DEGREES OF INCOMPUTABILITY, REALIZABILITY AND CONSTRUCTIVE REVERSE MATHEMATICS

TAKAYUKI KIHARA

ABSTRACT. There is a way of assigning a realizability notion to each degree of incomputability. In our setting, we make use of Weihrauch degrees (degrees of incomputability/discontinuity of partial multi-valued functions) to obtain Lifschitz-like relative realizability predicates. In this article, we present sample examples on how to lift some separation results on Weihrauch degrees to those over intuitionistic Zermelo-Fraenkel set theory **IZF**.

1. INTRODUCTION

1.1. **Summary.** This article is a contribution to *constructive reverse mathematics* initiated by Ishihara, cf. [18, 19]. For the overview of previous works on constructive reverse mathematics, we refer the reader to Diener [12] (and also [20]). However, contrary to [12], in our article, we do not include the axiom of countable choice $\mathsf{AC}_{\omega,Y}$ in our base system of constructive reverse mathematics. Here, $\mathsf{AC}_{X,Y}$ is the statement that

$$\forall x \in X \exists y \in Y \varphi(x,y) \longrightarrow \exists f \colon X \to Y \forall x \in X \varphi(x, f(x)),$$

and ω is the set of all natural numbers. Why do we think that the axiom of countable choice $\mathsf{AC}_{\omega,Y}$ should not be implicitly assumed? This is because including $\mathsf{AC}_{\omega,Y}$ makes it difficult to compare the results with classical reverse mathematics à la Friedman-Simpson [42]. For instance, weak König's lemma WKL is equivalent to the intermediate value theorem IVT over Biship-style constructive mathematics **BISH** (which entails countable choice), whereas WKL is strictly stronger than IVT in classical reverse mathematics.

In order to avoid this difficulty, several base systems other than **BISH** have been proposed. For instance, Troelstra's elementary analysis $\mathbf{EL}_0$ is widely used as a base system of constructive reverse mathematics [36]. Here, "$\mathbf{EL}_0$ plus the law of excluded middle" is exactly the base system $\mathbf{RCA}_0$ of classical reverse mathematics; and moreover, "$\mathbf{EL}_0$ plus the axiom of countable choice" is considered as Biship-style constructive mathematics **BISH**. Thus, $\mathbf{EL}_0$ lies in the intersection of $\mathbf{RCA}_0$ and **BISH**.

Our aim is to separate various non-constructive principles which are equivalent under countable choice $\mathsf{AC}_{\omega,Y}$, and our main tool is (a topological version of) Weihrauch reducibility (cf. [8]). The relationship between Weihrauch reducibility and intuitionistic systems are extensively studied, e.g. in [45, 51, 52]. Thus, some of separations may be automatically done using the previous works. Our main purpose in this article is to develop an intuitive and easily customizable framework constructing $\mathsf{AC}_{\omega,Y}$-free models separating such non-constructive principles. In practice, establishing a convenient framework for separation arguments is sometimes more useful than giving a completeness result (for instance, look at classical reverse mathematics, where computability-theorists make

1

use of Turing degree theory in order to construct ω-models separating various principles over $\mathbf{RCA}_0$, cf. [41]). Our work gives several sample separation results, not just over $\mathbf{EL}_0$, but even over intuitionistic Zermelo-Fraenkel set theory $\mathbf{IZF}$.

1.1.1. *Variants of weak König's lemma.* In this article, we discuss a hierarchy between LLPO and WKL which collapse under the axiom $\mathsf{AC}_{\omega,2}$ of countable choice (with binary values). Consider the following three principles:

- The binary expansion principle BE states that every regular Cauchy real has a binary expansion.
- The intermediate value theorem IVT states that for any continuous function $f\colon [0,1] \to [-1,1]$ if $f(0)$ and $f(1)$ have different signs then there is a regular Cauchy real $x \in [0,1]$ such that $f(x) = 0$.
- Weak König's lemma WKL states that every infinite binary tree has an infinite path.

Here, a regular Cauchy real is a real x which is represented by a sequence $(q_n)_{n\in\omega}$ of rational numbers such that $|q_n - q_m| < 2^{-n}$ for any $m \geq n$. As mentioned above, under countable choice $\mathsf{AC}_{\omega,2}$, we have

$$\mathsf{WKL} \leftrightarrow \mathsf{IVT} \leftrightarrow \mathsf{BE} \leftrightarrow \mathsf{LLPO}.$$

Even if countable choice, Markov's principle, etc. are absent, we have the forward implications; see Berger et al. [6]. However we will see that the implications are strict under the absence of countable choice.

Robust division principle RDIV. We also examine the *division* principles for reals. It is known that *Markov's principle* (double negation elimination for Σ^0_1-formulas) is equivalent to the statement that, given regular Cauchy reals $x, y \in [0,1]$, we can divide x by y whenever y is nonzero; that is, if y is nonzero then $z = x/y$ for some regular Cauchy real z.

However, it is algorithmically undecidable if a given real y is zero or not. To overcome this difficulty, we consider any regular Cauchy real z such that if y is nonzero then $z = x/y$. If we require $x \leq y$ then such a z always satisfies $x = yz$ whatever y is (since $y = 0$ and $x \leq 0$ implies $x = 0$, so any z satisfies $0 = 0z$). To avoid the difficulty of deciding if y is nonzero or not, we always assume this additional requirement, and then call it the *robust division* principle RDIV. In other words, RDIV is the following statement:

$$(\forall x, y \in [0,1])\ [x \leq y \to\ (\exists z \in [0,1])\ x = yz].$$

One may think that the condition $x \leq y$ is not decidable, but we can always replace x with $\min\{x, y\}$ without losing anything. Namely, the robust division principle RDIV is equivalent to the following:

$$(\forall x, y \in [0,1])(\exists z \in [0,1])\ \min\{x, y\} = yz.$$

The principle RDIV is known to be related to problems of finding Nash equilibria in bimatrix games [38] and of executing Gaussian elimination [25]. The following implications are known:

$$\text{WKL} \longrightarrow \text{IVT} \begin{array}{c} \nearrow \text{BE} \searrow \\ \\ \searrow \text{RDIV} \nearrow \end{array} \text{LLPO}$$
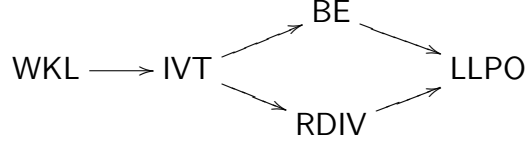
1.2. **Main Theorem.** In this article, to construct $\text{AC}_{\omega,2}$-free models of **IZF**, we incorporate Weihrauch reducibility [8] into Chen-Rathjen's Lifschitz-like realizability [10]. In general, we introduce the notion of a *j-operator* on a relative partial combinatory algebra. We call the pair of a relative partial combinatory algebra $\mathbb{P}$ and a *j-operator* **j** a *site*. Then we show that given a site $(\mathbb{P}, \mathbf{j})$ always yields a realizability predicate which validates **IZF**. As sample results of our method, we describe realizability interpretations of the following:

Theorem 1.1. *Each of the following is realizable (with respect to the realizability interpretation over a suitable site).*

(1) **IZF** + LLPO + ¬RDIV + ¬BE.
(2) **IZF** + RDIV + ¬BE.
(3) **IZF** + BE + ¬RDIV.
(4) **IZF** + RDIV + BE + ¬IVT.
(5) **IZF** + IVT + ¬WKL.

2. BACKGROUND AND KEY IDEAS

2.1. **Constructive reverse mathematics.** *Reverse mathematics* is the field that seeks to find the minimum axioms necessary to prove a given mathematical theorem [42]. Classical reverse mathematics measures the complexity of *induction* and *comprehension* principles which are necessary for proving a given theorem (the former is particularly important from a foundational perspective). Constructive reverse mathematics measures the complexity of the law of *excluded middle* which is necessary for proving a given theorem [12, 20]. We need benchmark principles to measure such complexities. One such benchmark is the arithmetical hierarchy of the law of excluded middle.

2.1.1. *Law of excluded middle.* We here consider central nonconstructive principles, the *law of excluded middle* LEM: $P \vee \neg P$, the *double negation elimination* DNE: $\neg\neg P \rightarrow P$, and *de Morgan's law* DML: $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$.

For each principle $\mathsf{L} \in \{\mathsf{LEM}, \mathsf{DNE}, \mathsf{DML}\}$, let $\Gamma\text{-}\mathsf{L}$ denote the principle L restricted to Γ-formulas P, Q. Since what deal with in this paper is a principle on the reals, we also deal with the hierarchy of the law of excluded middle that takes (expressions of) reals as parameters. When emphasizing this, the principle L is written as $\mathsf{L}_{\mathbb{R}}$. For example, $\Sigma_1^0\text{-}\mathsf{LEM}_{\mathbb{R}}$ is the scheme $\varphi(\bar{x}) \vee \neg\varphi(\bar{x})$ for Σ_1^0 formulas $\varphi(\bar{x})$ with real parameters $\bar{x}$.

More precisely, we examine the arithmetical hierarchy for these nonconstructive principles (cf. Akama et al. [1]) for formulas of second order arithmetic with function variables $\bar{v}$ (i.e, variables from $\mathbb{N}^{\mathbb{N}}$). For instance, we consider the class Σ_n^0 of formulas of the form

$$\exists a_1 \forall a_2 \ldots \mathsf{Q}a_n R(a_1, a_2, \ldots, a_n, \bar{v}),$$

where $Q = \exists$ if n is odd, $Q = \forall$ if n is even, each a_i ranges over $\mathbb{N}$, and R is a decidable formula. Note that every quantifier-free formula is decidable. These principles are also described as nonconstructive principles on the regular reals $\mathbb{R}$. For instance:

- $\Sigma_1^0\text{-DNE}_\mathbb{R}$ is known as the *Markov principle* MP, which is also equivalent to the following statement: For any regular Cauchy reals $x, y \in \mathbb{R}$, if $y \neq 0$ then you can divide x by y; that is, there is a regular Cauchy real $z \in \mathbb{R}$ such that $x = yz$.
- $\Sigma_1^0\text{-DML}_\mathbb{R}$ is equivalent to the *lessor limited principle of omniscience* LLPO, which states that for any regular Cauchy reals $x, y \in \mathbb{R}$, either $x \leq y$ or $y \leq x$ holds.
- $\Pi_1^0\text{-LEM}_\mathbb{R}$ is equivalent to the *limited principle of omniscience* LPO, which states that for any regular Cauchy reals $x, y \in \mathbb{R}$, either $x = y$ or $x \neq y$ holds.
- $\Pi_1^0\text{-LEM}_\mathbb{R} \leftrightarrow \Sigma_1^0\text{-LEM}_\mathbb{R}$ holds under Markov's principle.
- $\Sigma_2^0\text{-DML}_\mathbb{R}$ is equivalent (assuming Markov's principle, cf. Proposition 4.8) to the *pigeonhole principle* for infinite sets (or RT_2^1 in the context of reverse mathematics).

The arithmetical hierarchy of the law of excluded middle is first studied in the context of limit computable mathematics (one standpoint of constructive mathematics which allows *trial and errors* [16], which may be considered as a special case of a realizability interpretation of constructive mathematics w.r.t. a suitable partial combinatory algebra, see also Section 3.2.1). After that, the arithmetical hierarchy of law of excluded middle became extensively studied in various contexts; see e.g. [1, 4, 5, 13, 14, 35].

Example 2.1. The equivalence between $\Sigma_2^0\text{-DNE}_\mathbb{R}$ and Hilbert's basis theorem (Dickson's lemma), see [16, Section 5], can also be considered as an early result in constructive reverse mathematics.

In [5], it is shown that some variant of $\Sigma_3^0\text{-DML}_\mathbb{R}$ is equivalent to Ramsey's theorem for computable coloring of pairs, $\text{RT}_2^2(\Sigma_0^0)$, over Heyting arithmetic **HA**.

For the relationship among non-constructive principles, see also Figure 1, where IVT_{lin} and $\text{BE}_\mathbb{Q}$ are some auxiliary (less important) principles introduced in Section 4.1.2. We explain some nontrivial implications: For $\Sigma_2^0\text{-LEM}_\mathbb{R} \to \text{IVT}$, the standard proof of the intermediate value theorem employs the nested interval argument with the conditional branching on whether $f(x) = 0$ for some $x \in \mathbb{Q}$. This shows that $\Sigma_2^0\text{-LEM}_\mathbb{R}$ implies IVT, and hence BE. For $\Pi_1^0\text{-LEM}_\mathbb{R} \to \text{RDIV}$, one can decide if y is zero or not by using $\Pi_1^0\text{-LEM}_\mathbb{R}$. If $y = 0$ then put $z = 0$; otherwise define $z = x/y$. Thus, $\Pi_1^0\text{-LEM}_\mathbb{R}$ implies RDIV. For other nontrivial implications, see Section 4.1.1.

2.2. Function presentations of theorems.

2.2.1. *Theorems as multivalued functions.* A simple, but important, observation is that most principles which appear in Theorem 1.1 are of the $\forall\exists$-forms.

$$\varphi \equiv \forall x \in X \ [\eta(x) \to \exists y \in Y. \theta(x, y)]$$

Here, η and θ may also involve quantifiers. The basic idea is to regard this type of formula φ as a (multivalued) function F_φ that takes x as input and outputs y. In other words, a (possibly false) statement $\varphi \equiv \forall x \in X \ [\eta(x) \to \exists y \in Y. P(x, y)]$ is transformed
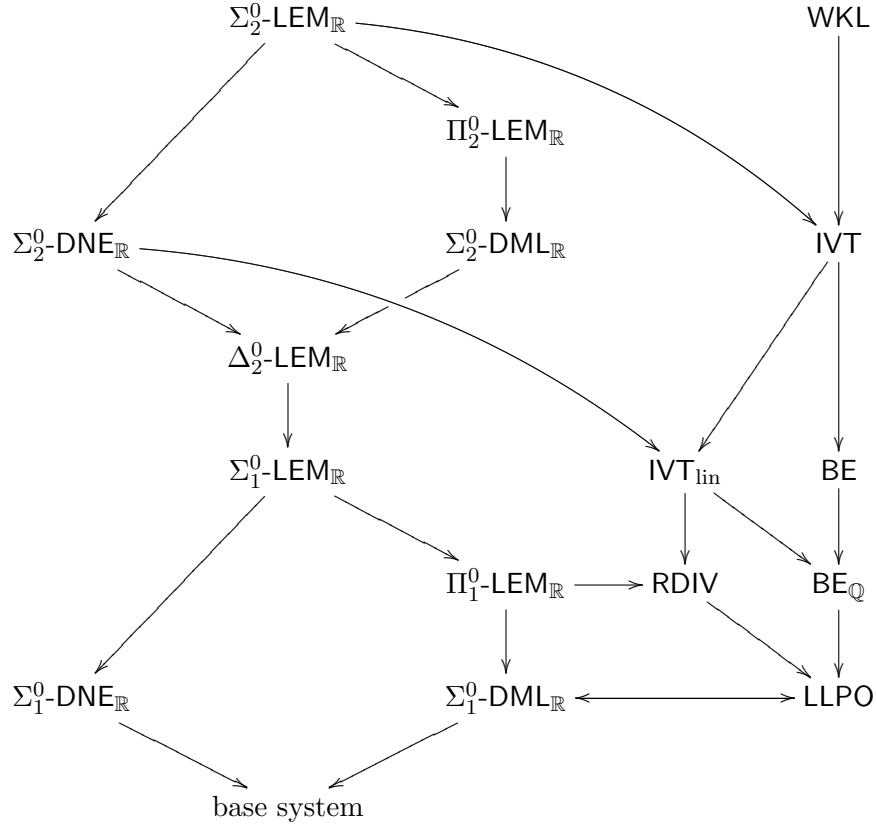
FIGURE 1. Nonconstructive principles over a base intuitionistic system

into the following F_φ:

$$\operatorname{dom}(F_\varphi) = \{x \in X : \eta(x)\}$$
$$F_\varphi(x) = \{y \in Y : \theta(x, y)\}$$

Think of F_φ as a multivalued function: Each $y \in F_\varphi(x)$ is regarded as an output of $F_\varphi(x)$. In this way, we view each $\forall\exists$-principle φ as a *partial multi-valued function* (*multifunction*). Here, we consider formulas as partial multifunctions rather than relations in order to distinguish a *hardest* instance $F_\varphi(x) = \emptyset$ (corresponding to a *false sentence*), and an *easiest* instance $x \in X \setminus \operatorname{dom}(F_\varphi)$ (corresponding to a *vacuous truth*). Only if $F_\varphi(x)$ is a singleton for any $x \in \operatorname{dom}(F_\varphi)$, we think of F_φ as a single-valued function. We use the symbol $F :\subseteq X \rightrightarrows Y$ to mean that F is a partial multifunction from X to Y.

The basic idea for measuring the complexity of a theorem φ is to analyze how complicated the function F_φ is. In a mathematical system based on constructive logic, the complexity of an $\forall\exists$-theorem can be predicted to be dominated by the complexity of constructing a existential witness. Therefore, it seems appropriate to measure the complexity of a *choice* function f of F_φ, i.e., a function f such that $f(x) \in F_\varphi(x)$ for all $x \in \operatorname{dom}(F_\varphi)$.

2.2.2. *Degrees of discontinuity.* If X, Y are topological spaces, we can discuss whether F_φ has a continuous choice function, or if not, how discontinuous a choice function is. All principles mentioned above, except for Markov's principle, are *discontinuous* principles; that is, there are no continuous choice functions for these principles. Thus, all of these principles fail in a well-known model of **IZF** (or a topos) in which all functions on ω^ω are continuous. Here, it is important to note that, for example, a statement that "for any regular Cauchy real ..." refers to the space of rational sequences (i.e., the Baire space, where $\mathbb{Q} \simeq \omega$ is endowed with the discrete topology) as the domain.

Example 2.2. The law of excluded middle $\Pi^0_2\text{-}\mathsf{LEM}_\mathbb{R}$ for Π^0_2 formulas (as a single-valued function) clearly corresponds to *Dirichlet's nowhere continuous function* $\chi_\mathbb{Q}$.

Example 2.3. For the binary expansion principle BE, any total binary expansion $\mathbb{R} \to 2^\omega$ is discontinuous, and indeed, there is no continuous function which, given a regular Cauchy representation of a real, returns its binary expansion.

Example 2.4. As already mentioned, some proof of the intermediate value theorem IVT can be given by determining whether a real is rational or not, i.e., using Dirichlet's function, and the rest can be done by continuous reasoning. However, the standard proof of IVT is more reminiscent of Thomae's function than Dirichlet's function.

$$\text{Thomae}(x) = \begin{cases} \frac{1}{q} & (x \text{ is of the form } \frac{p}{q} \text{ for coprime } p, q) \\ 0 & (x \text{ is irrational}) \end{cases}$$

Thomae's function has the property of being *discontinuous on the rationals* and *continuous on the irrationals*. The proof of IVT (i.e., for any continuous f, if $f(0)f(1) < 0$ then $f(x) = 0$ for some $x \in [0, 1]$) is given by the so-called interval-halving method, but if a solution x is a dyadic rational $m/2^n$, the process terminates in a finite number of steps discontinuously, and if x is a dyadic irrational, the process continues continuously forever. In other words, the standard proof of IVT has the same property as Thomae's function, which is *discontinuous on the dyadic rationals* and *continuous on the dyadic irrationals.*

Indeed, using this argument, one can formally prove that an upper bound on the discontinuity of IVT is given by Thomae's function (see [26]). Of course, the discontinuity of the standard proof is only as complicated as that of Thomae's function, so there may be other proofs with lower discontinuity, making this merely an upper bound.

There are differences among the *degrees of discontinuity* for these principles.

Remark. From the end of the 19th century to the beginning of the 20th century, various notions were introduced to measure the degree of discontinuity: A function is of Baire class 0 if it is continuous. A function is of Baire class $n + 1$ if it is the pointwise limit of functions of Baire class n. A function is Γ-measurable if the preimage of an open set is in Γ. A function is σ-continuous if some countable partition of the domain makes it continuous when restricted to each piece. By the Lebesgue-Hausdorff theorem, on a good topological space (such as the Baire space ω^ω), being of Baire n and being $\underset{\sim}{\Sigma}^0_{n+1}$-measurable are equivalent (see e.g. [21, Theorem 24.3]).

Example 2.5. Dirichlet's function is σ-continuous (i.e., decomposable into countably many continuous functions), and of Baire class 2 (i.e., a double pointwise limit of continuous functions), but not of Baire class 1. On the other hand, Thomae's function is of Baire class 1, so Thomae's function is a more precise upper bound for IVT.

Example 2.6. Weak König's lemma WKL has no σ-continuous choice, but has a choice of Baire class 1. Note also that WKL (even IVT) has no $\underset{\sim}{\boldsymbol{\Delta}}^0_2$-measurable choice, where there are functions of Baire class 1 (equivalently F_σ-measurable) which are not $\underset{\sim}{\boldsymbol{\Delta}}^0_2$-measurable. On the other hand, RDIV has a $\underset{\sim}{\boldsymbol{\Delta}}^0_2$-measurable choice.

These observations indicate that these principles are all different. However, one of the questions is, given a level $\mathbf{d}$ of discontinuity, how one can construct a model of constructive mathematics in which the functions on ω^ω are exactly those with discontinuity level at most $\mathbf{d}$.

2.2.3. *Theorems as realizability problems.* The discussion in Section 2.2.2 can be justified using Kleene's functional realizability interpretation. Realizability is the act of keeping track of witnesses (e.g., existential witnesses) for the correctness of a formula at all subformula levels (see e.g. [43]). For example, a realizer of an $\forall\exists$-theorem $\varphi \equiv \forall x \exists y \theta(x, y)$ contains information about a function which, given x, returns an existential witness y. In other words, a realizer always provides a choice function for F_φ.

There are several variants of realizability. Roughly speaking, Kleene's number realizability only validates the principles which have computably choices and refutes other non-computable principles, and Kleene's functional realizability validates the principles which have continuously choices and refutes other discontinuous principles. For the formal definition of realizability, see Section 3.2.2.

Then, can we formalize the discussion in Section 2.2.2 as a variant of realizability? For example, instead of realizability by continuous functions, one may consider realizability by Baire 1 functions or σ-continuous functions. At least for the former, it is clear that this does not yield a reasonable realizability since Baire 1 functions are not closed under composition. Furthermore, while the idea of considering the degrees of discontinuity is very promising, classical notions such as the Baire class are too coarse. A more useful tool is the notion of reducibility (relative computability/continuity), which has developed in computability theory.

What is dealt with in computability theory is the degree of difficulty of "problems." We consider an $\forall\exists$-theorem as a problem of finding an existential witness. In other words, consider a multifunction $F \colon \subseteq X \rightrightarrows Y$ as the problem of finding at least one solution $y \in F(x)$ for each instance $x \in \mathrm{dom}(F)$. Then one can consider the degree of difficulty of the problem F. This is given by a preorder relation $G \leq F$, which roughly means that F is more difficult than G or of equal difficulty. In other words, G is F-relatively solvable (if one knows a solution to F, one can also solve G). This preorder relation provides a much more delicate and precise measure than classical notions such as the Baire class.

