [BHW21], the authors, using a different notation, defined the functions $\mathsf{IS}_G : \mathbf{Gr} \to 2$ and $\mathsf{S}_G : \mathbf{Gr} \to 2$ for some fixed graph G as

$$\mathsf{IS}_G(H) = 1 \iff G \subseteq_{\mathbf{is}} H \text{ and } \mathsf{S}_G(H) = 1 \iff G \subseteq_{\mathbf{s}} H$$

Let R (ray) be s.t. $V(\mathsf{R}) = \mathbb{N}$ and $E(\mathsf{R}) = \{(i, i+1) : i \in \mathbb{N}\}$.

Theorem ([BHW21])

$\mathsf{WF} \equiv_W \mathsf{IS}_{\mathsf{R}} \equiv_W \mathsf{S}_{\mathsf{R}}$ *and if* G *is finite,* $\mathsf{LPO} \equiv_W \mathsf{IS}_G \equiv_W \mathsf{S}_G$.

Open question

We first deal with IS_G.

Open question

We first deal with IS_G. Notice the huge gap between LPO and WF (i.e. $\mathbf{\Sigma}^0_1$ vs $\mathbf{\Pi}^1_1$).

Open question

We first deal with IS_G. Notice the huge gap between LPO and WF (i.e. $\mathbf{\Sigma}^0_1$ vs $\mathbf{\Pi}^1_1$). The authors left open the following question:

is there an infinite computable graph G s.t. $\mathsf{LPO} <_{\mathsf{W}} \mathsf{IS}_G <_{\mathsf{W}} \mathsf{WF}$?

Open question

We first deal with IS_G. Notice the huge gap between LPO and WF (i.e. $\mathbf{\Sigma}^0_1$ vs $\mathbf{\Pi}^1_1$). The authors left open the following question:

is there an infinite computable graph G s.t. $\mathsf{LPO} <_{\mathsf{W}} \mathsf{IS}_G <_{\mathsf{W}} \mathsf{WF}$?

We gave a negative answer and the proof follows from the following two lemmas.

Open question

We first deal with IS_G. Notice the huge gap between LPO and WF (i.e. $\mathbf{\Sigma}^0_1$ vs $\mathbf{\Pi}^1_1$). The authors left open the following question:

is there an infinite computable graph G s.t. $\mathsf{LPO} <_{\mathsf{W}} \mathsf{IS}_G <_{\mathsf{W}} \mathsf{WF}$?

We gave a negative answer and the proof follows from the following two lemmas.

Lemma ((1) C., Pauly)

For any computable graph G s.t. $\mathsf{R} \subseteq_{\mathbf{s}} G$, $\mathsf{WF} \equiv_W \mathsf{IS}_G$.

Open question

We first deal with IS_G. Notice the huge gap between LPO and WF (i.e. $\mathbf{\Sigma}^0_1$ vs $\mathbf{\Pi}^1_1$). The authors left open the following question:

is there an infinite computable graph G s.t. $\mathsf{LPO} <_{\mathsf{W}} \mathsf{IS}_G <_{\mathsf{W}} \mathsf{WF}$?

We gave a negative answer and the proof follows from the following two lemmas.

Lemma ((1) C., Pauly)

For any computable graph G s.t. $\mathsf{R} \subseteq_{\mathbf{s}} G$, $\mathsf{WF} \equiv_W \mathsf{IS}_G$.

Lemma ((2) C., Pauly)

Let K_ω be the complete graph on ω vertices and let G be an infinite computable graph s.t. $K_\omega \not\subseteq_{\mathbf{s}} G$. Then $\mathsf{IS}_G \equiv_W \mathsf{WF}$.

Some observation on the proofs

In both cases $\mathsf{IS}_G \leq_{\mathsf{W}} \mathsf{WF}$ follows from [BHW21].

