Traceability of the Jump

Keng Meng, Ng
Victoria University of Wellington
Motivation

From algorithmic randomness.

- Approaches that tries to capture the intuitive meaning of “complexity” and “randomness”.

- In particular, the remarkable class of the $K$-trivials. (Compressibility)

- Relationship with the classical computability notions, such as degrees?

- Questions from relativization; Completions of pseudojump operators.
**Motivation**

- From algorithmic randomness.
  - Approaches that tries to capture the intuitive meaning of “complexity” and “randomness”.
    - In particular, the remarkable class of the $K$-trivials. (Compressibility)
    - Relationship with the classical computability notions, such as degrees?
  - Questions from relativization; Completions of pseudojump operators.
Motivation

- From algorithmic randomness.
  - Approaches that tries to capture the intuitive meaning of “complexity” and “randomness”.
  - In particular, the remarkable class of the $K$-trivals. (Compressibility)
  - Relationship with the classical computability notions, such as degrees?
- Questions from relativization;
  Completions of pseudojump operators.
Motivation

- From algorithmic randomness.
  - Approaches that tries to capture the intuitive meaning of “complexity” and “randomness”.
  - In particular, the remarkable class of the $K$-trivials. (Compressibility)
  - Relationship with the classical computability notions, such as degrees?

- Questions from relativization; Completions of pseudojump operators.
Some Concepts (I)

- A Turing Machine is a fixed set of instructions.
- An oracle TM has an extra read only tape.
- $A \subseteq \mathbb{N}$ is recursive (or computable) in $B \subseteq \mathbb{N}$. We write $A \leq_T B$.
- $A$ is recursively enumerable (r.e.) in $B$.
- Write $W^X_\varphi$ as the $e^{th}$ set, recursively enumerable in $X$. 
Some Concepts (I)

- A Turing Machine is a fixed set of instructions.

- An oracle TM has an extra read only tape.

- $A \subseteq \mathbb{N}$ is recursive (or computable) in $B \subseteq \mathbb{N}$. We write $A \leq_T B$.

- $A$ is recursively enumerable (r.e.) in $B$.

- Write $W_e^X$ as the $e^{th}$ set, recursively enumerable in $X$. 
Some Concepts (I)

- A Turing Machine is a fixed set of instructions.

- An oracle TM has an extra read only tape.

- \( A \subseteq \mathbb{N} \) is recursive (or computable) in \( B \subseteq \mathbb{N} \). We write \( A \leq_T B \).

- \( A \) is recursively enumerable (r.e.) in \( B \).

- Write \( W^X_\epsilon \) as the \( \epsilon \text{th} \) set, recursively enumerable in \( X \).
Some Concepts (I)

- A Turing Machine is a fixed set of instructions.

- An oracle TM has an extra read only tape.

- $A \subseteq \mathbb{N}$ is recursive (or computable) in $B \subseteq \mathbb{N}$. We write $A \leq_T B$.

- $A$ is recursively enumerable (r.e.) in $B$.

- Write $W^X_\theta$ as the $e^{th}$ set, recursively enumerable in $X$.  

Keng Meng, Ng Victoria University of Wellington

Traceability of the Jump
Some Concepts (II)

- Halting problem $\emptyset' = \{ e : e^{th} \text{ program halts} \}$.

- Similarly, relativization of above to define $A'$ for any set $A$.

- (Shoenfield Limit Lemma) $A \leq_T \emptyset'$ iff $A \in \Delta^0_2$.

- $A$ is low if $A' \equiv_T \emptyset'$.
  (In some sense, low computational power).

- $A$ is high if $A' \equiv_T \emptyset''$. 
Some Concepts (II)

- Halting problem $\emptyset' = \{ e : e^{th} \text{ program halts} \}$.

- Similarly, relativization of above to define $A'$ for any set $A$.

- (Shoenfield Limit Lemma) $A \leq_T \emptyset'$ iff $A \in \Delta_2^0$.

- $A$ is low if $A' \equiv_T \emptyset'$.
  (In some sense, low computational power).

- $A$ is high if $A' \equiv_T \emptyset''$.
SOME CONCEPTS (II)

- Halting problem $\emptyset' = \{ e : e^{th} \text{ program halts} \}$.

- Similarly, relativization of above to define $A'$ for any set $A$.

- (Shoenfield Limit Lemma) $A \leq_T \emptyset'$ iff $A \in \Delta^0_2$.

- $A$ is low if $A' \equiv_T \emptyset'$.
  (In some sense, low computational power).

- $A$ is high if $A' \equiv_T \emptyset''$. 
**Stronger Forms Of Reducibilities (I)**

- Truth-table Reducibility:
  
  We say $A \leq_{tt} B$ if there is a recursive $f$, such that
  
  $$x \in A \iff B \models \text{the } f(x)^{th} \text{ truth table}$$

  for every $x$.

- $A$ is super-low if $A' \equiv_{tt} \emptyset'$.

- $A$ is super-high if $A' \equiv_{tt} \emptyset''$.
Stronger Forms of Reducibilities (I)

- **Truth-table Reducibility**: We say $A \leq_{tt} B$ if there is a recursive $f$, such that
  
  $$x \in A \iff B \models \text{the } f(x)^{th} \text{ truth table}$$

  for every $x$.

- $A$ is super-low if $A' \equiv_{tt} \emptyset'$.

- $A$ is super-high if $A' \equiv_{tt} \emptyset''$. 
STRONGER FORMS OF REDUCIBILITIES (II)

- We say $X$ is $\omega$-r.e. if there are recursive $f$ and $g$,

  $\lim_{s \to \infty} f(n, s) = X(n),$

  $\# \text{ changes in } f(n, s) \leq g(n),$

  for all $n$.

- One can show that $A$ is super-low iff $A'$ is $\omega$-r.e.

- Hence, the standard construction of a low set also produces a super-low set.
STRONGER FORMS OF REDUCIBILITIES (II)

- We say $X$ is $\omega$-r.e. if there are recursive $f$ and $g$,

$$\lim_{s \to \infty} f(n, s) = X(n),$$

$$\# \text{ changes in } f(n, s) \leq g(n),$$

for all $n$.

- One can show that $A$ is super-low iff $A'$ is $\omega$-r.e.

- Hence, the standard construction of a low set also produces a super-low set.
**Stronger Forms of Reducibilities (II)**

- We say $X$ is $\omega$-r.e. if there are recursive $f$ and $g$, 

  $$\lim_{s \to \infty} f(n, s) = X(n),$$

  $$\# \text{ changes in } f(n, s) \leq g(n),$$

  for all $n$.

- One can show that $A$ is super-low iff $A'$ is $\omega$-r.e.

- Hence, the standard construction of a low set also produces a super-low set.
**Some Concepts from Randomness**

- Let $\sigma, \tau, \text{etc}$ denote finite binary strings.

- Let $U$ be the universal prefix-free machine.

  $$U : 2^{<\omega} \leftrightarrow 2^{<\omega},$$

  $\text{dom}(U)$ is an anti-chain under the extension relation.

- Prefix-free Kolmogorov complexity:

  $$K(\sigma) = \min\{|\tau| : U(\tau) \downarrow = \sigma\}.$$

- Measurement of compressibility.
Some Concepts from Randomness

- Let $\sigma, \tau$, etc denote finite binary strings.
- Let $U$ be the universal prefix-free machine.

$$U : 2^{<\omega} \leftrightarrow 2^{<\omega},$$

$\text{dom}(U)$ is an anti-chain under the extension relation.

- Prefix-free Kolmogorov complexity:

$$K(\sigma) = \min \{|\tau| : U(\tau) \downarrow = \sigma\}.$$

- Measurement of compressibility.
Some Concepts from Randomness

- Let $\sigma, \tau$, etc denote finite binary strings.
- Let $U$ be the universal prefix-free machine.

$$U : 2^{<\omega} \leftrightarrow 2^{<\omega},$$

$dom(U)$ is an anti-chain under the extension relation.

- Prefix-free Kolmogorov complexity:

$$K(\sigma) = \min\{|\tau| : U(\tau) \downarrow = \sigma\}.$$

- Measurement of compressibility.
Some Concepts from Randomness

- Let $\sigma$, $\tau$, etc denote finite binary strings.

- Let $U$ be the universal prefix-free machine.

$$U : 2^{<\omega} \leftrightarrow 2^{<\omega},$$

$dom(U)$ is an anti-chain under the extension relation.

- Prefix-free Kolmogorov complexity:

$$K(\sigma) = \min\{|\tau| : U(\tau) \downarrow = \sigma\}.$$

- Measurement of compressibility.
**K-triviality (I)**

- For any string \( \sigma \),
  \[
  K(\sigma) \geq K(|\sigma|) + O(1).
  \]

- A real \( \alpha \in 2^\omega \) is \( K \)-trivial, if all of it’s initial segments are highly compressible.
  \[
  K(\alpha|_n) \leq K(0^n) + O(1),
  \]
  for every \( n \).

- \( \alpha \) is “random” would mean that it is hard to compress.
- \( K \)-trivial would be opposite of what it means to be “random.”
**K-triviality (I)**

- For any string $\sigma$,
  \[ K(\sigma) \geq K(|\sigma|) + O(1). \]

- A real $\alpha \in 2^\omega$ is **K-trivial**, if all of its initial segments are highly compressible.
  \[ K(\alpha|n) \leq K(0^n) + O(1), \]
  for every $n$.

- $\alpha$ is “random” would mean that it is hard to compress.
- **K-trivial** would be opposite of what it means to be “random”.

