Two Propositions Between WWKL_0 and WKL_0

Wei Wang

Institute of Logic and Cognition, Sun Yat-sen University

CTFM 2019, Wuhan

Based on joint works of Chitat Chong, Wei Li, WW and Yue Yang, and of Barmaplias, WW and Xia.

The Two Propositions

P: every positive binary tree has a perfect subtree.

P^+: every positive binary tree has a positive perfect subtree.

Definitions

Cantor space

The Cantor space 2^ω is the set of countable binary sequences.

The canonical topology of Cantor space 2^ω has a base consisting of

$$[\sigma] = \{X \in 2^\omega : \sigma \prec X\}, \sigma \in 2^{<\omega},$$

where $2^{<\omega}$ denotes the set of finite binary sequences and $\sigma \prec X$ means that σ is an initial segment of X.

The Lebesgue measure μ on Cantor space is a measure such that:

$$\mu([\sigma]) = 2^{-|\sigma|}.$$

A set $\mathcal{C} \subseteq 2^\omega$ is null iff $\mu(\mathcal{C}) = 0$, conull iff $\mu(\mathcal{C}) = 1$, positive iff $\mu(\mathcal{C}) > 0$.

Definitions

Closed sets and trees

A (binary) tree T is a subset of $2^{<\omega}$ s.t. $\sigma \prec \tau \in T$ implies $\sigma \in T$. A leaf of a tree T is some $\sigma \in T$ without extensions in T. A branch of a tree T is an element of Cantor space whose finite initial segments are always in T. $[T]$ is the set of branches of T. The set $[T]$ of a tree T is always a closed subset of Cantor space. T is positive iff $[T]$ is positive as a subset of 2^{ω}.

On the other hand, a closed subset $\mathcal{C}$ of Cantor space can be coded by a tree

$$T = \{\sigma : \exists X \in \mathcal{C}(\sigma \prec X)\}$$

in the sense that $\mathcal{C} = [T]$. There could be $S \neq T$ with $[S] = [T]$, e.g., T is defined from some $\mathcal{C}$ as above and S contains T and some extra leaves.

A perfect subset of Cantor space is a closed set without isolated points. A perfect (binary) tree is an infinite binary tree isormorphic to $2^{<\omega}$. Note that a tree T could be non-perfect even if $[T]$ is a perfect subset of Cantor space.

Motivation from Algorithmic Randomness

In algorithmic randomness, elements of positive subsets of 2^ω have been extensively studied.

As a widely observed phenomenon in algorithmic randomness, almost every element of 2^ω has weak computational strength. E.g., given a non-computable X, the following set is conull:

$$\{Y \in 2^\omega : Y \text{ cannot compute } X\}. \tag{1}$$

So, it is natural to go a step further to study perfect subsets of positive sets from a computability viewpoint. And for this sake, perfect trees are more convenient than perfect subsets of 2^ω.

An easy observation: if a tree contains a perfect subtree then it contains a perfect subtree computing the halting problem. In particular, every positive tree contains a perfect subtree computing the halting problem (in contrast to (1)).

Motivation from Reverse Mathematics

WKL_0 consists of RCA_0 and the statement that every infinite binary tree has a branch. Over RCA_0, WKL_0 is equivalent to many important theorems, like Brouwer's Fixpoint theorem and Gödel's Completeness theorem.

WKL_0 has a corollary so-called WWKL_0, which plays an important role in the reverse mathematics of the part of analysis related to measure theory. WWKL_0 consists of RCA_0 and the statement that every positive binary tree has a branch.

WWKL_0 is closely related to algorithmic randomness, in that for every Martin-Löf random sequence X there is a standard model of WWKL_0 whose second order elements are all computable in X.

WKL_0 is strictly stronger than WWKL_0, and WWKL_0 is strictly stronger than RCA_0.

Clearly, P and P^+ can be regarded as variants of WWKL_0 and seem stronger than WWKL_0.