Some observation on the proofs

In both cases $\mathsf{IS}_G \leq_{\mathsf{W}} \mathsf{WF}$ follows from [BHW21].
To prove $\mathsf{WF} \leq_{\mathsf{W}} \mathsf{IS}_G$ when $\mathsf{R} \subseteq_{\mathbf{s}} G$ given $T \subseteq \mathbb{N}^{<\mathbb{N}}$ an input for WF and $(v_i)_{i\in\mathbb{N}}$ be a computable enumeration of $V(G)$ we compute H so that $V(H) = T$ and

Some observation on the proofs

In both cases $\mathsf{IS}_G \leq_{\mathsf{W}} \mathsf{WF}$ follows from [BHW21].
To prove $\mathsf{WF} \leq_{\mathsf{W}} \mathsf{IS}_G$ when $\mathsf{R} \subseteq_{\mathbf{s}} G$ given $T \subseteq \mathbb{N}^{<\mathbb{N}}$ an input for WF and $(v_i)_{i\in\mathbb{N}}$ be a computable enumeration of $V(G)$ we compute H so that $V(H) = T$ and

$$E(H) = \{(\sigma, \tau) : (v_{|\sigma|}, v_{|\tau|}) \in E(G) \wedge (\sigma \sqsubset \tau \vee \tau \sqsubset \sigma)\}.$$

Some observation on the proofs

In both cases $\mathsf{IS}_G \leq_{\mathsf{W}} \mathsf{WF}$ follows from [BHW21].
To prove $\mathsf{WF} \leq_{\mathsf{W}} \mathsf{IS}_G$ when $\mathsf{R} \subseteq_{\mathbf{s}} G$ given $T \subseteq \mathbb{N}^{<\mathbb{N}}$ an input for WF and $(v_i)_{i\in\mathbb{N}}$ be a computable enumeration of $V(G)$ we compute H so that $V(H) = T$ and

$$E(H) = \{(\sigma, \tau) : (v_{|\sigma|}, v_{|\tau|}) \in E(G) \wedge (\sigma \sqsubset \tau \vee \tau \sqsubset \sigma)\}.$$

To prove $\mathsf{WF} \leq_{\mathsf{W}} \mathsf{IS}_G$ when $K_\omega \not\subseteq_{\mathbf{s}} G$ we proceed in the same fashion with the only difference that

$$E(H) = \{(\sigma, \tau) : (v_{|\sigma|}, v_{|\tau|}) \in E(G) \vee (\sigma|\tau)\}.$$

A negative answer

We have just shown that:

$$(1)\mathsf{R} \subseteq_{\mathbf{s}} G \implies \mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF} \text{ and } (2)K_\omega \not\subseteq_{\mathbf{s}} G \implies \mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF}.$$

A negative answer

We have just shown that:

$$(1)\mathsf{R} \subseteq_{\mathbf{s}} G \implies \mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF} \text{ and } (2)K_\omega \not\subseteq_{\mathbf{s}} G \implies \mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF}.$$

If $K_\omega \subseteq_{\mathbf{s}} G$ then also $\mathsf{R} \subseteq_{\mathbf{s}} G$, hence (1) applies. Otherwise if $K_\omega \not\subseteq_{\mathbf{s}} G$, (2) applies.

A negative answer

We have just shown that:

$$(1)\mathsf{R} \subseteq_{\mathbf{s}} G \implies \mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF} \text{ and } (2)K_\omega \not\subseteq_{\mathbf{s}} G \implies \mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF}.$$

If $K_\omega \subseteq_{\mathbf{s}} G$ then also $\mathsf{R} \subseteq_{\mathbf{s}} G$, hence (1) applies. Otherwise if $K_\omega \not\subseteq_{\mathbf{s}} G$, (2) applies.

Theorem (C., Pauly)

For every infinite computable graph G, $\mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF}$.

A negative answer

We have just shown that:

$$(1)\mathsf{R} \subseteq_{\mathbf{s}} G \implies \mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF} \text{ and } (2)K_\omega \not\subseteq_{\mathbf{s}} G \implies \mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF}.$$

If $K_\omega \subseteq_{\mathbf{s}} G$ then also $\mathsf{R} \subseteq_{\mathbf{s}} G$, hence (1) applies. Otherwise if $K_\omega \not\subseteq_{\mathbf{s}} G$, (2) applies.