---

Keng Meng, Ng Victoria University of Wellington

Traceability of the Jump
For any string \( \sigma \),

\[
K(\sigma) \geq K(|\sigma|) + \mathcal{O}(1).
\]

A real \( \alpha \in 2^\omega \) is \( K \)-trivial, if all of it’s initial segments are highly compressible.

\[
K(\alpha|_n) \leq K(0^n) + \mathcal{O}(1),
\]

for every \( n \).

\( \alpha \) is “random” would mean that it is hard to compress.

\( K \)-trivial would be opposite of what it means to be “random”.
Solovay first constructed a non-recursive $K$-trivial ($\Delta^0_2$), later Zambella, Calude, Cole (r.e.).

As is now well-known - construction via cost functions by Downey, Hirschfeldt, Nies, Stephan:

- Assign a cost $c(x, s)$ for each $x$ and $s$.
- We make the enumeration at stage $s$, if $c(x, s) < \text{some bound we can afford.}$
Solovay first constructed a non-recursive $K$-trivial ($\Delta^0_2$), later Zambella, Calude, Cole (r.e.).

As is now well-known - construction via cost functions by Downey, Hirschfeldt, Nies, Stephan:

- Assign a cost $c(x, s)$ for each $x$ and $s$.
- We make the enumeration at stage $s$, if $c(x, s) < $ some bound we can afford.
Solovay first constructed a non-recursive $K$-trivial ($\Delta^0_2$), later Zambella, Calude, Cole (r.e.).

As is now well-known - construction via cost functions by Downey, Hirschfeldt, Nies, Stephan:

- Assign a cost $c(x, s)$ for each $x$ and $s$.
- We make the enumeration at stage $s$, if $c(x, s) <$ some bound we can afford.
$K$-TRIVIALITY (II)

- Solovay first constructed a non-recursive $K$-trivial ($\Delta^0_2$), later Zambella, Calude, Cole (r.e.).

- As is now well-known - construction via cost functions by Downey, Hirschfeldt, Nies, Stephan:
  - Assign a cost $c(x, s)$ for each $x$ and $s$.
  - We make the enumeration at stage $s$, if $c(x, s) < \text{some bound we can afford}$. 
K-triviality (III)

▷ (Downey et al.) A straightforward construction of a (promptly simple) $K$-trivial:

**Proof.**

Put $x$ into $A$ at stage $s$, if

\[
x \in W_{e,s}
\]

\[
W_{e,s} \cap A_{s-1} = \emptyset
\]

\[
x \geq 2e
\]

\[
c(x, s) < \frac{1}{2e+1}
\]
K-TRIVIALITY (III)

- (Downey et al.) A straightforward construction of a (promptly simple) $K$-trivial:

**Proof.**
Put $x$ into $A$ at stage $s$, if

$$x \in W_{e,s}$$

$$W_{e,s} \cap A_{s-1} = \emptyset$$

$$x \geq 2e$$

$$c(x, s) < \frac{1}{2^{e+1}}$$
**K-TRIVIALITY (III)**

(Downey et al.) A straightforward construction of a (promptly simple) $K$-trivial:

**Proof.**

Put $x$ into $A$ at stage $s$, if

$x \in W_{e,s}$

$W_{e,s} \cap A_{s-1} = \emptyset$

$x \geq 2e$

$c(x, s) < \frac{1}{2^{e+1}}$
The $K$-trivials have aroused great interest, and coincide with various other classes.

- (Nies) Low for $K$
  (i.e. $K^A(\sigma) = K(\sigma) + \mathcal{O}(1)$)

- (Nies) Low for ML-random
  (i.e. $A$-random=1-random)

- (Downey et al.) Every $\Omega$-operator takes $A$ to a c.e. real
The $K$-trivials have aroused great interest, and coincide with various other classes.

- (Nies) Low for $K$
  (i.e. $K^A(\sigma) = K(\sigma) + O(1)$)

- (Nies) Low for ML-random
  (i.e. $A$-random = 1-random)

- (Downey et al.) Every $\Omega$-operator takes $A$ to a c.e. real
The $K$-trivials have aroused great interest, and coincide with various other classes.

- (Nies) Low for $K$
  (i.e. $K^A(\sigma) = K(\sigma) + \mathcal{O}(1)$)

- (Nies) Low for ML-random
  (i.e. $A$-random=1-random)

- (Downey et al.) Every $\Omega$-operator takes $A$ to a c.e. real
The $K$-trivials have aroused great interest, and coincide with various other classes.

- (Nies) Low for $K$
  (i.e. $K^A(\sigma) = K(\sigma) + O(1)$)

- (Nies) Low for ML-random
  (i.e. $A$-random=1-random)

- (Downey et al.) Every $\Omega$-operator takes $A$ to a c.e. real
**Degrees Containing $K$-trivials**

- (Chaitin) Every $K$-trivial real is $\Delta^0_2$.

- (Downey et al.) Every $K$-trivial real is Turing incomplete (in fact, non-high), by the Decanter Method.

- (Downey et al.) $K$-trivials closed under $\oplus$ and $\leq_T$.

- (Nies Low$_2$-top Theorem) Bounded by an r.e. low$_2$ set.
Degrees Containing $K$-trivials

- (Chaitin) Every $K$-trivial real is $\Delta^0_2$.

- (Downey et al.) Every $K$-trivial real is Turing incomplete (in fact, non-high), by the Decanter Method.

- (Downey et al.) $K$-trivials closed under $\oplus$ and $\leq_T$.

- (Nies Low$_2$-top Theorem) Bounded by an r.e. low$_2$ set.
Degrees Containing $K$-trivials

- (Chaitin) Every $K$-trivial real is $\Delta^0_2$.

- (Downey et al.) Every $K$-trivial real is Turing incomplete (in fact, non-high), by the Decanter Method.

- (Downey et al.) $K$-trivials closed under $\oplus$ and $\leq_T$.

- (Nies Low$_2$-top Theorem) Bounded by an r.e. low$_2$ set.
Degrees Containing $K$-trivials

- (Chaitin) Every $K$-trivial real is $\Delta^0_2$.

- (Downey et al.) Every $K$-trivial real is Turing incomplete (in fact, non-high), by the Decanter Method.

- (Downey et al.) $K$-trivials closed under $\oplus$ and $\leq_T$.

- (Nies Low$_2$-top Theorem) Bounded by an r.e. low$_2$ set.
Computable Traceability (I)

- An order is a total function $h : \mathbb{N} \rightarrow \mathbb{N}$, such that $h$ is recursive, non-decreasing and unbounded.

- (Terwijn and Zambella)
  A degree $a$ is said to be computably traceable, if there is an order $h$ such that for every $f \leq_T a$, there is a strong array of finite sets $\{G_n\}_{n \in \mathbb{N}}$ (called the computable trace), such that
  
  1. $|G_n| \leq h(n)$, and
  2. $f(n) \in G_n$.

- Computably traceable is a uniform version of being hyperimmune-free.
**Computable Traceability (I)**

- An order is a total function $h : \mathbb{N} \rightarrow \mathbb{N}$, such that $h$ is recursive, non-decreasing and unbounded.

- (Terwijn and Zambella)
  A degree $a$ is said to be computably traceable, if there is an order $h$ such that for every $f \leq_T a$, there is a strong array of finite sets $\{G_n\}_{n \in \mathbb{N}}$ (called the computable trace), such that
  
  \begin{align*}
  (1) \quad |G_n| &\leq h(n), \text{ and} \\
  (2) \quad f(n) &\in G_n.
  \end{align*}

- Computably traceable is a uniform version of being hyperimmune-free.
Computable Traceability (I)

- An order is a total function $h : \mathbb{N} \rightarrow \mathbb{N}$, such that $h$ is recursive, non-decreasing and unbounded.

- (Terwijn and Zambella) A degree $a$ is said to be computably traceable, if there is an order $h$ such that for every $f \leq_T a$, there is a strong array of finite sets $\{G_n\}_{n \in \mathbb{N}}$ (called the computable trace), such that
  
  1. $|G_n| \leq h(n)$, and
  2. $f(n) \in G_n$.

- Computably traceable is a uniform version of being hyperimmune-free.
Theorem (Terwijn and Zambella)

If \( a \) is computably traceable, then it can be computably traced by any (arbitrarily slow growing) order.

Theorem (Terwijn and Zambella)

\( a \) is computably traceable iff \( a \) is low for Schnorr tests.
Computable Traceability (II)

Theorem (Terwijn and Zambella)

If $a$ is computably traceable, then it can be computably traced by any (arbitrarily slow growing) order.

Theorem (Terwijn and Zambella)

$a$ is computably traceable iff $a$ is low for Schnorr tests.
R.E. Traceability

- (Ishmukhametov) A is said to r.e. traceable if we replace computable by r.e. in the definition of computably traceable.
  - strong version = weak version.
  - A is said to be array recursive, if there is an $f \leq_{tt} \emptyset'$ such that $f$ dominates all $A$-recursive function.

- **Theorem (Ishmukhametov)**

  On the r.e. degrees, r.e. traceability = array recursive = having a strong minimal cover.
  - computably traceable = r.e. traceable + hyperimmune free.
R.E. Traceability

- (Ishmukhametov) $A$ is said to be r.e. traceable if we replace computable by r.e. in the definition of computably traceable.
- strong version = weak version.
- $A$ is said to be array recursive, if there is an $f \leq_{tt} \emptyset'$ such that $f$ dominates all $A$-recursive function.