The Propositions and WKL_0

Theorem (Chong, Li, Wang, Yang)

1. *There exists a computable infinite tree $T \subset 2^{<\omega}$ whose perfect subtrees always compute the halting problem.*
2. *Every computable positive tree $T \subseteq 2^{<\omega}$ contains a positive perfect subtree P which is* low *(i.e., the halting problem relative to P, P', is computable in $\emptyset'$, the standard halting problem).*
3. $\mathsf{WKL}_0 \to \mathsf{P}^+ \to \mathsf{P} \to \mathsf{WWKL}_0$.

Claus 1 means that P would become much less interesting if the positiveness assumption on T were omitted.

Clauses 2 and 3 can be regarded as analogues of Low Basis Theorem (which is a computability form of WKL_0).

The Propositions and WKL_0

Proof

Notation: For a finite binary sequence σ, $|\sigma|$ denotes its length. Let T_σ be the subtree of T whose nodes are all comparable with σ.

- Every positive tree T computes a subtree S and a density function $d : 2^{<\omega} \to \mathbb{Q}$ s.t. $\mu([S]) > q$ for some positive rational $q < \mu([T])$ and if $[S_\sigma] \neq \emptyset$ then $\mu([S_\sigma]) > d(\sigma)$.
- Define a computable increasing function $g : \omega \to \omega$ s.t. if $[S_\sigma] \neq \emptyset$ then σ has two extensions $\tau_0, \tau_1 \in S$ s.t. $|\tau_i| = g(|\sigma|)$ and $[S_{\tau_i}] \neq \emptyset$.
- Now the perfect subtrees P of S s.t. $\mu([P]) \geq q$ and the nodes of P split no later than g form a Π^0_1 class, which allows an application of Low Basis Theorem to obtain the 2nd clause.
- The above proof formalized in second order arithmetic yields $\mathsf{WKL}_0 \vdash \mathsf{P}^+$.

Perfect Subsets of Arbitrary Sets

Theorem (CLWY)

Fix a noncomputable X (e.g., the halting problem).

1. *Every positive binary tree (regardless of its complexity) contains a perfect subtree which does not compute X.*
2. *Every positive subset of Cantor space contains a perfect subset which can be coded by a perfect tree not computing X.*

Perfect Subsets of Arbitrary Sets

Proof

Let S be a positive tree. We build the desired subtree $G \subset S$ by a variant of Mathias forcing.

A forcing condition is a pair (F, T) s.t. F is a finite binary tree, T is a binary tree not computing X, and for every leaf σ of F the tree $(S \cap T)_\sigma$ (i.e., take the intersection of S and T, then remove nodes incomparable with σ) is positive.

A condition (F_1, T_1) extends (F_0, T_0) iff F_1 end-extends F_0 (i.e., $F_0 \subseteq F_1$ and every new node in F_1 extends a leaf of F_0) and $T_1 \subseteq T_0$.

The key here is the following observation. Given a condition (F, T) and a positive rational q s.t. $\mu([(S \cap T)_\sigma]) > q$ for all leaves σ of F, the set of trees R, s.t. $\mu([(R \cap T)_\sigma]) > q$ for all leaves σ of F, form a $\Pi^{0,T}_1$ class and contains S.

Then it can be shown that a sufficiently generic sequence $((F_n, T_n) : n \in \omega)$ produces a perfect $P = \bigcup_n F_n \subseteq T$ as desired.

Two Questions

We have

$$\mathsf{WKL}_0 \to \mathsf{P}^+ \to \mathsf{P} \to \mathsf{WWKL}_0.$$

Are these arrows reversible?

Does every positive subset of Cantor space contain a positive perfect subset coded by a perfect tree with low computational strength?

Separating WKL_0 and P^+

Theorem (Patey)

Every positive tree contains a perfect subtree which does not compute a completion of PA (the first order Peano arithmetic).

