The infinite loop operation and the axiom of dependent choice

Takayuki Kihara

Nagoya University, Japan

The 21th CCA
Kyoto September 24, 2025

This is a talk on Weihrauch reducibility
and its applications to (semi-)constructive mathematics.

Table of Contents

Naïve Idea.

- *Weihrauch reducibility* is a tool for the computability-theoretic classification of ∀∃-theorems.

- Think of an ∀∃-theorem $\forall x \in X \exists y \in Y \varphi(x, y)$ as a *partial multi-valued* function.
  — Input: a code of $x \in X$.
  — Output: a code of $y \in Y$ such that $\varphi(x, y)$ holds.
  ▷ Unless otherwise specified, a code is an element in $\mathbb{N}^{\mathbb{N}}$.

- A Weihrauch problem is a partial multifunction $:\subseteq \mathbb{N}^{\mathbb{N}} \rightrightarrows \mathbb{N}^{\mathbb{N}}$.

For realizability theorists:

- A Weihrauch problem is a predicate $\varphi : X \to \Omega$, where X is a (partitioned) modest set.
  ▷ $X \to \Omega$ is the same thing as $X \rightrightarrows \mathbb{N}^{\mathbb{N}}$.
  ▷ As a problem: "Given a code of x, find a realizer for $\varphi(x)$."

A multifunction $F :\subseteq \mathbb{N}^{\mathbb{N}} \rightrightarrows \mathbb{N}^{\mathbb{N}}$ is *Weihrauch reducible* to $G :\subseteq \mathbb{N}^{\mathbb{N}} \rightrightarrows \mathbb{N}^{\mathbb{N}}$ if there exist partial computable functions φ, ψ on $\mathbb{N}^{\mathbb{N}}$ such that

- given an instance $x \in \mathbf{dom}(F)$,
- the inner reduction φ yields an instance $\tilde{x} \in \mathbf{dom}(G)$,
- and then, given a solution $y \in G(\tilde{x})$,
- the outer reduction ψ yields a solution $\tilde{y} \in F(x)$.

	Oracle	*Computer*
1:	$x \in \mathbf{dom}(F)$	
2:		$\varphi(x) = \tilde{x} \in \mathbf{dom}(G)$
3:	$y \in G(\tilde{x})$	
4:		$\psi(x, y) = \tilde{y} \in F(x)$

▷ This corresponds to a relative computability that makes a *single query* to an oracle.

How to make many queries?

- Brattka-Gherardi-Marcone (2012) introduced the operation $G \star H$;
  - ▹ The computation first makes a query to H once, and then makes a query to G once.
- $F \leq_W G \star H \iff$ ***Computer*** has a computable winning strategy for the following game:

	Oracle	*Computer*
1:	$x_0 \in \mathrm{dom}(F)$	
2:		$z_0 \in \mathrm{dom}(H)$
3:	$x_1 \in H(z_0)$	
4:		$z_1 \in \mathrm{dom}(G)$
5:	$x_2 \in G(z_1)$	
6:		$z_2 \in F(x_0)$

▹ F is n-queries reducible to $G \iff F \leq_W \underbrace{G \star \cdots \star G}_{n}$

Hirschfeldt-Jockusch's reduction game (2016):

Oracle		*Computer*
$x_0 \in \mathbf{dom}(F)$		
	Query:	$z_0 \in \mathbf{dom}(G)$
$x_1 \in G(z_0)$		
	Query:	$z_1 \in \mathbf{dom}(G)$
$x_2 \in G(z_1)$		
$\vdots$		$\vdots$
	Query:	$z_{n-1} \in \mathbf{dom}(G)$
$x_n \in G(z_{n-1})$		
	Halt:	$z_n \in F(x_0)$

- *Oracle*'s moves are $x_0, x_1, x_2, \ldots$
- *Computer*'s moves are of the forms $(\mathtt{Query}, z_i)$ or $(\mathtt{Halt}, z_i)$.
  - `Query` is a signal to ask a query to oracle.
  - `Halt` is a signal to terminate the computation.
- F is *G.Weihrauch reducible* to G (written $F \leq_{GW} G$)
  $\iff$ *Computer* has a computable winning strategy.

Def. (Neumann-Pauly 2018). For a Weihrauch problem F, define a new Weihrauch problem $F^\diamond$ as follows:

- $\sigma \in \mathbf{dom}(F^\diamond) \iff \sigma$ is ***Computer***'s continuous strategy that eventually halts at some round, regardless of ***Oracle***'s strategy.
- $F^\diamond(\sigma)$ is the value of ***Computer***'s last move (e.g., z_n in the play below).

