# An effective perfect-set theorem

David Belanger, joint with Keng Meng (Selwyn) Ng

CTFM 2016 at Waseda University, Tokyo

Institute for Mathematical Sciences
National University of Singapore

# The perfect set theorem for closed sets

## For closed sets

If $F$ is an uncountable, closed subset of $2^\omega$, then $F$ contains a homeomorphic copy of $2^\omega$.

## For trees

If $T$ is a binary tree with uncountably many paths, then $T$ contains a homeomorphic copy $P$ of the full binary tree $2^{<\omega}$.

- This is provable in $\mathrm{ATR}_0$. (Simpson)
- If Cantor-Bendixson rank is $\alpha$, then $P \leq_T \mathbf{0}^{(2\alpha+1)}$.
- Computable trees have C-B rank $\leq \omega_1^{CK}$, and this limit is attained. (Kreisel)
- The Cantor-Bendixson theorem is equivalent to $\Pi^1_1$-$\mathrm{CA}_0$. (H. Friedman)

Basic motivation: Draw or exploit an analogy between the Perfect Set Theorem and Weak König's Lemma.

# The perfect set theorem for closed sets

**For closed sets**

If $F$ is an uncountable, closed subset of $2^\omega$, then $F$ contains a homeomorphic copy of $2^\omega$.

**For trees**

If $T$ is a binary tree with uncountably many paths, then $T$ contains a homeomorphic copy $P$ of the full binary tree $2^{<\omega}$.

- This is provable in $\mathrm{ATR}_0$. (Simpson)
- If Cantor-Bendixson rank is $\alpha$, then $P \leq_T \mathbf{0}^{(2\alpha+1)}$.
- Computable trees have C-B rank $\leq \omega_1^{CK}$, and this limit is attained. (Kreisel)
- The Cantor-Bendixson theorem is equivalent to $\Pi^1_1\text{-}\mathrm{CA}_0$. (H. Friedman)

Basic motivation: Draw or exploit an analogy between the Perfect Set Theorem and Weak König's Lemma.

# Basic question for today

**Question**

How difficult is the problem: Given a computable tree $T$ with uncountably many paths, find a perfect subtree?

- The usual reduction of $\mathrm{ATR}_0$ to this problem works by coding into a countable $\Pi^0_1$ class—countable, but not effectively countable.
- We restrict ourselves to trees of finite Cantor-Bendixson rank.
- A weaker version of our results can be derived from Cenzer, Clote, Smith, Soare, Wainer: "Members of countable $\Pi^0_1$ classes."

# Basic question for today

**Question**

How difficult is the problem: Given a computable tree $T$ with uncountably many paths, find a perfect subtree?

- The usual reduction of $\mathrm{ATR}_0$ to this problem works by coding into a countable $\Pi^0_1$ class—countable, but not effectively countable.
- We restrict ourselves to trees of finite Cantor-Bendixson rank.
- A weaker version of our results can be derived from Cenzer, Clote, Smith, Soare, Wainer: "Members of countable $\Pi^0_1$ classes."

# Examples of the perfect-set problem

## Example 0

Let $T$ be any (nonempty, computable, binary) tree with no dead ends and no isolated paths. Then $T$ is a perfect subtree of itself.

## Example $\frac{1}{2}$

Suppose $T$ is the union of a perfect tree and some 'dead-end' pieces, i.e., some $\sigma$ satisfying:

$$(\exists \ell > |\sigma|)[\sigma \text{ has no extensions in T of length } \ell].$$

Since the halting set $\mathbf{0}'$ can detect these pieces, $\mathbf{0}'$ is strong enough to compute the perfect subtree.

# A converse to Example $\frac{1}{2}$

## Proposition

*There is a computable $T$ consisting of a perfect tree and some dead-end pieces such that any perfect subtree $P \subseteq T$ computes the halting set $\mathbf{0}'$.*

## Proof.

Recall the definition of the Halting Set

$$0' = \{e \in \omega : \text{the e-th Turing machine halts}\},$$

and its recursive approximation

$$0'_s = \{e < s : \text{the e-th TM halts in } < \text{s steps}\},$$

and its least modulus function

$$m_{0'}(x) = \mu s.[0' \upharpoonright x = 0'_s \upharpoonright x].$$

# A converse to Example $\frac{1}{2}$

**Proof (continued).**

$$m_{0'}(x) = \mu s.[0' \restriction x = 0'_s \restriction x].$$

- Construct a tree with exactly $2^x$ nodes at level $m_{0'}(x) + x$, each with two extensions at level $m_{0'}(x+1) + x + 1$.
- There is a perfect subtree by definition.
- Now suppose $P \subseteq T$ is perfect; then the function

$$f(x) = \mu \ell.[\text{P has } \geq 2^x \text{ many nodes at level } \ell]$$

dominates $m_{0'}(x)$ and hence can be used to compute $0'$.