Theorem (Ishmukhametov)

On the r.e. degrees, r.e. traceability = array recursive = having a strong minimal cover.

- computably traceable = r.e. traceable + hyperimmune free.
R.E. Traceability

- (Ishmukhametov) $A$ is said to be r.e. traceable if we replace computable by r.e. in the definition of computably traceable.
- strong version = weak version.
- $A$ is said to be array recursive, if there is an $f \leq_{tt} \emptyset'$ such that $f$ dominates all $A$-recursive function.

Theorem (Ishmukhametov)

On the r.e. degrees, r.e. traceability = array recursive = having a strong minimal cover.
- computably traceable = r.e. traceable + hyperimmune free.
R.E. TRACEABILITY

- (Ishmukhametov) A is said to r.e. traceable if we replace computable by r.e. in the definition of computably traceable.
- strong version = weak version.
- A is said to be array recursive, if there is an $f \leq_{tt} \emptyset'$ such that $f$ dominates all $A$-recursive function.

**Theorem (Ishmukhametov)**

On the r.e. degrees, r.e. traceability = array recursive = having a strong minimal cover.
- computably traceable = r.e. traceable + hyperimmune free.
R.E. Traceability

- (Ishmukhametov) A is said to r.e. traceable if we replace computable by r.e. in the definition of computably traceable.
- strong version = weak version.
- A is said to be array recursive, if there is an \( f \leq_{tt} \emptyset' \) such that \( f \) dominates all A-recursive function.

**Theorem (Ishmukhametov)**

On the r.e. degrees, r.e. traceability = array recursive = having a strong minimal cover.

- computably traceable = r.e. traceable + hyperimmune free.
We let $J^X(e)$ denote the value of the jump at input $e$.

(Nies) $A$ is said to be jump traceable, if

1. There exists an order $h$,
2. There exists a u.r.e sequence $\{W_{g(e)}\}_{e \in \mathbb{N}}$ (called the jump trace) that traces $J^A(e)$ with bound $h(e)$.

Modification of recursive traceability; Introduced to study lowness properties.

Keng Meng, Ng Victoria University of Wellington
We let $J^X(e)$ denote the value of the jump at input $e$.

(Nies) $A$ is said to be jump traceable, if

1. There exists an order $h$,
2. There exists a u.r.e sequence $\{W_{g(e)}\}_{e \in \mathbb{N}}$ (called the jump trace) that traces $J^A(e)$ with bound $h(e)$.

Modification of recursive traceability; Introduced to study lowness properties.
We let $J^X(e)$ denote the value of the jump at input $e$ 
$\{e\}^X(e)$.

(Nies) $A$ is said to be jump traceable, if

1. $\exists$ an order $h$,
2. $\exists$ a u.r.e sequence $\{W_{g(e)}\}_{e \in \mathbb{N}}$ (called the jump trace) 
that traces $J^A(e)$ with bound $h(e)$.

Modification of recursive traceability; 
Introduced to study lowness properties.
Lowness of the Jump Traceables

Every jump traceable set is generalized low\(_1\).
\((A' \leq_T A \oplus \emptyset')\)

**Theorem (Nies)**
An r.e. set \(A\) is jump traceable iff \(A\) is super-low.

**Theorem (Nies)**
However, neither direction holds for the \(\Delta^0_2\) sets.
Lowness of the Jump Traceables

- Every jump traceable set is generalized low$_1$.  
  \[(A' \leq_T A \oplus \emptyset')\]

**Theorem (Nies)**

An r.e. set $A$ is jump traceable iff $A$ is super-low.

**Theorem (Nies)**

However, neither direction holds for the $\Delta^0_2$ sets.
Lowness of the Jump Traceables

Every jump traceable set is generalized low₁.

\((A' \leq_T A \oplus \emptyset')\)

**Theorem (Nies)**

An r.e. set \(A\) is jump traceable iff \(A\) is super-low.

**Theorem (Nies)**

However, neither direction holds for the \(\Delta^0_2\) sets.
\[ \emptyset' \text{ is the join of two super-low r.e. sets.} \]
Hence the r.e. jump traceables do not form an ideal (unlike the \( K \)-trivials).

\[ \text{However, there is an r.e. super-low set } A, \text{ such that for all other r.e. super-low sets } W, \text{ the join } A \oplus W \text{ is still super-low.} \]
$\emptyset'$ is the join of two super-low r.e. sets.
Hence the r.e. jump traceables do not form an ideal (unlike the $K$-trivials).

However, there is an r.e. super-low set $A$, such that for all other r.e. super-low sets $W$, the join $A \oplus W$ is still super-low.
Introduction (I)

- Related work goes back to Bickford and Mills. An r.e. degree $a$ is deep if for all r.e. $b$,
  \[(a \cup b)' = b'.\]

- The search for natural definable ideals. Unfortunately, the deep r.e. degrees form a trivial ideal:

**Theorem (Lempp and Slaman)**

*The only deep r.e. degree is $0$.***
Introduction (I)

- Related work goes back to Bickford and Mills. An r.e. degree $a$ is deep if for all r.e. $b$,

$$(a \cup b)' = b'.$$

- The search for natural definable ideals.

Unfortunately, the deep r.e. degrees form a trivial ideal:

Theorem (Lempp and Slaman)

The only deep r.e. degree is $0$. 

Keng Meng, Ng Victoria University of Wellington

Traceability of the Jump
Introduction (I)

- Related work goes back to Bickford and Mills. An r.e. degree $a$ is deep if for all r.e. $b$,

$$ (a \cup b)' = b'. $$

- The search for natural definable ideals. Unfortunately, the deep r.e. degrees form a trivial ideal:

**Theorem (Lempp and Slaman)**

*The only deep r.e. degree is $0$.***
Introduction (II)

- We can modify deepness by requiring $a$ to preserve the jump on some subclass of the r.e. degrees.

- (Cholak, Groszek and Slaman) An r.e. degree $a$ is almost deep, if for every low r.e. $b$, the join $a \cup b$ is still low.

- Hence, an almost deep degree preserves the jump on all low r.e. degrees. This is not possible for low$_2$ r.e. degrees.
We can modify deepness by requiring $a$ to preserve the jump on some subclass of the r.e. degrees.

(Cholak, Groszek and Slaman) An r.e. degree $a$ is almost deep, if for every low r.e. $b$, the join $a \cup b$ is still low.

Hence, an almost deep degree preserves the jump on all low r.e. degrees. This is not possible for low$_2$ r.e. degrees.
INTRODUCTION (II)

- We can modify deepness by requiring $a$ to preserve the jump on some subclass of the r.e. degrees.

- (Cholak, Groszek and Slaman) An r.e. degree $a$ is almost deep, if for every low r.e. $b$, the join $a \cup b$ is still low.

- Hence, an almost deep degree preserves the jump on all low r.e. degrees. This is not possible for low$_2$ r.e. degrees.
Introduction (iii)

Theorem (Cholak, Groszek and Slaman)

There is a non-recursive r.e. almost deep degree.

- This gives a definable ideal (using the jump).

- We consider a modification of the almost deep, by replacing “low” with “super-low” (i.e. jump traceable).
Theorem (Cholak, Groszek and Slaman)

There is a non-recursive r.e. almost deep degree.

- This gives a definable ideal (using the jump).

- We consider a modification of the almost deep, by replacing “low” with “super-low” (i.e. jump traceable).
**Theorem**

There is a non-recursive r.e. set $A$, such that if $W$ is r.e. super-low, then $A \oplus W$ is super-low.

**Requirements.**

$P$ : (Non-recursive) If $W$ is infinite $\Rightarrow A \neq \overline{W}$,

$N$ : (Super-deepness) If $f$ approximates $W'$ with at most $g$ many mind changes, then $A \oplus W$ is super-low.
Theorem

There is a non-recursive r.e. set $A$, such that if $W$ is r.e. super-low, then $A \oplus W$ is super-low.

Requirements.

$\mathcal{P}$ : (Non-recursive) If $W$ is infinite $\Rightarrow A \neq \overline{W}$,

$\mathcal{N}$ : (Super-deepness) If $f$ approximates $W'$ with at most $g$ many mind changes, then $A \oplus W$ is super-low.
**Basic Strategy (Idea)**

- Basically we split $\mathcal{N}$ into sub-requirements $\mathcal{N}_0, \mathcal{N}_1, \cdots$, where $\mathcal{N}_e$ tries to predict if $J^{A \oplus W}(e) \downarrow$ or $J^{A \oplus W}(e) \uparrow$.

- To do this, we will control $J^W(n)$ and challenge our opponent to respond.

- If $\langle f, g \rangle$ is indeed a correct $\omega$-r.e. witness for $W'$, then our opponent has no choice but to play his predictions $f(n, s)$ as dictated by us.

Force him to make a mistake, every time we make one.
**Basic Strategy (Idea)**

- Basically we split $\mathcal{N}$ into sub-requirements $\mathcal{N}_0, \mathcal{N}_1, \cdots$, where $\mathcal{N}_e$ tries to predict if $J^{A \oplus W}(e) \downarrow$ or $J^{A \oplus W}(e) \uparrow$.

- To do this, we will control $J^W(n)$ and challenge our opponent to respond.