Theorem (C., Pauly)

For every infinite computable graph G, $\mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF}$.

What about S_G?

A negative answer

We have just shown that:

$$(1)\mathsf{R} \subseteq_{\mathbf{s}} G \implies \mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF} \text{ and } (2)K_\omega \not\subseteq_{\mathbf{s}} G \implies \mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF}.$$

If $K_\omega \subseteq_{\mathbf{s}} G$ then also $\mathsf{R} \subseteq_{\mathbf{s}} G$, hence (1) applies. Otherwise if $K_\omega \not\subseteq_{\mathbf{s}} G$, (2) applies.

Theorem (C., Pauly)

For every infinite computable graph G, $\mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF}$.

What about S_G? We first define the *jump* of a problem.

A negative answer

We have just shown that:

$$(1)\mathsf{R} \subseteq_{\mathbf{s}} G \implies \mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF} \text{ and } (2)K_\omega \not\subseteq_{\mathbf{s}} G \implies \mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF}.$$

If $K_\omega \subseteq_{\mathbf{s}} G$ then also $\mathsf{R} \subseteq_{\mathbf{s}} G$, hence (1) applies. Otherwise if $K_\omega \not\subseteq_{\mathbf{s}} G$, (2) applies.

Theorem (C., Pauly)

For every infinite computable graph G, $\mathsf{IS}_G \equiv_{\mathsf{W}} \mathsf{WF}$.

What about S_G? We first define the *jump* of a problem. Given a problem f we define its jump f' as f with the only difference that f' receives in input a name that converges to an input of f. Similarly, we can define the n-fold jump $f^{(n)}$ of a problem f (notice that $f^{(0)} = f$).

Infinite and computable G implies LPO'

In [BHW21] the authors already showed that $\mathsf{S}_{\mathsf{R}} \equiv_{\mathsf{W}} \mathsf{WF}$, and, if G is finite, $\mathsf{S}_G \equiv_{\mathsf{W}} \mathsf{LPO}$.

Infinite and computable G implies LPO'

In [BHW21] the authors already showed that $\mathsf{S}_\mathsf{R} \equiv_\mathsf{W} \mathsf{WF}$, and, if G is finite, $\mathsf{S}_G \equiv_\mathsf{W} \mathsf{LPO}$. Pauly also observed the existence of a graph G s.t. $\mathsf{LPO}' \equiv_\mathsf{W} \mathsf{S}_G$.

Infinite and computable G implies LPO'

In [BHW21] the authors already showed that $\mathsf{S}_\mathsf{R} \equiv_\mathsf{W} \mathsf{WF}$, and, if G is finite, $\mathsf{S}_G \equiv_\mathsf{W} \mathsf{LPO}$. Pauly also observed the existence of a graph G s.t. $\mathsf{LPO}' \equiv_\mathsf{W} \mathsf{S}_G$.
It is convenient to consider $\mathsf{LPO}^{(n)}$ as the principle answering a $\mathbf{\Sigma}^0_{n+1}$ (or equivalently $\mathbf{\Pi}^0_{n+1}$) question.

Infinite and computable G implies LPO'

In [BHW21] the authors already showed that $\mathsf{S}_\mathsf{R} \equiv_\mathsf{W} \mathsf{WF}$, and, if G is finite, $\mathsf{S}_G \equiv_\mathsf{W} \mathsf{LPO}$. Pauly also observed the existence of a graph G s.t. $\mathsf{LPO}' \equiv_\mathsf{W} \mathsf{S}_G$.
It is convenient to consider $\mathsf{LPO}^{(n)}$ as the principle answering a $\mathbf{\Sigma}^0_{n+1}$ (or equivalently $\mathbf{\Pi}^0_{n+1}$) question.
Using the same construction used to prove that $\mathsf{S}_G \equiv_\mathsf{W} \mathsf{WF}$ with $K_\omega \not\subseteq_\mathbf{s} G$ we obtain the following.

Proposition (C., Pauly)

Let G be an infinite computable graph. Then $\mathsf{LPO}' \leq_\mathsf{W} \mathsf{S}_G$.