Our goal is to define a notion of realizability relative to a given problem F. The process is as follows: In the following sections, we formalize the notion of F-relative computation, then introduce the notion of universal computation $F^{\circlearrowright}$ for F-relative

computability, which becomes a j-operator, from which we construct a site, and finally, the realizability over the site provides an interpretation of constructive mathematics corresponding to F-relative computability.

3. KEY TECHNICAL NOTIONS

3.1. **Weihrauch reducibility.** The main tool for measuring the degrees of difficulty of partial multifunctions in this article is Weihrauch reducibility (see e.g. [8]). Below we assume that $\mathbf{X}$ is either ω or ω^ω.

3.1.1. *Weihrauch reducibility.* Let $F, G \colon \mathbf{X} \rightrightarrows \mathbf{X}$ be partial multifunctions. We say that *F is Weihrauch reducible to G* (written $F \leq_{\mathsf{W}} G$) if there are partial computable functions Φ and Ψ such that, for any $x \in \mathrm{dom}(F)$, $y \in G(\Phi(x))$ implies $\Psi(x, y) \in F(x)$. For basics on Weihrauch reducibility, we refer the reader to [8]. The definition of Weihrauch reducibility $F \leq_{\mathsf{W}} G$ can be viewed as the following perfect information two-player game:

$$
\begin{array}{lllll}
\text{I:} & x_0 \in \mathrm{dom}(F) & & x_1 \in G(y_0) & \\
\text{II:} & & y_0 \in \mathrm{dom}(G) & & y_1 \in F(x_0)
\end{array}
$$

More precisely, each player chooses an element from $\mathbf{X}$ at each round, and Player II wins if there is a computable strategy τ for II which yields a play described above. Note that y_0 depends on x_0, and y_1 depends on x_0 and x_1, and moreover, a computable strategy τ for II yields partial computable maps $\Phi \colon x_0 \mapsto y_0$ and $\Psi \colon (x_0, x_1) \mapsto y_1$. Usually, Φ is called an *inner reduction* and Ψ is called an *outer reduction.* If reductions Φ and Ψ are allowed to be continuous (when $\mathbf{X} = \omega^\omega$), then we say that *$F$ is continuously Weihrauch reducible to G* (written $F \leq_{\mathsf{W}}^c G$).

In (classical or intuitionistic) logic, when showing $P \to Q$, the contraction rule allows us to use the antecedent P more than once. However, the above reducibility notion essentially requires us to show Q by using P only once. To overcome this difficulty, there is an operation $H \star G$ on partial multifunctions G, H which allows us to use H after the use of G. For partial multi-valued functions G and H, define the composition $H \circ G$ as follows:

$$
H \circ G(x) = \begin{cases} \bigcup \{ H(y) : y \in G(x) \} & \text{if } G(x) \downarrow \subseteq \mathrm{dom}(H), \\ \uparrow & \text{otherwise.} \end{cases}
$$

Note that the above definition of the composition of partial multifunctions is different from the usual definition of the composition of relations (so the partial multifunctions are completely different from the relations). Then it is shown that $\min_{\leq_{\mathsf{W}}} \{ H_0 \circ G_0 : G_0 \leq_{\mathsf{W}} G \wedge H_0 \leq_{\mathsf{W}} H \}$ always exists, and a representative is denoted by $H \star G$; see Brattka-Pauly [9]. There is an explicit description of a representative $H \star G$, and it is indeed needed to prove results on $\star$. To explain the explicit definition, we consider the following play for "$F \leq_{\mathsf{W}} H \star G$":

$$
\begin{array}{lllllll}
\text{I:} & x_0 \in \mathrm{dom}(F) & & x_1 \in G(y_0) & & x_2 \in H(y_1) & \\
\text{II:} & & y_0 \in \mathrm{dom}(G) & & y_1 \in \mathrm{dom}(H) & & y_2 \in F(x_0)
\end{array}
$$

Again, a computable strategy for Player II yields functions $y_0 = \Phi_0(x_0)$, $y_1 = \Phi_1(x_0, x_1)$, and $y_2 = \Psi(x_0, x_1, x_2)$. That is, II's strategy yields two inner reductions

Φ_0, Φ_1 and an outer reduction Ψ, where Φ_0 makes a query to G, and then, after seeing a result $x_1 \in G(y_0)$, the second reduction Φ_1 makes a query to H.

Now, the partial multifunction $H \star G$ takes, as an input, Player I's first move x_0 and a code for Player II's strategy τ (a code for $\langle \Phi_0, \Phi_1, \Psi \rangle$) such that II's play according to the strategy τ obeys the rule unless I's play violates the rule at some previous round, where we say that Player I obeys the rule if $x_1 \in G(y_0)$ and $x_2 \in H(y_1)$, and Player II obeys the rule if $y_0 \in \mathrm{dom}(G)$ and $y_1 \in \mathrm{dom}(H)$. Then, $H \star G(x_0, \tau)$ returns $\Psi(x_0, x_1, x_2)$ for I's some play (x_1, x_2) which obeys the rule. Note that there are many possible values for $\Psi(x_0, x_1, x_2)$, which means that $H \star G$ is multi-valued. Without loss of generality, one can remove x_0 from the definition of $H \star G$ by considering a continuous strategy τ.

This operation $(G, H) \mapsto H \star G$ is called the *compositional product*; see also Brattka-Pauly [9]. For game-theoretic descriptions, see also Hirschfeldt-Jockusch [17] and Goh [15]. This idea extends to reduction games introduced below.

3.1.2. *$\star$-closure and reduction games.* In order to express "arbitrary use of an antecedent", Hirschfeldt-Jockusch [17] introduced *reduction games* and *generalized Weihrauch reducibility*. For partial multifunctions $F, G \colon \subseteq \mathbf{X} \rightrightarrows \mathbf{X}$, let us consider the following perfect information two-player game $\mathfrak{G}(F, G)$:

$$
\begin{array}{llll}
\text{I:} & x_0 & x_1 & x_2 & \cdots \\
\text{II:} & & y_0 & y_1 & y_2 & \cdots
\end{array}
$$

Each player chooses an element from $\mathbf{X}$ at each round. Here, Players I and II need to obey the following rules.

- First, Player I chooses $x_0 \in \mathrm{dom}(F)$.
- At the nth round, Player II reacts with $y_n = \langle a, u_n \rangle$.
  - The choice $a = 0$ indicates that Player II makes a new query u_n to G. In this case, we require $u_n \in \mathrm{dom}(G)$.
  - The choice $a = 1$ indicates that Player II declares victory with u_n.
- At the $(n+1)$th round, Player I responds to the query made by Player II at the previous stage. This means that $x_{n+1} \in G(u_n)$.

Then, *Player II wins the game* $\mathfrak{G}(F, G)$ if either Player I violates the rule before Player II violates the rule or Player II obeys the rule and declares victory with $u_n \in F(x_0)$.

Hereafter, we require that Player II's moves are chosen in a continuous manner (if $\mathbf{X} = \omega^\omega$). In other words, Player II's strategy is a code τ of a partial *continuous* function $h_\tau \colon \subseteq \mathbf{X}^{<\omega} \to \mathbf{X}$. On the other hand, Player I's strategy is any partial function $\sigma \colon \subseteq \mathbf{X}^{<\omega} \to \mathbf{X}$ (which is not necessarily continuous). Given such strategies σ and τ yield a play $\sigma \otimes \tau$ in the following manner:

$$
\begin{aligned}
(\sigma \otimes \tau)(0) &= \sigma(\langle \rangle), \\
(\sigma \otimes \tau)(2n+1) &= h_\tau(\langle (\sigma \otimes \tau)(2m) \rangle_{m \le n}), \\
(\sigma \otimes \tau)(2n+2) &= \sigma(\langle (\sigma \otimes \tau)(2m+1) \rangle_{m \le n}).
\end{aligned}
$$

Player II's strategy τ is *winning* if Player II wins along $\sigma \otimes \tau$ whatever Player I's strategy σ is. We write $F \le_{\supset W} G$ if Player II has a computable winning strategy for $\mathfrak{G}(F, G)$. We write $F \le_{\supset W}^c G$ if Player II has a (continuous) winning strategy for $\mathfrak{G}(F, G)$. Hirschfeldt-Jockusch [17] showed that the relation $\le_{\supset W}$ is transitive. In a traditional context, the notion $\le_{\supset W}$ is called generalized Weihrauch reducibility.

Remark. This notion $\leq_{\supset W}$ should be considered to be the most fundamental reducibility notion, which performs a relative computation using a finite number of queries to an oracle, and perhaps a simpler name would be preferable.

If f, g are total functions on ω, then one can see that $f \leq_{\supset W} g$ if and only if f is Turing reducible to g. For this reason, $\leq_{\supset W}$ is called Turing reducibility in [24].

Note that the rule of the above game does not mention F except for Player I's first move. Hence, if we skip Player I's first move, we can judge if a given play follows the rule without specifying F. For a partial multifunction $G \colon \subseteq \mathbf{X} \rightrightarrows \mathbf{X}$, we define $G^{\supset} \colon \subseteq \mathbf{X} \rightrightarrows \mathbf{X}$ as follows:

- $(x_0, \tau) \in \mathrm{dom}(G^{\supset})$ if and only if τ is Player II's strategy, and for Player I's any strategy σ with first move x_0, if Player II declares victory at some round along the play $\sigma \otimes \tau$.
- Then, $u \in G^{\supset}(x_0, \tau)$ if and only if Player II declares victory with u at some round along the play $\sigma \otimes \tau$ for some σ with first move x_0.

Here, the statement "Player II declares victory" does not necessarily mean "Player II wins". Indeed, the above definition is made before F is specified, so the statement "Player II wins" does not make any sense. Again, one can remove x_0 from an input for $G^{\supset}$ by considering a continuous strategy. The following is obvious by definition.

Observation 3.1 (see [37]). $F \leq_W G^{\supset} \iff F \leq_{\supset W} G$.

Remark. We may think of $G^{\supset}$ as a *universal machine* for G-relative computation. The relative computation process is displayed as a game, a program is Player II's strategy, and an input is Player I's first move. This means that $G^{\supset}(x_0, \sigma)$ is the result of the computation of a program σ with an oracle G and an input x_0. Using the notation of traditional computability theory, it would be appropriate to write $\Phi_\sigma^G(x_0) := G^{\supset}(x_0, \sigma)$.

Note that Neumann-Pauly [37] used generalized register machines to define the $\star$-closure $G^{\diamondsuit}$ of G. This notion is also studied by Pauly-Tsuiki [39]. Neumann-Pauly [37] showed that $F \leq_W G^{\diamondsuit}$ if and only if $F \leq_{\supset W} G$. Therefore, we have $G^{\supset} \equiv_W G^{\diamondsuit}$, but the game-theoretic definition is mathematically simpler and clearer than $\diamondsuit$ (and the symbol $\diamondsuit$ may be used in many contexts while $\supset$ is the standard symbol for the game quantifier), so we use $G^{\supset}$ throughout this paper.

By transitivity of $\leq_{\supset W}$, we have $F^{\supset} \star F^{\supset} \equiv_W F^{\supset}$. Moreover, Westrick [50] showed that $\diamondsuit$ (so $\supset$) gives the *least fixed point* of the operation $F \mapsto F \star F$ whenever $F \geq_W \mathbf{1}$, where $\mathbf{1}$ is the Weihrauch degree of the identity function on $\mathbf{X}$. More precisely, if $\mathbf{1} \leq_W F$ then $F \leq_W F \star F$ implies $F^{\supset} \leq_W F$. For more details on the closure operator $\diamondsuit$, see Westrick [50].

3.1.3. *Recursion trick.* To prove various separation results on Weihrauch reducibility, *Kleene's recursion theorem* has been frequently used as a simple, but very strong, proof machinery, cf. [17, 25, 26]. We employ this machinery throughout this article, so we separate this argument, and call it *recursion trick*. In Kihara-Pauly [25], recursion trick is described in the context of two player games, where Player I is called Pro (the proponent) and II is called Opp (the opponent) because our purpose is to show Weihrauch separation $\not\leq_W$. The formal description of recursion trick for short reduction games is described in Kihara-Pauly [26].

We first describe the idea of recursion trick in a slightly longer reduction game, i.e., for $F \leq_{\mathsf{W}} G \star H$. Note that, in this kind of reduction, any play of the reduction game ends at the third round of Player II:

$$
\begin{array}{cccc}
\text{I:} & x_0 & x_1 & x_2 \\
\text{II:} & y_0 & y_1 & y_2
\end{array}
$$

If Player II's strategy is given, we may consider that the value of y_i depends on Player I's play; that is, $y_0 = y_0(x_0)$, $y_1 = y_1(x_0, x_1)$, and $y_2 = y_2(x_0, x_1, x_2)$. If we describe an algorithm constructing x_0' from (y_0, y_1, y_2) (with a parameter x_0), by Kleene's recursion theorem, the parameter x_0 can be interpreted as *self-reference*; that is, we may assume that $x_0' = x_0$. Hence, one can remove x_0 from the list of parameters in y_0, y_1 and y_2. In summary, to show that $F \nleq_{\mathsf{W}} G \star H$, given y_0, y_1, y_2, it suffices to construct x_0 such that Player I wins along the following play

$$
\begin{array}{cccc}
\text{I:} & x_0 & x_1 & x_2 \\
\text{II:} & y_0 & y_1(x_1) & y_2(x_1, x_2)
\end{array}
$$

for some x_1 and x_2, that is, Player II violates the rule before I violates the rule, or Player I obeys the rule and $y_2 \notin F(x_0)$. The essence of recursion trick is that *it allows us to construct Player I's first move x_0 later than II's moves y_0, y_1, y_2.* However, note that I's move x_0 has to obey the rule even if II violates the rule.

In general reduction games, II's strategy yields $y_n(x_0, x_1, \dots, x_n)$. Again we describe an algorithm $(y_n)_{n \in \omega} \mapsto x_0'$, and by Kleene's recursion theorem, consider x_0 as self-reference. As before, this allows us to construct Player I's first move x_0 later than II's moves $(y_n)_{n \in \omega}$, but I's move x_0 has to obey the rule even if II violates the rule.

Formal description of recursion trick: For $\mathbf{X} = \omega^\omega$, we rigorously describe what recursion trick is. Fix an oracle Z. Given Player II's Z-computable strategy τ, we get a uniform sequence of Z-computable functions $y_n^\tau(x_0, x_1, \dots, x_n)$ which determine the nth move of Player II. Now, assume that Player I's first move is the eth partial Z-computable function φ_e^Z on ω. Then, the reduction game proceeds as follows:

$$
\begin{array}{ccccc}
\text{I:} & \varphi_e^Z & x_1 & x_2 & \dots \\
\text{II:} & y_0^\tau(\varphi_e^Z) & y_1^\tau(\varphi_e^Z, x_1) & y_2^\tau(\varphi_e^Z, x_1, x_2) & \dots
\end{array}
$$

Here, each move in the above play may be a *partial* function on ω, while each move in the original reduction game is a *total* function on ω.

Assume that we have constructed a Z-computable algorithm $e \mapsto x_0(e)$ constructing Player I's first move $x_0(e)$ using information of II's play $\langle \tau(\varphi_e^Z, x_1, \dots, x_n) \rangle_{n \in \omega}$. Clearly, there is a Z-computable function u such that $x_0(e) \simeq \varphi_{u(e)}^Z$. Moreover, assume that

($\star$) The move $x_0(e)$ is designed to defeat Player II's strategy $\langle \tau(\varphi_e^Z, x_1, \dots, x_n) \rangle_{n \in \omega}$; that is, for any e, there exist subsequent moves $(x_1, x_2, \dots)$ such that Player I wins along the following play:

$$
\begin{array}{ccccc}
\text{I:} & x_0(e) \simeq \varphi_{u(e)}^Z & x_1 & x_2 & \dots \\
\text{II:} & y_0^\tau(\varphi_e^Z) & y_1^\tau(\varphi_e^Z, x_1) & y_2^\tau(\varphi_e^Z, x_1, x_2) & \dots
\end{array}
$$

Here, we do not need to find such $(x_1, x_2, \dots)$ effectively. By Kleene's recursion theorem, there exists r such that $x_0(r) \simeq \varphi_{u(r)}^Z \simeq \varphi_r^Z$. Note that, in the above play with $e = r$, Player II just follows the strategy τ. By the assumption ($\star$), I's play

$(x_0(r), x_1, x_2, \dots)$ defeats Player II who follows τ, which means that τ is not a winning strategy.

In summary, if for any computable strategy τ for Player II (in the reduction game $\mathfrak{G}(F, G)$) one can construct a computable algorithm $e \mapsto x_0(e)$ satisfying $(\star)$, then we obtain $F \not\leq_{\mathsf{W}} G^{\diamondsuit}$. Moreover, if this is possible relative to any oracle Z, then we also obtain the continuous Weihrauch separation $F \not\leq_{\mathsf{W}}^c G^{\diamondsuit}$. We often describe such a construction $e \mapsto x_0(e)$ as an algorithm describing $x_0(\tilde{\tau})$ from given $\langle \tilde{\tau}(x_1, \dots, x_n) \rangle$, where $\tilde{\tau}(x_1, \dots, x_n) \simeq \tau(\varphi_e^Z, x_1, \dots, x_n)$ for some e.

3.2. **Realizability.** In this section, we construct a realizability predicate from a given Weihrauch degree, and moreover, if a given Weihrauch degree has a good closure property, then we show that the induced realizability predicate validates all axioms of intuitionistic Zermelo-Fraenkel set theory **IZF**.

3.2.1. *Partial combinatory algebra.* In order to give a computability-theoretic model of constructive mathematics, it is convenient to abstract the property of computability as a kind of algebra.

Definition 3.2 (see also [49]). A partial magma is a pair $(M, *)$ of a set M and a partial binary operation $*$ on M. We often write xy instead of $x * y$, and as usual, we consider $*$ as a left-associative operation, that is, xyz stands for $(xy)z$. A partial magma is *combinatory complete* if, for any term $t(x_1, x_2, \dots, x_n)$, there is $a_t \in M$ such that $a_t x_1 x_2 \dots x_{n-1} \downarrow$ and $a_t x_1 x_2 \dots x_n \simeq t(x_1, x_2, \dots, x_n)$. For terms $t(x, y) = x$, and $u(x, y, z) = xz(yz)$, the corresponding elements $a_t, a_u \in M$ are usually written as k and s. A combinatory complete partial magma is called a *partial combinatory algebra* (abbreviated as *pca*).

Example 3.3 (The algebra of computable functions). Kleene's first algebra $\mathsf{K}_1 = (\omega, *)$ is defined by $e * n = \varphi_e(n)$, where φ_e is the eth partial computable function. Then K_1 forms a partial combinatory algebra. The combinatory completeness follows from the so-called smn-theorem.

Example 3.4 (The algebra of continuous functions). Kleene's second algebra $\mathsf{K}_2 = (\omega^\omega, *)$ is defined by $x * y = \varphi_e^z(y)$, where $x = e^\frown z$ and φ_e is the eth partial computable function relative to $z \in \omega^\omega$. Then K_2 forms a partial combinatory algebra.

However, this notion is not sufficient for our purposes. The most basic setting in modern computability theory is not "computable objects + computable morphisms," nor "topological objects + topological morphisms," but "topological objects + computable morphism [2]."

Definition 3.5 (see also [49, Sections 2.6.9 and 4.5]). A *relative pca* is a triple $\mathbb{P} = (\mathsf{P}, \underset{\sim}{\mathsf{P}}, *)$ such that $\mathsf{P} \subseteq \underset{\sim}{\mathsf{P}}$, both $(\underset{\sim}{\mathsf{P}}, *)$ and $(\mathsf{P}, * \restriction \mathsf{P})$ are pcas, and share combinators s and k.

In this article, the boldface algebra $\underset{\sim}{\mathsf{P}}$ is always the set ω^ω of all infinite sequences.

Example 3.6 (The algebra of type-two computable functions). Kleene-Vesley's algebra is $\mathsf{K}_2^{\mathsf{eff}} = ((\omega^\omega)^{\mathsf{eff}}, \omega^\omega, *)$, where $*$ is the same as K_2 in Example 3.4, and $(\omega^\omega)^{\mathsf{eff}}$ is the set of all computable sequences. Then $\mathsf{K}_2^{\mathsf{eff}}$ forms a relative pca.

Note that the functions of the forms $x \mapsto e * x$ for some $e \in (\omega^\omega)^{\mathsf{eff}}$ are exactly partial computable functions on ω^ω.

Example 3.7. In descriptive set theory, the idea of a relative pca is ubiquitous, which usually occurs as a pair of *lightface* and *boldface* pointclasses. A large number of nontrivial, deep, examples of relative pcas have been (implicitly) studied in descriptive set theory [33]: By the *good parametrization lemma* in descriptive set theory (see Moschovakis [33, Lemma 3H.1]), any Σ^*-pointclass Γ — in particular, any Spector pointclass (see Moschovakis [33, Lemma 4C]), so many infinitary computation models including infinite time Turing machines (ITTMs) — yields a (relative) pca, since the good parametrization lemma is, roughly speaking, a generalized smn-theorem, so such a pointclass admits currying. Here, the *partial Γ-computable function application* form a lightface pca, and the *partial $\underset{\sim}{\Gamma}$-measurable function application* form a boldface pca. For the details, see [22].

Example 3.8. Let us give some computability/descriptive-set-theoretic examples of relative pcas.

(1) If $\Gamma = \Sigma^0_1$, the induced lightface pca is equivalent to *Kleene's first algebra* (associated with Kleene's number realizability), and the boldface pca is *Kleene's second algebra* (associated with Kleene's functional realizability). The relative pca induced from $\Gamma = \Sigma^0_1$ is Kleene-Vesley's algebra.

(2) The pointclass $\Sigma^0_1(\emptyset^{(n)})$ is employed by Akama et al. [1] to show that the arithmetical hierarchy of the law of excluded middle does not collapse over Heyting arithmetic **HA**. Here, note that Σ^0_n does not yield a pca on ω^ω since Σ^0_n-computable functions (on ω^ω) and $\underset{\sim}{\Sigma}^0_n$-measurable functions are not closed under composition. In descriptive set theory, Borel pointclass whose associated measurable functions are closed under composition are studied under the name of Borel amenability, cf. [34]; for instance, the nth level Borel functions (a.k.a. $\Sigma_{n,n}$-functions) are closed under composition, and the induced reducibility notion is studied, e.g. by [27].

(3) The pointclass Π^1_1 is the best-known example of a Spector pointclass, and the induced lightface pca obviously yields hyperarithmetical realizability. For the boldface pca, the associated total realizable functions are exactly the Borel measurable functions. If one considers Kleene realizability with such a pca (i.e., one obtained by a Spector pointclass), then it trivially validates $\Sigma^0_n\text{-}\mathsf{LEM}_\mathbb{R}$ for any $n \in \omega$.

(4) Bauer [3] also studied the pca (and the induced realizability topos) obtained from infinite time Turing machines (ITTMs). Here, recall that ITTMs form a Spector pointclass.

One may also consider descriptive set theoretic versions of Scott's graph model using lightface/boldface pointclasses.

3.2.2. *Realizability interpretation.* One of the most important interpretations of constructive mathematics is the realizability interpretation; see e.g. [43]. The most basic Kleene's realizability interpretation is an interpretation of constructive mathematics using partial computable functions on ω. Or, it can be said to be an interpretation using

K_1 (Definition 3.3). This idea is generalized to an interpretation using arbitrary relative pca.