Oracle	***Computer***
	Query: $z_0 \in \mathbf{dom}(F)$
$x_1 \in F(z_0)$	
	Query: $z_1 \in \mathbf{dom}(F)$
$x_2 \in F(z_1)$	
$\vdots$	$\vdots$
	Query: $z_{n-1} \in \mathbf{dom}(F)$
$x_n \in F(z_{n-1})$	
	Halt: z_n

▷ *Prop. (Neumann-Pauly 2018).* $F \leq_{GW} G \iff F \leq_W G^\diamond$.

Def. (Brattka 2025, Yoshimura). For a Weihrauch problem F, define a new Weihrauch problem F^∞ as follows:

- $\sigma \in \mathrm{dom}(F^\infty) \iff \sigma$ is *Computer*'s continuous strategy.
- $F^\infty(\sigma)$ is the history of the resulting computation.

Oracle	*Computer*
	$z_0 \in \mathrm{dom}(F)$
$x_1 \in F(z_0)$	
	$z_1 \in \mathrm{dom}(F)$
$x_2 \in F(z_1)$	
$\vdots$	$\vdots$
	$z_{n-1} \in \mathrm{dom}(F)$
$x_n \in F(z_{n-1})$	
$\vdots$	$\vdots$

Abstract Definitions (Pradic-Price, Ahman-Bauer):

- A Weih. problem F can be expressed as a polynomial functor.

$$P_F := \lambda X. \sum_{x \in \mathrm{dom}(F)} X^{F(x)}$$

- $G \star F$ is the composition of polynomial functors F, G:

$$P_{G \star F} \simeq P_F \circ P_G$$

- $F^\diamond$ is the initial algebra for a polynomial functor (W-type)

$$P_{F^\diamond} \simeq \mu(1 + P_F)$$

    - 1 is a constant symbol for "Halt."

- F^∞ is the final coalgebra for a polynomial functor (M-type)

$$P_{F^\infty} \simeq \nu P_F$$

Description of the initial algebra μP for a polynomial functor P:

- W-type: the well-founded trees.
- the (internal) colimit of transfinite iterated applications of F:

$$0 \xrightarrow{!} P0 \xrightarrow{P!} P^2 0 \xrightarrow{P^2!} \cdots \longrightarrow P^\omega 0 = \operatorname*{colim}_{n<\omega} P^n 0 \longrightarrow \cdots$$

(For $F^\diamond$, represent HJ's reduction game (2016) using a strategy tree.
Lee-van Oosten (2011/2013)'s definition is closer to the W-type.)

Description of the final coalgebra νP for a polynomial functor P:

- M-type: the (possibly ill-founded) trees.
- the (internal) limit of ω-iterated applications of P:

$$\nu P \simeq \lim_{n<\omega} P^n 1 \longrightarrow \cdots \xrightarrow{P^2!} P^2 1 \xrightarrow{P!} P1 \xrightarrow{!} 1$$

(In other words, F^∞ can be written either as *trees* or as an *inverse limit*.
Yoshimura (2016)'s is coalgebraic, and Brattka (2025) uses the inverse limit.)

Table of Contents

Kleene's realizability interpretation (1945)

- A realizer for $A \wedge B$ is a pair of realizers of A and B.
- A realizer for $A \vee B$ is a pair of a tag indicating which of A or B is correct and a realizer of the formula for the correct side.
- A realizer for $A \to B$ is (a code of) a computable function that, given a realizer of A, outputs a realizer of B.
- A realizer for $\exists x \in I.\, A(x)$ is a pair of a code of a witness $c \in I$ of the existence and a realizer of the formula $A(c)$.
- A realizer for $\forall x \in I.\, A(x)$ is (a code of) a computable function that, given a code of an element $c \in I$, outputs a realizer of $A(c)$.

- This interpretation can obviously be made relative to an oracle.
  - Given an oracle α, replace "computable" with "α-computable".
  - An oracle α is not necessarily single-valued or total.
  - An oracle changes semantics.

- $F :\subseteq \mathbb{N} \rightrightarrows \mathbb{N}$ a Weihrauch problem (partial multi-valued function).
- An F-realizer for $A \to B$ is *Computer*'s computable winning strategy for the following game:

Oracle	*Computer*
x_0 realizes A	
	Query: $z_0 \in \mathrm{dom}(F)$
$x_1 \in F(z_0)$	
	Query: $z_1 \in \mathrm{dom}(F)$
$x_2 \in F(z_1)$	
$\vdots$	$\vdots$
	Query: $z_n \in \mathrm{dom}(F)$
$x_{n+1} \in F(z_n)$	
	Halt: z_{n+1} realizes B

- *Oracle*'s moves are $x_0, x_1, x_2, \ldots$
- *Computer*'s moves are of the forms $(\mathtt{Query}, z_i)$ or $(\mathtt{Halt}, z_i)$.
  - ▷ `Query` is a signal to ask a query to oracle.
  - ▷ `Halt` is a signal to terminate the computation.