Is there any computable graph G s.t. $\mathsf{S}_G \equiv_\mathsf{W} f$ for $f \not\equiv_\mathsf{W} \mathsf{WF}, \mathsf{LPO}, \mathsf{LPO}'$?

S_G and $\mathsf{LPO}^{(n)}$

Theorem (C., Pauly)

For every $n \in \mathbb{N}$ there exists a computable graph G_n s.t. $\mathsf{LPO}^{(n)} \equiv_W \mathsf{S}_{G_n}$.

S_G and $\mathsf{LPO}^{(n)}$

Theorem (C., Pauly)

For every $n \in \mathbb{N}$ there exists a computable graph G_n s.t. $\mathsf{LPO}^{(n)} \equiv_W \mathsf{S}_{G_n}$.

We define G_n as follows:

- if $n = 2k$, $V(G_{2k}) = \{\sigma \in \mathbb{N}^{<\mathbb{N}} : |\sigma| \leq k\}$;
- if $n = 2k+1$, $V(G_{2k+1}) = \{\sigma \in \mathbb{N}^{<\mathbb{N}} : 0 < |\sigma| \leq k+1\}$.

In both cases,
$(\sigma, \tau) \in E(G_n) \iff (\sigma = \tau[|\tau| - 1] \vee \tau = \sigma[|\sigma| - 1])$.

G_n, informally

- G_0 is just a single vertex
- G_1 is the disconnected union of infinitely many copies of G_0, i.e. infinitely many disconnected vertices,
- G_2 is the graph obtained connecting to a new vertex all the components of G_1, i.e. a tree of height one with infinitely many children,
- G_3 is the disconnected union of infinitely many copies of G_2,
- G_4 is the graph obtained connecting to a new vertex all the components of G_3, ...

G_n, informally

- G_0 is just a single vertex
- G_1 is the disconnected union of infinitely many copies of G_0, i.e. infinitely many disconnected vertices,
- G_2 is the graph obtained connecting to a new vertex all the components of G_1, i.e. a tree of height one with infinitely many children,
- G_3 is the disconnected union of infinitely many copies of G_2,
- G_4 is the graph obtained connecting to a new vertex all the components of G_3, ...

Key observation: $G_n \subseteq_{\mathbf{s}} H$ iff $\varphi_n(H)$ where if n is even φ_n is a $\mathbf{\Sigma}^0_{n+1}$-complete formula, while if n is odd φ_n is a $\mathbf{\Pi}^0_{n+1}$-complete one.

G_n, informally

- G_0 is just a single vertex
- G_1 is the disconnected union of infinitely many copies of G_0, i.e. infinitely many disconnected vertices,
- G_2 is the graph obtained connecting to a new vertex all the components of G_1, i.e. a tree of height one with infinitely many children,
- G_3 is the disconnected union of infinitely many copies of G_2,
- G_4 is the graph obtained connecting to a new vertex all the components of G_3, ...

Key observation: $G_n \subseteq_{\mathbf{s}} H$ iff $\varphi_n(H)$ where if n is even φ_n is a $\mathbf{\Sigma}^0_{n+1}$-complete formula, while if n is odd φ_n is a $\mathbf{\Pi}^0_{n+1}$-complete one. For example $G_3 \subseteq_{\mathbf{s}} H$ iff
$(\exists^\infty v \in V(H))(\exists^\infty w \in V(H))((v,w) \in E(H))$.

When G is in the graph

Finding the induced subgraph

We first introduce the following well known multi-valued function.

- $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$: given in input $T \subseteq \mathbb{N}^{<\mathbb{N}}$ s.t. T is ill-founded output some $p \in [T]$.

Finding the induced subgraph

We first introduce the following well known multi-valued function.

- $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$: given in input $T \subseteq \mathbb{N}^{<\mathbb{N}}$ s.t. T is ill-founded output some $p \in [T]$.

Recall that WF answers the question whether a tree T is well-founded or not. In the previous slides we showed that $\mathsf{WF} \equiv_{\mathsf{W}} \mathsf{IS}_G$ where G is an infinite computable graph: in some sense, IS_G is the counterpart of WF in the induced subgraph terms.