- If $\langle f, g \rangle$ is indeed a correct $\omega$-r.e. witness for $W'$, then our opponent has no choice but to play his predictions $f(n, s)$ as dictated by us. Force him to make a mistake, every time we make one.
Basic Strategy (Idea)

- Basically we split $\mathcal{N}$ into sub-requirements $\mathcal{N}_0, \mathcal{N}_1, \cdots$, where $\mathcal{N}_e$ tries to predict if $J^{A \oplus W}(e) \downarrow$ or $J^{A \oplus W}(e) \uparrow$.

- To do this, we will control $J^W(n)$ and challenge our opponent to respond.

- If $\langle f, g \rangle$ is indeed a correct $\omega$-r.e. witness for $W'$, then our opponent has no choice but to play his predictions $f(n, s)$ as dictated by us.

Force him to make a mistake, every time we make one.
**Basic Strategy (Idea)**

- Basically we split $\mathcal{N}$ into sub-requirements $\mathcal{N}_0, \mathcal{N}_1, \ldots$, where $\mathcal{N}_e$ tries to predict if $J^{A \oplus W}(e) \downarrow$ or $J^{A \oplus W}(e) \uparrow$.

- To do this, we will control $J^W(n)$ and challenge our opponent to respond.

- If $\langle f, g \rangle$ is indeed a correct $\omega$-r.e. witness for $W'$, then our opponent has no choice but to play his predictions $f(n, s)$ as dictated by us.

  Force him to make a mistake, every time we make one.
**Basic Strategy**

**(Step 1)** Wait for $J^{A \oplus W}(e)[s] \downarrow$.
- We place $n$ into $W'$ and restrain $A$.

**(Step 2)** Wait for opponent to switch $f(n, s)$ to predict $J^W(n) \downarrow$.
- We then follow our opponent and guess that $J^{A \oplus W}(e) \downarrow$.

**(Step 3)** The only way that we can be wrong, is for a $W$ change.
- Go back to Step 1, unless opponent has changed more than $g(n)$ times.
**Basic Strategy**

**Step 1**  \( J^{A \oplus W}(e)[s] \downarrow. \)
- We place \( n \) into \( W' \) and restraint \( A \).

**Step 2**  \( f(n, s) \) to predict \( J^W(n) \downarrow. \)
- We then follow our opponent and guess that \( J^{A \oplus W}(e) \downarrow. \)

**Step 3**  The only way that we can be wrong, is for a \( W \) change.
- Go back to Step 1, unless opponent has changed more than \( g(n) \) times.
Basic Strategy

(Step 1) Wait for $J^{A \oplus W}(e)[s] \downarrow$.
   ▶ We place $n$ into $W'$ and restrain $A$.

(Step 2) Wait for opponent to switch $f(n, s)$ to predict $J^W(n) \downarrow$.
   ▶ We then follow our opponent and guess that $J^{A \oplus W}(e) \downarrow$.

(Step 3) The only way that we can be wrong, is for a $W$ change.
   ▶ Go back to Step 1, unless opponent has changed more than $g(n)$ times.
**Strong Jump Traceability (I)**

**Theorem (Nies)**

Every $K$-trivial is jump traceable.

- This is by the golden run method; Clearly it is a proper subclass.

- In fact, if $A$ is $K$-trivial, then it can be jump traced at order $\sim n \log n$.

- This leads Figueira, Nies and Stephan to investigate new class...
**Strong Jump Traceability (I)**

**Theorem (Nies)**

*Every $K$-trivial is jump traceable.*

- This is by the golden run method; Clearly it is a proper subclass.

- In fact, if $A$ is $K$-trivial, then it can be jump traced at order $\sim n \log n$.

- This leads Figueira, Nies and Stephan to investigate new class...
(Recall) For r.e. sets, jump traceability $\equiv$ super-lowness.

Figueira et al. defined stronger forms of both, and proved that they coincide on the r.e. sets:

(Figueira, Nies and Stephan) $A$ is strongly jump traceable, if $A$ is jump traceable via all order functions.

(Figueira, Nies and Stephan) $A$ is well-approximable, if $A$ is $\omega$-r.e. via all order functions.
**Strong Jump Traceability (II)**

- (Recall) For r.e. sets, jump traceability $\equiv$ super-lowness.

- Figueira et al. defined stronger forms of both, and proved that they coincide on the r.e. sets:
  
  - (Figueira, Nies and Stephan) $A$ is strongly jump traceable, if $A$ is jump traceable via all order functions.

  - (Figueira, Nies and Stephan) $A$ is well-approximable, if $A$ is $\omega$-r.e. via all order functions.
(Recall) For r.e. sets, jump traceability $\equiv$ super-lowness.

Figueira et al. defined stronger forms of both, and proved that they coincide on the r.e. sets:

- (Figueira, Nies and Stephan) $A$ is strongly jump traceable, if $A$ is jump traceable via all order functions.

- (Figueira, Nies and Stephan) $A$ is well-approximable, if $A$ is $\omega$-r.e. via all order functions.
SOME FACTS

- The strongly jump traceables are downwards closed under $\leq_T$.

- (Nies) For r.e. sets, $A$ is strongly jump traceable iff $A'$ is well-approximable.

THEOREM (Figueira, Nies and Stephan)
There is a promptly simple, strongly jump traceable r.e. set.
Some Facts

- The strongly jump traceables are downwards closed under $\leq_T$.

- (Nies) For r.e. sets, $A$ is strongly jump traceable iff $A'$ is well-approximable.

Theorem (Figueira, Nies and Stephan)

There is a promptly simple, strongly jump traceable r.e. set.
Proof of Existence of S.J.T. (I)

- We build an r.e. $A$ satisfying the requirements:
  
  $P_e : \ |W_e| = \infty \Rightarrow A \cap W_e \neq \emptyset$,
  
  $N_e : \ h_e$ is an order $\Rightarrow A$ is jump traceable via $h_e$.

- $N_e$ will build the trace $\{V_{e,k}\}_{k \in \mathbb{N}}$.

- We describe the plan to satisfy $N_0$. 
**Proof of existence of S.J.T. (I)**

- We build an r.e. $A$ satisfying the requirements:

  $\mathcal{P}_e : \ |W_e| = \infty \Rightarrow A \cap W_e \neq \emptyset,$

  $\mathcal{N}_e : h_e$ is an order $\Rightarrow A$ is jump traceable via $h_e$.

- $\mathcal{N}_e$ will build the trace $\{V_{e,k}\}_{k \in \mathbb{N}}$.

- We describe the plan to satisfy $\mathcal{N}_0$. 
Proof of existence of S.J.T. (II)

Let $I_k = \{ x \in \mathbb{N} \mid h_0(x) = k \}.$

- If $k \in I_1,$ whenever $J^A(k)[s] \downarrow$
  Enumerate the value into $V_{0,k}$ and freeze $A$ on the use.

- If $k \in I_2,$ we could do the same.
  But we can’t increase the $A$-restraint for $I_2, I_3, \cdots.$

- But for $k \in I_2,$ we are allowed two attempts to trace $J^A(k).$
Proof of existence of S.J.T. (II)

- Let $l_k = \{ x \in \mathbb{N} \mid h_0(x) = k \}$.

- If $k \in l_1$, whenever $J^A(k)[s] \downarrow$
  Enumerate the value into $V_{0,k}$ and freeze $A$ on the use.

- If $k \in l_2$, we could do the same.
  But we can’t increase the $A$-restraint for $l_2, l_3, \ldots$.

- But for $k \in l_2$, we are allowed two attempts to trace $J^A(k)$. 
Proof of Existence of S.J.T. (II)

Let \( I_k = \{ x \in \mathbb{N} \mid h_0(x) = k \} \).

- If \( k \in I_1 \), whenever \( J^A(k)[s] \downarrow \)
  Enumerate the value into \( V_{0,k} \) and freeze \( A \) on the use.

- If \( k \in I_2 \), we could do the same. But we can’t increase the \( A \)-restraint for \( I_2, I_3, \ldots \).

- But for \( k \in I_2 \), we are allowed two attempts to trace \( J^A(k) \).
PROOF OF EXISTENCE OF S.J.T. (II)

▶ Let $I_k = \{ x \in \mathbb{N} \mid h_0(x) = k \}$.

▶ If $k \in I_1$, whenever $J^A(k)[s] \downarrow$
  Enumerate the value into $V_{0,k}$ and freeze $A$ on the use.

▶ If $k \in I_2$, we could do the same.
  But we can’t increase the $A$-restraint for $I_2, I_3, \cdots$.

▶ But for $k \in I_2$, we are allowed two attempts to trace $J^A(k)$.

Keng Meng, Ng Victoria University of Wellington
Traceability of the Jump
Proof of existence of S.J.T. (III)

- So, $\mathcal{N}_0$ will allow $\mathcal{P}_0$ below it to destroy the traced value $J^A(k)[s]$.

- And it blocks all others from doing so by increasing the $A$-restraint on $\mathcal{P}_1, \mathcal{P}_2, \cdots$.

- Similarly if $k \in I_3$ it allows $\mathcal{P}_0$ and $\mathcal{P}_1$ to injure it. Increases the $A$-restraint on $\mathcal{P}_2, \mathcal{P}_3, \cdots$.

- Each positive requirement enumerates only once (promptly), and has only finite restraint on it.
So, $\mathcal{N}_0$ will allow $\mathcal{P}_0$ below it to destroy the traced value $J^A(k)[s]$.

And it blocks all others from doing so by increasing the $A$-restraint on $\mathcal{P}_1, \mathcal{P}_2, \cdots$.

Similarly if $k \in I_3$ it allows $\mathcal{P}_0$ and $\mathcal{P}_1$ to injure it. Increases the $A$-restraint on $\mathcal{P}_2, \mathcal{P}_3, \cdots$.