Hirschfeldt-Jockusch's reduction game (2016):

Oracle	*Computer*
$x_0 \in \mathrm{dom}(G)$	
	Query: $z_0 \in \mathrm{dom}(F)$
$x_1 \in F(z_0)$	
	Query: $z_1 \in \mathrm{dom}(F)$
$x_2 \in F(z_1)$	
$\vdots$	$\vdots$
	Query: $z_{n-1} \in \mathrm{dom}(F)$
$x_n \in F(z_{n-1})$	
	Halt: $z_n \in G(x_0)$

- *Oracle*'s moves are $x_0, x_1, x_2, \dots$
- *Computer*'s moves are of the forms $(\mathtt{Query}, z_i)$ or $(\mathtt{Halt}, z_i)$.
  - `Query` is a signal to ask a query to oracle.
  - `Halt` is a signal to terminate the computation.
- G is *G.Weihrauch reducible* to F (written $G \leq_{GW} F$)

  $\iff$ *Computer* has a computable winning strategy.

Theorem (essentially Yoshimura 2016)

The following are equivalent for a "well-behaved" Weihrauch problem F:

1. The axiom of dependent choice is F-realizable.
2. $F^\diamond$ is closed under the infinite loop $(F^\diamond)^\infty \equiv_W F^\diamond$.

- *Note:* Due to an insufficient understanding of $(-)^\diamond$ at that point, Yoshimura was unable to describe his theorem in this form.
- I found an error in the proof I wrote immediately after Dagstuhl, so I'm temporarily adding a "well-behavedness" assumption. This well-behavedness assumption poses no problem for specific problems we'll address from here on.

Related???

Fact

Let $\mathcal{E}$ be a topos. The following are equivalent:

1. $\mathcal{E}$ satisfies the axiom of dependent choice:
   - ▷ $\forall x \in X\, \exists y \in X\, \varphi(x,y)$
     $\to \forall x_0 \in X\, \exists f \in X^\omega\, (f(0) = x_0\ \&\ \forall n \in \omega\, \varphi(f(n), f(n+1)))$.
2. Any ω^{op}-limit (i.e., inverse limit) preserves surjectivity:
   - ▷ $g : X \to X$ surjective $\implies \pi_0 : \lim_{n<\omega} g^{(n)} \to X$ surjective.

Table of Contents

Previous works:

- Yoshimura (2016): $UC_{\mathbb{N}^\mathbb{N}}$ and $C_{\mathbb{N}^\mathbb{N}}$ are closed under $(-)^\infty$.
- Brattka (2025): $C_{2^\mathbb{N}}$ (so WKL) is closed under $(-)^\infty$.

$LLPO_k$:

- Input: (a code of) $(n_i)_{i<k} \in \mathbb{S}^k$ such that $n_i = 1$ for at most one i.
- Output: $i < k$ such that $n_i = 0$.

$LLPO := LLPO_2$

- Brattka (2025): $LLPO^{\infty\infty} \equiv_W WKL^\infty \equiv_W WKL$.

Question (Dagstuhl, Mar. 2025):

- What is the "∞-closure" of $LLPO_k$?

DNR_k:

- Input: (a code of) a partial function $f :\subseteq \mathbb{N} \to k$.
- Output: $g \in k^{\mathbb{N}}$ such that $g(n) \neq f(n)$ for any $n \in \mathbf{dom}(f)$.

Theorem (K.)

$LLPO_k^{\infty\infty} \equiv_W DNR_k^{\infty} \equiv_W DNR_k$.

Corollary:

- The ∞-closure of $LLPO_{k+1} <_W$ the ∞-closure of $LLPO_k$.
- ▹ Indeed: $LLPO_k \not\leq_W LLPO_{k+1}^{\infty}$.

- $LLPO_k \not\leq_W LLPO^{\infty}_{k+1}$.
- Recall: DC is related to infinite loop (inverse limit).
  Thus, considering $LLPO^{\infty}_{k+1}$-realizability, we obtain:

Theorem (K.)

$IZF + DC + MP$ does not prove $LLPO_{k+1} \to LLPO_k$.

