Seetapun's Theorem revisited

Yang Yue

Department of Mathematics
National University of Singapore

January 28, 2016

Ramsey's Theorem

Definition

For $A \subseteq \mathbb{N}$, let $[A]^n$ denote the set of all n-element subsets of A.

Theorem (Ramsey, 1930)

Suppose $f : [\mathbb{N}]^n \to \{0, 1, \ldots, k-1\}$. Then there is an infinite set $H \subseteq \mathbb{N}$ such that f is a constant on $[H]^n$.

This version is denoted by RT^n_k.

(often described in terms of coloring/homogeneous and problem/solution.)

Ramsey's Theorem

Definition

For $A \subseteq \mathbb{N}$, let $[A]^n$ denote the set of all n-element subsets of A.

Theorem (Ramsey, 1930)

Suppose $f : [\mathbb{N}]^n \to \{0, 1, \ldots, k-1\}$. Then there is an infinite set $H \subseteq \mathbb{N}$ such that f is a constant on $[H]^n$.

This version is denoted by RT^n_k.

(often described in terms of coloring/homogeneous and problem/solution.)

Ramsey's Theorem

Definition

For $A \subseteq \mathbb{N}$, let $[A]^n$ denote the set of all n-element subsets of A.

Theorem (Ramsey, 1930)

Suppose $f : [\mathbb{N}]^n \to \{0, 1, \ldots, k-1\}$. Then there is an infinite set $H \subseteq \mathbb{N}$ such that f is a constant on $[H]^n$.

This version is denoted by RT^n_k.

(often described in terms of coloring/homogeneous and problem/solution.)

Ramsey's Theorem

Definition

For $A \subseteq \mathbb{N}$, let $[A]^n$ denote the set of all n-element subsets of A.

Theorem (Ramsey, 1930)

Suppose $f : [\mathbb{N}]^n \to \{0, 1, \ldots, k-1\}$. Then there is an infinite set $H \subseteq \mathbb{N}$ such that f is a constant on $[H]^n$.

This version is denoted by RT^n_k.

(often described in terms of coloring/homogeneous and problem/solution.)

Goal(s) of Reverse Mathematics

- Goal: What set existence axioms are needed to prove the theorems of ordinary, classical (countable) mathematics?
- Other motivations: Analyze the relative strength of mathematical theorems; and
- discover new proofs and insights.

Goal(s) of Reverse Mathematics

- Goal: What set existence axioms are needed to prove the theorems of ordinary, classical (countable) mathematics?
- Other motivations: Analyze the relative strength of mathematical theorems; and
- discover new proofs and insights.

Goal(s) of Reverse Mathematics

- Goal: What set existence axioms are needed to prove the theorems of ordinary, classical (countable) mathematics?
- Other motivations: Analyze the relative strength of mathematical theorems; and
- discover new proofs and insights.

Measuring Strengths

- We use subsystems of second order arithmetic $\mathbb{Z}_2$.
- Today we only look at
  - RCA_0: $PA^- + \Sigma_1$-induction and Δ_0-comprehension;
  - WKL_0: RCA_0 and every infinite binary tree has an infinite path;
  - ACA_0: RCA_0 and arithmetical comprehension: for φ arithmetic, $\exists X \forall n (n \in X \leftrightarrow \varphi(n))$.
- (Subsystems in first order arithmetic has also been used, $I\Sigma_1 < B\Sigma_2 < I\Sigma_2 < \cdots$.)

Measuring Strengths

- We use subsystems of second order arithmetic $\mathbb{Z}_2$.
- Today we only look at
  - RCA_0: $PA^- + \Sigma_1$-induction and Δ_0-comprehension;
  - WKL_0: RCA_0 and every infinite binary tree has an infinite path;
  - ACA_0: RCA_0 and arithmetical comprehension: for φ arithmetic, $\exists X \forall n (n \in X \leftrightarrow \varphi(n))$.
- (Subsystems in first order arithmetic has also been used, $I\Sigma_1 < B\Sigma_2 < I\Sigma_2 < \cdots$.)