Finding the induced subgraph

We first introduce the following well known multi-valued function.

- $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$: given in input $T \subseteq \mathbb{N}^{<\mathbb{N}}$ s.t. T is ill-founded output some $p \in [T]$.

Recall that WF answers the question whether a tree T is well-founded or not. In the previous slides we showed that $\mathsf{WF} \equiv_{\mathsf{W}} \mathsf{IS}_G$ where G is an infinite computable graph: in some sense, IS_G is the counterpart of WF in the induced subgraph terms. What about the counterpart of $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$?

Finding the induced subgraph

We first introduce the following well known multi-valued function.

- $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$: given in input $T \subseteq \mathbb{N}^{<\mathbb{N}}$ s.t. T is ill-founded output some $p \in [T]$.

Recall that WF answers the question whether a tree T is well-founded or not. In the previous slides we showed that $\mathsf{WF} \equiv_{\mathsf{W}} \mathsf{IS}_G$ where G is an infinite computable graph: in some sense, IS_G is the counterpart of WF in the induced subgraph terms. What about the counterpart of $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$?
For some fixed computable graph G, let $\mathsf{FindIS}_G :\subseteq \mathbf{Gr} \rightrightarrows \mathbf{Gr}$ be s.t.

$$\mathsf{FindIS}_G(H) := \{G' : G' \cong G\}$$

where $\mathrm{dom}(\mathsf{FindIS}_G) = \{H : G \subseteq_{\mathbf{is}} H\}$.

$\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$ and FindIS_G

Proposition (C., Pauly)

If G is computable, then $\mathsf{FindIS}_G \leq_W \mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$. In case G is also finite, FindIS_G is computable.

$\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$ and FindIS_G

Proposition (C., Pauly)

If G is computable, then $\mathsf{FindIS}_G \leq_W \mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$. In case G is also finite, FindIS_G is computable.

The proofs are essentially the same of $\mathsf{IS}_G \leq_W \mathsf{WF}$ for G computable and of $\mathsf{LPO} \equiv_W \mathsf{IS}_G$ for G finite.

$\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$ and FindIS_G

Proposition (C., Pauly)

If G is computable, then $\mathsf{FindIS}_G \leq_W \mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$. In case G is also finite, FindIS_G is computable.

The proofs are essentially the same of $\mathsf{IS}_G \leq_W \mathsf{WF}$ for G computable and of $\mathsf{LPO} \equiv_W \mathsf{IS}_G$ for G finite. Is it the case that for an infinite computable graph G, $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \equiv_W \mathsf{FindIS}_G$?

Proposition (C., Pauly)

If G is computable, then $\mathsf{FindIS}_G \leq_{\mathsf{W}} \mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$. In case G is also finite, FindIS_G is computable.

The proofs are essentially the same of $\mathsf{IS}_G \leq_{\mathsf{W}} \mathsf{WF}$ for G computable and of $\mathsf{LPO} \equiv_{\mathsf{W}} \mathsf{IS}_G$ for G finite. Is it the case that for an infinite computable graph G, $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \equiv_{\mathsf{W}} \mathsf{FindIS}_G$?
Notation: $\deg^G(v) = n$ denotes "v has n-many neighbors in G".

$\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$ and FindIS_G

Proposition (C., Pauly)

If G is computable, then $\mathsf{FindIS}_G \leq_W \mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$. In case G is also finite, FindIS_G is computable.

The proofs are essentially the same of $\mathsf{IS}_G \leq_W \mathsf{WF}$ for G computable and of $\mathsf{LPO} \equiv_W \mathsf{IS}_G$ for G finite. Is it the case that for an infinite computable graph G, $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \equiv_W \mathsf{FindIS}_G$?
Notation: $\deg^G(v) = n$ denotes "v has n-many neighbors in G".

Theorem (C., Pauly)

Let G be an infinite computable graph s.t.

- $|\{v \in V(G) : \deg^G(v) < \aleph_0\}| < \aleph_0$ *or*
- $|\{v \in V(G) : \deg^G(v) = \aleph_0\}| < \aleph_0$.