$\square$

# The Cantor-Bendixson derivative and rank

**For closed sets**

If $F$ is closed, define $\partial F = F - \{\text{isolated points of } F\}$.
If $\alpha$ is least such that $\delta^{\alpha+1} F = \delta_\alpha F$, we say $F$ has Cantor-Bendixson rank $\alpha$.

Famously used to prove:

**Cantor-Bendixson Theorem**

Every closed $F$ has rank $< \omega_1$; in particular, $F$ is a union of a perfect set and a countable set.

We are concerned with $\Pi^0_1$ classes of finite rank.

# Rank as an upper bound

## Theorem (Folklore)

*If $T$ is a tree whose paths $[T]$ have rank $n$, then $0^{(2n+1)}$ can find a perfect subtree, where $0^{(1)} = 0'$ is the Halting Problem, $0^{(2)} = 0''$ is the relativized 'Halting Problem's halting problem,' etc.*

## Proof.

- Use $0''$ to trim off the roots of isolated paths:

$$\{\sigma : (\forall \ell_1 > |\sigma|)(\exists \ell_2 > \ell_1)[\text{at most one } \tau \supset \sigma \text{ at level } \ell_1 \text{ has an extension at level } \ell_2]\}.$$

- Use $0^{(4)}$ to iterate this process a second time.

$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots$

- Use $0^{(2n)}$ to remove all isolated paths.
- Use one more jump to remove the remaining dead-ends.

Now we have our perfect tree. $\square$

# Rank as an upper bound, part 2

**Theorem (Folklore)**

*If $T$ is a tree whose paths $[T]$ have rank $n$, then $0^{(2n+1)}$ can find a perfect subtree.*

**Alternate proof.**

- Use $0'$ to remove dead-ends $\sigma$ as in Example $\frac{1}{2}$:
$$(\exists \ell > |\sigma|)[\sigma \text{ has no extensions of length } \ell].$$
- Use $0''$ to remove roots $\sigma$ of isolated paths, which is now simpler:
$$(\forall \ell > |\sigma|)[\sigma \text{ has at most one extension of length } \ell].$$
- Use $0'''$ to remove the new dead ends.

$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots$

- Use $0^{(2n)}$ to remove the last isolated paths.
- Use $0^{(2n+1)}$ to remove the last dead ends.

Now we have our perfect tree.

# Cantor-Bendixson rank as a lower bound

## For trees

If $T$ is a tree whose paths $[T]$ have rank $n$ then:

- $T$ has rank $n$ if $T$ has no isolated paths;
- $T$ has rank $n\frac{1}{2}$ otherwise.

You can also define rank using an appropriate *half-derivative*.

## Main Theorem

If $T$ has rank $q \in \{0, \frac{1}{2}, 1, 1\frac{1}{2}, 2, \ldots\}$, then $\mathbf{0}^{(2q)}$ is exactly enough to find a perfect subtree.

- We have seen this for $q = 0$ and $q = \frac{1}{2}$.
- We have seen that $\mathbf{0}^{(2q)}$ is an upper bound.
- Remains to show that $\mathbf{0}^{(2q)}$ is necessary for $q \geq 1$.

# Cantor-Bendixson rank as a lower bound

## For trees

If $T$ is a tree whose paths $[T]$ have rank $n$ then:

- $T$ has rank $n$ if $T$ has no isolated paths;
- $T$ has rank $n\frac{1}{2}$ otherwise.

You can also define rank using an appropriate *half-derivative*.

## Main Theorem

If $T$ has rank $q \in \{0,\ \frac{1}{2},\ 1,\ 1\frac{1}{2},\ 2,\ \ldots\}$, then $\mathbf{0}^{(2q)}$ is exactly enough to find a perfect subtree.

- We have seen this for $q = 0$ and $q = \frac{1}{2}$.
- We have seen that $\mathbf{0}^{(2q)}$ is an upper bound.
- Remains to show that $\mathbf{0}^{(2q)}$ is necessary for $q \geq 1$.

# Proof outline

Recall: $T$ rank $n\frac{1}{2}$ means $[T]$ rank $n$ and $T$ has dead ends.

### Lemma 0

There is a $0^{(n)}$-computable tree $T$ of rank $\frac{1}{2}$ with uncountably many paths such that every perfect subtree computes $0^{(n+1)}$.

### Lemma $\frac{1}{2}$

If $T$ is a $0'$-computable tree of rank $\frac{1}{2}$, there is a computable tree $T^*$ of rank 1 such that every perfect subtree of $T^*$ computes a perfect subtree of $T$.