Definition 3.9. Let $\mathbb{P} = (\mathsf{P}, \underset{\sim}{\mathbf{P}}, *)$ be a relative pca. Given $a \in \underset{\sim}{\mathbf{P}}$ and a formula φ, the realizability relation $a \Vdash_{\mathbb{P}} \varphi$ is inductively defined as follows:

$$a \Vdash_{\mathbb{P}} A \iff A \text{ is true} \quad \text{(for atomic } A).$$
$$(a, b) \Vdash_{\mathbb{P}} \varphi \wedge \psi \iff a \Vdash_{\mathbb{P}} \varphi \text{ and } b \Vdash_{\mathbb{P}} \psi.$$
$$(i, a) \Vdash_{\mathbb{P}} \varphi \vee \psi \iff \text{if } i = 0 \text{ then } a \Vdash_{\mathbb{P}} \varphi \text{ else } a \Vdash_{\mathbb{P}} \psi.$$
$$e \Vdash_{\mathbb{P}} \varphi \to \psi \iff \forall a \in \underset{\sim}{\mathbf{P}} \, [a \Vdash_{\mathbb{P}} \varphi \longrightarrow ea \Vdash_{\mathbb{P}} \psi].$$
$$(\mathtt{x}, a) \Vdash_{\mathbb{P}} \exists x \in X.\varphi(x) \iff a \Vdash_{\mathbb{P}} \varphi(x), \text{ where } \mathtt{x} \text{ is a code of } x \in X.$$
$$e \Vdash_{\mathbb{P}} \forall x \in X.\varphi(x) \iff \forall \mathtt{x} \in \underset{\sim}{\mathbf{P}} \, [\mathtt{x} \text{ is a code of } x \in X \longrightarrow e\mathtt{x} \Vdash_{\mathbb{P}} \varphi(x)].$$

Then φ is $\mathbb{P}$-realizable if $a \Vdash_{\mathbb{P}} \varphi$ for some $a \in \mathsf{P}$. Here, we assume that each type X is interpreted as a coded set (an assembly, a represented space, etc.)

Example 3.10. This notion can clearly be relativized to an oracle. For instance, if an oracle $\alpha \colon \omega \to \omega$ is given, instead of the pca $\mathsf{K}_1 = (\omega, *)$ of computable functions, one can consider the pca $(\omega, *_\alpha)$ of α-computable functions, where $*_\alpha$ is the α-computable function application. In general, any relative pca $\mathbb{P}$ can be relativized to a partial oracle $\alpha \colon \subseteq \underset{\sim}{\mathbf{P}} \to \underset{\sim}{\mathbf{P}}$; see [48].

However, most of the $\forall\exists$-theorems in mathematics are not single-valued (that is, a witness for the existence is not unique). Therefore, what we really want to consider is the realizability interpretation relative to a partial *multi-valued* oracle $\alpha \colon \subseteq \underset{\sim}{\mathbf{P}} \rightrightarrows \underset{\sim}{\mathbf{P}}$.

This goes beyond the framework of relative pca: It is not possible to combine the pair of a relative pca and a multi-valued oracle into a single relative pca, but that is fine. Rather than collapsing a computation system and an oracle, it is sometimes better to consider them separately for clarity.

We declare that the pair $(\mathbb{P}, F)$ of a relative pca $\mathbb{P}$ and a partial multi-valued function $F \colon \subseteq \underset{\sim}{\mathbf{P}} \rightrightarrows \underset{\sim}{\mathbf{P}}$ is the fundamental notion. When the oracle part F satisfies certain conditions, we call $(\mathbb{P}, F)$ a *site* (see also Definition 3.16).

We use the notation for relative computation as close to that of ordinary computability theory, as: $\Phi_e^F(x_0) = F^{\circlearrowleft}(x_0, e)$. The realizability interpretation is relativized to a multi-valued oracle F as follows.

Definition 3.11. Let $\mathbb{P}$ be a relative pca, and $F \colon \subseteq \underset{\sim}{\mathbf{P}} \rightrightarrows \underset{\sim}{\mathbf{P}}$ be a partial multifunction. The realizability relation $a \Vdash_F \varphi$ is inductively defined as follows:

$$e \Vdash_F \varphi \to \psi \iff \forall a \, [a \Vdash_F \varphi \longrightarrow \Phi_e^F(a) \downarrow \text{ and } (\forall b \in \Phi_e^F(a)) \, b \Vdash_F \psi].$$
$$e \Vdash_F \forall x \in X.\varphi(x) \iff \forall \mathtt{x} \, [\mathtt{x} \text{ is a code of } x \in X$$
$$\longrightarrow \Phi_e^F(\mathtt{x}) \downarrow \text{ and } \forall b \in \Phi_e^F(\mathtt{x}). \, b \Vdash_F \varphi(\mathtt{x})].$$

For other logical connectives, the definitions are the same as in Definition 3.9.

Observation 3.12. $F \equiv_{\supseteq\mathsf{W}} G$ *then F-realizability and G-realizability are equivalent.*

Let us connect the relativization to a multivalued oracle with a known feasibility interpretation. If we define $\mathbf{j} = F^{\circlearrowright}$, Definition 3.11 can be rephrased as follows:

$$e \Vdash_{\mathbf{j}} \varphi \to \psi \iff \forall a \, [a \Vdash_{\mathbf{j}} \varphi \longrightarrow \mathbf{j}(e, a) \downarrow \text{ and } (\forall b \in \mathbf{j}(e, a)) \, b \Vdash_{\mathbf{j}} \psi].$$

$$e \Vdash_{\mathbf{j}} \forall x \in X . \varphi(x) \iff \forall \mathtt{x} \, [\mathtt{x} \text{ is a code of } x \in X$$
$$\longrightarrow \mathbf{j}(e, \mathtt{x}) \downarrow \text{ and } \forall b \in \mathbf{j}(e, \mathtt{x}) . \, b \Vdash_{\mathbf{j}} \varphi(\mathtt{x})].$$

We call this the *realizability interpretation over the site* $(\mathbb{P}, \mathbf{j})$. Some readers may recognize a similar realizability notion. That is Lifschitz' realizability.

Example 3.13 (Lifschitz realizability)**.** The key idea in Lifschitz [30] (and van Oosten [47]) of validating $\mathsf{CT}_0! + \neg\mathsf{CT}_0$ is the use of *multifunction applications*, rather than single-valued applications. From the computability-theoretic perspective, their key idea is regarding Π^0_1 *sets* (or Π^0_1 *classes*) as basic concepts rather than computable functions. More precisely, over the Kleene first algebra $(\omega, *)$, Lifschitz considered the partial multifunction $\mathbf{j}_{\mathrm{L}} :\subseteq \omega \rightrightarrows \omega$ defined by

$$\mathbf{j}_{\mathrm{L}}(\langle e, b \rangle) = \{n \in \omega : n < b \, \wedge \, e * n \uparrow\},$$

where $\mathbf{j}_{\mathrm{L}}(\langle e, b \rangle) \downarrow$ if and only if the set is nonempty. Obviously, Lifschitz's multifunction $\mathbf{j}_{\mathrm{L}}$ gives an effective enumeration of all bounded Π^0_1 subsets of ω. Van Oosten [47] extended this notion to the Kleene second algebra $(\omega^\omega, *)$ by considering the following partial multifunction $\mathbf{j}_{\mathrm{vO}} :\subseteq \omega^\omega \rightrightarrows \omega^\omega$:

$$\mathbf{j}_{\mathrm{vO}}(\langle g, h \rangle) = \{x \in \omega^\omega : (\forall n \in \omega) \, x(n) < h(n) \, \wedge \, g * x \uparrow\},$$

It is again obvious that van Oosten's multifunction $\mathbf{j}_{\mathrm{vO}}$ gives a representation of all compact subsets of ω^ω, and $\mathbf{j}_{\mathrm{vO}}(\langle g, h \rangle)$ is an effectively compact Π^0_1 class relative to $\langle g, h \rangle$.

Compare their realizability interpretations [30, 47] with ours. One can see that Lifschitz' realizability [30] is equivalent to the realizability interpretation over the site $(\mathsf{K}_1, \mathbf{j}_{\mathrm{L}})$, and van Oosten's realizability [47] is equivalent to the realizability interpretation over the site $(\mathsf{K}_2^{\mathsf{eff}}, \mathbf{j}_{\mathrm{vO}})$.

Remark. Lifschitz' realizability and van Oosten's realizability can also be given as the realizability interpretations relative to the weakest oracles that solves LLPO and WKL, respectively. As a logical principle, LLPO is de Morgan's law for Σ_1 formulas:

$$\neg(\varphi \wedge \psi) \longrightarrow \neg\varphi \vee \neg\psi \qquad (\varphi, \psi \colon \Sigma_1 \text{ formulas})$$

As a problem, LLPO is a partial multifunction on ω such that

(1) $e \in \mathrm{dom}(\mathsf{LLPO})$ iff either $\varphi_e(0) \uparrow$ or $\varphi_e(1) \uparrow$.
(2) $i \in \mathsf{LLPO}(e)$ iff $i < 2$ and $\varphi_e(i) \uparrow$.

For computability-theorists, it is obvious that $\mathsf{LLPO}^{\circlearrowright} \equiv_{\mathsf{W}} \mathbf{j}_{\mathrm{L}}$ (see e.g. [39, 52]). Thus, by Observation 3.12, Lifschitz realizability is equivalent to LLPO-realizability; that is, realizability over the site $(\mathsf{K}_1, \mathsf{LLPO}^{\circlearrowright})$.

As a mathematical principle, weak König's lemma WKL states that every infinite binary tree has an infinite path. As a problem, WKL is a partial multifunction on ω^ω such that

(1) $T \in \mathrm{dom}(\mathsf{WKL})$ iff $T \subseteq 2^{<\omega}$ and T is an infinite tree.
(2) $p \in \mathsf{WKL}(T)$ iff p is an infinite path through T; that is, $p \upharpoonright n \in T$ for any $n \in \omega$.

Again, it is easy to see that $\mathsf{WKL}^{\supset} \equiv_{\mathsf{W}} \mathsf{WKL} \equiv_{\mathsf{W}} \mathbf{j}_{\mathrm{vO}}$ (see e.g. [17, Proposition 4.9]). Thus, by Observation 3.12, van Oosten's version of Lifschitz realizability is equivalent to WKL-realizability; that is, realizability over the site $(\mathsf{K}_2^{\mathsf{eff}}, \mathsf{WKL}^{\supset})$.

Remark. After finding Definition 3.11, one might mistakenly assume that it is a straightforward generalization of Lifschitz's realizability [30], but it is not so simple at all. It is essential to recognize Lifschitz's realizability as a realizability interpretation relative to a multi-valued oracle with an appropriate reducibility notion for multifunctions. However, the discovery of the correct reducibility notion (that is, the correct definition of Φ_e^F for a partial multifunction F) had to wait until Hirschfeldt-Jockusch [17] and Neumann-Pauly [37]. This was not accessible in Lifschitz's time (until recently, in fact).

Remark. Yoshimura [52] came closest to Definition 3.11, but he did not have the definition of relative computation Φ_e^F. He had arrived at the definition of Lifschitz-like realizability interpretation $\Vdash_{\mathbf{j}}$ over $(\mathbb{P}, \mathbf{j})$ (although he did not use the term "site") independently of us.

We must be careful to note that there is a big difference between oracle realizability $\Vdash_F$ and Lifschitz-like realizability $\Vdash_{\mathbf{j}}$ (if $\mathbf{j}$ is not of the form $F^{\supset}$). The former uses the relative computation allowing an arbitrary finite number of accesses to the oracle F, while the latter only accesses the oracle $\mathbf{j}$ once. For this reason, Yoshimura [52] formulated an interpretation of finite-type *modal* Heyting arithmetic, where the modal operator $\Diamond$ is interpreted by a single query to a (multi-valued) oracle.

3.2.3. *Set theory.* McCarty [32] introduced a realizability interpretation of the intuitionistic Zermero-Fraenkel set theory **IZF** which validates Church's thesis CT_0. Chen-Rathjen [10] combined McCarty-realizability with Lifschitz-realizability [30] to give a realizability interpretation of **IZF** which validates $\mathsf{CT}_0! + \neg\mathsf{CT}_0 + \mathsf{LLPO} + \neg\mathsf{LPO} + \neg\mathsf{AC}_{\omega,2}$.

To define a set-theoretic realizability interpretation, we follow the argument in Chen-Rathjen [10]: First, we consider a set-theoretic universe with urlements $\mathbb{N}$. For a relative pca $\mathbb{P} = (\mathsf{P}, \underset{\sim}{\mathbf{P}})$, as in the usual set-theoretic forcing argument, we consider a $\mathbb{P}$-*name*, which is any set x satisfying the following condition:

$$x \subseteq \{(p, u) : p \in \underset{\sim}{\mathbf{P}} \text{ and } (u \in \mathbb{N} \text{ or } u \text{ is a } \mathbb{P}\text{-name})\}.$$

Roughly speaking, $(p, u) \in x$ indicates that p is a witness for "$u \in x$." The $\mathbb{P}$-names are used as our universe. To be precise, we get the cumulative hierarchy of $\mathbb{P}$-names as follows:

$$V_0^{\mathbb{P}} = \emptyset, \quad V_\alpha^{\mathbb{P}} = \bigcup_{\beta < \alpha} \mathcal{P}(\underset{\sim}{\mathbf{P}} \times (V_\beta^{\mathbb{P}} \cup \mathbb{N})), \text{ and } V_{\mathrm{set}}^{\mathbb{P}} = \bigcup_{\alpha \in \mathrm{Ord}} V_\alpha^{\mathbb{P}}.$$

Note that the urelements $\mathbb{N}$ are disjoint from $V_{\mathrm{set}}^{\mathbb{P}}$. We define $V^{\mathbb{P}} = V_{\mathrm{set}}^{\mathbb{P}} \cup \mathbb{N}$. McCarty [32] used this notion to give a Kleene-like realizability interpretation for **IZF**. Then, as mentioned above, Chen-Rathjen [10] incorporated Lifschitz's multifunction $\mathbf{j}_{\mathrm{L}}$ into McCarty realizability. We now generalize Chen-Rathjen's realizability predicate to *any* partial multifunction $\mathbf{j}$.

Definition 3.14. Fix a partial multifunction $\mathbf{j} \colon\subseteq \underset{\sim}{\mathbf{P}} \rightrightarrows \underset{\sim}{\mathbf{P}}$. For $e \in \underset{\sim}{\mathbf{P}}$ and a sentence φ of **IZF** from parameters from $V^{\mathbb{P}}$, we define a relation $e \Vdash_{\mathbf{j}} \varphi$. For relation symbols:

$$e \Vdash_{\mathbf{j}} R(\bar{a}) \iff \mathbb{N} \models R(\bar{a})$$
$$e \Vdash_{\mathbf{j}} \mathbf{N}(a) \iff a \in \mathbb{N} \ \& \ e = \underline{a}$$
$$e \Vdash_{\mathbf{j}} \mathbf{Set}(a) \iff a \in V^{\mathbb{P}}_{\text{set}}$$

where R is a primitive recursive relation. For set-theoretic symbols:

$$e \Vdash_{\mathbf{j}} a \in b \iff \forall^+ d \in \mathbf{j}(e) \exists c \, [(\pi_0 d, c) \in b \ \wedge \ \pi_1 d \Vdash_{\mathbf{j}} a = c]$$
$$e \Vdash_{\mathbf{j}} a = b \iff (a, b \in \mathbb{N} \wedge a = b) \vee (\mathbf{Set}(a) \wedge \mathbf{Set}(b) \wedge$$
$$\forall^+ d \in \mathbf{j}(e) \forall p, c \, [(p, c) \in a \ \rightarrow \ \pi_0 dp \Vdash_{\mathbf{j}} c \in b] \wedge$$
$$\forall^+ d \in \mathbf{j}(e) \forall p, c \, [(p, c) \in b \ \rightarrow \ \pi_1 dp \Vdash_{\mathbf{j}} c \in a].$$

Here, we write $\forall^+ d \in \mathbf{j}(e)$ if $e \in \text{dom}(\mathbf{j})$ and $\forall d \in \mathbf{j}(e)$. For logical connectives:

$$e \Vdash_{\mathbf{j}} A \wedge B \iff \pi_0 e \Vdash_{\mathbf{j}} A \ \wedge \ \pi_1 e \Vdash_{\mathbf{j}} B$$
$$e \Vdash_{\mathbf{j}} A \vee B \iff \forall^+ d \in \mathbf{j}(e) \, [(\pi_0 d = 0 \wedge \pi_1 d \Vdash_{\mathbf{j}} A) \vee (\pi_0 d = 1 \wedge \pi_1 d \Vdash_{\mathbf{j}} B)]$$
$$e \Vdash_{\mathbf{j}} \neg A \iff (\forall a \in \underset{\sim}{\mathbf{P}}) \, a \nVdash_{\mathbf{j}} A$$
$$e \Vdash_{\mathbf{j}} A \rightarrow B \iff (\forall a \in \underset{\sim}{\mathbf{P}}) \, [a \Vdash A \ \rightarrow \ ea \Vdash_{\mathbf{j}} B].$$

For quantifiers:

$$e \Vdash_{\mathbf{j}} \forall x A \iff (\forall^+ d \in \mathbf{j}(e))(\forall c \in V^{\mathbb{P}}) \, e \Vdash_{\mathbf{j}} A[c/x]$$
$$e \Vdash_{\mathbf{j}} \exists x A \iff (\forall^+ d \in \mathbf{j}(e))(\exists c \in V^{\mathbb{P}}) \, e \Vdash_{\mathbf{j}} A[c/x].$$

Then we say that a formula φ is **j**-*realizable over* $\mathbb{P}$ if there is $e \in \mathsf{P}$ such that $e \Vdash_{\mathbf{j}} \varphi$.

Unfortunately, a **j**-realizability predicate does not necessarily validate axioms of **IZF**. Next, we will discuss what condition for a multifunction **j** is needed in order to ensure all axioms of **IZF**.

3.2.4. *j-operator.* The difficulty with Lifschitz-like realizability is that it only makes one access to an oracle **j**. Therefore, for this to behave correctly as oracle realizability, the condition that one access and any finite number of accesses to **j** are equivalent must be satisfied. This leads to the following notion:

Definition 3.15. Let $\mathbb{P} = (\mathsf{P}, \underset{\sim}{\mathbf{P}}, *)$ be a relative pca. We say that a partial multifunction $\mathbf{j} \colon\subseteq \underset{\sim}{\mathbf{P}} \rightrightarrows \underset{\sim}{\mathbf{P}}$ is a *j-operator* on $\mathbb{P}$ if

(1) There is $\mathsf{u} \in \mathsf{P}$ such that for any $a, x \in \underset{\sim}{\mathbf{P}}$, $a\mathbf{j}(x) = \mathbf{j}(\mathsf{u}ax)$.
(2) There is $\eta \in \mathsf{P}$ such that for any $x \in \underset{\sim}{\mathbf{P}}$, $x = \mathbf{j}(\eta x)$.
(3) There is $\mu \in \mathsf{P}$ such that for any $x \in \underset{\sim}{\mathbf{P}}$, $\mathbf{jj}(x) = \mathbf{j}(\mu x)$.

Here, $\mathbf{jj}$ is an abbreviation for the composition $\mathbf{j} \circ \mathbf{j}$, where recall the definition of the composition of multifunctions from Section 3.1.1. Also, if F is a multifunction on $\underset{\sim}{\mathbf{P}}$ and $a, x \in \underset{\sim}{\mathbf{P}}$, then define $aF(x) = \{ay : y \in F(x)\}$.

Definition 3.16. A pair $(\mathbb{P}, \mathbf{j})$ of a relative pca $\mathbb{P}$ and a j-operator $\mathbf{j}$ is called a *site*.

Definition 3.17. Let $(\mathbb{P}, \mathbf{j})$ be a site. A partial multifunction $f : \subseteq \underline{\mathbf{P}} \rightrightarrows \underline{\mathbf{P}}$ is $(\mathbb{P}, \mathbf{j})$-*trackable* if there is $a \in \underline{\mathbf{P}}$ such that, for any $x \in \mathrm{dom}(f)$, we have $\underset{\sim}{\mathbf{j}}(ax) \subseteq f(x)$. If $a \in \mathsf{P}$, we say that f is $(\mathbb{P}, \mathbf{j})$-*computable*.

Remark. This notion (for operators satisfying (1) and (2)) is implicitly studied in the work on the jump of a represented space, e.g. by de Brecht [11]. From the viewpoint of the jump of a represented space, one may think of $\mathbf{j}$ as an endofunctor on the category **Rep** of represented spaces and computable functions. Indeed, any j-operator $\mathbf{j}$ yields a *monad* on the category **Rep**: The condition (3) gives us a monad multiplication, and from (2) we get a unit. Thus, the $\mathbf{j}$-computable functions on represented spaces are exactly the *Kleisli morphisms* for this monad.

Remark. In the context of Lifschitz realizability, the property (1) in our definition of a j-operator corresponds to [30, Lemma 4], [47, Lemma 5.7], and [10, Lemma 4.4], the property (2) corresponds to [30, Lemma 2], [47, Lemma 5.4], and [10, Lemma 4.2], and (3) corresponds to [30, Lemma 3], [47, Lemma 5.6], and [10, Lemma 4.5].

One can obtain a lot of examples of j-operators from partial multi-valued oracles.

Lemma 3.18. *Let* $\mathbb{P} = (P, \underline{\mathbf{P}}, *)$ *be a relative pca, and let* $F : \subseteq \underset{\sim}{\mathbf{P}} \rightrightarrows \underset{\sim}{\mathbf{P}}$ *be a partial multifunction. Then,* $F^{\circlearrowright}$ *is a j-operator on* $\mathbb{P}$.

Proof. By definition, it is easy to check that $F^{\circlearrowright}$ satisfies the conditions (1) and (2). The idempotence (3) follows from transitivity of $\leq_{\circlearrowright\mathsf{W}}$ [17]. $\qquad\square$

In fact, by Westrick's theorem [50], $F^{\circlearrowright}$ is the weakest j-operator generated by F. Moreover, the $(\mathbb{P}, F^{\circlearrowright})$-computable partial multifunctions coincide with the partial multifunctions $\leq_{\circlearrowright\mathsf{W}} F$.

Observation 3.19. *Let* $\mathbf{j}$ *be a j-operator on* $\mathbb{P}$ *and* $f : \subseteq \underset{\sim}{\mathbf{P}} \rightrightarrows \underset{\sim}{\mathbf{P}}$ *be a partial multifunction.*

 (1) $f \leq_{\mathsf{W}} \mathbf{j}$ *iff f is* $(\mathbb{P}, \mathbf{j})$-*computable.*
 (2) $f \leq_{\mathsf{W}} \mathbf{j}$ *iff* $f^{\circlearrowright} \leq_{\mathsf{W}} \mathbf{j}$ *iff* $f \leq_{\circlearrowright\mathsf{W}} \mathbf{j}$.