Then $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \equiv_W \mathsf{FindIS}_G$.

Finding the induced subgraph

Notice that in the reduction $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \leq_{\mathsf{W}} \mathsf{FindIS}_G$, the input for FindIS_G is computed as in $\mathsf{WF} \leq_{\mathsf{W}} \mathsf{IS}_G$ for $K_\omega \not\subseteq_{\mathbf{s}} G$.

Finding the induced subgraph

Notice that in the reduction $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \leq_{\mathsf{W}} \mathsf{FindIS}_G$, the input for FindIS_G is computed as in $\mathsf{WF} \leq_{\mathsf{W}} \mathsf{IS}_G$ for $K_\omega \not\subseteq_{\mathbf{s}} G$.
The only G's left out from the theorem above are those s.t.
$|\{v \in V(G) : \deg^G(v) < \aleph_0\}| = \aleph_0$ and
$|\{v \in V(G) : \deg^G(v) = \aleph_0\}| = \aleph_0$.

Finding the induced subgraph

Notice that in the reduction $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \leq_{\mathsf{W}} \mathsf{FindIS}_G$, the input for FindIS_G is computed as in $\mathsf{WF} \leq_{\mathsf{W}} \mathsf{IS}_G$ for $K_\omega \not\subseteq_{\mathbf{s}} G$.
The only G's left out from the theorem above are those s.t. $|\{v \in V(G) : \deg^G(v) < \aleph_0\}| = \aleph_0$ and $|\{v \in V(G) : \deg^G(v) = \aleph_0\}| = \aleph_0$. In this case we were only able to obtain a continuous reduction (i.e. w.r.t. some oracle).

Finding the induced subgraph

Notice that in the reduction $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \leq_{\mathsf{W}} \mathsf{FindIS}_G$, the input for FindIS_G is computed as in $\mathsf{WF} \leq_{\mathsf{W}} \mathsf{IS}_G$ for $K_\omega \not\subseteq_{\mathbf{s}} G$.
The only G's left out from the theorem above are those s.t. $|\{v \in V(G) : \deg^G(v) < \aleph_0\}| = \aleph_0$ and $|\{v \in V(G) : \deg^G(v) = \aleph_0\}| = \aleph_0$. In this case we were only able to obtain a continuous reduction (i.e. w.r.t. some oracle).

What about FindS_G?

Finding the induced subgraph

Notice that in the reduction $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \leq_{\mathsf{W}} \mathsf{FindIS}_G$, the input for FindIS_G is computed as in $\mathsf{WF} \leq_{\mathsf{W}} \mathsf{IS}_G$ for $K_\omega \not\subseteq_{\mathbf{s}} G$.
The only G's left out from the theorem above are those s.t. $|\{v \in V(G) : \deg^G(v) < \aleph_0\}| = \aleph_0$ and $|\{v \in V(G) : \deg^G(v) = \aleph_0\}| = \aleph_0$. In this case we were only able to obtain a continuous reduction (i.e. w.r.t. some oracle).

What about FindS_G? As before $\mathsf{FindS}_G \leq_{\mathsf{W}} \mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$ and if G is finite FindS_G is computable.

Finding the induced subgraph

Notice that in the reduction $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \leq_{\mathsf{W}} \mathsf{FindIS}_G$, the input for FindIS_G is computed as in $\mathsf{WF} \leq_{\mathsf{W}} \mathsf{IS}_G$ for $K_\omega \not\subseteq_{\mathbf{s}} G$.
The only G's left out from the theorem above are those s.t. $|\{v \in V(G) : \deg^G(v) < \aleph_0\}| = \aleph_0$ and $|\{v \in V(G) : \deg^G(v) = \aleph_0\}| = \aleph_0$. In this case we were only able to obtain a continuous reduction (i.e. w.r.t. some oracle).

What about FindS_G? As before $\mathsf{FindS}_G \leq_{\mathsf{W}} \mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$ and if G is finite FindS_G is computable.
We first consider the case of computable G's s.t. $\mathsf{R} \subseteq_{\mathbf{s}} G$.