Each positive requirement enumerates only once (promptly), and has only finite restraint on it.
Proof of Existence of S.J.T. (III)

▶ So, $\mathcal{N}_0$ will allow $\mathcal{P}_0$ below it to destroy the traced value $J^\mathcal{A}(k)[s]$.

▶ And it blocks all others from doing so by increasing the $\mathcal{A}$-restraint on $\mathcal{P}_1, \mathcal{P}_2, \cdots$.

▶ Similarly if $k \in l_3$ it allows $\mathcal{P}_0$ and $\mathcal{P}_1$ to injure it. Increases the $\mathcal{A}$-restraint on $\mathcal{P}_2, \mathcal{P}_3, \cdots$.

▶ Each positive requirement enumerates only once (promptly), and has only finite restraint on it.
Proof of existence of S.J.T. (III)

- So, $N_0$ will allow $P_0$ below it to destroy the traced value $J^A(k)[s]$.

- And it blocks all others from doing so by increasing the $A$-restraint on $P_1, P_2, \cdots$.

- Similarly if $k \in I_3$ it allows $P_0$ and $P_1$ to injure it. Increases the $A$-restraint on $P_2, P_3, \cdots$.

- Each positive requirement enumerates only once (promptly), and has only finite restraint on it.
Introduction
Part 1 - Tracing The Jump
Part 2 - Relativization

Some More Facts

- Being $K$-trivial implies that $A$ doesn’t help in the compression when used as oracle.

- However, we can characterize strongly jump traceables by the fact that $C^A$ is very close to $C$:

Theorem (Figueira, Nies and Stephan)

$A$ is strongly jump traceable iff

$(\forall$ orders $h)(\forall^{\infty} x) C(x) \leq C^A(x) + h(C^A(x))$. 

Keng Meng, Ng Victoria University of Wellington

Traceability of the Jump
Being $K$-trivial implies that $A$ doesn’t help in the compression when used as oracle.

However, we can characterize strongly jump traceables by the fact that $C^A$ is very close to $C$:

\[
\forall \text{ orders } h \forall \infty x \quad C(x) \leq C^A(x) + h(C^A(x)).
\]

**Theorem (Figueira, Nies and Stephan)**

$A$ is strongly jump traceable iff 
\[
(\forall \text{ orders } h)(\forall \infty x) \quad C(x) \leq C^A(x) + h(C^A(x)).
\]
**Some More Facts**

- Being $K$-trivial implies that $A$ doesn’t help in the compression when used as oracle.

- However, we can characterize strongly jump traceables by the fact that $C^A$ is very close to $C$:

**Theorem (Figueira, Nies and Stephan)**

A is strongly jump traceable iff

$$\forall \text{ orders } h)(\forall \infty x) \ C(x) \leq C^A(x) + h(C^A(x)).$$
(Recall) $K$-trivials $\subsetneq$ jump traceables.

The search for a combinatorial characterization for the $K$-trivials lead Miller and Nies to ask if

$$K\text{-trivials} \equiv \text{strongly jump traceables}.$$ 

This new class turns out to be a proper subclass of the $K$-trivials:

**Theorem (Cholak, Downey and Greenberg)**

A is r.e. and strongly jump traceable $\Rightarrow$ A is $K$-trivial.
A PROPER SUBCLASS OF THE \( K \)-TRIVIALS (I)

- (Recall) \( K \)-trivials \( \subset \) jump traceables.
- The search for a combinatorial characterization for the \( K \)-trivials lead Miller and Nies to ask if

\[
\text{\( K \)-trivials} \overset{?}{=} \text{strongly jump traceables.}
\]

- This new class turns out to be a proper subclass of the \( K \)-trivials:

**Theorem (Cholak, Downey and Greenberg)**

\( A \) is r.e. and strongly jump traceable \( \Rightarrow \) \( A \) is \( K \)-trivial.
A Proper Subclass of the \( K \)-trivials (I)

▷ (Recall) \( K \)-trivials \( \subsetneq \) jump traceables.

▷ The search for a combinatorial characterization for the \( K \)-trivials lead Miller and Nies to ask if

\[
\text{\( K \)-trivials } \equiv \text{ strongly jump traceables.}
\]

▷ This new class turns out to be a proper subclass of the \( K \)-trivials:

**Theorem (Cholak, Downey and Greenberg)**

\[
\text{A is r.e. and strongly jump traceable } \Rightarrow \text{ A is } K \text{-trivial.}
\]
A Proper Subclass of the \( K \)-trivials (II)

**Theorem (Cholak, Downey and Greenberg)**

There is a \( K \)-trivial that is not strongly jump traceable
(at order \( \sim \log \log n \)).

- This suggests that some combinatorial characterization
  might still be found at some growth rate between the two.

- It is an interesting open question as to whether or not the
  strongly jump traceables form an ideal (just like the
  \( K \)-trivials).
A PROPER SUBCLASS OF THE $K$-TRIVIALS (II)

**Theorem (Cholak, Downey and Greenberg)**

There is a $K$-trivial that is not strongly jump traceable (at order $\sim \log \log n$).

- This suggests that some combinatorial characterization might still be found at some growth rate between the two.

- It is an interesting open question as to whether or not the strongly jump traceables form an ideal (just like the $K$-trivials).
**A Proper Subclass of the \(K\)-trivials (II)**

**Theorem (Cholak, Downey and Greenberg)**

*There is a \(K\)-trivial that is not strongly jump traceable (at order \(\sim \log \log n\)).*

- This suggests that some combinatorial characterization might still be found at some growth rate between the two.

- It is an interesting open question as to whether or not the strongly jump traceables form an ideal (just like the \(K\)-trivials).
**Theorem**

There is a non-recursive r.e. set $A$, such that if $W$ is r.e. strongly jump traceable, then so is $A \oplus W$.

**Proof.**

Follows more or less from the uniformity of the proof of an almost super deep.

i.e. If $\langle f, g \rangle$ is a correct $\omega$-r.e. witness for $W'$, we have made $(A \oplus W)' \omega$-r.e. with bound $\hat{g}$.
A Proper Subclass of the $K$-trivials (III)

**Theorem**

There is a non-recursive r.e. set $A$, such that if $W$ is r.e. strongly jump traceable, then so is $A \oplus W$.

**Proof.**

Follows more or less from the uniformity of the proof of an almost super deep.

i.e. If $\langle f, g \rangle$ is a correct $\omega$-r.e. witness for $W'$, we have made $(A \oplus W)' \omega$-r.e. with bound $\hat{g}$.
Introduction

The most natural example of a non recursive set is $\emptyset'$.
Relativization produces $X \mapsto X'$ (r.e. in and above $X$).

Generalization of this notion:
For each $e$, the $e^{th}$ pseudo-jump operator

$$H_e : X \mapsto X \oplus W_e^X.$$

In some sense, $H_e(X)$ relates to $X$ in the same way as how $W_e$ relates to $\emptyset$. 
The most natural example of a non recursive set is $\emptyset'$. Relativization produces $X \mapsto X'$ (r.e. in and above $X$).

Generalization of this notion:
For each $e$, the $e^{th}$ pseudo-jump operator

$$H_e : X \mapsto X \oplus W_e^X.$$

In some sense, $H_e(X)$ relates to $X$ in the same way as how $W_e$ relates to $\emptyset$. 
The most natural example of a non recursive set is $\emptyset'$. Relativization produces $X \mapsto X'$ (r.e. in and above $X$).

Generalization of this notion:
For each $e$, the $e^{th}$ pseudo-jump operator

$$H_e : X \mapsto X \oplus W_e^X.$$ 

In some sense, $H_e(X)$ relates to $X$ in the same way as how $W_e$ relates to $\emptyset$. 

Keng Meng, Ng Victoria University of Wellington
We say that $A$ completes the pseudo-jump operator $H_e$, if $H_e(A) \equiv_T \emptyset'$.

**Theorem (Jockusch and Shore)**
Every pseudo-jump operator has an r.e. non-recursive completion.

- Implies that $\emptyset'$ always has the relativized property.
  - Relativize the construction of a low r.e. set $\Rightarrow$ High r.e. A.
  - High construction $\Rightarrow$ Get a low$_2$ set.
  - Jump traceable construction $\Rightarrow$ Get a super-high set.
We say that $A$ completes the pseudo-jump operator $H_e$, if $H_e(A) \equiv_T \emptyset'$.

**Theorem (Jockusch and Shore)**

*Every pseudo-jump operator has an r.e. non-recursive completion.*

- Implies that $\emptyset'$ always has the relativized property.
  - Relativize the construction of a low r.e. set $\Rightarrow$ High r.e. $A$.
  - High construction $\Rightarrow$ Get a low$_2$ set.
  - Jump traceable construction $\Rightarrow$ Get a super-high set.
Pseudo-jump Inversion (I)

We say that $A$ completes the pseudo-jump operator $H_e$, if $H_e(A) \equiv_T \emptyset'$.

Theorem (Jockusch and Shore)
Every pseudo-jump operator has an r.e. non-recursive completion.

- Implies that $\emptyset'$ always has the relativized property.
  - Relativize the construction of a low r.e. set $\Rightarrow$ High r.e. $A$.
  - High construction $\Rightarrow$ Get a low$_2$ set.
  - Jump traceable construction $\Rightarrow$ Get a super-high set.
**Pseudo-jump Inversion (I)**

- We say that $A$ completes the pseudo-jump operator $H_e$, if $H_e(A) ≡_T \emptyset'$.