- IZF: Intuitionistic ZF set theory
- DC: dependent choice
- MP: Markov's principle
  (double negation elimination for Σ^0_1 formulas)

Logical Principles

- $WLEM_k$: If $\varphi(n)$ holds for at most one $n < k$, then there exists $n < k$ for which $\neg\varphi(n)$ holds.
- $LLPO_k$: The restriction of $WLEM_k$ to Σ^0_1 formulas.
- There is a complexity hierarchy inbetween $LLPO_k$ and $WLEM_k$:

$$LLPO_k <_W \Sigma_2\text{-}LLPO_k <_W \cdots <_W \Pi^1_1\text{-}LLPO_k <_W \cdots <_W WLEM_k$$

Key Point

- Γ-$LLPO_k$ can be presented as a Weihrauch problem.
  - Input: an index of a Γ formula $\varphi(n, x)$ with a parameter x, where $\varphi(n, x)$ holds for at most one $n < k$.
  - Output: some n s.t. $\neg\varphi(n, x)$ holds.
- $WLEM_k$ cannot be presented as a Weihrauch problem.
  - There is no represented space of *all* propositions.
    - We need something like the object Ω of *all* propositions.
  - Thus, the notation $LLPO_k <_W WLEM_k$ does not make sense.

Def. (Bauer 2022, K. 2023). An extended Weihrauch problem is a partial multifunction $F :\subseteq \mathbb{N} \times \Lambda \rightrightarrows \mathbb{N}$.

- Λ is a set.
- Each $p \in \mathbb{N}$ is called a *public input* (available for $\mathbf{Computer}$).
- Each $\alpha \in \Lambda$ is called a *secret input* (*not* available for $\mathbf{Computer}$).

Note:

This is just a predicate $F : A \to \Omega$, where A is regular projective.

Example:

Think of $\mathbf{WLEM}_k$ as the following extended Weihrauch problem:

- Public input: Nothing.
- Secret input: A formula $\varphi(n, x)$ with a parameter x, where $\varphi(n, x)$ holds for at most one $n < k$.
- Output: some n s.t. $\neg\varphi(n, x)$ holds.

(K. 2023) *Reduction game* for *extended Weihrauch problems*:

Merlin	Arthur	Nimue
$(\mathbf{p}_0 \mid x_0) \in \mathbf{dom}(F)$		
	Query: $(\mathbf{q}_0 \mid$	$z_0) \in \mathbf{dom}(G)$
$\mathbf{p}_1 \in G(z_0)$		
	Query: $(\mathbf{q}_1 \mid$	$z_1) \in \mathbf{dom}(G)$
$\mathbf{p}_2 \in G(z_1)$		
$\vdots$	$\vdots$	$\vdots$
	Query: $(\mathbf{q}_n \mid$	$z_n) \in \mathbf{dom}(G)$
$\mathbf{p}_{n+1} \in G(z_n)$		
	Halt: $\mathbf{q}_{n+1}$	$\in F(\mathbf{p}_0 \mid x_0)$

- Merlin's move is $(\mathbf{p}, x)$ or $\mathbf{p}$; Nimue's move is z; and Arthur's move is either $(\mathbf{Query}, q)$ or $(\mathbf{Halt}, q)$.
  - ▷ Arthur cannot see "*secret*" inputs and can only see "*public*" inputs, and Arthur can only perform computable procedures
  - ▷ Merlin and Nimue can see both "*public*" and "*secret*" inputs, and Merlin and Nimue can perform any procedure.
- F is *LT-reducible* to G $\iff$ Arthur-Nimue has a winning strategy where Arthur's strategy is computable.

Thm. (Hendtlass-Lubarsky 2016)

$\mathit{IZF} + \mathit{DC}$ does not prove $\mathit{WLEM}_{k+1} \to \mathit{LLPO}_k$.

Prop. (essentially, K. 2023 or folklore?)

(Any reasonable system) $+\,\mathit{MP}$ does prove $\mathit{WLEM}_{k+1} \to \mathit{LLPO}_k$.

Thm. (essentially, K. 2023)

$\mathit{IZF} + \mathit{MP}$ does not prove $\mathit{WLEM}_{k+2} \to \mathit{LLPO}_k$.

— *Why?*

- (K. 2023) $\mathit{LLPO}_k \leq_{\mathrm{LT}} \mathit{WLEM}_{k+1}$ and $\mathit{LLPO}_k \not\leq_{\mathrm{LT}} \mathit{WLEM}_{k+2}$.

— *Indeed, we now have:*

- (K.) Indeed, $\mathit{LLPO}_k \not\leq_{\mathrm{LT}} \mathit{WLEM}^{\infty}_{k+2}$.

Theorem (K.)

$\mathit{IZF} + \mathit{DC} + \mathit{MP}$ does not prove $\mathit{WLEM}_{k+2} \to \mathit{LLPO}_k$.

- Recall: DC is related to infinite loop (inverse limit) for multi-valued oracles.