Finding the induced subgraph

Notice that in the reduction $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \leq_{\mathsf{W}} \mathsf{FindIS}_G$, the input for FindIS_G is computed as in $\mathsf{WF} \leq_{\mathsf{W}} \mathsf{IS}_G$ for $K_\omega \not\subseteq_{\mathbf{s}} G$.
The only G's left out from the theorem above are those s.t. $|\{v \in V(G) : \deg^G(v) < \aleph_0\}| = \aleph_0$ and $|\{v \in V(G) : \deg^G(v) = \aleph_0\}| = \aleph_0$. In this case we were only able to obtain a continuous reduction (i.e. w.r.t. some oracle).

What about FindS_G? As before $\mathsf{FindS}_G \leq_{\mathsf{W}} \mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$ and if G is finite FindS_G is computable.
We first consider the case of computable G's s.t. $\mathsf{R} \subseteq_{\mathbf{s}} G$. We proved that for those G's s.t. finding an embedding from R to a graph containing R is computable, $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \equiv_{\mathsf{W}} \mathsf{FindIS}_G$. This is the case when G is "$\mathbb{Z}$", the full binary tree, ...

Finding the induced subgraph

Notice that in the reduction $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \leq_{\mathsf{W}} \mathsf{FindIS}_G$, the input for FindIS_G is computed as in $\mathsf{WF} \leq_{\mathsf{W}} \mathsf{IS}_G$ for $K_\omega \not\subseteq_{\mathbf{s}} G$.
The only G's left out from the theorem above are those s.t. $|\{v \in V(G) : \deg^G(v) < \aleph_0\}| = \aleph_0$ and $|\{v \in V(G) : \deg^G(v) = \aleph_0\}| = \aleph_0$. In this case we were only able to obtain a continuous reduction (i.e. w.r.t. some oracle).

What about FindS_G? As before $\mathsf{FindS}_G \leq_{\mathsf{W}} \mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$ and if G is finite FindS_G is computable.
We first consider the case of computable G's s.t. $\mathsf{R} \subseteq_{\mathbf{s}} G$. We proved that for those G's s.t. finding an embedding from R to a graph containing R is computable, $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \equiv_{\mathsf{W}} \mathsf{FindIS}_G$. This is the case when G is "$\mathbb{Z}$", the full binary tree, ...does this hold for R?

$\mathsf{FindS}_{\mathsf{R}}$

- $\lim_2$: given in input a converging sequence in $\{0,1\}$, outputs its limit.

Proposition (C., Pauly)

$\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \equiv_W \lim_2 * \mathsf{FindS}_{\mathsf{R}}$.

Here $*$ denotes the *compositional product*. For two multivalued functions $f, g, f * g$ denotes that, given a (name for an) input for g, we apply g, we compute a name for an input of f, and finally we apply f.

$\mathsf{FindS}_{\mathsf{R}}$

- $\lim_2$: given in input a converging sequence in $\{0,1\}$, outputs its limit.

Proposition (C., Pauly)

$\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \equiv_W \lim_2 * \mathsf{FindS}_{\mathsf{R}}$.

Here $*$ denotes the *compositional product*. For two multivalued functions $f, g, f * g$ denotes that, given a (name for an) input for g, we apply g, we compute a name for an input of f, and finally we apply f.
The proposition above shows that $\mathsf{FindS}_{\mathsf{R}}$ needs very little to reach $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$, on the other hand, the question whether $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}} \equiv_W \mathsf{FindS}_{\mathsf{R}}$ is still open.

Other subgraphs

$G \bigoplus H$ denotes the disconnected union of G and H.

Other subgraphs

$G \bigoplus H$ denotes the disconnected union of G and H. Is it possible to reach $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$ is we consider graphs in which every component is finite, i.e. $G = \bigoplus_{n \in \mathbb{N}} F_n$?

Other subgraphs

$G \bigoplus H$ denotes the disconnected union of G and H. Is it possible to reach $\mathsf{C}_{\mathbb{N}^\mathbb{N}}$ is we consider graphs in which every component is finite, i.e. $G = \bigoplus_{n \in \mathbb{N}} F_n$?