Measuring Strengths

- We use subsystems of second order arithmetic $\mathbb{Z}_2$.
- Today we only look at
  - RCA_0: $PA^- + \Sigma_1$-induction and Δ_0-comprehension;
  - WKL_0: RCA_0 and every infinite binary tree has an infinite path;
  - ACA_0: RCA_0 and arithmetical comprehension: for φ arithmetic, $\exists X \forall n (n \in X \leftrightarrow \varphi(n))$.
- (Subsystems in first order arithmetic has also been used, $I\Sigma_1 < B\Sigma_2 < I\Sigma_2 < \cdots$.)

Measuring Strengths

- We use subsystems of second order arithmetic $\mathbb{Z}_2$.
- Today we only look at
  - RCA_0: $PA^- + \Sigma_1$-induction and Δ_0-comprehension;
  - WKL_0: RCA_0 and every infinite binary tree has an infinite path;
  - ACA_0: RCA_0 and arithmetical comprehension: for φ arithmetic, $\exists X \forall n (n \in X \leftrightarrow \varphi(n))$.
- (Subsystems in first order arithmetic has also been used, $I\Sigma_1 < B\Sigma_2 < I\Sigma_2 < \cdots$.)

Measuring Strengths

- We use subsystems of second order arithmetic $\mathbb{Z}_2$.
- Today we only look at
  - RCA_0: $PA^- + \Sigma_1$-induction and Δ_0-comprehension;
  - WKL_0: RCA_0 and every infinite binary tree has an infinite path;
  - ACA_0: RCA_0 and arithmetical comprehension: for φ arithmetic, $\exists X \forall n (n \in X \leftrightarrow \varphi(n))$.
- (Subsystems in first order arithmetic has also been used, $I\Sigma_1 < B\Sigma_2 < I\Sigma_2 < \cdots$.)

Models

Models are often specified by their second order part:

- RCA_0: A Turing ideal.
- WKL_0: Scott sets, models can be subcollections of low sets.
- ACA_0: Jump idea.

Models

Models are often specified by their second order part:

- RCA_0: A Turing ideal.
- WKL_0: Scott sets, models can be subcollections of low sets.
- ACA_0: Jump idea.

Models

Models are often specified by their second order part:

- RCA_0: A Turing ideal.
- WKL_0: Scott sets, models can be subcollections of low sets.
- ACA_0: Jump idea.

Back to Ramsey Theorem

- Since 1972, recursion theorists have been studying the "effective" content of Ramsey's Theorem.
- Basic Question: Suppose $f : [\mathbb{N}]^n \to \{0, \ldots, k-1\}$ is recursive. What can we say about the complexity of infinite f-homogeneous sets H?
- Effective studies and reverse mathematics are related.

Back to Ramsey Theorem

- Since 1972, recursion theorists have been studying the "effective" content of Ramsey's Theorem.
- Basic Question: Suppose $f : [\mathbb{N}]^n \to \{0, \ldots, k-1\}$ is recursive. What can we say about the complexity of infinite f-homogeneous sets H?
- Effective studies and reverse mathematics are related.

Back to Ramsey Theorem

- Since 1972, recursion theorists have been studying the "effective" content of Ramsey's Theorem.
- Basic Question: Suppose $f : [\mathbb{N}]^n \to \{0, \ldots, k-1\}$ is recursive. What can we say about the complexity of infinite f-homogeneous sets H?
- Effective studies and reverse mathematics are related.

Jockusch's results (phrased in reverse math)

Theorem (1972)

1. $\mathsf{ACA}_0 \vdash \mathsf{RT}^n_k$.
2. $\mathsf{RCA}_0 + \mathsf{RT}^3_2$ *implies* ACA_0.
3. WKL_0 *does not imply* RT^2_2.