Proof. (1) This follows from the self-universality of a j-operator; i.e., the condition (1) in Definition 3.15. (2) See Westrick [50], and also [22]. $\qquad\square$

For a j-operator $\mathbf{j}$ on a relative pca $\mathbb{P} = (\mathsf{P}, \underline{\mathbf{P}}, *)$, one can introduce a new partial application $*_{\mathbf{j}}$ on $\underline{\mathbf{P}}$ defined by $a *_{\mathbf{j}} b \simeq a^! * b$ if $\mathbf{j}(a) = \{a^!\}$; otherwise $a *_{\mathbf{j}} b$ is undefined. Hereafter, we always write $a^!$ for the unique element of $\mathbf{j}(a)$ whenever $\mathbf{j}(a)$ is a singleton. Then, consider the following "deterministic part":

$$\mathsf{P}_{\mathbf{j}} = \{a^! : a \in \mathsf{P} \text{ and } \mathbf{j}(a) \text{ is a singleton}\}.$$

Lemma 3.20. $\mathbb{P}_{\mathbf{j}} = (\mathsf{P}_{\mathbf{j}}, \underset{\sim}{\mathbf{P}}, *_{\mathbf{j}})$ *is a relative pca. Moreover, (4) there is $\varepsilon \in \mathsf{P}_{\mathbf{j}}$ such that for any $a \in \underset{\sim}{\mathbf{P}}$, if $a^!$ is defined, then $\varepsilon *_{\mathbf{j}} a = a^!$.*

Proof. To see that $\mathsf{P}_{\mathbf{j}}$ is closed under $*_{\mathbf{j}}$, for $a, b \in \mathsf{P}$, we have $a *_{\mathbf{j}} b = a^! b$. Then, $a^! b = (\lambda x.xb)a^! = (\mathsf{u}(\lambda x.xb)a)^!$, where u is from (1). As $\mathsf{u}(\lambda x.xb)a \in \mathsf{P}$, we have $a^! b \in \mathsf{P}_{\mathbf{j}}$.

Given a combinator k in P, define $\mathsf{k}_{\mathbf{j}} = \eta(\lambda x.\eta(\mathsf{k}x))$, where η is from (2). Then, $\mathsf{k}_{\mathbf{j}} *_{\mathbf{j}} a *_{\mathbf{j}} b = (\mathsf{k}_{\mathbf{j}}^! a)^! b = (\iota(\mathsf{k}a))^! b = \mathsf{k}ab = a$. Given a combinator s in P, define $\mathsf{s}_{\mathbf{j}} =$

$\eta(\lambda x.\eta(\lambda y.\eta(\mathsf{s}xy)))$. Then, $\mathsf{s_j} *_\mathbf{j} a *_\mathbf{j} b *_\mathbf{j} c = ((\mathsf{s_j^!}a)^!b)^!c = ((\eta(\lambda y.\eta(\mathsf{s}ay)))^!b)^!c = (\eta(\mathsf{s}ab))^!c = \mathsf{s}abc = ac(bc)$.

Thus, $\mathbb{P_j}$ is a relative pca, and so we get a new $\lambda_\mathbf{j}$ by combinatory completeness. For the second assertion, define $\varepsilon = \lambda_\mathbf{j}x.\mathsf{u}(\mathsf{k}x) *_\mathbf{j} 0$. Then, $\varepsilon *_\mathbf{j} a = \mathsf{u}(\mathsf{k}a) *_\mathbf{j} 0 = (\mathsf{u}(\mathsf{k}a))^!0 = \mathsf{k}a^!0 = a^!$. $\square$

Remark. In the context of Lifschitz realizability, the condition (4) in Lemma 3.20 corresponds to [30, Lemma 1], [47, Lemma 5.3], and [10, Lemma 4.3].

We next show that the update of a given pca preserves the class of realizable functions.

Lemma 3.21. *A partial multifunction $f : \subseteq \mathbf{\underset{\sim}{P}} \rightrightarrows \mathbf{\underset{\sim}{P}}$ is $(\mathbb{P_j}, \mathbf{j})$-trackable iff it is $(\mathbb{P}, \mathbf{j})$-trackable. Similarly, f is $(\mathbb{P_j}, \mathbf{j})$-computable iff it is $(\widetilde{\mathbb{P}}, \mathbf{j})$-computable.*

Proof. We only consider the latter assertion. By definition, every $(\mathbb{P}, \mathbf{j})$-computable partial multifunction is $(\mathbb{P_j}, \mathbf{j})$-computable. Conversely, if F is $(\mathbb{P_j}, \mathbf{j})$-computable, then there is $a \in \mathsf{P_j}$ such that $\mathbf{j}(ax) \subseteq F(x)$. By the definition of $\mathsf{P_j}$, there is $b \in \mathsf{P}$ such that $a = b^!$. Then $b^!x = (\lambda y.yx)b^! = (\mathsf{u}(\lambda y.yx)b)^!$, where u is from (1). Therefore, $\mathbf{j}(ax) = \mathbf{j}(b^!x) = \mathbf{jj}(\mathsf{u}(\lambda y.yx)b) = \mathbf{j}(\mu(\mathsf{u}(\lambda y.yx)b))$ where μ is from (3). Since $\mu(\mathsf{u}(\lambda y.yx)b) \in \mathsf{P}$, this shows that F is $(\mathbb{P}, \mathbf{j})$-computable. $\square$

Consequently, Lemma 3.18 still holds even if we replace $\mathbb{P}$ with the new pca $\mathbb{P_j}$. Hereafter, we always assume the underlying pca $\mathbb{P}$ is $\mathbb{P_j}$; so in particular, the properties (1), (2), (3) and (4) hold.

Lifschitz [30] showed that the above four properties (1), (2), (3) and (4) ensure that all axioms of the Heyting arithmetic **HA** are $\mathbf{j_L}$-realizable over K_1. Moreover, van Oosten [47] showed that all axioms of Troelstra's elementary analysis **EL** are $\mathbf{j_{vO}}$-realizable over $\mathsf{K}_2^{\mathsf{eff}}$. Then Chen-Rathjen [10] showed that all axioms of **IZF** are $\mathbf{j_L}$-realizable over K_1. Now, the properties (1), (2), (3), and (4) ensure the following general result.

Theorem 3.22. *Let $(\mathbb{P}, \mathbf{j})$ be a site. Then, all axioms of **IZF** are $\mathbf{j}$-realizable over $\mathbb{P_j}$.* $\square$

3.2.5. *Topos-theoretic perspective.* A topos-theoretic explanation may convince some readers of the fact that **IZF** is $\mathbf{j}$-realizable (Theorem 3.22) without referring to [30, 47, 10]. If the reader is already convinced by the concrete discussion in the previous section (with the references [30, 47, 10]), they can skip this section.

As noted by the author [22], a j-operator on $\mathbb{P}$ in the sense of Definition 3.15 yields a Lawvere-Tierney topology on the relative realizability topos $\mathsf{RT}(\mathbb{P})$. A *Lawvere-Tierney topology* on a topos is an (internally) $\wedge$-preserving closure operator on the truth-value object Ω. It is known that a Lawvere-Tierney topology on a topos $\mathcal{E}$ corresponds to a subtopos of $\mathcal{E}$; see e.g. [31, Chapter V]. In particular, the site $(\mathbb{P}, \mathbf{j})$ always presents a subtopos of $\mathsf{RT}(\mathbb{P})$.

Let us explain more concretely. Let Ω be the power set of $\mathbf{\underset{\sim}{P}}$. A partial multifunction $F : \subseteq \mathbf{\underset{\sim}{P}} \rightrightarrows \mathbf{\underset{\sim}{P}}$ yields $j_F : \Omega \to \Omega$ as follows:

$$j_F(p) = \{\langle e, n \rangle : n \in \mathrm{dom}(\Phi_e^F) \text{ and } \Phi_e^F(n) \subseteq p\}$$

This gives a Lawvere-Tierney topology on $\mathsf{RT}(\mathbb{P})$; see [22]. Note that j_F can also be considered as a closure operation on the realizability tripos over $\mathbb{P}$. That is, think of Ω^X as the Heyting pre-algebra, where $\varphi \leq_X \psi$ iff there is $a \in \mathsf{P}$ such that $n \in \varphi(x)$

implies $an \in \psi(x)$ for any $x \in X$. This is called a realizability tripos [49]. Then F yields a new preorder $\leq_X^F$ on Ω^X as follows: For predicates φ, ψ on X, define $\varphi \leq_X^F \psi$ as $\varphi(x) \leq_X j_F(\psi(x))$. Equivalently,

$$\exists e \in \mathsf{P} \forall x \in X \forall n \in \varphi(x) \ [\Phi_e^F(n) \downarrow \text{ and } \forall b \in \Phi_e^F(n). \ b \in \psi(x)]$$

Then $(\Omega^X, \leq_X^F)$ yields a new tripos. The interpretations of logical connectives are given as follows:

$$\top_F = \top, \qquad \bot_F = \bot,$$
$$\varphi \wedge_F \psi = \varphi \wedge \psi, \qquad \varphi \vee_F \psi = \varphi \vee \psi, \qquad \varphi \to_F \psi = \varphi \to j_F(\psi),$$
$$\exists^F = \exists, \qquad \forall^F = \forall \circ j_F$$

This is a variant of the Gödel-Gentzen translation, called the (Kuroda-style) j_F-translation (see e.g. [29, 46, 40]). This gives exactly the same definition as Definition 3.11, so one can conclude that F-realizability is the semantics of the subtopos of $\mathsf{RT}(\mathbb{P})$ consisting of the j_F-sheaves (since an inclusion of triposes coincides with an inclusion of toposes; see also van Oosten [49, Theorem 2.5.11]).

Remark. Our term "site" introduced in Definition 3.16 originates from topos theory. In topos theory, a site is a presentation of a Grothendieck topos, specifically, the pair $(\mathcal{C}, J)$ of a small category $\mathcal{C}$ and a Grothendieck topology J. In other words, a site $(\mathcal{C}, J)$ is a presentation of the topos of J-sheaves over $\mathcal{C}$, which is a subtopos of the presheaf topos $\mathsf{Psh}(\mathcal{C})$ over $\mathcal{C}$. In realizability theory, we consider a relative pca $\mathbb{P}$ to play the role of a base category $\mathcal{C}$, and an oracle $\mathbf{j}$ to play the role of a Grothendieck topology (see also [22]), which is why we borrowed the term "site."

3.3. **Weihrauch analysis of realizability interpretation.**

3.3.1. *The internal Baire space.* Our main focus is on interpreting mathematical statements in the language of second-order arithmetic. Here, $\mathbb{N}$ and $\mathbb{N}^{\mathbb{N}}$ are basic sorts in the language, so we want to find out how they are interpreted. One may use Definition 3.14 directly, but the interpretation in a realizability topos (or its subtopos) is clearer, so we explain that first.

Let $(\mathbb{P}, \mathbf{j})$ be a site. We explicitly describe the internal logic of the topos of $\mathbf{j}$-sheaves. Of course, the use of a topos for the interpretation of second-order arithmetic is like using a sledgehammer to crack a nut. It is sufficient to consider its full subcategory consisting of assemblies.

Definition 3.23. A $\mathbb{P}$-*assembly* (or a multi-represented space) is a pair of a set X and a function $\mathbf{E}_X \colon X \to \Omega$, where Ω is the power set of $\mathbf{P}$, and $\mathbf{E}_X(x) \neq \emptyset$. Each $p \in \mathbf{E}_X(x)$ is called a code of x. For $\mathbb{P}$-assemblies X and Y, a function $f \colon X \to Y$ is $(\mathbb{P}, \mathbf{j})$-trackable if there is $a \in \mathbf{P}$ such that, for any code p of $x \in X$, any element of $\mathbf{j}(ap)$ is a code of $f(x) \in Y$. If $a \in \mathsf{P}$, we say that f is $(\mathbb{P}, \mathbf{j})$-computable.

If we set $F(p) = \mathbf{E}_Y(f(x))$ for $p \in \mathbf{E}_X(x)$, this terminology is consistent with Definition 3.17. The Kleisli category over the site $(\mathbb{P}, \mathbf{j})$ consists of $\mathbb{P}$-assemblies as objects and $(\mathbb{P}, \mathbf{j})$-computable functions as morphisms. We often omit $\mathbb{P}$ and simply refer to $(\mathbb{P}, \mathbf{j})$-trackable (-computable) as a $\mathbf{j}$-trackable (-computable) function.

Definition 3.24. The exponential assembly Y^X is the set of **j**-trackable functions from X to Y, where a code of $f \in Y^X$ is a **j**-tracker of f.

Example 3.25. The natural numbers object N is given as follows: The underlying set is ω, and a code of $n \in \mathsf{N}$ is the Church numeral $\underline{n}$; i.e., $\mathbf{E}_\mathsf{N}(n) = \{\underline{n}\}$.

Then the *internal Baire space* N^N is given as the exponential assembly. To be explicit, the underlying set is the set of all **j**-trackable functions on ω, and $\mathsf{f} \in \underset{\sim}{\mathbf{P}}$ is a code of $f \in \mathsf{N}^\mathsf{N}$ iff $\mathbf{j}(\mathsf{f}\underline{n}) = \{\underline{f(n)}\}$ for any $n \in \omega$.

Example 3.26. If $\mathbb{P} = \mathsf{K}_2^{\mathsf{eff}}$ then $\underset{\sim}{\mathbf{P}} = \omega^\omega$. One may consider ω^ω as an assembly given by the identical coding $\mathbf{E}_{\omega^\omega}(f) = \underset{\sim}{f}$ for any $f \in \omega^\omega$. We call this the *external Baire space.*

In the case where $\underset{\sim}{\mathbf{j}} = \mathrm{id}$, we have $\omega^\omega \simeq \mathsf{N}^\mathsf{N}$, but this is not always the case in general.

Remark. Of course, one may obtain the internal Baire space directly using the realizability interpretation (Definition 3.14) without going through the category of assemblies: First $e \Vdash_\mathbf{j} x \in \mathbb{N}^\mathbb{N}$ means that $e \Vdash_\mathbf{j} \forall n \in \mathbb{N} \exists! m \in \mathbb{N} \langle n, m \rangle \in x$. Then, it is easy to check that there is a unique $\tilde{x} \in \mathbb{N}^\mathbb{N}$ such that $\langle n, \tilde{x}(n) \rangle$ is realizable. By Chen-Rathjen [10, Lemma 4.7], given such an e, one can effectively produce a p_e such that $p_e *_\mathbf{j} n = \{\tilde{x}(n)\}$, i.e., $(p_e^! n)^! = \tilde{x}(n)$. Conversely, from such p_e, one can effectively recover the realizer e. That is, we consider p_x as a name of $\tilde{x} \in \mathbb{N}^\mathbb{N}$ if $(p_x^! n)^! = \tilde{x}(n)$. In essence, this gives a $\mathbb{P}_\mathbf{j}$-tracker for $\tilde{x}$. Therefore, by Lemma 3.21, one can see that this is equivalent to considering the above exponential assembly N^N.

3.3.2. *Internal versus external Baire spaces.* We connect the realizability interpretation and Weihrauch reducibility. When interpreting a formula in second-order arithmetic, we must be careful that the interpretation of type $\mathbb{N}^\mathbb{N}$ is the internal Baire space N^N, whereas our Weihrauch reducibility is based on the external Baire space ω^ω. If we can identify N^N with ω^ω as in $\mathsf{K}_2^{\mathsf{eff}}$, then there is no need to worry about this.

Definition 3.27. Let $\mathbf{j}$ be a j-operator on $\mathsf{K}_2^{\mathsf{eff}}$. We say that $\mathbf{j}$ satisfies the *relative unique Church thesis* (rCT!) if $\mathrm{id} \colon \omega^\omega \to \mathsf{N}^\mathsf{N}$ is an isomorphism in the Kleisli category over $(\mathsf{K}_2^{\mathsf{eff}}, \mathbf{j})$.

Remark. As in Lemma 3.20, the computability of $\mathrm{id} \colon \omega^\omega \to \mathsf{N}^\mathsf{N}$ always holds, so the nontrivial part is the computability of $\mathrm{id} \colon \mathsf{N}^\mathsf{N} \to \omega^\omega$. That is, $\mathbf{j}$ satisfies rCT! iff there is a **j**-computable function which, given a **j**-tracker of $f \colon \mathsf{N} \to \mathsf{N}$, returns $f \in \omega^\omega$.

Example 3.28. $\mathbf{j} = \mathrm{id}$ satisfies rCT!. This is because $\mathrm{id} \colon \omega^\omega \to \mathsf{N}^\mathsf{N}$ is an isomorphism in the category of $\mathsf{K}_2^{\mathsf{eff}}$-assemblies.

Definition 3.29. Let $\mathbf{j}$ be a j-operator on $\mathsf{K}_2^{\mathsf{eff}}$. Then define $\mathbf{j}^! \colon\subseteq \omega^\omega \to \omega^\omega$ as follows:

$$\mathbf{j}^!(\alpha)(n) = m \iff \mathbf{j}(\alpha_n) = \{\underline{m}\}$$

where for $\alpha \in \omega^\omega$, $\alpha_n \in \omega^\omega$ is given by $\alpha_n(m) = \alpha(\langle n, m \rangle)$, and $\alpha \in \mathrm{dom}(\mathbf{j}^!)$ iff $\mathbf{j}^!(\alpha)(n)$ is defined for all $n \in \omega$.

Remark. Using the terminology of the theory of Weihrauch degrees, $\mathbf{j}^!$ is the parallelization of the deterministic first-order part of $\mathbf{j}$.

Observation 3.30. *If* $\mathbf{j}_0 \leq_\mathsf{W} \mathbf{j}_1$ *then* $\mathbf{j}_0^! \leq_\mathsf{W} \mathbf{j}_1^!$.

Lemma 3.31. $\mathbf{j}$ *satisfies* rCT! *if and only if* $\mathbf{j}^! \leq_W \mathbf{j}$.

Proof. ($\Rightarrow$) Let $\alpha = (\alpha_n)_{n \in \omega} \in \text{dom}(\mathbf{j}^!)$ be given. One can easily compute $e \in \omega^\omega$ such that $\alpha_n = e\alpha\underline{n}$ holds in $\mathsf{K}_2^{\text{eff}}$ for any $n \in \omega$. As $\alpha \in \text{dom}(\mathbf{j}^!)$, we have $\mathbf{j}(e\alpha\underline{n}) = \mathbf{j}(\alpha_n) = \{\mathbf{j}^!(\alpha)(n)\}$. By the definition of N^N in Example 3.25, this means that $e\alpha$ is a code of $\mathbf{j}^!(\alpha) \in \mathsf{N}^\mathsf{N}$. Now, by our assumption, id $\colon \mathsf{N}^\mathsf{N} \to \omega^\omega$ is $\mathbf{j}$-computable, so there is $p \in \omega^\omega$ such that any $z \in \mathbf{j}(p(e\alpha))$ is a code of $\mathbf{j}^!(\alpha) \in \omega^\omega$. By the definition of the external Baire space ω^ω in Example 3.26, this exactly means $\mathbf{j}(p(e\alpha)) = \{\mathbf{j}^!(\alpha)\}$. Therefore, we obtain $\mathbf{j}^! \leq_W \mathbf{j}$.

($\Leftarrow$) As in Lemma 3.20, one can check that id $\colon \omega^\omega \to \mathsf{N}^\mathsf{N}$ is always computable. To show that id $\colon \mathsf{N}^\mathsf{N} \to \omega^\omega$ is $\mathbf{j}$-computable, suppose that we are given a code z for $x \in \mathsf{N}^\mathsf{N}$. By the definition of the internal Baire space, for any $n \in \omega$, we have $\mathbf{j}(z\underline{n}) = \{\underline{x(n)}\}$. Take α such that $\alpha_n = z\underline{n}$. Then we have $\mathbf{j}(\alpha_n) = \{\underline{x(n)}\}$, so we obtain that $\mathbf{j}^!(\alpha) = x$. By the assumption, we have $\mathbf{j}^! \leq_W \mathbf{j}$, so by Observation 3.19, there exists some $w \in \omega^\omega$ such that $\mathbf{j}(w\alpha) = \{\mathbf{j}^!(\alpha)\} = \{x\}$ holds. Now, if we consider $v \in \omega^\omega$ such that $vz = \alpha$, then $\lambda z.w(vz)$ tracks id $\colon \mathsf{N}^\mathsf{N} \to \omega^\omega$. $\square$

Recall from Lemma 3.18 that $F^{\supset}$ is a j-operator for any $F \colon\subseteq \omega^\omega \rightrightarrows \omega^\omega$. By an abuse of notation, we also use $F^!$ as an abbreviation for $(F^{\supset})^!$.

Lemma 3.32. *If* $\mathbf{j} \leq_W$ WKL, *then* $\mathbf{j}$ *satisfies* rCT!.

Proof. By Observation 3.30 and Lemma 3.31, it suffices to show that $\mathsf{WKL}^! \leq_W \mathsf{WKL}$, but this is obvious since $\mathsf{WKL}^! \equiv_W$ id (that is, $\mathsf{WKL}^!$ is computable), which follows from an easy observation that the deterministic part of WKL (so its parallelization) is trivial (see e.g. [7, Corollary 8.8]). Note that this observation corresponds to the fact that $\mathsf{WKL}^{\supset}$-realizability satisfies the unique Church thesis $\mathsf{CT}_0!$. $\square$

Example 3.33 (Non-example)**.** The discrete limit operator $\lim_\omega \colon\subseteq \omega^\omega \to \omega$ does not satisfy rCT!: This is a partial single-valued function defined by $\lim_\omega(\alpha) = \lim_{n \to \infty} \alpha(n)$ if the limit exists in the discrete space ω. Its deterministic first-order part is $\lim_\omega$ itself, but its parallelization is the limit operator $\lim_{\omega^\omega} \colon\subseteq (\omega^\omega)^\omega \to \omega^\omega$, where $\lim_{\omega^\omega}(\alpha) = \lim_{n \to \infty} \alpha(n)$ if the limit exists in the Baire space ω^ω. Then we have $\lim_\omega <_W \lim_\omega^! \equiv_W \lim_{\omega^\omega}$.

Remark. For any $\mathbf{j}$, $\mathbf{j}^! \colon\subseteq \omega^\omega \to \omega^\omega$ is "parallelizable" partial single-valued function. The continuous Weihrauch degrees of parallelizable total single-valued functions are completely classified by the author [23] (under meta-theory $\mathsf{ZF} + \mathsf{DC} + \mathsf{AD}$, where DC is the axiom of dependent choice, and AD is the axiom of determinacy): The structure of parallelized continuous Weihrauch degrees of single-valued functions is well-ordered (with the order type Θ), and the smallest nontrivial one already entails the Turing jump operator (a universal Baire class 1 function). In particular, under meta-theory ZFC, by Martin's Borel determinacy theorem, there are ω_1 many Borel single-valued functions (which are exactly universal Baire class α functions).

3.3.3. *$\forall\exists$-theorems and Weihrauch degrees.* We say that a partial multifunction $F \colon\subseteq \underset{\sim}{\mathbf{P}} \rightrightarrows \underset{\sim}{\mathbf{P}}$ is $\neg\neg$-*dense* if $\mathbf{j}(x) \neq \emptyset$ for any $x \in \text{dom}(F)$. Note that if F is $\neg\neg$-dense then so is $\mathbf{j} = F^{\supset}$. In this section, we assume that $\mathbf{j}$ is a $\neg\neg$-dense j-operator on $\mathsf{K}_2^{\text{eff}}$.