Theorem (C., Pauly)

Let G be as above:

- $(\forall^\infty i)(\exists^\infty j)(F_i \subseteq_{\mathbf{s}} F_j)$ *implies that* FindS_G *is computable.*
- *Let C_n be the cyclic graph of n vertices. Then for $G' = \bigoplus_{n \geq 3} C_n$ (notice that G' does not satisfy the condition above) we have that* $\mathsf{FindS}_{G'} \equiv_W \mathsf{C}_{\mathbb{N}^\mathbb{N}}$.

Other subgraphs

$G \bigoplus H$ denotes the disconnected union of G and H. Is it possible to reach $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$ is we consider graphs in which every component is finite, i.e. $G = \bigoplus_{n \in \mathbb{N}} F_n$?

Theorem (C., Pauly)

Let G be as above:

- $(\forall^{\infty} i)(\exists^{\infty} j)(F_i \subseteq_{\mathbf{s}} F_j)$ *implies that* FindS_G *is computable.*
- *Let C_n be the cyclic graph of n vertices. Then for $G' = \bigoplus_{n \geq 3} C_n$ (notice that G' does not satisfy the condition above) we have that* $\mathsf{FindS}_{G'} \equiv_W \mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$.

We conclude this section mentioning that with the same graphs used to prove that for every n, $\mathsf{LPO}^{(n)} \equiv_W \mathsf{S}_{G_n}$, we obtained a similar result for FindS_{G_n}.

The opposite problem

Some results

Recall that $\mathsf{IS}_G(H) = 1 \iff G \subseteq_{\mathbf{is}} H$.

Some results

Recall that $\mathsf{IS}_G(H) = 1 \iff G \subseteq_{\mathbf{is}} H$. We define the "opposite problem" as

$$\mathsf{IS}^G = 1 \iff H \subseteq_{\mathbf{is}} G.$$

Some results

Recall that $\mathsf{IS}_G(H) = 1 \iff G \subseteq_{\mathbf{is}} H$. We define the "opposite problem" as

$$\mathsf{IS}^G = 1 \iff H \subseteq_{\mathbf{is}} G.$$

Notation: K_ω is the complete graph with infinitely many vertices, K_i is the complete graph of i vertices and D the totally disconnected graph.

Some results

Recall that $\mathsf{IS}_G(H) = 1 \iff G \subseteq_{\mathbf{is}} H$. We define the "opposite problem" as

$$\mathsf{IS}^G = 1 \iff H \subseteq_{\mathbf{is}} G.$$

Notation: K_ω is the complete graph with infinitely many vertices, K_i is the complete graph of i vertices and D the totally disconnected graph. In [BHW21] the authors proved that $\mathsf{IS}^{K_\omega} \equiv_{\mathsf{W}} \mathsf{IS}^{D} \equiv_{\mathsf{W}} \mathsf{LPO}$ and left open the following question: is there a graph G s.t. $\mathsf{LPO} <_{\mathsf{W}} \mathsf{IS}^G$?

Theorem (C., Pauly)

Let $G = \bigoplus_{n \geq 3} K_n$. *Then* $\mathsf{IS}^G \equiv_{\mathsf{W}} \mathsf{LPO}''$.

Still open: is there a computable graph G s.t.$\mathsf{IS}^G \equiv_{\mathsf{W}} \mathsf{LPO}'$ or s.t. $\mathsf{LPO}'' <_{\mathsf{W}} \mathsf{IS}^G$?

Thanks for Your attention!

Bibliography

[BHW21] Zach BeMent, Jeffry L. Hirst, and Asuka Wallace. "Reverse mathematics and Weihrauch analysis motivated by finite complexity theory". In: *Computability* 10.4 (2021). arXiv 2105.01719, pp. 343–354. DOI: 10.3233/COM-210310. URL: https://doi.org/10.3233/COM-210310.

[HL96] Jeffry L. Hirst and Steffen Lempp. "Infinite Versions of Some Problems from Finite Complexity Theory". In: *Notre Dame Journal of Formal Logic* 37.4 (1996), pp. 545 –553. DOI: 10.1305/ndjfl/1040046141.