**Theorem (Jockusch and Shore)**

*Every pseudo-jump operator has an r.e. non-recursive completion.*

- Implies that $\emptyset'$ always has the relativized property.
  - Relativize the construction of a low r.e. set $\Rightarrow$ High r.e. $A$.
  - High construction $\Rightarrow$ Get a low$_2$ set.
  - Jump traceable construction $\Rightarrow$ Get a super-high set.
We say that $A$ completes the pseudo-jump operator $H_e$, if $H_e(A) \equiv_T \emptyset'$. 

**Theorem (Jockusch and Shore)**

*Every pseudo-jump operator has an r.e. non-recursive completion.*

- Implies that $\emptyset'$ always has the relativized property.
  - Relativize the construction of a low r.e. set $\Rightarrow$ High r.e. $A$.
  - High construction $\Rightarrow$ Get a low$_2$ set.
  - Jump traceable construction $\Rightarrow$ Get a super-high set.
Pseudo-Jump Inversion (II)

- Look at the relativizations of the two classes
  strongly jump traceables $\subset K$-trivials.

- $A$ is almost complete if $\emptyset'$ is $K$-trivial relative to it.
  
  $K^A(\emptyset'|_n) \leq K^A(n) + O(1)$ for all $n$.

- $A$ is ultra-high if $\emptyset'$ is strongly jump traceable relative to it.

- Their existence + incompleteness follow from the pseudo-jump theorem.
PSEUDO-JUMP INVERSION (II)

- Look at the relativizations of the two classes

  strongly jump traceables $\subset K$-trivials.

- $A$ is almost complete if $\emptyset'$ is $K$-trivial relative to it.

  $$K^A(\emptyset' \upharpoonright n) \leq K^A(n) + O(1) \text{ for all } n.$$  

  $A$ is ultra-high if $\emptyset'$ is strongly jump traceable relative to it.

- Their existence + incompleteness follow from the pseudo-jump theorem.
PSEUDO-JUMP INVERSION (II)

▶ Look at the relativizations of the two classes

strongly jump traceables $\not\subseteq K$-trivials.

▶ $A$ is almost complete if $\emptyset'$ is $K$-trivial relative to it.

$$K^A(\emptyset'|n) \leq K^A(n) + O(1) \text{ for all } n.$$  

$A$ is ultra-high if $\emptyset'$ is strongly jump traceable relative to it.

▶ Their existence + incompleteness follow from the pseudo-jump theorem.
Introduction

(Dobrinen and Simpson) \( a \) is uniformly almost everywhere (u.a.e.) dominating if there is \( f \leq_T a \), such that

\[
\mu \left( \{ X \in 2^\omega : \forall g \leq_T X \Rightarrow g \leq^* f \} \right) = 1
\]

Much stronger than a dominant function.

(Martin) Degrees computing dominant functions are just the high ones.

How about for u.a.e. domination? At least high.
Introduction

(Dobrinen and Simpson) $a$ is uniformly almost everywhere (u.a.e.) dominating if there is $f \leq_T a$, such that

$$
\mu\left(\{X \in 2^\omega : \forall g \leq_T X \Rightarrow g \leq^* f\}\right) = 1
$$

Much stronger than a dominant function.

(Martin) Degrees computing dominant functions are just the high ones.

How about for u.a.e. domination? At least high.
Characterizing the u.a.e. dominating (I)

- (Kurtz) There is such a function of degree $0'$. Hence every degree $\geq 0'$ is u.a.e. dominating.

- (Dobrinen and Simpson) Conjecture:
  u.a.e. dominating degrees = high degrees?
  u.a.e. dominating degrees = complete degrees?

- Answer: Somewhere in the middle.
(Kurtz) There is such a function of degree $0'$. Hence every degree $\geq 0'$ is u.a.e. dominating.

(Dobrinen and Simpson) Conjecture:
- u.a.e. dominating degrees = high degrees?
- u.a.e. dominating degrees = complete degrees?

Answer: Somewhere in the middle.
CHARACTERIZING THE U.A.E. DOMINATING (I)

- (Kurtz) There is such a function of degree $0'$. Hence every degree $\geq 0'$ is u.a.e. dominating.

- (Dobrinen and Simpson) Conjecture: u.a.e. dominating degrees = high degrees? u.a.e. dominating degrees = complete degrees?

- Answer: Somewhere in the middle.
Characterizing the u.a.e. dominating (ii)

- (Cholak, Greenberg and Miller) There is such a function of degree $< 0'$. So, u.a.e. dominating degrees $\neq$ complete degrees.

- (Binns et al., indeptly Greenberg and Miller) There is a high degree that is not u.a.e. dominating. So, u.a.e. dominating degrees $\neq$ high degrees.
Characterizing the u.a.e. dominating (II)

- (Cholak, Greenberg and Miller)
  There is such a function of degree $\prec 0'$. 
  So, u.a.e. dominating degrees $\neq$ complete degrees.

- (Binns et al., indeptly Greenberg and Miller)
  There is a high degree that is not u.a.e. dominating. 
  So, u.a.e. dominating degrees $\neq$ high degrees.
Characterizing the u.a.e. dominating (iii)

Theorem (Binns et al. via work of Nies)

If $A \in \Delta^0_2$ and has u.a.e. dominating degree
$\Rightarrow A$ is almost complete.

Corollary

u.a.e. dominating degrees $\neq$ high degrees.

Proof.

Every almost complete is super-high.
Theorem (Binns et al. via work of Nies)

If $A \in \Delta_2^0$ and has u.a.e. dominating degree

$\Rightarrow$ $A$ is almost complete.

Corollary

u.a.e. dominating degrees $\neq$ high degrees.

Proof.

Every almost complete is super-high.
**Theorem (Binns et al.)**

*u.a.e. dominating = almost complete.*

**Corollary**

*u.a.e. dominating degrees ≠ complete degrees.*

**Proof.**

There is an incomplete, almost complete degree.

---

**Traceability of the Jump**
Theorem (Binns et al.)

* u.a.e. dominating = almost complete.

Corollary

* u.a.e. dominating degrees ≠ complete degrees.

Proof.

There is an incomplete, almost complete degree.
Almost Complete Degrees

Theorem (Nies and Shore)
There is an r.e. almost complete $A$, and an r.e. $K$-trivial $B$ such that $A \nleq_T B$.

- Can we avoid upper cones (of r.e. sets) in general?

- (Barmpalias) Constructed a cappable almost complete.

- The more general question:
Can we make an r.e. minimal pair of almost complete?
Theorem (Nies and Shore)

There is an r.e. almost complete $A$, and an r.e. $K$-trivial $B$ such that $A \nless_T B$.

- Can we avoid upper cones (of r.e. sets) in general?

- (Barmpalias) Constructed a cappable almost complete.

- The more general question:
  Can we make an r.e. minimal pair of almost complete?
Almost Complete Degrees

Theorem (Nies and Shore)
There is an r.e. almost complete A, and an r.e. K-trivial B such that \( A \nind_T B \).

- Can we avoid upper cones (of r.e. sets) in general?

- (Barmpalias) Constructed a cappable almost complete.

- The more general question:
  Can we make an r.e. minimal pair of almost complete?
THEOREM

There is an r.e. minimal pair of super-high.

REQUIREMENTS.

We need to enumerate $A$, $B$ and $tt$-functionals $\Gamma, \Delta$ satisfying the requirements:

$\mathcal{N}_e$ : If $\Phi^A_e = \Phi^B_e = h$ is total, then $h$ is recursive,

$\mathcal{P}^A_e$ : $\text{Tot}(e) = \Gamma^{A'}(e)$,

$\mathcal{P}^B_e$ : $\text{Tot}(e) = \Delta^{B'}(e)$.

where $\text{Tot} = \{ i \mid i^{th} \text{ partial rec. function is total} \}$. 
Requirements

**Theorem**

There is an r.e. minimal pair of super-high.

**Requirements.**

We need to enumerate $A$, $B$ and $tt$-functionals $\Gamma, \Delta$ satisfying the requirements:

- $\mathcal{N}_e$: If $\Phi^A_e = \Phi^B_e = h$ is total, then $h$ is recursive,
- $\mathcal{P}^A_e$: $\text{Tot}(e) = \Gamma^A'(e)$,
- $\mathcal{P}^B_e$: $\text{Tot}(e) = \Delta^B'(e)$.

where $\text{Tot} = \{i \mid i^{th} \text{ partial rec. function is total}\}$. 
The Positive Strategy (I)

- $P^A_e$ builds $\Gamma^A(e)$:
  Wants to control the configuration of an initial segment of $A'$.

- Fix two numbers $\eta_f$ and $\eta_\infty$ targeted for $A'$.
  
  - Every time $\text{Tot}(e)$ looks like 0,
    Put $\eta_f$ into $A'[s]$ with some use $\geq s$.
  
  - Every time $\text{Tot}(e)$ looks like 1,
    Take $\eta_f$ out of $A'[s + 1]$ by making an enumeration into $A$,
    Put $\eta_\infty$ into $A'[s]$.
THE POSITIVE STRATEGY (I)

- $P^A_e$ builds $\Gamma^A'(e)$:
  Wants to control the configuration of an initial segment of $A'$.