Remark. In this case, an almost negative statement ψ is **j**-realizable iff it is true, and one may compute a term realizing ψ (see e.g. [44, Proposition 4.4.5]). Here, an almost negative formula φ is a formula which does not contain $\vee$, and $\exists$ appears only immediately in front of an atomic subformula. Since the realizability of a Δ_0^0-formula (a formula in second-order arithmetic containing no unbounded quantifiers) also corresponds to the truth, one can treat a Δ_0^0-formula as an atomic formula without any harm. In particular, Σ_1^0- and Π_1^0-formulas can be considered almost negative formulas.

Now, consider an $\forall\exists$-statement

$$\varphi \equiv \forall x \in \mathbb{N}^{\mathbb{N}} \, [\eta(x) \to \exists y \in \mathbb{N}^{\mathbb{N}} \, \theta(x, y)]$$

for some almost negative formulas η and θ. We call such a formula φ a tidy $\forall\exists$-principle.

Example 3.34. WKL is a tidy $\forall\exists$-principle: The premise $\eta(T)$ is the conjunction of the statement that T is a binary tree, and the formula $\forall \ell \exists \sigma \in 2^\ell. \ \sigma \in T$. Hence, η is Π_1^0, so almost negative. The conclusion is $\theta(p) \equiv \forall n. \ p \upharpoonright n \in T$. This is clearly almost negative.

Similarly, IVT, BE, RDIV and LLPO are tidy $\forall\exists$-principles: The premises of BE, RDIV and LLPO are Π_1^0, and the premise of IVT is Σ_1^0. The conclusions are Π_1^0.

Example 3.35. Σ_1^0-DNE$_\mathbb{R}$, Σ_1^0-DML$_\mathbb{R}$ and Π_1^0-LEM$_\mathbb{R}$ introduced in Section 2.1.1 are tidy $\forall\exists$-principles. For example, the premise of Π_1^0-LEM$_\mathbb{R}$ is Π_1^0. For the conclusion $\theta(\langle x, y\rangle, i) \equiv (i = 0 \to x = y$ and $i \neq 0 \to x \neq y)$, clearly, $x = y$ is Π_1^0 and $x \neq y$ is Σ_1^0, so θ is almost negative.

By the above Remark, one can easily see that a tidy $\forall\exists$-principle φ is **j**-realizable iff there is an element $e \in (\omega^\omega)^{\mathsf{eff}}$ such that any $a \in \mathbf{j}(e)$ witnessing that if p codes $x \in \mathsf{N}^{\mathsf{N}}$ such that $\eta(x)$ is true then any $q \in \mathbf{j}(ap)$ codes $y \in \mathsf{N}^{\mathsf{N}}$ such that $\theta(x, y)$ is true.

The partial multifunction corresponding to the formula φ is given as follows.

$$\mathrm{dom}(F_\varphi) = \{x \in \omega^\omega : \eta(x) \text{ is true}\},$$
$$F_\varphi(x) = \{y \in \omega^\omega : \theta(x, y) \text{ is true}\}.$$

Of course, we must pay attention to the difference between ω^ω and N^{N}. However, as long as we are considering j-operators that satisfy rCT!, we can ignore this difference.

Lemma 3.36. *Let* **j** *be a* $\neg\neg$*-dense j-operator on* $\mathsf{K}_2^{\mathsf{eff}}$*, and assume that* **j** *satisfies* rCT!. *Then a tidy $\forall\exists$-principle φ is **j**-realizable if and only if $F_\varphi \leq_{\mathsf{W}}$* **j**.

Proof. Since **j** satisfies rCT!, $\mathrm{id} \colon \mathsf{N}^{\mathsf{N}} \to \omega^\omega$ is **j**-computable. Thus, by the idempotence and $\neg\neg$-density of **j**, φ is **j**-realizable iff there is $e \in (\omega^\omega)^{\mathsf{eff}}$ such that for any $x \in \omega^\omega$ if $\eta(x)$ is true then $\theta(x, y)$ is true for any $y \in \mathbf{j}(ex)$. That is, $x \in \mathrm{dom}(F_\varphi)$ implies $y \in F_\varphi(x)$ for any $y \in \mathbf{j}(ex)$. This means $F_\varphi \leq_{\mathsf{W}}$ **j** by Observation 3.19. $\qquad\square$

4. Key principles

We introduce some non-constructive principles in the context of Weihrauch degrees, and show some implications and equivalences among them.
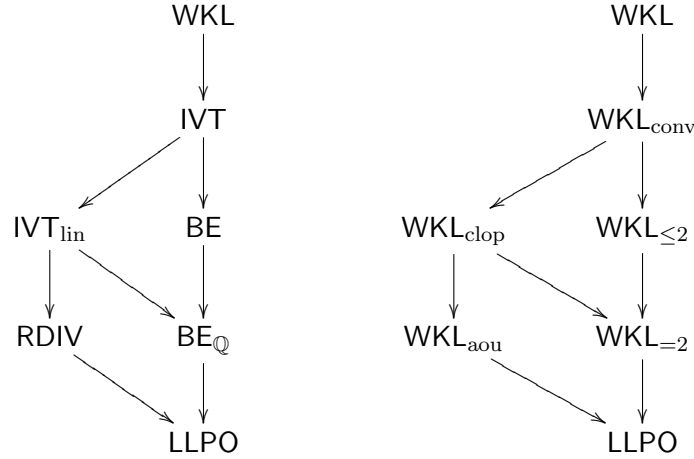
FIGURE 2. Nonconstructive principles as variants of weak König's lemma

4.1. **Weak König's lemma.** In this section, we consider weak variants of weak König's lemma. A *binary tree* is a subset of $2^{<\omega}$ closed under taking initial segments. A binary tree T is *infinite* if for any ℓ there is a node $\sigma \in T$ of length ℓ. We consider the following properties for trees.

- A *2-tree* is a binary tree T such that, for any $\ell \in \mathbb{N}$, T has exactly two strings of length ℓ.
- A *rational* 2-*tree* is a 2-tree which does *not* have infinitely many branching nodes.
- An *all-or-unique tree* or simply an *aou-tree* is a binary tree T such that, for any $\ell \in \mathbb{N}$, T has either all binary strings or only one string of length ℓ.
- A *convex tree* is a binary tree T such that, for any $\ell \in \omega$, if two strings σ and τ of length ℓ are contained in T, then all length ℓ strings between σ and τ are also contained in T.
- A binary tree T is *constructively non-clopen* if, for any finitely many binary strings $\sigma_0, \ldots, \sigma_\ell$, there is a binary string τ such that the following does not hold: $\tau \in T$ if and only if τ extends σ_i for some $i \leq \ell$.
- A *clopen tree* is a binary tree T which is not constructively non-clopen. If $\ell = 0$, then we say that T is a *basic clopen tree*.

The notion of a clopen tree is a bit weird, but we use this notion as it is useful for understanding the relationship among non-constructive principles. For a collection $\mathcal{T}$ of trees, we mean by *weak König's lemma for $\mathcal{T}$* the statement that every infinite tree in $\mathcal{T}$ has an infinite path. Then, we denote by $\mathsf{WKL}_{\leq 2}$, $\mathsf{WKL}_{=2}$, $\mathsf{WKL}_{\mathrm{aou}}$, $\mathsf{WKL}_{\mathrm{conv}}$, and $\mathsf{WKL}_{\mathrm{clop}}$ weak König's lemma for 2-trees, rational 2-trees, aou-trees, convex trees, and convex clopen trees, respectively.

Remark. It is known that, over elementary analysis EL_0, the binary expansion principle BE is equivalent to weak König's lemma for 2-trees, $\mathsf{WKL}_{\leq 2}$, and the intermediate value theorem IVT is equivalent to weak König's lemma for convex trees, $\mathsf{WKL}_{\mathrm{conv}}$; see [6]. For the relationship among these principles, see Figure 2 and Section 4.1.1 for the proofs.

Observation 4.1. $\mathsf{WKL}_{\leq 2}$, $\mathsf{WKL}_{=2}$, $\mathsf{WKL}_{\mathrm{aou}}$, $\mathsf{WKL}_{\mathrm{conv}}$ *and* $\mathsf{WKL}_{\mathrm{clop}}$ *are tidy* $\forall\exists$*-principles.*

Proof. Clearly, $\mathsf{WKL}_{\leq 2}$, $\mathsf{WKL}_{\mathrm{aou}}$ and $\mathsf{WKL}_{\mathrm{conv}}$ are tidy $\forall\exists$-principles.

Similarly, $\mathsf{WKL}_{=2}$ is a tidy $\forall\exists$-principle: The formula describing that T does not have infinite many branching nodes is

$$\neg\forall\ell\exists\sigma \ (|\sigma| \geq \ell \text{ and } \sigma, \sigma^\frown 0, \sigma^\frown 1 \in T).$$

Clearly, the existential quantifier $\exists$ appears only in the front of a Δ_0^0 subformula. Hence, the premise of $\mathsf{WKL}_{=2}$ is almost negative.

$\mathsf{WKL}_{\mathrm{clop}}$ is also a tidy $\forall\exists$-principle: For a formula describing being non-constructively non-clopen, the existential quantifier $\exists\tau$ appears only in the front of a Δ_0^0 subformula. Hence, the premise of $\mathsf{WKL}_{\mathrm{clop}}$ is almost negative. $\square$

In the context of Weihrauch degrees, these notions are formalized as multifunctions. First, *weak König's lemma for 2-trees*, $\mathsf{WKL}_{\leq 2}\colon\subseteq 2^{2^{<\omega}} \rightrightarrows 2^\omega$, is defined as follows:

$$\mathrm{dom}(\mathsf{WKL}_{\leq 2}) = \{T \subseteq 2^{<\omega} : T \text{ is a 2-tree}\},$$

$$\mathsf{WKL}_{\leq 2}(T) = [T] := \{p \in 2^\omega : p \text{ is an infinite path through } T\}.$$

Classically, a 2-tree has either *one or two* infinite paths, and a rational 2-tree has *exactly two* infinite paths. Then, define the multifunction $\mathsf{WKL}_{=2}$ as $\mathsf{WKL}_{\leq 2}$ restricted to the rational 2-trees.

Remark. In the study of Weihrauch degrees, this kind of notion is treated as a choice principle. Consider the *choice principle for finite closed sets in X with at most n elements $\mathsf{C}_{X,\#\leq n}$ (with exactly n elements $\mathsf{C}_{X,\#=n}$)*, which states that, given (a code of) a closed subset P of a space X if P is nonempty, but has at most n elements (exactly n elements), then P has an element; see [28]. The principles $\mathsf{WKL}_{\leq 2}$ and $\mathsf{C}_{X,\#\leq 2}$ are not constructively equivalent, where $X = 2^\omega$. However, if we consider $\mathsf{WKL}_{\leq 2}$ as a multifunction, then they are Weihrauch equivalent. Similarly, $\mathsf{WKL}_{=2}$ and $\mathsf{C}_{X,\#=2}$ are Weihrauch equivalent.

Next, the *weak König's lemma for convex trees* is a multifunction $\mathsf{WKL}_{\mathrm{conv}}\colon\subseteq 2^{2^{<\omega}} \rightrightarrows 2^\omega$ defined as follows:

$$\mathrm{dom}(\mathsf{WKL}_{\mathrm{conv}}) = \{T \subseteq 2^{<\omega} : T \text{ is a convex tree}\},$$

$$\mathsf{WKL}_{\mathrm{conv}}(T) = [T] := \{p \in 2^\omega : p \text{ is an infinite path through } T\}.$$

In the context of Weihrauch degrees, a clopen tree is exactly a binary tree T such that $[T]$ is clopen, and a basic clopen tree is a tree of the form $\{\tau \in 2^{<\omega} : \tau \succ \sigma\}$ for some binary string σ. It is clear that every basic clopen tree is a convex clopen tree. Then, define the multifunction $\mathsf{WKL}_{\mathrm{clop}}$ as $\mathsf{WKL}_{\mathrm{conv}}$ restricted to the convex clopen trees.

Remark. As mentioned above, the intermediate value theorem IVT is known to be constructively equivalent to weak König's lemma for convex trees, which is also Weihrauch equivalent to the convex choice XC_X for $X = 2^\omega$, where XC_X states that, given (a code of) a closed subset P of X, if P is convex, then P has an element; see [28, 26].

Finally, as a multifunction, *weak König's lemma for aou-trees*, $\mathsf{WKL}_{\mathrm{aou}}\colon\subseteq 2^{2^{<\omega}} \rightrightarrows 2^\omega$, is defined as follows:

$$\mathrm{dom}(\mathsf{WKL}_{\mathrm{aou}}) = \{T \subseteq 2^{<\omega} : T \text{ is an aou-tree}\},$$

$$\mathsf{WKL}_{\mathrm{aou}}(T) = [T] := \{p \in 2^\omega : p \text{ is an infinite path through } T\}.$$

Remark. As a multifunction, the robust division principle RDIV is Weihrauch equivalent to the *all-or-unique choice* AoUC (or equivalently, the *totalization of unique choice*), which takes, as an input (a code of), a closed subset P of Cantor space 2^ω such that either $P = X$ or P is a singleton, and then any element of P is a possible output; see [25]. This is, of course, Weihrauch equivalent to weak König's lemma for aou-trees.

4.1.1. *Implications.* As seen above, the principles introduced in Section 1.1.1 are written as some variants of weak König's lemma; see Figure 2. We show some nontrivial implications in Figure 2.

Proposition 4.2. $\mathsf{WKL}_{\mathrm{clop}} \to \mathsf{WKL}_{\mathrm{aou}}$.

Proof. Let T be an aou-tree. If T is found to be unique σ at height s, then let $H(T) = \{\tau : \tau \succ \sigma\}$, and for any infinite path p, define $K(p) = (p(0), p(1), \ldots, p(s-2), q(s-1), q(s), q(s+1), \ldots)$, where q is the unique infinite path through T. If it is not witnessed, i.e., $T = 2^{<\omega}$, then $H(T)$ becomes $2^{<\omega}$, and $K(p) = p$ for any infinite path p.

More formally, we define a basic clopen tree T^* and a function $k \colon \omega \times 2 \to 2$ as follows: First put the empty string into T^*. Given $s > 0$, if T contains all binary strings of length s then T^* contains all strings of length s, and define $k(s-1, i) = i$ for each $i < 2$. If T has only one binary string σ of length s but it is not true for all $t < s$, then σ is also the unique binary string of length s in T^*, and define $k(s-1, i) = \sigma(s-1)$ for each $i < 2$. Otherwise, for each $\tau \in T^*$ of length $s-1$, put $\tau 0$ and $\tau 1$ into T, and define $k(s-1, i) = \sigma(s-1)$ for each $i < 2$. Then, T^* is clearly a basic clopen tree. Therefore, by $\mathsf{WKL}_{\mathrm{clop}}$, the tree T^* has an infinite path p. Then, define $p^*(n) = k(n, p(n))$. Then, p^* is of the form $\langle p(0), p(1), \ldots, p(t-1), \sigma_t(t), \sigma_{t+1}(t+1), \ldots \rangle$, where σ_t is a string of length $t+1$ in T. It is easy to check that p^* is an infinite path through T. $\square$

Proposition 4.3. $\mathsf{WKL}_{\mathrm{clop}} \to \mathsf{WKL}_{=2}$.

Proof. Let T be a rational 2-tree. We define a basic clopen tree T^* as follows: For each $s > 0$, T has exactly two nodes ℓ_s, r_s of length s. For $s = 0$, let $\ell_s = r_s$ be the empty string. If both ℓ_s and r_s extend ℓ_{s-1} and r_{s-1}, respectively, then for each $\tau \in T$ of length $s-1$ put $\tau 0$ and $\tau 1$ into T. Otherwise, ℓ_s and r_s is of the form $\ell_s = b_s 0$ and $r_s = b_s 1$ for some b_s. Then, put both $\ell_s = b_s 0$ and $r_s = b_s 1$ into T^*. Clearly, T^* is the clopen tree of the form $T^* = \{\tau \in 2^\omega : \tau \succ b\}$, where b is the last branching node of T.

By $\mathsf{WKL}_{\mathrm{clop}}$, the tree T^* has an infinite path p. For each s, check if $p \restriction s + 1 \in T$ or not. If it is true, define $p^*(s) = p(s)$. Otherwise, if s is the least such number, then $p \restriction s \in T$, and there is no branching node above $p \restriction s$. Thus, for any $t \geq s$, there is the unique string $\sigma_t \in T$ of length t extending $p \restriction s$. Hence, for such σ_t, define $p^*(t) = \sigma_t(t)$. Then, it is easy to see that p^* is an infinite path through T. $\square$

Proposition 4.4. $\Sigma_2^0\text{-}\mathsf{DNE}_{\mathbb{R}} \to \mathsf{WKL}_{\mathrm{clop}}$.

Proof. Let T be a basic clopen tree. Then, we have the double negation of the existence of σ such that $\tau \in T$ for all $\tau \succeq \sigma$. By $\Sigma_2^0\text{-}\mathsf{DNE}_{\mathbb{R}}$, we get such a σ. Then, $\sigma^\frown \langle 0, 0, \ldots \rangle$ is an infinite path through T. $\square$

4.1.2. *Auxiliary principles.* As seen in Section 4.1, several mathematical principles can be described as weak variants of WKL. For instance, IVT, BE and RDIV are equivalent to $\mathsf{WKL}_{\mathrm{conv}}$, $\mathsf{WKL}_{\leq 2}$ and $\mathsf{WKL}_{\mathrm{aou}}$, respectively. Focusing on weak variants of WKL,

as we have already seen, there are principles such as $\mathsf{WKL}_{=2}$ and $\mathsf{WKL}_{\mathrm{clop}}$. They are well positioned as benchmarks for measuring the strength of principles. Let us look at how these additional principles relate to $\mathsf{IVT}, \mathsf{BE}$ and RDIV. The two principles introduced below may look weird, and we do not think it has any constructive meaning, but they are introduced solely as auxiliary principles to link the benchmark principles with mathematical principles such as IVT, BE and RDIV; see also Figures 1 and 2.

Binary expansion for regular Cauchy rationals $\mathsf{BE}_{\mathbb{Q}}$. We first consider the binary expansion principle $\mathsf{BE}_{\mathbb{Q}}$ for regular Cauchy rationals (more precisely, for non-irrationals), which states that, for any regular Cauchy real, if it happens to be a rational, then it has a binary expansion:

$$(\forall x \in [0,1]) \left[\neg(\forall a, b \in \mathbb{Z}, \ ax \neq b) \ \longrightarrow \ \exists f \colon \mathbb{N} \to \{0,1\}, \ x = \sum_{n=1}^{\infty} 2^{-f(n)} \right].$$

Intermediate value theorem for piecewise linear maps $\mathsf{IVT}_{\mathrm{lin}}$. Next, we consider a (weird) variant of the intermediate value theorem. A rational piecewise linear map on $\mathbb{R}$ is determined by finitely many rational points $\bar{p} = (p_0, \ldots, p_\ell)$ in the plane. Then, any continuous function on $\mathbb{R}$ can be represented as the limit of a sequence $(f_s)_{s \in \omega}$ of rational piecewise linear maps coded by $(\bar{p}^s)_{s \in \omega}$, where any p_i^{s+1} belongs to the 2^{-s-1}-neighborhood of the graph of f_s. That is, the sequence $(\bar{p}^s)_{s \in \omega}$ codes an approximation procedure of a continuous function f. Then, consider the case that a code $(\bar{p}^s)_{s \in \omega}$ of an approximation stabilizes, in the sense that $\bar{p}^s = \bar{p}^t$ for sufficiently large s, t. Then, of course, it converges to a rational piecewise linear map.

In other words, we represent a continuous function as usual, but we consider the case that such a continuous function *happens to be a rational piecewise linear map*. Formally, if f is coded by $(\bar{p}^s)_{s \in \omega}$ which does not change infinitely often (that is, $\neg\forall s \exists t > s \ \bar{p}^t \neq \bar{p}^s$) then we call f *a rational piecewise linear map as a discrete limit*. The principle $\mathsf{IVT}_{\mathrm{lin}}$ states that if $f \colon [0,1] \to \mathbb{R}$ is a rational piecewise linear map as a discrete limit such that $f(0) < 0 < f(1)$ then there is $x \in [0,1]$ such that $f(x) = 0$.

Proposition 4.5. $\mathsf{BE}_{\mathbb{Q}} \longleftrightarrow \mathsf{WKL}_{=2}$.

Proof. For the forward direction, from a given rational 2-tree T, one can easily construct a regular Cauchy real which happens to be a dyadic rational whose binary expansions are exactly infinite paths through T. For the backward direction, from a regular Cauchy real α, it is also easy to construct a 2-tree T whose infinite paths are binary expansions of α as usual. Although the binary expansion of a rational α is periodic, i.e., of the form $\sigma^\frown \tau^\frown \tau^\frown \tau^\frown \ldots$, this is insufficient for ensuring T to be rational, so we need to construct another rational 2-tree T^* from T.

Use ℓ_s^* to denote the leftmost node of length s in T^*, and r_s^* to denote the rightmost node. We inductively ensure that for any s, every node of length $s+1$ in T is a successor of a node of length s in T^*. Fix s. Let b_{s+1} be the last branching node of length $\leq s+1$ in T. Consider all decompositions $b_{s+1} = \sigma^\frown \tau^\frown \tau^\frown \ldots^\frown \tau^\frown (\tau \restriction j)$, and then take the shortest $\sigma^\frown \tau$. Put $t = 2|b_{s+1}| + 1$. If either $b_t = b_{s+1}$ or $\sigma^\frown \tau$ is updated from the previous one, then define $\ell_s^* = \ell_s$ and $r_s^* = r_s$. Otherwise, $b_{s+1} \neq b_t$, so either $\ell_{s+1} \preceq b_t$ or $r_{s+1} \preceq b_t$ holds. If the former holds then, by induction hypothesis, ℓ_{s+1} is a successor of u_s^*, where $u \in \{\ell, r\}$. Put $\bar{u} = r$ if $u = \ell$; otherwise $\bar{u} = \ell$. Then, define $u_{s+1}^* = \ell_{s+1}$

and $\overline{u}_{s+1}^* = \overline{u}_s^* 0$. This ensures that $\ell_{s+1}^* \preceq b_t \preceq \ell_t, r_t$. If $r_{s+1} \preceq b_t$ holds, replace ℓ_{s+1} with r_{s+1} and vice versa. This procedure adds no new branching node. By periodicity of a binary expansion, $\sigma^\frown\tau$ cannot be updated infinitely often. Moreover, there cannot be infinitely many different b_{s+1} such that $b_t = b_{s+1}$ happens. This is because $b_t = b_{s+1}$ implies that $\ell_t = b_{s+1} 01^{|b_{s+1}|}$ and $r_t = b_{s+1} 10^{|b_{s+1}|}$, which witnesses that there is no seed $\sigma^\frown\tau$ of periodicity of α in b_{s+1}. Hence, if there are infinitely many such b_{s+1}, then α cannot be periodic. Thus, T^* is a rational 2-tree.