$\mathsf{RCA}_0 + \mathsf{P} \nvdash \mathsf{WKL}_0$.

Theorem (Barmaplias, Wang, Xia)

Fix a non-computable X. Every positive tree contains a positive perfect subtree which computes neither a completion of PA nor X.

Hence $\mathsf{RCA}_0 + \mathsf{P}^+ \nvdash \mathsf{WKL}_0$.

By the regularity of Lebesgue measure, the above computatibility results also apply for arbitrary positive subsets of Cantor space.

Separating WKL_0 and P^+

Proof: lower density function

The proof of the computability result of BWX uses a refined forcing of CLWY.

A lower density function (l.d.f.) is a function d s.t. its domain is a finite binary tree, d takes real values, and if $\sigma \in \operatorname{dom} d$ then

$$d(\sigma)2^{-|\sigma|} \leq \sum_{\sigma\langle i\rangle \in \operatorname{dom} d} d(\sigma\langle i\rangle)2^{-|\sigma|-1}.$$

An infinite tree T is d-dense iff $\mu([T_\sigma]) \geq d(\sigma)2^{-|\sigma|}$ for each $\sigma \in \operatorname{dom} d$. Given two l.d.f. d and d', $d' \leq d$ iff $\operatorname{dom} d'$ end-extends $\operatorname{dom} d$ (as finite trees) and if $\sigma \in \operatorname{dom} d$ then $d'(\sigma) \geq d(\sigma)$. So if T is d'-dense and $d' \leq d$ then T is d-dense as well.

Separating WKL_0 and P^+

Proof: forcing conditions

A condition is a pair $p = (d_p, T_p)$ s.t. T_p is an infinite d_p-dense tree and $\mu([(T_p)_\sigma]) > 0$ for every $\sigma \in T_p$.

$q = (d_q, T_q) \leq p$ iff $d_q \leq d_p$ and T_q is a subtree of T_p.

Note that if we let $F_p = \operatorname{dom} d_p$ then (F_p, T_p) is a condition in the forcing of CLWY. However, the computability condition in CLWY is removed here, thanks to an observation of Patey.

Fix a non-computable X and a positive tree T. We may assume that every T_σ is positive.

Working with conditions whose second components are subtrees of T, a series of density lemmas show that a sufficiently generic sequence $(p_n : n \in \omega)$ produces a tree $P = \bigcup_n F_{p_n} = \bigcup_n \operatorname{dom} d_{p_n}$ with desired properties (positive, perfect, being a subtree of T, neither PA nor computing X).

Separating P and WWKL_0

Theorem (BWX)

There exists a positive computable tree T s.t. the following set is null:

$$\{X : X \text{ computes a perfect subtree of } T\}.$$

So sufficiently random sequences cannot compute perfect subtrees of T.

Hence WWKL_0 is strictly weaker than P.

This can also be seen as another evidence that random sequences have weak computational strength.

The key to the construction of T is the following technical lemma.

Lemma

Let Φ be an oracle Turing machine s.t. if $k \in \omega$ and X is any oracle with $\Phi(X;k)\downarrow$ then $\Phi(X;k)$ is a set of 2^k many pairwise incomparable finite binary sequences. Then for any $k \in \omega$ and any $\epsilon > 0$, there exists a c.e. set $V \subset 2^{<\omega}$ of pairwise incomparable sequences s.t.

1. $\sum_{\sigma \in V} 2^{-|\sigma|} \leq \epsilon$;
2. $\{X : \Phi(X;k)\downarrow$ *contains no extension of any element of* $V\}$ *has measure at most* $1/(k\epsilon)$.

$\Phi(X;k)$ is supposed to be the k-th layer of a perfect tree computable in X. So, by removing a subset of 2^ω with measure $\leq \epsilon$ from $[T]$, we can prevent a set of oracles with measure $\geq 1 - 1/(k\epsilon)$ from computing a complete binary subtree of T with height k.

Thank you for your attention.