### Lemma 1

As above, with $T$ of rank 1 and $T^*$ of rank $1\frac{1}{2}$.

Start with $T$ as in Lemma 0. Alternate between versions of Lemma $\frac{1}{2}$ and Lemma 1 until you get a computable $T^*$ of rank $n/2$ whose perfect subtrees each compute $0^{(n+1)}$.

# Proof outline

Recall: $T$ rank $n\frac{1}{2}$ means $[T]$ rank $n$ and $T$ has dead ends.

**Lemma 0**

There is a $0^{(n)}$-computable tree $T$ of rank $\frac{1}{2}$ with uncountably many paths such that every perfect subtree computes $0^{(n+1)}$.

**Lemma $\frac{1}{2}$**

If $T$ is a $0'$-computable tree of rank $\frac{1}{2}$, there is a computable tree $T^*$ of rank 1 such that every perfect subtree of $T^*$ computes a perfect subtree of $T$.

**Lemma 1**

As above, with $T$ of rank 1 and $T^*$ of rank $1\frac{1}{2}$.

Start with $T$ as in Lemma 0. Alternate between versions of Lemma $\frac{1}{2}$ and Lemma 1 until you get a computable $T^*$ of rank $n/2$ whose perfect subtrees each compute $0^{(n+1)}$.

# Proof outline

Recall: $T$ rank $n\frac{1}{2}$ means $[T]$ rank $n$ and $T$ has dead ends.

**Lemma 0**

There is a $0^{(n)}$-computable tree $T$ of rank $\frac{1}{2}$ with uncountably many paths such that every perfect subtree computes $0^{(n+1)}$.

**Lemma $\frac{1}{2}$**

If $T$ is a $0'$-computable tree of rank $\frac{1}{2}$, there is a computable tree $T^*$ of rank 1 such that every perfect subtree of $T^*$ computes a perfect subtree of $T$.

**Lemma 1**

As above, with $T$ of rank 1 and $T^*$ of rank $1\frac{1}{2}$.

Start with $T$ as in Lemma 0. Alternate between versions of Lemma $\frac{1}{2}$ and Lemma 1 until you get a computable $T^*$ of rank $n/2$ whose perfect subtrees each compute $0^{(n+1)}$.

# Proving the theorem

**Lemma 0**

There is a $0^{(n)}$-computable tree $T$ of rank $\frac{1}{2}$ with uncountably many paths such that every perfect subtree computes $0^{(n+1)}$.

**Proof.**

- Let $m_{0^{(n+1)}}$ be the least modulus function of $0^{(n+1)}$ when approximated using $0^{(n)}$ as an oracle.
- Similar to before, construct a $0^{(n)}$-computable tree with exactly $2^x$ nodes at each level $m_{0^{(n+1)}}(x) + x$, each with exactly two extensions at $m_{0^{(n+1)}}(x+1) + x + 1$.
- Then every perfect subtree computes a function dominating $m_{0^{(n+1)}}$.
- Such a function computes $0^{(n+1)}$. (Proof: First show it computes $0'$, then that it computes $0''$, etc.)

□

# Proving the theorem

## Lemma 0

There is a $0^{(n)}$-computable tree $T$ of rank $\frac{1}{2}$ with uncountably many paths such that every perfect subtree computes $0^{(n+1)}$.

## Proof.

- Let $m_{0^{(n+1)}}$ be the least modulus function of $0^{(n+1)}$ when approximated using $0^{(n)}$ as an oracle.
- Similar to before, construct a $0^{(n)}$-computable tree with exactly $2^x$ nodes at each level $m_{0^{(n+1)}}(x) + x$, each with exactly two extensions at $m_{0^{(n+1)}}(x+1) + x + 1$.
- Then every perfect subtree computes a function dominating $m_{0^{(n+1)}}$.
- Such a function computes $0^{(n+1)}$. (Proof: First show it computes $0'$, then that it computes $0''$, etc.)

$\square$

# Proving the theorem

**Lemma $\frac{1}{2}$**

If $T$ is a $0'$-computable tree of rank $\frac{1}{2}$, there is a computable tree $T^*$ of rank 1 such that every perfect subtree of $T^*$ computes a perfect subtree of $T$.

- Let $(T_s)_{s\in\omega}$ be a recursive approximation to $T$.
- We build $T^*$ as a ternary tree in $\{0,1,b\}^{<\omega}$.
- For every finite or infinite string $\sigma \in \{0,1,b\}^{<\omega}$, let $\bar{\sigma}$ denote the string you get after removing all $b$.
- Example: If $\sigma = 01bbb11b01b$ then $\bar{\sigma} = 011101$.