Note that if T is already rational then $T^* = T$. Otherwise, T has infinitely many branching nodes, and thus, we check arbitrary long decompositions, so the last branching node of T^* is the first branching node in T after finding the correct seed $\sigma^\frown\tau$ of periodicity of α. In particular, any infinite path through T^* extends $\sigma^\frown\tau$. Now, given an infinite path p through T^*, for each s, check if $p \upharpoonright s \in T$. If s is the least number such that $p \upharpoonright s + 1 \notin T$ then T is not rational, and thus p extends the first branching node $b \in T$ after finding the correct seed $\sigma^\frown\tau$. In particular, we must have $\sigma^\frown\tau \preceq b \preceq p \upharpoonright s \in T$. Now, consider the all decompositions $p \upharpoonright s = \mu^\frown\nu^\frown\nu^\frown \ldots ^\frown\nu^\frown(\nu \upharpoonright j)$. There are only finitely many such decompositions, and if a decomposition is incorrect, it is witnessed after seeing T up to some finite height (which is effectively calculated from s). Thus, after checking such a height, we see that all surviving decompositions are equivalent. Therefore, a seed $\mu^\frown\nu$ of such a decomposition is actually equivalent to the correct seed $\sigma^\frown\tau$; hence $\mu^\frown\nu^\frown\nu^\frown\nu^\frown \ldots$ is an infinite path through T. $\square$

Proposition 4.6. $\mathsf{IVT}_{\mathrm{lin}} \longleftrightarrow \mathsf{WKL}_{\mathrm{clop}}$.

Proof. For the forward direction, one can assign an interval I_σ to each binary string σ in a standard manner: let $\tilde{\sigma} = \sigma[2/1]$ be the result of replacing every occurrence of 1 in σ with 2, and then the endpoints of I_σ are defined by $0.\tilde{\sigma}1$ and $0.\tilde{\sigma}2$ under the ternary expansions. Given a basic clopen tree $T = \{\tau : \tau \succeq \sigma\}$, it is easy to construct a piecewise linear map whose zeros are exactly I_σ.

For the backward direction, let (f_s) be a descrete approximation of a rational piecewise linear map f with $f(0) < 0 < f(1)$. Note that each f_s is determined by finitely many points in the plane. However, for each s, we only need to pay attention on at most four rational points $(x_i, f_s(x_i))_{i<4}$, where $x_0 \leq x_1 < x_2 \leq x_3$, and $-2^{-s} > f_s(x_0) \leq f_s(x_1) \leq 0 \leq f(x_2) \leq f(x_3) > 2^{-s}$. If $f(x_1) = f(x_2)$ then consider $I_s = [x_1, x_2]$. Otherwise, by linearity, one can compute a unique rational $x_1 < a_s < x_3$ such that $f(a_s) = 0$. Define I_s as a sufficiently small rational neighborhood of a_s, and one can ensure that (I_s) is a decreasing sequence.

Then, we define the tree T^* of binary expansions of elements of $\bigcap_n I_n$. If the rational interval I_s is updated because of $f(x_1) = f(x_2)$, then one can easily compute finitely many binary strings $\sigma_0, \ldots, \sigma_\ell$ of length s such that any binary expansion of a real in I_s extends some σ_i. Then the nodes of length s in T^* are exactly $\sigma_0, \ldots, \sigma_\ell$. If we currently guess $f(a_s) = 0$, then we proceed the argument in the proof of Proposition 4.5 to construct a basic clopen tree which determines a binary expansion of a_s.

Given an infinite path p through T^*, for each s, check if $p \upharpoonright s$ has an extension in I_s. If not, let s be the least number such that $p \upharpoonright s + 1 \notin T$. Note that if I_t is updated because of $f(x_1) = f(x_2)$ for some $t > s$ then we take some nodes $\sigma_0, \ldots, \sigma_\ell$ of length t and $p \upharpoonright s \preceq p \upharpoonright t = \sigma_i$ for some i, but σ_i extends to some element in $I_t \subseteq I_s$. Thus, this never happens. If we guess $f(a_s) = 0$, then as in the proof of Proposition 4.5 we

get a binary expansion of a_s, so a regular Cauchy representation of a_s. Otherwise, the current guess I_s is already a correct solution, so just take any extension of $p \restriction s$ which belongs to I_s. $\square$

4.2. Law of excluded middle as a Weihrauch problem.

Hereafter, we consider non-constructive principles $\Sigma_1^0\text{-DML}_\mathbb{R}$, $\Sigma_2^0\text{-DNE}_\mathbb{R}$, and $\Sigma_2^0\text{-DML}_\mathbb{R}$ as partial multifunctions. First, recall that de Morgan's law $\Sigma_1^0\text{-DML}_\mathbb{R}$ is formulated as LLPO.

As a multifunction, $\Sigma_2^0\text{-DNE}_\mathbb{R}$ is equivalent to the discrete limit operation Lim_ω (or equivalently, learnability with finite mind changes) in the Weihrauch context, where the discrete limit function Lim_ω is formalized as follows: Given a sequence $(a_i)_{i \in \mathbb{N}}$ of natural numbers, if $a_i = a_j$ holds for sufficiently large i, j, then Lim_ω returns the value of such an a_i. That is, Lim_ω is exactly the limit operation on the discrete space ω.

Proposition 4.7. $\Sigma_2^0\text{-DNE}_\mathbb{R} \equiv_\mathsf{W} \mathsf{Lim}_\omega$.

Proof. Assume that a sentence $\neg\neg\exists n \forall m R(n, m, \alpha)$ is given, where R is decidable. Search for the least n such that $\forall m R(n, m, \alpha)$ is true. Begin with $n = 0$, and if $\forall m R(n, m, \alpha)$ fails, then one can eventually find m such that $R(n, m, \alpha)$ by Markov's principle. If such an m is found, proceed this algorithm with $n+1$. If $\neg\neg\exists n \forall m R(n, m, \alpha)$ is true, then this algorithm eventually halts with the correct n such that $\forall m R(n, m, \alpha)$ is true. $\square$

First note that de Morgan's law $\Sigma_2^0\text{-DML}_\mathbb{R}$ for Σ_2^0 formulas is essentially equivalent to the pigeonhole principle on $\mathbb{N}$, or Ramsey's theorem RT_2^1 for singletons and two colors. Here, RT_2^1 (as a multifunction) states that, given a function $f \colon \mathbb{N} \to 2$ (which is called a 2-*coloring*), returns $h \colon \mathbb{N} \to 2$ such that f is constant on $H = \{n : h(n) = 1\}$, where H is called a *homogeneous set*.

Proposition 4.8. $\Sigma_2^0\text{-DML}_\mathbb{R} \equiv_\mathsf{W} \mathsf{RT}_2^1$.

Proof. For $\mathsf{RT}_2^1 \leq_\mathsf{W} \Sigma_2^0\text{-DML}_\mathbb{R}$, given a coloring $f \colon \mathbb{N} \to 2$, consider (the characteristic function of) the set $C_i = \{n : f(n) = i\}$. Either C_0 or C_1 is unbounded, so $\Sigma_2^0\text{-DML}_\mathbb{R}$ chooses an index j such that C_j is unbounded (where boundedness is a Σ_2^0 property). Then, C_j must be an infinite homogeneous set.

For $\Sigma_2^0\text{-DML}_\mathbb{R} \leq_\mathsf{W} \mathsf{RT}_2^1$, assume that Σ_2^0 formulas $\psi_i \equiv \exists a \forall b \varphi_i(a, b)$ are given. If $\psi_0 \wedge \psi_1$ is false, then by Markov's principle, for any a, either $\exists b \neg \varphi_0(a, b)$ or $\exists b \neg \varphi_1(a, b)$. Hence, given a, search for the least b such that either $\neg \varphi_0(a, b)$ or $\neg \varphi_1(a, b)$ holds. For such b, if $\varphi_0(a, b)$ fails, then put $f(a) = 0$; otherwise put $f(a) = 1$. By RT_2^1, we get an infinite homogeneous set H for f. Then given any $a \in H$, compute the value $f(a) = j$. This j is a solution to $\Sigma_2^0\text{-DML}_\mathbb{R}$ since for any $a \in H$ we have $\exists b \neg \varphi_j(a, b)$. $\square$

5. Main Proof

5.1. Weihrauch separations.

In order to prove Theorem 1.1, we will lift separation results on Weihrauch degrees to unprovability results over **IZF**. In this section, we formalize and prove statements on Weihrauch degrees which are needed to prove Theorem 1.1.

5.1.1. *The strength of choice for finite sets.* We first examine the strength of the game reinforcement of weak König's lemma for 2-trees, $(\mathsf{WKL}_{\leq 2})^{\supsetneq}$. It is not hard to see that $(\mathsf{WKL}_{\leq 2})^{\supsetneq}$ can be reduced to the finite parallelization $\mathsf{WKL}^*_{\leq 2}$ of $\mathsf{WKL}_{\leq 2}$; see Le Roux-Pauly [28] (where $\mathsf{WKL}_{\leq n} \star \mathsf{WKL}_{\leq m} \leq_{\mathrm{W}} \mathsf{WKL}_{\leq nm}$ is claimed) or Pauly-Tsuiki [39]: $\mathsf{C}^{\supsetneq}_{2^\omega, \#\leq 2} \equiv_{\mathsf{W}} \bigsqcup_n \mathsf{C}_{2^\omega, \#\leq n}$.

Fact 5.1. $(\mathsf{WKL}_{\leq 2})^{\supsetneq} \equiv_{\mathsf{W}} (\mathsf{WKL}_{\leq 2})^*$.

We first show that $(\mathsf{WKL}_{\leq 2})^{\supsetneq}$ is not strong enough for solving weak König's lemma $\mathsf{WKL}_{\mathrm{aou}}$ for all-or-unique trees:

Proposition 5.2. $\mathsf{WKL}_{\mathrm{aou}} \not\leq^{\mathrm{c}}_{\mathsf{W}} (\mathsf{WKL}_{\leq 2})^{\supsetneq}$.

Proof. By Fact 5.1, it suffices to show that $\mathsf{WKL}_{\mathrm{aou}} \not\leq_{\mathsf{W}} (\mathsf{WKL}_{\leq 2})^*$. We now consider the following play:

$$
\begin{array}{llll}
\text{I:} & T & & x = \langle x_0, \ldots, x_n \rangle \\
\text{II:} & & \langle S_0, \ldots, S_n \rangle & \alpha_x
\end{array}
$$

Player II tries to construct a sequence $S = \langle S_0, \ldots, S_n \rangle$ of 2-trees, and a path α_x through I's aou-tree T, where the second move α_x depends on Player I's second move x. As usual, each partial continuous function f is coded as an element p of $(\omega \cup \{\bot\})^\omega$, so one can read a finite portion $p[s] = \langle p(0), p(1), \ldots, p(s-1) \rangle$ of p by stage s. More precisely, $p(s) = \langle n, m \rangle$ indicates that we obtain the information $f(n) \downarrow = m$ at stage $s+1$, but if $p(s) = \bot$ we get no information.

By *recursion trick* (Section 3.1.3), it is sufficient to describe an algorithm constructing an aou-tree T from given S, α such that Player I wins along the above play for some x. Now, the algorithm is given as follows:

(1) At each stage s, either T contains all binary strings of length s or T only has a single node of length s. If we do not act at stage s (while our algorithm works), we always assume that T contains all binary strings of length $s+1$.

(2) Wait for n (in Player II's first move) being determined. Then wait for $\alpha_x \upharpoonright n+2$ being defined for any correct x; that is, any x with $x_i \in [S_i]$. By compactness, if it happens, it is witnessed by some finite stage.

(3) As we consider $n+1$ many 2-trees $S_0, \ldots, S_n$, there are at most 2^{n+1} many correct x's, so there are at most 2^{n+1} many α_x's. Clearly, there are 2^{n+2} nodes of length $n+2$, so choose a binary string σ of length $n+2$ which is different from any $\alpha_x \upharpoonright n+2$. Then, we declare that $T = \{\sigma^\frown 0^\omega\}$, and our algorithm halts.

Note that if the procedure arrives at (3) then α_x cannot be a path through T. Otherwise, the algorithm waits at (2) forever, which means that α_x is not total for some correct x, and by (1) we have $T = 2^{<\omega}$, so Player I obeys the rule.

To apply recursion trick, even if Player II violates the rule, the first move T of Player I needs to obey the rule. To ensure this, one may assume that we proceed the step (3) in our algorithm only if Player II's partial trees look like 2-trees. Moreover, our algorithm performs some action on T only once; hence, in any case, T must be an aou-tree. Hence, Player I wins. $\square$

5.1.2. *The strength of all-of-unique choice.* We next examine the strength of the game reinforcement of weak König's lemma for all-or-unique trees, $(\mathsf{WKL}_{\mathrm{aou}})^{\supsetneq}$. Kihara-Pauly

[25] showed that the hierarchy of compositional products of the finite parallelization AoUC^* of the all-or-unique choice AoUC collapses after the second level (where recall that AoUC is Weihrauch equivalent to $\mathsf{WKL}_{\mathrm{aou}}$). In particular, we have the following:

Fact 5.3. $(\mathsf{WKL}_{\mathrm{aou}})^{\supsetneq} \equiv_{\mathsf{W}} (\mathsf{WKL}_{\mathrm{aou}})^* \star (\mathsf{WKL}_{\mathrm{aou}})^*$

We first verify that $(\mathsf{WKL}_{\mathrm{aou}})^{\supsetneq}$ is not strong enough for solving weak König's lemma $\mathsf{WKL}_{=2}$ for rational 2-trees:

Proposition 5.4. $\mathsf{WKL}_{=2} \not\leq_{\mathsf{W}}^{c} (\mathsf{WKL}_{\mathrm{aou}})^{\supsetneq}$.

Proof. By Kihara-Pauly [25], it suffices to show that $\mathsf{WKL}_{\leq 2} \not\leq_{\mathsf{W}} (\mathsf{WKL}_{\mathrm{aou}})^* \star (\mathsf{WKL}_{\mathrm{aou}})^*$. We now consider the following play:

$$
\begin{array}{llll}
\text{I:} & T & x = \langle x_0, \ldots, x_n \rangle & y = \langle y_0, \ldots, y_{\ell(x)} \rangle \\
\text{II:} & \langle S_0, \ldots, S_n \rangle & \langle R_x^0, \ldots, R_x^{\ell(x)} \rangle & \alpha_{x,y}
\end{array}
$$

Player II tries to construct two sequences $S = \langle S_0, \ldots, S_n \rangle$ and $R(x) = \langle R_x^0, \ldots, R_x^{\ell(x)} \rangle$ of aou-trees, and a path $\alpha_{x,y}$ through I's rational 2-tree T, where the second move $R(x)$ depends on Player I's second move x, and the third move $\alpha_{x,y}$ depends on Player I's second and third moves (x, y). As in Proposition 5.2, each partial continuous function f is coded as an element p of $(\omega \cup \{\bot\})^\omega$, so one can read a finite portion $p[s] = \langle p(0), p(1), \ldots, p(s-1) \rangle$ of p by stage s.

By *recursion trick* (Section 3.1.3), it is sufficient to describe an algorithm constructing a rational 2-tree T from given S, R, α such that Player I wins along the above play for some x and y. Now, the algorithm is given as follows:

(1) At each stage s, T has two nodes ℓ_s, r_s of length s, where ℓ_s is left to r_s. Let $v_s = \ell_s \wedge r_s$ be the current branching node of T. If we do not act at stage s, we always put $\ell_{s+1} = \ell_s 1$ and $r_{s+1} = r_s 0$.

(2) Wait for $\alpha_{x,y}(|v_s|)$ being defined for any correct x and y; that is, any (x, y) with $x_i \in [S_i]$ and $y_j \in [R_x^j]$. By compactness, if it happens, it is witnessed by some finite stage.

(3) Suppose that it happens at stage s. If $\alpha_{x,y}(|v_s|) = 0$ for some correct x and y then $\alpha_{x,y}$ is incomparable with r_s, so we put $\ell_{s+1} = r_s 0$ and $r_{s+1} = r_s 1$. Otherwise, $\alpha_{x,y}$ for any correct x and y is incomparable with ℓ_s, so we put $\ell_{s+1} = \ell_s 0$ and $r_{s+1} = r_s 1$. In any case, this action ensures that $\alpha_{x,y}$ for some correct x and y is not a path through T.

(4) In order for Player II to win, II needs to make (x, y) incorrect; that is, II needs to remove either x_i from S_i for some i or y_j from R_x^j for some j.

(5) If Player II decided to remove x_i from S_i, then this action forces S_i to have at most one path as S_i is aou-tree. If it happens, go back to (2). Note that we can arrive at (5) at most $n + 1$ times, since we only have $n + 1$ trees $S_0, \ldots, S_n$.

(6) If Player II decided to remove y_j from R_x^j, then this action also forces R_x^j to have at most one path as R_x^j is aou-tree. By continuity of R, there is a clopen neighborhood C of x such that R_z^j has a single path for any $z \in [\sigma]$. We may assume that ℓ takes a constant value c on C by making such a neighborhood C sufficiently small. Then we now go to the following $(2')$ with this C.

(2′) Wait for $\alpha_{x,y}(v_s)$ being defined for any correct $x \in C$ and y; that is, any (x, y) with $x_i \in [S_i] \cap C$ and $y_j \in [R_x^j]$. By compactness, if it happens, it is witnessed by some finite stage.

(3′) Suppose that it happens at stage s. If $\alpha_{x,y}(|\sigma|) = 0$ for some correct $x \in U$ and y then $\alpha_{x,y}$ is incomparable with r_s, so we put $\ell_{s+1} = r_s 0$ and $r_{s+1} = r_s 1$. Otherwise, $\alpha_{x,y}$ for any correct x and y is incomparable with ℓ_s, so we put $\ell_{s+1} = \ell_s 0$ and $r_{s+1} = r_s 1$. In any case, this action ensures that $\alpha_{x,y}$ for some correct x and y is not a path through T.

(7) As before, in order for Player II to win, II needs to make (x, y) incorrect; that is, II needs to remove either x_i from $S_i \cap C$ for some i or y_j from R_x^j for some j. In other words, Player II chooses either (5) or (6′) below at the next step. Here, after going to (5), the parameter C is initialized.

(6′) The action of this step is the same as (6), but a clopen neighborhood C' of x is chosen as a subset of C. The action of (6′) forces some R_z^j to have at most one path for any $z \in C'$. Then go back to (2′). Note that, as the value c of ℓ on C is determined at the step (6), we only have $c + 1$ many trees $T_z^0, \ldots, T_z^c$ for any $z \in C$, so we can arrive at (6′) at most c many times unless going to (5).

Consequently, the procedure arrives (5) or (6′) at most finitely often. This means that Player I eventually defeats Player II; that is, either $\alpha_{x,y}$ is not a path through T for some x, y or else Player II violates the rule by making something which is not an aou-tree.

To apply recursion trick, even if Player II violates the rule, the first move T of Player I needs to obey the rule. To ensure this, one may assume that we proceed our algorithm at stage s only if Player II's partial trees look like aou-trees at all levels below s. Then if Player II violates the rule then our algorithm performs only finitely many actions, so T must be a rational 2-tree. $\qquad \square$

5.1.3. *The strength of their product.* We now consider the product $\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{aou}}$ of choice for finite sets and all-or-unique choice. To examine the strength of $(\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{aou}})^{\circlearrowleft}$, we consider the following compactness argument:

In general, to show either $F \leq_{\mathsf{W}} G^{\circlearrowleft}$ or $F \not\leq_{\mathsf{W}} G^{\circlearrowleft}$, we need to consider a reduction game of arbitrary length. However, if everything involved in this game is compact, then one can pre-determine a bound of the length of a given game. To explain more details, let $G(y) \subseteq 2^\omega$ be compact uniformly in y. For instance, one can take $\mathsf{WKL}_{\leq 2}$, $\mathsf{WKL}_{\mathrm{aou}}$, $\mathsf{WKL}_{\mathrm{conv}}$, and WKL. These examples are actually effectively compact. Then, consider the reduction game $\mathfrak{G}(F, G)$:

$$
\begin{array}{llllll}
\text{I:} & x_0 & & x_1 & & x_2 & & \ldots \\
\text{II:} & & \langle j_0, y_0 \rangle & & \langle j_1, y_1 \rangle & & \langle j_2, y_2 \rangle & \ldots
\end{array}
$$

Assume that Player II has a winning strategy τ, which yields continuous functions $\tilde{j}$ and $\tilde{y}$ such that $j_n = \tilde{j}(x_0, x_1, \ldots, x_n)$ and $y_n = \tilde{y}(x_0, x_1, \ldots, x_n)$. Now, consider the set $P_\tau(x_0)$ of Player I's all plays $x = (x_1, \ldots)$ (where the first move x_0 is fixed) against II's strategy τ, where x obeys the rule (and $x_n \in 2^\omega$ is arbitrary if $j_m = 1$ for some $m < n$; that is, Player II has already declared victory). More precisely, consider

$$
P_\tau(x_0) = \{x \in (2^\omega)^\omega : x_{n+1} \in g(\tilde{y}(x_0, \ldots, x_n)) \text{ or } (\exists m < n)\, \tilde{j}(x_0, \ldots, x_m) = 1\}.
$$

By continuity of $\tilde{j}$ and $\tilde{y}$, and compactness of $g(y)$, the set $P_\tau(x_0)$ is compact. Hence, again by continuity of $\tilde{j}$ and compactness of $P_\tau(x_0)$, one can easily verify that there is a bound n of the length of any play; that is, for any $x = (x_i) \in P_\tau(x_0)$ there is m such that $\tilde{j}(x_0, \ldots, x_m) = 1$. Moreover, $x_0 \mapsto n$ is continuous. If the strategy τ is computable, and $g(y)$ is effectively compact uniformly in y, then $P_\tau(x_0)$ is also effectively compact, and moreover, $x_0 \mapsto n$ is computable. This concludes that

$$F \leq_{\mathsf{W}} G^{\supset} \implies F \leq_{\mathsf{W}} \bigsqcup_n G^{(n)},$$

where $G^{(n)}$ is the nth iteration of the compositional product; that is, $G^{(1)} = G$ and $G^{(n+1)} = G \star G^{(n)}$. Furthermore, if $\mathrm{dom}(F)$ is effectively compact, one can also ensure that $F \leq_{\mathsf{W}} \bigsqcup_n G^{(n)}$ implies $F \leq_{\mathsf{W}} G^{(n)}$ for some $n \in \omega$. By the above argument, for instance, one can ensure the following:

Observation 5.5. $(\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{aou}})^{\supset} \equiv_{\mathsf{W}} \bigsqcup_n (\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{aou}})^{(n)}$.

Then we show that $(\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{aou}})^{\supset}$ is not strong enough for solving $\mathsf{WKL}_{\mathrm{clop}}$, weak König's lemma for clopen convex trees.

Proposition 5.6. $\mathsf{WKL}_{\mathrm{clop}} \not\leq_{\mathsf{W}}^c (\mathsf{WKL}_{\mathrm{aou}} \times \mathsf{WKL}_{\leq 2})^{\supset}$.