- Fix two numbers $\eta_f$ and $\eta_\infty$ targetted for $A'$.
  - Every time $\text{Tot}(e)$ looks like 0,
    Put $\eta_f$ into $A'[s]$ with some use $> s$.
  - Every time $\text{Tot}(e)$ looks like 1,
    Take $\eta_f$ out of $A'[s + 1]$ by making an enumeration into $A$.
    Put $\eta_\infty$ into $A'[s]$.  

Keng Meng, Ng Victoria University of Wellington  
Traceability of the Jump
THE POSITIVE STRATEGY (I)

- $\mathcal{P}_e^A$ builds $\Gamma^A(e)$:
  Wants to control the configuration of an initial segment of $A'$.

- Fix two numbers $\eta_f$ and $\eta_\infty$ targeted for $A'$.
  - Every time $\text{Tot}(e)$ looks like 0,
    Put $\eta_f$ into $A'[s]$ with some use $> s$.
  - Every time $\text{Tot}(e)$ looks like 1,
    Take $\eta_f$ out of $A'[s + 1]$ by making an enumeration into $A$.
    Put $\eta_\infty$ into $A'[s]$. 
**THE POSITIVE STRATEGY (I)**

- $P_e^A$ builds $\Gamma^A'(e)$:
  Wants to control the configuration of an initial segment of $A'$.

- Fix two numbers $\eta_f$ and $\eta_\infty$ targetted for $A'$.

  - Every time $Tot(e)$ looks like 0,
    Put $\eta_f$ into $A'[s]$ with some use $> s$.

  - Every time $Tot(e)$ looks like 1,
    Take $\eta_f$ out of $A'[s+1]$ by making an enumeration into $A$.
    Put $\eta_\infty$ into $A'[s]$. 
THE POSITIVE STRATEGY (II)

- If $\text{Tot}(e)$ plays the $\Sigma_2$ outcome, then
  \[ A'(\eta_f)A'(\eta_\infty) = 1x. \]

- If $\text{Tot}(e)$ plays the $\Pi_2$ outcome, then
  \[ A'(\eta_f)A'(\eta_\infty) = 01. \]

- The same strategy is used for the $B$ requirements.
THE POSITIVE STRATEGY (II)

- If $\text{Tot}(e)$ plays the $\Sigma_2$ outcome, then
  
  \[ A'(\eta_f)A'(\eta_\infty) = 1x. \]

- If $\text{Tot}(e)$ plays the $\Pi_2$ outcome, then

  \[ A'(\eta_f)A'(\eta_\infty) = 01. \]

- The same strategy is used for the $B$ requirements.
The Negative Requirements

- The negative requirement $\mathcal{N}$ will want to prevent enumerations into $A$ or $B$ to preserve the common value

$$\Phi^A(x)[s] = \Phi^B(x)[s].$$

- At times $\mathcal{P}^A$ might want to take $\eta_f$ out $A'$. He can’t do it if some $\Phi^A(x)$ computation has converged after $\eta_f$ was put into $A'$, and $\mathcal{N}$ wants to preserve that.

- Solution: Is to let $\mathcal{P}^A$ control four bits of $A'$ (instead of just two bits).

$(\eta^A_f, \eta^A_\infty)$ used when $\mathcal{N}$ is not holding $A$-restraint.

$(\eta^B_f, \eta^B_\infty)$ used when $\mathcal{N}$ is holding $A$-restraint.
THE NEGATIVE REQUIREMENTS

- The negative requirement $\mathcal{N}$ will want to prevent enumerations into $A$ or $B$ to preserve the common value

$$\Phi^A(x)[s] = \Phi^B(x)[s].$$

- At times $\mathcal{P}^A$ might want to take $\eta_f$ out $A'$. He can’t do it if some $\Phi^A(x)$ computation has converged after $\eta_f$ was put into $A'$, and $\mathcal{N}$ wants to preserve that.

- Solution: Is to let $\mathcal{P}^A$ control four bits of $A'$ (instead of just two bits).
  
  $(\eta^A_f, \eta^A_\infty)$ used when $\mathcal{N}$ is not holding $A$-restraint.
  
  $(\eta^B_f, \eta^B_\infty)$ used when $\mathcal{N}$ is holding $A$-restraint.
The Negative Requirements

- The negative requirement $\mathcal{N}$ will want to prevent enumerations into $A$ or $B$ to preserve the common value

$$\phi^A(x)[s] = \phi^B(x)[s].$$

- At times $\mathcal{P}^A$ might want to take $\eta_f$ out $A'$. He can’t do it if some $\phi^A(x)$ computation has converged after $\eta_f$ was put into $A'$, and $\mathcal{N}$ wants to preserve that.

- Solution: Is to let $\mathcal{P}^A$ control four bits of $A'$ (instead of just two bits).

$(\eta_f^A, \eta_\infty^A)$ used when $\mathcal{N}$ is not holding $A$-restraint.

$(\eta_f^B, \eta_\infty^B)$ used when $\mathcal{N}$ is holding $A$-restraint.
Generally if $\mathcal{P}^A$ lives below $e$ negative requirements, it will need code $\text{Tot}(e)$ into $A'$ using a truth table of width $\sim 2^e$.

Hence $\emptyset'$ would be jump traceable relative to $A$ (as well as $B$) via order $\sim 2^e$.

This is a very generous bound. Can we do better?
Generally if $P^A$ lives below $e$ negative requirements, it will need code $\text{Tot}(e)$ into $A'$ using a truth table of width $\sim 2^e$.

Hence $\emptyset'$ would be jump traceable relative to $A$ (as well as $B$) via order $\sim 2^e$.

This is a very generous bound. Can we do better?
Consider a minimal pair requirement $\mathcal{N}$ and an $A$-positive requirement $\mathcal{P}^A$.

$\mathcal{N}$ is building a recursive function $h$ that captures the common value of $\Phi^A = \Phi^B = h$. 
THE SECOND PROBLEM (II)

\[ \mathcal{N} \text{ preserving } A \text{ computations} \]

\[ \text{dom}(h)[s] \]

B-use

A-use

Keng Meng, Ng Victoria University of Wellington
THE SECOND PROBLEM (II)

$\mathcal{P}^A$ puts $\eta^B_f$ into $A'$

$B$

$A$

$\eta^B_f$-use

$A$-use

$dom(h)[s]$

$B$-use
THE SECOND PROBLEM (II)

$\mathcal{N}$ recovers

\[
\begin{align*}
B & \quad A \\
& \quad \downarrow \downarrow \\
& \quad \uparrow \uparrow \\
& \quad \text{dom}(h)[s]
\end{align*}
\]

- $\eta_f^B$-use
- $\eta_f^A$-use
- B-use
- A-use

Keng Meng, Ng Victoria University of Wellington
Traceability of the Jump
THE SECOND PROBLEM (II)

$\mathcal{N}$ recovers

Solution: Delay extending $\text{dom}(h)$ until the computation is “believable”
Given a pseudo-jump operation, what are the properties of the completing sets?

**Theorem (Coles, Downey, Jockusch and LaForte)**

Any pseudo-jump operator $V$, such that $\forall r.e. X (X <_T V^X)$, always has Turing incomparable r.e. completions.

They also showed that if $V$ is non-trivial over the d.r.e. sets, then $V$ has a proper d.r.e. completion.
Pseudo-jump Completions

- Given a pseudo-jump operation, what are the properties of the completing sets?

**Theorem (Coles, Downey, Jockusch and LaForte)**

Any pseudo-jump operator $V$, such that $\forall r.e. X (X <_T V^X)$, always has Turing incomparable r.e. completions.

- They also showed that if $V$ is non-trivial over the d.r.e. sets, then $V$ has a proper d.r.e. completion.
Theorem (Impossibility of Cone Avoidance)

There is a pseudo-jump operator $V$, and an r.e. $C > T \emptyset$, such that

(I) $\forall$ r.e. $X$ ($X < T V^X$),

(II) If $V^W_e \equiv_T \emptyset'$, then $W_e \geq_T C$.

It is not known if the theorem can be strengthened to make $V$ non-trivial on all oracles.
(since any natural operator must be so).
THEOREM (IMPOSSIBILITY OF CONE AVOIDANCE)
There is a pseudo-jump operator $V$, and an r.e. $C >_T \emptyset$, such that

(I) $\forall$ r.e. $X$ ($X <_T V^X$),
(II) If $V^{W_e} \equiv_T \emptyset'$, then $W_e \geq_T C$.

It is not known if the theorem can be strengthened to make $V$ non-trivial on all oracles. (since any natural operator must be so).
Further Questions

- Take any pseudo-jump operator $V$ non-trivial on the r.e. sets.
- Since we can’t avoid cones, there cannot be a minimal pair of completions.
- But can we have a cappable completion?
- The ultra-high is a (natural) pseudo-jump operator, which had hopes of possibly having no cappable completion.
**FURTHER QUESTIONS**

- Take any pseudo-jump operator $V$ non-trivial on the r.e. sets.
- Since we can’t avoid cones, there cannot be a minimal pair of completions.
- But can we have a cappable completion?
- The ultra-high is a (natural) pseudo-jump operator, which had hopes of possibly having no cappable completion.
FURTHER QUESTIONS

- Take any pseudo-jump operator $V$ non-trivial on the r.e. sets.
- Since we can’t avoid cones, there cannot be a minimal pair of completions.
- But can we have a cappable completion?
- The ultra-high is a (natural) pseudo-jump operator, which had hopes of possibly having no cappable completion.
THEOREM

There is an r.e. ultra-high set which is half a minimal pair.

REQUIREMENTS.

We build the r.e. sets $A$ and $B$, satisfying the following requirements.