1. Put the empty string $\emptyset$ into $T^*$.
2. If $\sigma \in T^*$ and $\bar{\sigma}0 \in T_{|\sigma|}$, put $\sigma 0$ into $T^*$.
3. If $\sigma \in T^*$ and $\bar{\sigma}1 \in T_{|\sigma|}$, put $\sigma 1$ into $T^*$.
4. If $\sigma \in T^*$ and neither case applies, put $\sigma b$ into $T^*$.

# Proving the theorem

**Lemma $\frac{1}{2}$**

If $T$ is a $0'$-computable tree of rank $\frac{1}{2}$, there is a computable tree $T^*$ of rank 1 such that every perfect subtree of $T^*$ computes a perfect subtree of $T$.

1. Put the empty string $\emptyset$ into $T^*$.
2. If $\sigma \in T^*$ and $\bar{\sigma}0 \in T_{|\sigma|}$, put $\sigma 0$ into $T^*$.
3. If $\sigma \in T^*$ and $\bar{\sigma}1 \in T_{|\sigma|}$, put $\sigma 1$ into $T^*$.
4. If $\sigma \in T^*$ and neither case applies, put $\sigma b$ into $T^*$.

- Every $g \in [T]$ equals $\bar{f}$ for a unique $f \in [T^*]$.
- If $g_1, g_2 \in [T]$ then $g_1 = \bar{f}_1$ and $g_2 = \bar{f}_2$, and $\overline{f_1 \cap f_2} = g_1 \cap g_2$, where $\cdot \cap \cdot$ denotes the longest common initial segment.
- If $f \in [T^*]$ equals $\sigma b^\omega$, then $f$ isolated above $\sigma$.
- If $f \in [T^*]$ is not of this form, then $\bar{f} = g$ for some $g \in [T]$.
- No dead ends.

# Proving the theorem

**Lemma 1**

If $T$ is a $0'$-computable tree of rank 1, there is a computable tree $T^*$ of rank $1\frac{1}{2}$ such that every perfect subtree of $T^*$ computes a perfect subtree of $T$.

- Watch the computable approximation $(T_s)_s$ to $T$.
- We may assume no $T_s$ has dead ends.
- Build $T^*$ together with partial embeddings $\psi_s : T_s \to T^*$, with pointwise limit $\psi$.

1. If $\sigma = \sigma_0 i \in T_s \cap T_{s+1}$ has different successors in $T_{s+1}$ than in $T_s$, reassign $\psi_{s+1}(\sigma)$ to a maximal extension of $\psi_s(\sigma_0)$ in $T^*_s$, and add the appropriate successors to $\psi_{s+1}(\sigma)$ in $T_{s+1}$.
2. Do not extend $\tau \in T^*_s$ which are not an initial segment of some $\psi_{s+1}(\sigma)$.

# Proving the theorem

**Lemma 1**

If $T$ is a $0'$-computable tree of rank 1, there is a computable tree $T^*$ of rank $1\frac{1}{2}$ such that every perfect subtree of $T^*$ computes a perfect subtree of $T$.

1. If $\sigma = \sigma_0 i \in T_s \cap T_{s+1}$ has different successors in $T_{s+1}$ than in $T_s$, reassign $\psi_{s+1}(\sigma)$ to a maximal extension of $\psi_s(\sigma_0)$ in $T_s^*$, and add the appropriate successors to $\psi_{s+1}(\sigma)$ in $T_{s+1}$.
2. Do not extend $\tau \in T_s^*$ which are not an initial segment of some $\psi_{s+1}(\sigma)$.

- For every path $g \in [T^*]$ there is a unique $f \in [T]$ such that $\psi(\sigma) \subseteq g$ for every $\sigma \subseteq f$.
- For every pair $f_1, f_2$, the image $\psi(f_1 \cap f_2)$ is approximately $\psi(f_1) \cap \psi(f_2)$.
- If $P \subseteq T$ is perfect, we may use its splits to solve for the perfect tree $\psi^{-1}(P)$.

# Future work.

### Question

What is the exact difficulty of the perfect set problem for limit ranks $\lambda < \omega_1^{CK}$?

- Can you code $0^{(\omega)}$ or other $H$-sets directly into the trees?
- Given a uniform sequence of trees $T_0, T_1, \ldots$ can you combine them into a single tree, with smallish rank, whose perfect set problem solves those of every $T_k$?

### Question

What about rank $\omega_1^{CK}$? (Possible, due to Kreisel.)

### Question

What about $\Sigma_1^1$ classes in place of $\Pi_1^0$ classes?

The end.