Proof. By Observation 5.5, it suffices to show that $\mathsf{WKL}_{\mathrm{clop}} \not\leq_{\mathsf{W}} \bigsqcup_n (\mathsf{WKL}_{\mathrm{aou}} \times \mathsf{WKL}_{\leq 2})^{(n)}$. We now consider the following play:

I: T $x(1) = \langle x_A^1, x_B^1 \rangle$ $\ldots$ $x(n) = \langle x_A^n, x_B^n \rangle$

II: $n, \langle A, B \rangle$ $\langle A_{x(1)}, B_{x(1)} \rangle$ $\ldots$ $\alpha_{x(1)\ldots x(n)}$

Player II tries to construct a sequence of pairs of aou-trees $A_\bullet$ and 2-trees $B_\bullet$, and a path $\alpha_{x(1)\ldots x(n)}$ through a basic clopen tree T, where the ith move $\langle A_{x(1)\ldots x(i-1)}, B_{x(1)\ldots x(i-1)} \rangle$ depends on Player I's moves $x(1), \ldots, x(i-1)$. As in Proposition 5.2, each partial continuous function f is coded as an element p of $(\omega \cup \{\bot\})^\omega$, so one can read a finite portion $p[s] = \langle p(0), p(1), \ldots, p(s-1) \rangle$ of p by stage s.

By *recursion trick* (Section 3.1.3), it is sufficient to describe an algorithm constructing a basic clopen tree T from given $n, \langle A_\bullet, B_\bullet \rangle, \alpha$ such that Player I wins along the above play for some $x(1), \ldots, x(n)$. Now, the algorithm is given as follows:

(1) At each stage s, there is a binary string σ such that T contains all binary strings of length s extending σ. If we do not act at stage s, we always assume that T contains all binary strings of length $s + 1$ extending σ.

(2) Wait for n (in Player II's first move) being determined. Then wait for $\alpha_{x(1)\ldots x(n)} \restriction n + 2$ being defined for any correct $x(1), \ldots, x(n)$, which means that $x_A^{i+1} \in A_{x(1)\ldots x(i)}$ and $x_B^{i+1} \in B_{x(1)\ldots x(i)}$. By compactness, if it happens, it is witnessed by some finite stage.

(3) For any i, there are two possibilities for x_B^i for a given $B_{x(i-1)}$ since $B_\bullet$ is a 2-tree. So, we consider a non-deterministic play of Player I, which takes both possibilities for x_B^i, but for x_A^i chooses just one element. In other words, we consider (at most) 2^n many $x = x(1)\ldots x(n)$'s, so there are at most 2^{n+1} many α_x's. Clearly, there are 2^{n+2} nodes of length $n + 2$, so choose a binary string σ of length $n + 2$ which is different from any $\alpha_x \restriction n + 2$. Then, we declare that T only contains binary strings of length $s + 1$ extending σ at stage $s + 1$. This ensures that α_x is not a path through T.

(4) In order for Player II to win, II needs to remove x_A^i from $A_{x(1)\ldots x(i-1)}$ for some i. This action forces $A_{x(1)\ldots x(i-1)}$ to have at most one path as $A_\bullet$ is an aou-tree. Player I re-chooses x_A^i as a unique element in $A_{x(1)\ldots x(i-1)}$, and go back to (2) with $k(n+2)$ instead of $n+2$, where k is the number of times which we have reached the step (4) so far.

When we arrive at (4), Player II need to declare that some of $A_\bullet$ has at most one path. Thus, it is easy to ensure that the algorithm can arrive at (4) at most finitely often. Hence, x_A^i is eventually stabilized for any i, and by our action at (3), this ensures that α_x cannot be a path through T. The string σ is also eventually stabilized, and thus T becomes a clopen tree. Thus, Player I obeys the rule.

To apply recursion trick, even if Player II violates the rule, the first move T of Player I needs to obey the rule. To ensure this, one may assume that we proceed our algorithm at stage s only if Player II's partial trees look like pairs of aou-trees and 2-trees at all levels below s. Then if Player II violates the rule then our algorithm performs only finitely many actions, so T must be a basic clopen tree. In any case, Player I obeys the rule, and so Player I defeats Player II. $\qquad\square$

5.1.4. *The product of de Morgan's law and the double negation elimination.* We now examine the strength of $(\mathsf{RT}_2^1 \times \mathsf{Lim}_\omega)^\circleddash$, the game closure of the product of Ramsey's theorem for singletons and two-colors (the infinite pigeonhole principle) and the discrete limit operation. We say that a partial finite-valued function $F\colon \subseteq \omega^\omega \rightrightarrows \omega^\omega$ has a *semicontinuous bound* if there is a partial continuous function $g\colon \subseteq \omega^\omega \times \omega \to \omega^\omega$ such that $F(x) \subseteq \{g(x,n) : n < b(x)\}$ for any $x \in \mathrm{dom}(F)$, where $b\colon \subseteq \omega^\omega \to \omega$ is lower semicontinuous.

Lemma 5.7. $(\mathsf{RT}_2^1 \times \mathsf{Lim}_\omega)^\circleddash$ *has a semicontinuous bound.*

Proof. Let (x,τ) be an instance of $(\mathsf{RT}_2^1 \times \mathsf{Lim}_\omega)^\circleddash$. Consider a play of a reduction game according to II's strategy τ:

I: x $\sigma(0) = (j_0, k_0)$ $\sigma(1) = (j_1, k_1)$ $\ldots$
II: (c, α) $(c_{\sigma(0)}, \alpha_{\sigma(0)})$ $(c_{\sigma(0)\sigma(1)}, \alpha_{\sigma(0)\sigma(1)})$ $\ldots$

Here, Player II's moves are automatically generated from the strategy τ. For each round, Player I needs to answer a query c_* to RT_2^1 and a query α_* to Lim_ω made by Player II. As an answer to the RT_2^1-query c_* is either 0 or 1, and to the Lim_ω-query α_* is a natural number, one may assume that Player I's moves are restricted to $2 \times \omega$. We define the *tree T of Player I's possible plays* as $T = (2 \times \omega)^{<\omega}$.

We view each node $\sigma \in T$ as a sequence such that $j(n) \in 2$ and $k(n) \in \omega$ alternatively appear (i.e., $\sigma(2n) \in \{0,1\}$ and $\sigma(2n+1) \in \omega$) rather than a sequence of pairs from $2 \times \omega$. Then, a node of T of even length is called a RT_2^1-*node*, and a node of odd length is called a Lim_ω-*node*. The strategy τ automatically assigns a query c_σ to each RT_2^1-node $\sigma \in T_s$, where $c_\sigma\colon \mathbb{N} \to 2$ is a two-coloring. Then, each RT_2^1-node σ has at most two immediate successors $\sigma 0, \sigma 1$, and moving to the node σi indicates that Player I chose i as an answer to c_σ. Similarly, the strategy τ automatically assigns a query α_σ to each Lim_ω-node $\sigma \in T_s$, where $\alpha_\sigma \in \omega^\omega$ is a (convergent) sequence of natural numbers.

As mentioned above, Player I's next move is restricted to ω (whenever I obeys the rule), but the uniqueness of the limit, there is an exactly one correct answer. Since Lim_ω

is the discrete limit operation, if Player II obeys the rule, then α_σ must stabilize, that is, $\alpha_\sigma(s) = \alpha_\sigma(t)$ for any sufficiently large s, t, and this value is the correct answer to α_σ. Thus, if $\alpha_\sigma(s) = k$, we say that k is the *current true outcome* of the Lim_ω-node σ at stage s, and only the edge $[\sigma, \sigma k]$ is *passable* at this stage.

Similarly, for a RT^1_2-node, we declare that the edge $[\sigma, \sigma i]$ is *passable* if i looks like a correct answer (which means that $c_\sigma(n) = i$ for infinitely many n). More precisely, the edge $[\sigma, \sigma i]$ is *passable at stage s* if $c_\sigma(t_\sigma(s)) = i$, where, each node σ has its own clock t_σ, and it moves only when σ is accessible, where we say that σ is *accessible at stage s* if $|\sigma| < s$ and all edges in the path $[\langle\rangle, \sigma]$ are passable at stage s (where $\langle\rangle$ is the root of T). Note that there are only finitely many accessible nodes at each stage.

If Player I obeys the rule, then, as (x, τ) is an instance of $(\mathsf{RT}^1_2 \times \mathsf{Lim}_\omega)^\supset$, Player II (according to the strategy τ) declares victory at some round. This means that if we restrict the tree T to the nodes σ at which Player I obeys the rule (i.e., Player I gives a correct answer to each query made by Player II), then the restriction R forms a well-founded binary tree; that is, a finite binary tree. Moreover, any $\sigma \notin R$ is never accessible after some stage. Indeed, if $\sigma \in R$ and $\sigma i \notin R$, then the edge $[\sigma, \sigma i]$ is never passable after some stage, and there are only finite such nodes as R is finite.

Let A_s be the finite tree of all accessible nodes at stage s. The above argument shows that for a sufficiently large t, we always have $A_t \subseteq R$; hence $A = \bigcup_s A_s$ is finite. We also note that if $\sigma \in R$ then there are infinitely many t such that $\sigma \in A_t$. At stage s, for each $\sigma \in A_s$, check if Player II (according to the strategy τ) declares victory with some value u_σ at the next round. If so, enumerate u_σ into an auxiliary set B. Then we get an (x, τ)-computable increasing sequence $(B_s)_{s \in \omega}$ of finite sets such that $B = \bigcup_s B_s$ is finite, and contains all possible outputs u_σ of $(\mathsf{RT}^1_2 \times \mathsf{Lim}_\omega)^\supset(x, \tau)$. $\qquad\square$

We apply the above lemma to show that $(\mathsf{RT}^1_2 \times \mathsf{Lim}_\omega)^\supset$ is not strong enough for solving weak König's lemma for 2-trees:

Proposition 5.8. *If $F: \subseteq \omega^\omega \rightrightarrows \omega$ has a semi-continuous bound, then $\mathsf{WKL}_{\leq 2} \not\leq^c_{\mathsf{W}} F$. In particular, $\mathsf{WKL}_{\leq 2} \not\leq^c_{\mathsf{W}} (\mathsf{RT}^1_2 \times \mathsf{Lim}_\omega)^\supset$.*

Proof. If $\mathsf{WKL}_{\leq 2} \leq^c_{\mathsf{W}} F$, where F has a semi-continuous bound $(\tilde{g}, \tilde{b})$, then there are continuous functions H, K such that if $\tilde{T}$ is a 2-tree then, there is $n < b := \tilde{b}(H(\tilde{T}))$ such that $\tilde{g}(H(\tilde{T}), n)$ is defined, and $g(n) := K(\tilde{T}, \tilde{g}(H(\tilde{T}), n))$ is a path through $\tilde{T}$. If we describe an algorithm constructing T from (g, b) (with a parameter $\tilde{T}$), by Kleene's recursion theorem, $\tilde{T}$ can be interpreted as self-reference; that is, we may assume that $T = \tilde{T}$. This is the simplest version of *recursion trick* (Section 3.1.3).

Thus, it is sufficient to describe an algorithm constructing a 2-tree T from given (g, b):

(1) First we declare that each $n \in \mathbb{N}$ is active. At each stage s, T has two nodes ℓ_s, r_s of length s, where ℓ_s is left to r_s. Let $v_s = \ell_s \wedge r_s$ be the current branching node of T. If we do not act at stage s, we always put $\ell_{s+1} = \ell_s 1$ and $r_{s+1} = r_s 0$.

(2) Then, ask if $g(n)(|v_s|)$ is already defined by stage s for some active $n < b_s$, where b_s is the stage s approximation of b.

(3) If such n exists, then the least such n receives attention. For this n, if $g(n)(|v_s|) = 0$, then $g(n)$ is incomparable with the rightmost node r_s, so we put $\ell_{s+1} = r_s 0$ and $r_{s+1} = r_s 1$. Similarly, if $g(n)(|v_s|) = 1$, then $g(n)$ is incomparable with the leftmost node ℓ_s, so we put $\ell_{s+1} = r_s 0$ and $r_{s+1} = r_s 1$. This procedure ensures

that $g(n)$ is incomparable with ℓ_{s+1} and r_{s+1}. Then, we declare that n is not active anymore.

Assume that Player II obeys the rule, i.e., (b_s) converges to a finite value b. Then, since there are only finitely many n such that $g(n)$ is total and $n < b$, such n receives attention at some stage s. By our strategy described above, for any such n, if p is a path through T then we must have $g(n) \neq p$. This contradicts our assumption that $g(n)$ is a path through T for some $n < b$.

To apply recursion trick, even if Player II violates the rule (i.e., (b_s) does not converge), the first move T of Player I needs to obey the rule. In this case, our algorithm may arrive at (3) infinitely often, but this just implies that T converges to a tree which has a unique path. In any case, T is a 2-tree. Hence, we have $\mathsf{WKL}_{\leq 2} \not\leq^c_\mathsf{W} F$. $\qquad\square$

5.1.5. *More products.* We finally examine the strength of the game closure of $\mathsf{RT}^1_2 \times \mathsf{Lim}_\omega \times \mathsf{WKL}_{\leq 2}$. We show that it is not strong enough for solving weak König's lemma $\mathsf{WKL}_\mathrm{conv}$ for convex trees.

Proposition 5.9. $\mathsf{WKL}_\mathrm{conv} \not\leq^c_\mathsf{W} (\mathsf{RT}^1_2 \times \mathsf{Lim}_\omega \times \mathsf{WKL}_{\leq 2})^\supset$.

Proof. We now consider the following play:

I:	T		$\sigma(0) = (j_0, k_0, x_0)$		$\sigma(1) = (j_1, k_1, x_1)$	...
II:		(c, α, B)		$(c_{\sigma(0)}, \alpha_{\sigma(0)}, B_{\sigma(0)})$		...

Again, by *recursion trick* (Section 3.1.3), it is sufficient to describe an algorithm constructing a basic clopen tree T from Player II's given strategy $c_\bullet, \alpha_\bullet, B_\bullet, \dots$ such that Player I wins along the above play for some $j_0, k_0, x_0, j_1, k_1, x_1, \dots$ The difficulty here is again lack of compactness.

Recall the tree of I's possible plays from Proposition 5.7. In the present game, each node of our tree essentially codes Player I's play $\langle j_0, k_0, x_0, j_1, k_1, x_1, \dots \rangle$; however, now our tree is slightly different from the previous one. The first entry is $j = j_0$, which is an answer to the query c, and the second entry is $k = k_0$, which codes an answer to the query α (assigned to the node $\langle j \rangle$).

However, although the 2-tree B (assigned to the node $\langle jk \rangle$) has at most two paths, we have a lot of candidates for the answer $x(n) \in 2^\omega$. We assign infinitely many possible outcomes $(\infty, 0, 1, \dots)$ to the node $\langle jk \rangle$. The outcome ∞ indicates that B has at most one path, and the outcome n indicates that B_j has exactly two paths which branch at height n. Then we consider a non-deterministic play of Player I, which means that I chooses both paths x, x' through B whenever $\langle jk \rangle$ has finite outcome. Thus, the next entry of Player I's move may be a pair $\langle j_1, j'_1 \rangle$ of answers to two different queries c_{jkx} and $c_{jkx'}$. Then the tree is constructed by continuing this procedure.

We call a node of length $3n$ an RT^1_2-*node*, a node of length $3n + 1$ an Lim_ω-*node*, and a node of length $3n + 2$ a $\mathsf{WKL}_{\leq 2}$-*node*. Note that an RT^1_2-node has at most finitely many immediate successors, while a Lim_ω-node and a $\mathsf{WKL}_{\leq 2}$-node has infinitely many immediate successors. In general, many queries are assigned to each node. For each query on a RT^1_2-node, the outcomes are defined as in the proof of Proposition 5.8. For each query α on a Lim_ω-node, we assign infinitely many possible finite outcomes $n \in \mathbb{N}$ to this query, where the outcome n indicates that α stabilizes after n; that is, n is the least number such that $\alpha(k) = \alpha(n)$ for any $k \geq n$ (which is slightly different from

Proposition 5.8). The outcomes of each query B on a $\mathsf{WKL}_{\leq 2}$-node are the same as described as above: The outcome ∞ indicates that B_j has at most one path, and the outcome n indicates that B has exactly two paths which branch at height n. Then we consider a non-deterministic play of Player I, which means that I chooses both paths through B whenever the $\mathsf{WKL}_{\leq 2}$-query B has finite outcome. An outcome of a node σ is a sequence of outcomes of queries assigned to σ.

As in Proposition 5.7, each node σ has its own clock t_σ. Each $\mathsf{WKL}_{\leq 2}$-query B (which is assigned to a $\mathsf{WKL}_{\leq 2}$-node σ) has a unique *current true outcome* at stage $t = t_\sigma(s)$ in the following sense: The 2-tree B has exactly two strings ℓ_t and r_t of length t. If ℓ_t extends ℓ_{t-1} and r_t extends r_{t-1}, then the finite outcome n is currently true, where n is the height where ℓ_t and r_t branch; otherwise ∞ is currently true. Although the current true outcome is unique for each $\mathsf{WKL}_{\leq 2}$-query, the immediate successor may involve two RT^1_2-queries corresponding to II's responses to I's possible moves ℓ_t and r_t. In this sense, our tree consists of possible *non-deterministic* plays. Similarly, each Lim_ω-query α (which is assigned to a Lim_ω-node σ) has a unique current true outcome at stage $t = t_\sigma(s)$ in the following sense: The outcome n is currently true if α stabilizes between stages n and t; that is, n is the least number such that $\alpha(k) = \alpha(n)$ whenever $n \leq k \leq t$. Then, Player I's corresponding move to the outcome n is $\alpha(n)$. Thus, Lim_ω-moves are deterministic.

A *current true multi-path* at stage s is a finite string of length s defined as follows: At an RT^1_2-node, choose a passable edge for each assigned RT^1_2-query (where an RT^1_2-query can have two passable edges). At a $\mathsf{WKL}_{\leq 2}$-node, choose a (unique) current true outcome for each assigned $\mathsf{WKL}_{\leq 2}$-query. Recall that we consider a non-deterministic play, so many queries may be assigned to each node, and we need to choose a passable edge and a current true outcome for each query. Therefore, there may be a lot of (but finitely many) current true multi-paths.

A *true multi-path* is defined as follows: At an RT^1_2-node, choose correct answers to the assigned RT^1_2-queries. At a Lim_ω-node, choose the unique true outcome for each assigned query. Similarly, at a $\mathsf{WKL}_{\leq 2}$-node, choose true outcomes; i.e., n if an assigned $\mathsf{WKL}_{\leq 2}$-query B has two nodes which branch at height n; otherwise ∞. Note that a true multi-path α is non-deterministic, but two-branching, and every single path γ through α yields Player I's play which obeys the rule (Note also that there are many true multi-paths because each RT^1_2-query may have two true outcomes, and every true multi-path contains many paths because each true $\mathsf{WKL}_{\leq 2}$-outcome may yield two moves). As Player I obeys the rule along γ, if Player II wins, then Player II declares victory along γ. This means that a true multi-path α is well-founded; hence, finite as α is binary. The finiteness of a true multi-path α ensures that α becomes a current true multi-path at infinitely many stages.

Now, the algorithm constructing a basic clopen tree T is given as follows:

(1) At each stage s, there is a binary string σ such that T contains all binary strings of length s extending σ. If we do not act at stage s, we always assume that T contains all binary strings of length $s + 1$ extending σ.
(2) Wait for some stage when Player II declares victory with $(u_i)_{i \in I}$ along all plays $(\gamma_i)_{i \in I}$ on a current true path, and $u_i(|\sigma| + |I|)$ is defined for any $i \in I$.

(3) If there is such a current true path α, then choose τ extending σ such that $\tau(|\sigma| + |I|) \neq u_i(|\sigma| + |I|)$ for any $i \in I$. Then we declare that, at stage $s + 1$, all binary strings in T of length $s + 1$ extend τ. This action ensures that u_i is not a path through T for any $i \in I$.

(4) If Player II wins, for each single path γ of α, some outcome in γ must be incorrect. Assume that some $\mathsf{WKL}_{\leq 2}$-outcome is incorrect. If a node $\sigma \prec \gamma$ made an incorrect outcome n, the corresponding $\mathsf{WKL}_{\leq 2}$-query B has ℓ_s and r_s which branch at length n, but at some later stage $t > s$, either $\ell_t, r_t \succ \ell_s$ or $\ell_t, r_t \succ r_s$. However, the multi-path after σ is automatically generated from ℓ_s, r_s by definition, so ℓ_t, r_t yield a sub-multi-path of α. A similar argument holds for ∞. Thus, some RT_2^1- or Lim_ω-outcome has to be incorrect.

In (4), as in the proof of Proposition 5.7, accessible nodes are eventually contained in the positions where Player I obeys the rule. This means that all RT_2^1- and Lim_ω-outcomes along a current true multi-path are correct after some stage. Therefore, Player II cannot win.

To apply recursion trick, even if Player II violates the rule, the first move T of Player I needs to obey the rule. In this case, our algorithm may arrive at (3) infinitely often, but this just implies that T converges to a tree which has a unique path. In any case, T is a convex tree, which means that Player I obeys the rule, and so Player I defeats Player II. $\square$

5.2. **Lifting Weihrauch separations to IZF-separations.**

We now translate Weihrauch separation results in Section 5.1 into the context of realizability. Recall that φ is **j**-realizable over $\mathbb{P}$ (or φ is realizable over the site $(\mathbb{P}, \mathbf{j})$) if there is $e \in \mathsf{P}$ such that $e \Vdash_{\mathbf{j}} \varphi$. We say that φ is *boldface* **j**-*realizable* over $\mathbb{P}$ if there is $e \in \mathbf{P}$ such that $e \Vdash_{\mathbf{j}} \varphi$. Let us recall the following key observations:

- By definition, $\neg\varphi$ is **j**-realizable over $\mathbb{P}$ if and only if φ is not boldface **j**-realizable over $\mathbb{P}$.
- For any partial multifunction F, by Lemma 3.18 and Theorem 3.22, **IZF** is $F^{\circlearrowright}$-realizable.
- For any $\mathbf{j} \leq_{\mathsf{W}} \mathsf{WKL}$, a tidy $\forall\exists$-principle φ is **j**-realizable iff $F_\varphi \leq_{\mathsf{W}} \mathbf{j}$ by Lemma 3.32. One can also see that φ is boldface **j**-realizable iff $F_\varphi \leq_{\mathsf{W}}^c \mathbf{j}$.

Therefore, we hereby declare the following:

- All the principles we deal with below are tidy $\forall\exists$-principles.
- Below we only consider partial multifunction $F \leq_{\mathsf{W}} \mathsf{WKL}$. In this case, we have $\mathbf{j} := F^{\circlearrowright} \leq_{\mathsf{W}} \mathsf{WKL}^{\circlearrowright} \equiv_{\mathsf{W}} \mathsf{WKL}$ (see e.g. [17, Proposition 4.9]).

Proof of Theorem 1.1. We prove the following results which are stronger than Theorem 1.1:

(1) $\mathbf{IZF} + \Sigma_1^0\text{-}\mathsf{DML}_\mathbb{R} + \neg\mathsf{RDIV} + \neg\mathsf{BE}_\mathbb{Q}$ is $\mathsf{LLPO}^{\circlearrowright}$-realizable.