- $\mathcal{N}_e$: If $\Phi_e^A = \Phi_e^B = h$ is total, then $h$ is recursive,
- $\mathcal{P}_e^A$: If $\Phi_e^A$ is an order, make $\emptyset'$ $A$-jump-traceable via $\Phi_e^A$,
- $\mathcal{P}_e^B$: $|W_e| = \infty$, make $B \cap W_e \neq \emptyset$. 

Keng Meng, Ng Victoria University of Wellington
**Theorem**

There is an r.e. ultra-high set which is half a minimal pair.

**Requirements.**

We build the r.e. sets $A$ and $B$, satisfying the following requirements.

- $\mathcal{N}_e$: If $\Phi_e^A = \Phi_e^B = h$ is total, then $h$ is recursive,
- $\mathcal{P}_e^A$: If $\Phi_e^A$ is an order, make $\emptyset'$ $A$-jump-traceable via $\Phi_e^A$,
- $\mathcal{P}_e^B$: $|W_e| = \infty$, make $B \cap W_e \neq \emptyset$. 
**A-Positive Strategy (I)**

- If $\Phi_e^A$ is an order, we need to build a trace
  
  \[
  \{ V_k^A \}_{k \in \mathbb{N}} \text{ - an } A\text{-u.r.e. sequence,}
  \]

  such that for all $k$,

  (I) \( |V_k^A| \leq \Phi_e^A(k) \), and

  (II) \( J^{\emptyset'}(k) \in V_k^A \).

- Divide $P_e^A$ into infinitely many subrequirements.
  Each subrequirement responsible for “tracing” $J^{\emptyset'}(k)$ for a few $k$’s.
A-Positive Strategy (I)

- If $\Phi^A_\emptyset$ is an order, we need to build a trace
  \[
  \{ V^A_k \}_{k \in \mathbb{N}} - \text{an } A\text{-u.r.e. sequence},
  \]
  such that for all $k$,

  (I) \( |V^A_k| \leq \Phi^A_\emptyset(k) \), and
  (II) \( J^{\emptyset'}(k) \in V^A_k \).

- Divide $P^A_\emptyset$ into infinitely many subrequirements. Each subrequirement responsible for “tracing” \( J^{\emptyset'}(k) \) for a few $k$’s.
A-Positive Strategy (II)

Suppose $\mathcal{P}_{e,i}^A$ is responsible for tracing $J^\emptyset'(k)$.

Every time we see $J^\emptyset'(k)[s] \downarrow$, we enumerate the value into $V^A_k$ with $A$-use $> s$.

However, the value $J^\emptyset'(k)[s]$ might change. So, we need make enumerations into $A$ to clear $V^A_k$. Replace it with the new value.

So, each $A$-subrequirement needs to make infinitely many enumerations into $A$.
A-Positive Strategy (II)

- Suppose $P_{e,i}^A$ is responsible for tracing $J^0(k)$.

- Every time we see $J^0(k)[s] \downarrow$, we enumerate the value into $V_k^A$ with $A$-use $> s$.

- However, the value $J^0(k)[s]$ might change. So, we need make enumerations into $A$ to clear $V_k^A$. Replace it with the new value.

- So, each $A$-subrequirement needs to make infinitely many enumerations into $A$. 

---

Keng Meng, Ng Victoria University of Wellington
Traceability of the Jump
Suppose $\mathcal{P}^A_{e,i}$ is responsible for tracing $J^{\emptyset'}(k)$.

Every time we see $J^{\emptyset'}(k)[s] \downarrow$, we enumerate the value into $V^A_k$ with $A$-use $> s$.

However, the value $J^{\emptyset'}(k)[s]$ might change. So, we need make enumerations into $A$ to clear $V^A_k$. Replace it with the new value.

So, each $A$-subrequirement needs to make infinitely many enumerations into $A$. 

Keng Meng, Ng Victoria University of Wellington
Suppose $\mathcal{P}_{e,i}^A$ is responsible for tracing $J^\emptyset'(k)$.

Every time we see $J^\emptyset'(k)[s] \downarrow$, we enumerate the value into $V_k^A$ with $A$-use $> s$.

However, the value $J^\emptyset'(k)[s]$ might change. So, we need make enumerations into $A$ to clear $V_k^A$. Replace it with the new value.

So, each $A$-subrequirement needs to make infinitely many enumerations into $A$. 
CONFLICTS

- If we could always do this, then $|V^A_k| = 1$.

- Sometimes, we have to leave a wrong trace in $V^A_k$. Think of each location as a “box”. Hence we are allocated $\Phi^A_\epsilon(k)$ many boxes.

- Unlike the super-high case: our tracing order was $2^k$. In this case our tracing order is $\Phi^A_\epsilon(k)$ - arbitrarily slow-growing.

- Save on the number of boxes used.
If we could always do this, then $|V^A_k| = 1$.

Sometimes, we have to leave a wrong trace in $V^A_k$. Think of each location as a “box”. Hence we are allocated $\Phi^A_\varphi(k)$ many boxes.

Unlike the super-high case: our tracing order was $2^k$. In this case our tracing order is $\Phi^A_\varphi(k)$ - arbitrarily slow-growing.

Save on the number of boxes used.
CONFLICTS

- If we could always do this, then $|V_k^A| = 1$.

- Sometimes, we have to leave a wrong trace in $V_k^A$. Think of each location as a "box". Hence we are allocated $\Phi_e^A(k)$ many boxes.

- Unlike the super-high case: our tracing order was $2^k$. In this case our tracing order is $\Phi_e^A(k)$ - arbitrarily slow-growing.

- Save on the number of boxes used.
If we could always do this, then $|V_k^A| = 1$.

Sometimes, we have to leave a wrong trace in $V_k^A$. Think of each location as a “box”. Hence we are allocated $\Phi_e^A(k)$ many boxes.

Unlike the super-high case: our tracing order was $2^k$. In this case our tracing order is $\Phi_e^A(k)$ - arbitrarily slow-growing.

Save on the number of boxes used.
Suppose $\mathcal{P}_{e,i}^A$ lies on level 2:

$\begin{align*}
\mathcal{P}^B \\
\mathcal{N} & \quad \mathcal{N} \\
\mathcal{P}_{e,i}^A & \quad \mathcal{P}_{e,i}^A & \quad \mathcal{P}_{e,i}^A & \quad \mathcal{P}_{e,i}^A
\end{align*}$

And we are only allowed 2 boxes (instead of 4).
Suppose $\mathcal{P}_{e,i}^A$ lies on level 2:

\[
\begin{align*}
\mathcal{P}^B & \quad \mathcal{N} & \quad \hat{\mathcal{N}} \\
\mathcal{P}_{e,i}^A & \quad \mathcal{P}_{e,i}^A & \quad \mathcal{P}_{e,i}^A & \quad \mathcal{P}_{e,i}^A
\end{align*}
\]

And we are only allowed 2 boxes (instead of 4).
**Box Saving Plan**

- Idea: Allow negative $B$-restraint to transfer sideways from left to right.
- Observation: Both $\mathcal{N}$ and $\mathcal{N}$ are measuring the same length of agreement.

Traceability of the Jump

Keng Meng, Ng Victoria University of Wellington
Box Saving Plan

- Idea: Allow negative $B$-restraint to transfer sideways from left to right.
- Observation: Both $\mathcal{N}$ and $\mathcal{\hat{N}}$ are measuring the same length of agreement.
Suppose $\mathcal{N}$ was visited, and $\mathcal{P}_{e,i}^A$ fills up the two boxes allocated to it.
Box Saving Plan

▶ Suppose $\mathcal{N}$ was visited, and $\mathcal{P}^A_{e,i}$ fills up the two boxes allocated to it.

▶ When $\mathcal{\hat{N}}$ is next visited, the versions of $\mathcal{P}^A_{e,i}$ believing in $\mathcal{\hat{N}}$ have no boxes left to use (if $\mathcal{N}$ is holding $\hat{A}$-restraint).
Box Saving Plan

But \( \hat{\mathcal{N}} \) can wait until the all the \( B \)-computations have recovered up to whatever \( \mathcal{N} \) is preserving.

Now restraint \( B \) on the use of these computations. \( \mathcal{N} \) no longer cares what we put in \( A \).
But $\hat{N}$ can wait until the all the $B$-computations have recovered up to whatever $N$ is preserving.

Now restraint $B$ on the use of these computations. $N$ no longer cares what we put in $A$.  

Keng Meng, Ng Victoria University of Wellington  
Traceability of the Jump
This plan is wrecked every time $\mathcal{P}^B$ makes an enumeration. Only finitely often.

If $\mathcal{P}^B$ makes infinitely many enumerations, such as making $B$ high, no longer works.
This plan is wrecked every time $\mathcal{P}^B$ makes an enumeration. Only finitely often.

If $\mathcal{P}^B$ makes infinitely many enumerations, such as making $B$ high, no longer works.
Further Questions

- Can we make a high cappable ultra-high?
- Is there a minimal pair of ultra-high? Or almost complete?
- In general, a minimal pair of super-high via any arbitrary order?
Can we make a high cappable ultra-high?

Is there a minimal pair of ultra-high? Or almost complete?

In general, a minimal pair of super-high via any arbitrary order?
Further Questions

- Can we make a high cappable ultra-high?

- Is there a minimal pair of ultra-high? Or almost complete?

- In general, a minimal pair of super-high via any arbitrary order?
Thank the organizers for this workshop.
Audience for their attention.