(2) $\mathbf{IZF} + \neg\Pi_1^0\text{-}\mathsf{LEM}_\mathbb{R} + \mathsf{BE}_\mathbb{Q} + \neg\mathsf{BE} + \neg\mathsf{RDIV}$ is $(\mathsf{WKL}_{=2})^{\circlearrowright}$-realizable.

(3) $\mathbf{IZF} + \neg\Pi_1^0\text{-}\mathsf{LEM}_\mathbb{R} + \mathsf{RDIV} + \neg\mathsf{BE}_\mathbb{Q}$ is $(\mathsf{WKL}_{\mathrm{aou}})^{\circlearrowright}$-realizable.

(4) $\mathbf{IZF} + \neg\Pi_1^0\text{-}\mathsf{LEM}_\mathbb{R} + \mathsf{IVT}_{\mathrm{lin}} + \neg\mathsf{BE}$ is $(\mathsf{WKL}_{\mathrm{clop}})^{\circlearrowright}$-realizable.

(5) $\mathbf{IZF} + \mathsf{BE} + \neg\mathsf{RDIV}$ is $(\mathsf{WKL}_{\leq 2})^{\circlearrowright}$-realizable.

(6) $\mathbf{IZF} + \neg\Pi_1^0\text{-}\mathsf{LEM}_\mathbb{R} + \mathsf{BE}_\mathbb{Q} + \mathsf{RDIV} + \neg\mathsf{IVT}_{\mathrm{lin}} + \neg\mathsf{BE}$ is $(\mathsf{WKL}_{=2} \times \mathsf{WKL}_{\mathrm{aou}})^{\circlearrowright}$-realizable.

(7) $\mathbf{IZF} + \neg\Pi_1^0\text{-}\mathsf{LEM}_\mathbb{R} + \mathsf{BE} + \mathsf{RDIV} + \neg\mathsf{IVT}_{\mathrm{lin}}$ is $(\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{aou}})^{\circlearrowright}$-realizable.

(8) **IZF** $+ \neg\Pi_1^0\text{-LEM}_{\mathbb{R}} + \text{BE} + \text{IVT}_{\text{lin}} + \neg\text{IVT}$ is $(\text{WKL}_{\leq 2} \times \text{WKL}_{\text{clop}})^{\supset}$-realizable.

(9) **IZF** $+ \neg\Pi_1^0\text{-LEM}_{\mathbb{R}} + \text{IVT} + \neg\text{WKL}$ is $(\text{WKL}_{\text{conv}})^{\supset}$-realizable.

(1) We use $\text{LLPO}^{\supset}$-realizability (i.e., Lifshitz realizability; see below) to realize $\Sigma_1^0\text{-DML}_{\mathbb{R}} + \neg\text{RDIV} + \neg\text{BE}_{\mathbb{Q}}$.

Recall that de Morgan's law $\Sigma_1^0\text{-DML}_{\mathbb{R}}$ is formulated as LLPO, so it is clear that $\text{LLPO}^{\supset}$-realizability validates $\Sigma_1^0\text{-DML}_{\mathbb{R}}$. Moreover, $\text{LLPO}^{\supset}$ is known to be Weihrauch equivalent to the *choice principle* $\text{K}_{\mathbb{N}}$ *for compact sets in* $\mathbb{N}$, which is a multifunction that, given a bound $b \in \mathbb{N}$ and a sequence $a = (a_i)$ of natural numbers, if $V_{a,b} = \{n < b : (\forall i)\ n \neq a_i\}$ is nonempty, return any element from $V_{a,b}$.

Fact 5.10 ([39])**.** $\text{LLPO}^{\supset} \equiv_{\text{W}} \text{K}_{\mathbb{N}}$.

To prove (1), we need to show that $\text{LLPO}^{\supset}$-realizability refutes BE (equivalent to $\text{WKL}_{\leq 2}$) and RDIV (equivalent to WKL_{aou}). This essentially follows from the following:

Lemma 5.11. $\text{WKL}_{=2} \not\leq_{\text{W}}^c \text{K}_{\mathbb{N}}$ *and* $\text{WKL}_{\text{aou}} \not\leq_{\text{W}}^c \text{K}_{\mathbb{N}}$.

The former follows from Proposition 5.4 as $\text{LLPO} \leq_{\text{W}} \text{WKL}_{\text{aou}}$. The latter is a known fact, which also follows from Proposition 5.2 as $\text{LLPO} \leq_{\text{W}} \text{WKL}_{\leq 2}$.

Now, we discuss on realizability. By Fact 5.10, $\text{LLPO}^{\supset}$-realizability is equivalent to $\text{K}_{\mathbb{N}}$-realizability, which may be considered as *Lifschitz realizability* (over the Kleene-Vesley algebra), and so the $\star$-closure property of $\text{K}_{\mathbb{N}}$ has implicitly been known in the context of realizability (cf. Lifschitz [30, Lemma 3]). It is well-known that Lifschitz realizability validates LLPO. In our context, this is because LLPO is a tidy $\forall\exists$-principle and $\text{LLPO} \leq_{\text{W}} \text{WKL}$, so $\text{LLPO}^{\supset}$-realizability validates LLPO by Lemma 3.32.

If $\text{BE}_{\mathbb{Q}}$ is boldface $\text{LLPO}^{\supset}$-realizable, so is $\text{WKL}_{=2}$ by Proposition 4.5. As seen in Observation 4.1, $\text{WKL}_{=2}$ is a tidy $\forall\exists$-principle. Hence, by the boldface version of Lemma 3.32, we get $\text{WKL}_{=2} \leq_{\text{W}}^c \text{K}_{\mathbb{N}}$, which contradicts Lemma 5.11.

Similarly, if RDIV is boldface $\text{K}_{\mathbb{N}}$-realizable, then WKL_{aou} is also $\text{K}_{\mathbb{N}}$-realizable. As seen in Observation 4.1, WKL_{aou} is a tidy $\forall\exists$-principle. Hence, by the boldface version of Lemma 3.32, we get $\text{WKL}_{\text{aou}} \leq_{\text{W}}^c \text{K}_{\mathbb{N}}$, which contradicts Lemma 5.11. This verifies the item (1).

(2) We use $(\text{WKL}_{=2})^{\supset}$-realizability to realize $\neg\Pi_1^0\text{-LEM}_{\mathbb{R}} + \text{BE}_{\mathbb{Q}} + \neg\text{BE} + \neg\text{RDIV}$. By Observation 4.1, $\text{WKL}_{=2}$ is a tidy $\forall\exists$-principle and $\text{WKL}_{=2} \leq_{\text{W}} \text{WKL}$; hence, it is evident that $(\text{WKL}_{=2})^{\supset}$-realizability validates $\text{BE}_{\mathbb{Q}}$ by Proposition 4.5. Thus, to prove (2), it suffices to show the following:

Lemma 5.12. $\text{LPO} \not\leq_{\text{W}}^c (\text{WKL}_{=2})^{\supset}$, $\text{WKL}_{\leq 2} \not\leq_{\text{W}}^c (\text{WKL}_{=2})^{\supset}$ *and* $\text{WKL}_{\text{aou}} \not\leq_{\text{W}}^c (\text{WKL}_{=2})^{\supset}$.

The first assertion is trivial: It is known that $\text{WKL}^{\supset} \equiv_{\text{W}} \text{WKL}$. This means that $P \leq_{\text{W}} \text{WKL}$ implies $P^{\supset} \leq_{\text{W}} \text{WKL}^{\supset} \equiv_{\text{W}} \text{WKL}$. It is also well-known that $\text{LPO} \not\leq_{\text{W}} \text{WKL}$, where recall that LPO is equivalent to $\Pi_1^0\text{-LEM}_{\mathbb{R}}$. The second assertion follows from Proposition 5.8 since $\text{WKL}_{=2} \leq_{\text{W}} \text{Lim}_{\omega}$ by Propositions 4.3 and 4.4. The last assertion clearly follows from Proposition 5.2. This proves Lemma 5.12; hence, as in (1), one can easily verify the item (2).

(3) We use $(\text{WKL}_{\text{aou}})^{\supset}$-realizability to realize $\neg\Pi_1^0\text{-LEM}_{\mathbb{R}} + \text{RDIV} + \neg\text{BE}_{\mathbb{Q}}$. As WKL_{aou} is tidy and equivalent to RDIV, it is evident that $(\text{WKL}_{\text{aou}})^{\supset}$-realizability validates RDIV. Thus, to prove (3), it suffices to show the following:

Lemma 5.13. $\mathsf{LPO} \not\leq_{\mathsf{W}} (\mathsf{WKL}_{\mathrm{aou}})^{\supset}$, *and* $\mathsf{WKL}_{=2} \not\leq_{\mathsf{W}} (\mathsf{WKL}_{\mathrm{aou}})^{\supset}$.

The first assertion is trivial as in Lemma 5.12. The second assertion clearly follows from Proposition 5.4. This proves Lemma 5.13; hence, as in (1), one can easily verify the item (3).

(4) We use $(\mathsf{WKL}_{\mathrm{clop}})^{\supset}$-realizability to realize $\neg\Pi^0_1\text{-}\mathsf{LEM}_{\mathbb{R}} + \mathsf{IVT}_{\mathrm{lin}} + \neg\mathsf{BE}$. As seen in Observation 4.1, $\mathsf{WKL}_{\mathrm{clop}}$ is a tidy $\forall\exists$-principle. Hence, it is evident that $(\mathsf{WKL}_{\mathrm{clop}})^{\supset}$-realizability validates $\mathsf{IVT}_{\mathrm{lin}}$ by Proposition 4.6. To prove (4), it suffices to show the following:

Lemma 5.14. $\mathsf{LPO} \not\leq^c_{\mathsf{W}} (\mathsf{WKL}_{\mathrm{clop}})^{\supset}$, *and* $\mathsf{WKL}_{\leq 2} \not\leq^c_{\mathsf{W}} (\mathsf{WKL}_{\mathrm{clop}})^{\supset}$.

The first assertion is trivial as in Lemma 5.12. The second assertion follows from Proposition 5.8 since $\mathsf{WKL}_{\mathrm{clop}} \leq_{\mathsf{W}} \mathsf{Lim}_\omega$ by Proposition 4.4. This proves Lemma 5.14; hence, as in (1), one can easily verify the item (4).

(5) We use $(\mathsf{WKL}_{\leq 2})^{\supset}$-realizability to realize $\mathsf{BE}+\neg\mathsf{RDIV}$. As $\mathsf{WKL}_{\leq 2}$ is equivalent to BE, it is evident that $(\mathsf{WKL}_{\leq 2})^{\supset}$-realizability validates BE. Moreover, $(\mathsf{WKL}_{\leq 2})^{\supset}$-realizability refutes RDIV by Proposition 5.2. Hence, as in (1), one can easily verify the item (5).

(6) We use $(\mathsf{WKL}_{=2} \times \mathsf{WKL}_{\mathrm{aou}})^{\supset}$-realizability to realize $\neg\Pi^0_1\text{-}\mathsf{LEM}_{\mathbb{R}} + \mathsf{BE}_{\mathbb{Q}} + \mathsf{RDIV} + \neg\mathsf{IVT}_{\mathrm{lin}}+\neg\mathsf{BE}$. It is evident that $(\mathsf{WKL}_{=2}\times\mathsf{WKL}_{\mathrm{aou}})^{\supset}$-realizability validates $\mathsf{BE}_{\mathbb{Q}}+\mathsf{RDIV}$. To prove (6), it suffices to show the following:

Lemma 5.15. $\mathsf{LPO} \not\leq^c_{\mathsf{W}} (\mathsf{WKL}_{=2} \times \mathsf{WKL}_{\mathrm{aou}})^{\supset}$, $\mathsf{WKL}_{\mathrm{clop}} \not\leq^c_{\mathsf{W}} (\mathsf{WKL}_{=2} \times \mathsf{WKL}_{\mathrm{aou}})^{\supset}$, *and* $\mathsf{WKL}_{\leq 2} \not\leq^c_{\mathsf{W}} (\mathsf{WKL}_{=2} \times \mathsf{WKL}_{\mathrm{aou}})^{\supset}$.

The first assertion is trivial as in Lemma 5.12. The second assertion clearly follows from Proposition 5.6. The last assertion follows from Proposition 5.8 since $\mathsf{WKL}_{=2}, \mathsf{WKL}_{\mathrm{aou}} \leq_{\mathsf{W}} \mathsf{Lim}_\omega$ by Proposition 4.4. This proves Lemma 5.15; hence, as in (1), one can easily verify the item (6).

(7) We use $(\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{aou}})^{\supset}$-realizability to realize $\neg\Pi^0_1\text{-}\mathsf{LEM}_{\mathbb{R}}+\mathsf{BE}+\mathsf{RDIV}+\neg\mathsf{IVT}_{\mathrm{lin}}$. It is evident that $(\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{aou}})^{\supset}$-realizability validates $\mathsf{BE} + \mathsf{RDIV}$. To prove (7), it suffices to show the following:

Lemma 5.16. $\mathsf{LPO} \not\leq^c_{\mathsf{W}} (\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{aou}})^{\supset}$, *and* $\mathsf{WKL}_{\mathrm{clop}} \not\leq^c_{\mathsf{W}} (\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{aou}})^{\supset}$.

The first assertion is trivial as in Lemma 5.12. The second assertion follows from Proposition 5.9 since $\mathsf{WKL}_{\mathrm{clop}} \leq_{\mathsf{W}} \mathsf{Lim}_\omega$ by Proposition 4.4. This proves Lemma 5.16; hence, as in (1), one can easily verify the item (7).

(8) We use $(\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{clop}})^{\supset}$-realizability to realize $\neg\Pi^0_1\text{-}\mathsf{LEM}_{\mathbb{R}}+\mathsf{BE}+\mathsf{IVT}_{\mathrm{lin}}+\neg\mathsf{IVT}$. It is evident that $(\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{clop}})^{\supset}$-realizability validates $\mathsf{BE}+\mathsf{IVT}_{\mathrm{lin}}$ by Proposition 4.4. To prove (8), it suffices to show the following:

Lemma 5.17. $\mathsf{LPO} \not\leq^c_{\mathsf{W}} (\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{clop}})^{\supset}$, *and* $\mathsf{WKL}_{\mathrm{conv}} \not\leq^c_{\mathsf{W}} (\mathsf{WKL}_{\leq 2} \times \mathsf{WKL}_{\mathrm{clop}})^{\supset}$.

The first assertion is trivial as in Lemma 5.12. The second assertion clearly follows from Proposition 5.6. This proves Lemma 5.17; hence, as in (1), one can easily verify the item (8).

(9): Straightforward. $\square$

References

[1] Yohji Akama, Stefano Berardi, Susumu Hayashi, and Ulrich Kohlenbach. An arithmetical hierarchy of the law of excluded middle and related principles. In *19th IEEE Symposium on Logic in Computer Science (LICS 2004), 14-17 July 2004, Turku, Finland, Proceedings*, pages 192–201. IEEE Computer Society, 2004.

[2] Andrej Bauer. *The Realizability Approach to Computable Analysis and Topology*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, 2000.

[3] Andrej Bauer. An injection from the Baire space to natural numbers. *Math. Structures Comput. Sci.*, 25(7):1484–1489, 2015.

[4] Stefano Berardi. Some intuitionistic equivalents of classical principles for degree 2 formulas. *Ann. Pure Appl. Logic*, 139(1-3):185–200, 2006.

[5] Stefano Berardi and Silvia Steila. Ramsey's theorem for pairs and K colors as a sub-classical principle of arithmetic. *J. Symb. Log.*, 82(2):737–753, 2017.

[6] Josef Berger, Hajime Ishihara, Takayuki Kihara, and Takako Nemoto. The binary expansion and the intermediate value theorem in constructive reverse mathematics. *Arch. Math. Logic*, 58(1-2):203–217, 2019.

[7] Vasco Brattka and Guido Gherardi. Weihrauch degrees, omniscience principles and weak computability. *J. Symbolic Logic*, 76(1):143–176, 2011.

[8] Vasco Brattka, Guido Gherardi, and Arno Pauly. Weihrauch complexity in computable analysis. In *Handbook of computability and complexity in analysis*, Theory Appl. Comput., pages 367–417. Springer, Cham, [2021] ©2021.

[9] Vasco Brattka and Arno Pauly. On the algebraic structure of Weihrauch degrees. *Logical Methods in Computer Science*, 14(4), 2018.

[10] Ray-Ming Chen and Michael Rathjen. Lifschitz realizability for intuitionistic Zermelo-Fraenkel set theory. *Arch. Math. Logic*, 51(7-8):789–818, 2012.

[11] Matthew de Brecht. Levels of discontinuity, limit-computability, and jump operators. In *Logic, computation, hierarchies*, volume 4 of *Ontos Math. Log.*, pages 79–107. De Gruyter, Berlin, 2014.

[12] Hannes Diener. Constructive reverse mathematics, 2018. Habilitationsschrift, University of Siegen.

[13] Makoto Fujiwara and Taishi Kurahashi. Prenex normal form theorems in semi-classical arithmetic. *J. Symb. Log.*, 86(3):1124–1153, 2021.

[14] Makoto Fujiwara and Taishi Kurahashi. Refining the arithmetical hierarchy of classical principles. *MLQ Math. Log. Q.*, 68(3):318–345, 2022.

[15] Jun Le Goh. Compositions of multivalued functions. *Computability*, 9(3-4):231–247, 2020.

[16] Susumu Hayashi and Masahiro Nakata. Towards limit computable mathematics. In *Types for proofs and programs (Durham, 2000)*, volume 2277 of *Lecture Notes in Comput. Sci.*, pages 125–144. Springer, Berlin, 2002.

[17] Denis R. Hirschfeldt and Carl G. Jockusch, Jr. On notions of computability-theoretic reduction between Π_2^1 principles. *J. Math. Log.*, 16(1):1650002, 59, 2016.

[18] Hajime Ishihara. Constructive reverse mathematics: compactness properties. In *From sets and types to topology and analysis*, volume 48 of *Oxford Logic Guides*, pages 245–267. Oxford Univ. Press, Oxford, 2005.

[19] Hajime Ishihara. Reverse mathematics in Bishop's constructive mathematics. *Philosophia Scientae*, 6:43–59, 2006.

[20] Hajime Ishihara. An introduction to constructive reverse mathematics. In *Handbook of constructive mathematics*, volume 185 of *Encyclopedia Math. Appl.*, pages 636–660. Cambridge Univ. Press, Cambridge, 2023.

[21] A. Kechris. *Classical Descriptive Set Theory*. Springer-Verlag, 1995.

[22] Takayuki Kihara. Rethinking the notion of oracle: a prequel to Lawvere-Tierney topologies for computability theorists. arXiv:2202.00188, 2022.

[23] Takayuki Kihara. Topological reducibilities for discontinuous functions and their structures. *Israel J. Math.*, 252(1):461–500, 2022.

[24] Takayuki Kihara. Lawvere-Tierney topologies for computability theorists. *Trans. Amer. Math. Soc. Ser. B*, 10:48–85, 2023.

[25] Takayuki Kihara and Arno Pauly. Dividing by zero – how bad is it, really? In Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier, editors, *41st Int. Sym. on Mathematical Foundations of Computer Science (MFCS 2016)*, volume 58 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 58:1–58:14. Schloss Dagstuhl, 2016.

[26] Takayuki Kihara and Arno Pauly. Finite choice, convex choice and sorting. In *Theory and applications of models of computation*, volume 11436 of *Lecture Notes in Comput. Sci.*, pages 378–393. Springer, Cham, 2019.

[27] Takayuki Kihara and Victor Selivanov. Wadge-like degrees of Borel bqo-valued functions. *Proc. Amer. Math. Soc.*, 150(9):3989–4003, 2022.

[28] Stéphane Le Roux and Arno Pauly. Finite choice, convex choice and finding roots. *Log. Methods Comput. Sci.*, 11(4):4:6, 30, 2015.

[29] Sori Lee. Subtoposes of the effective topos. Master's thesis, Utrecht University, 2011. available at arXiv:1112.5325.

[30] Vladimir Lifschitz. CT_0 is stronger than CT_0!. *Proc. Amer. Math. Soc.*, 73(1):101–106, 1979.

[31] Saunders Mac Lane and Ieke Moerdijk. *Sheaves in Geometry and Logic*. Universitext. Springer-Verlag, New York, 1994. A first introduction to topos theory, Corrected reprint of the 1992 edition.

[32] David Charles McCarty. *Realizability and recursive mathematics*. PhD thesis, The University of Edinburgh, 1984.

[33] Yiannis N. Moschovakis. *Descriptive Set Theory*, volume 155 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, second edition, 2009.

[34] Luca Motto Ros. Borel-amenable reducibilities for sets of reals. *J. Symbolic Logic*, 74(1):27–49, 2009.

[35] Satoshi Nakata. Local operators in topos theory and separation of semi-classical axioms in intuitionistic arithmetic. In *32nd EACSL Annual Conference on Computer Science Logic*, volume 288 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 42, 21. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2024.

[36] Takako Nemoto. Systems for constructive reverse mathematics. In *Handbook of constructive mathematics*, volume 185 of *Encyclopedia Math. Appl.*, pages 661–699. Cambridge Univ. Press, Cambridge, 2023.

[37] Eike Neumann and Arno Pauly. A topological view on algebraic computation models. *J. Complexity*, 44:1–22, 2018.

[38] Arno Pauly. How incomputable is finding Nash equilibria? *J.UCS*, 16(18):2686–2710, 2010.

[39] Arno Pauly and Hideki Tsuiki. T^ω-representations of compact sets. *CoRR*, abs/1604.00258, 2016.

[40] Michael Rathjen and Andrew W. Swan. Lifschitz realizability as a topological construction. *J. Symb. Log.*, 85(4):1342–1375, 2020.

[41] Richard A. Shore. Reverse mathematics: the playground of logic. *Bull. Symbolic Logic*, 16(3):378–402, 2010.

[42] Stephen G. Simpson. *Subsystems of Second Order Arithmetic*. Perspectives in Logic. Cambridge University Press, 2009.

[43] A. S. Troelstra. Realizability. In *Handbook of proof theory*, volume 137 of *Stud. Logic Found. Math.*, pages 407–473. North-Holland, Amsterdam, 1998.

[44] A. S. Troelstra and D. van Dalen. *Constructivism in mathematics. Vol. I*, volume 121 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1988. An introduction.

[45] Patrick Uftring. Proof-theoretic characterization of Weihrauch reducibility. Master's thesis, Technische Universität Darmstadt, 2018.

[46] Benno van den Berg. A Kuroda-style *j*-translation. *Arch. Math. Logic*, 58(5-6):627–634, 2019.

[47] Jaap van Oosten. Lifschitz' realizability. *J. Symbolic Logic*, 55(2):805–821, 1990.

[48] Jaap van Oosten. A general form of relative recursion. *Notre Dame J. Formal Logic*, 47(3):311–318, 2006.

[49] Jaap van Oosten. *Realizability: an introduction to its categorical side*, volume 152 of *Studies in Logic and the Foundations of Mathematics*. Elsevier B. V., Amsterdam, 2008.

[50] Linda Westrick. A note on the diamond operator. *Computability*, 10(2):107–110, 2021.

[51] Kazuto Yoshimura. From Weihrauch lattice toward logic: Part I. submitted, 2014.

[52] Kazuto Yoshimura. General treatment of non-standard realizabilities. submitted, 2016.