Jockusch's results (phrased in reverse math)

Theorem (1972)

1. $ACA_0 \vdash RT^n_k$.
2. $RCA_0 + RT^3_2$ *implies* ACA_0.
3. WKL_0 *does not imply* RT^2_2.

Jockusch's results (phrased in reverse math)

Theorem (1972)

1. $\mathsf{ACA}_0 \vdash \mathsf{RT}^n_k$.
2. $\mathsf{RCA}_0 + \mathsf{RT}^3_2$ *implies* ACA_0.
3. WKL_0 *does not imply* RT^2_2.

Seetapun's Theorem

Theorem (Seetapun and Slaman, 1995)
RT^2_2 *is strictly weaker than* ACA_0*.*

It revived the area after more than 20 years silence.

Basic idea: (1) avoiding the upper cone; and (2) iterate.

In this talk, I will highlight only one important ingredient, and ignore other issues like iteration and relativization issues.

Seetapun's Theorem

Theorem (Seetapun and Slaman, 1995)
RT^2_2 *is strictly weaker than* ACA_0*.*

It revived the area after more than 20 years silence.

Basic idea: (1) avoiding the upper cone; and (2) iterate.

In this talk, I will highlight only one important ingredient, and ignore other issues like iteration and relativization issues.

Seetapun's Theorem

Theorem (Seetapun and Slaman, 1995)
RT^2_2 *is strictly weaker than* ACA_0.

It revived the area after more than 20 years silence.

Basic idea: (1) avoiding the upper cone; and (2) iterate.

In this talk, I will highlight only one important ingredient, and ignore other issues like iteration and relativization issues.

Seetapun's Theorem

Theorem (Seetapun and Slaman, 1995)
RT^2_2 *is strictly weaker than* ACA_0.

It revived the area after more than 20 years silence.

Basic idea: (1) avoiding the upper cone; and (2) iterate.

In this talk, I will highlight only one important ingredient, and ignore other issues like iteration and relativization issues.

Classical Cone Avoiding: Finding Splits

Proposition

Given C nonrecursive, there is a nonrecursive set $A \not\geq_T C$.

Proof (Sketch)
It is easy to make A nonrecursive.

We use Cohen forcing to satisfy requirements $\Phi_e^A \neq C$ for all $e \in \omega$.

Given an initial segment p, ask if there are extensions q_1 and q_2 of p such that for some x,

$$\Phi_e^{q_1}(x)\downarrow \neq \Phi_e^{q_2}(x)\downarrow .$$

(I will loosely call such q_1 and q_2 an *e-split* or simply a split.)

Note the question is Σ_1^0.

Classical Cone Avoiding: Finding Splits

Proposition

Given C nonrecursive, there is a nonrecursive set $A \not\geq_T C$.

Proof (Sketch)
It is easy to make A nonrecursive.

We use Cohen forcing to satisfy requirements $\Phi_e^A \neq C$ for all $e \in \omega$.

Given an initial segment p, ask if there are extensions q_1 and q_2 of p such that for some x,

$$\Phi_e^{q_1}(x) \downarrow \neq \Phi_e^{q_2}(x) \downarrow .$$

(I will loosely call such q_1 and q_2 an *e-split* or simply a split.)

Note the question is Σ_1^0.

Classical Cone Avoiding: Finding Splits

Proposition

Given C nonrecursive, there is a nonrecursive set $A \not\geq_T C$.

Proof (Sketch)
It is easy to make A nonrecursive.

We use Cohen forcing to satisfy requirements $\Phi_e^A \neq C$ for all $e \in \omega$.

Given an initial segment p, ask if there are extensions q_1 and q_2 of p such that for some x,

$$\Phi_e^{q_1}(x) \downarrow \neq \Phi_e^{q_2}(x) \downarrow .$$

(I will loosely call such q_1 and q_2 an *e-split* or simply a split.)

Note the question is Σ_1^0.

Classical Cone Avoiding: Finding Splits

Proposition

Given C nonrecursive, there is a nonrecursive set $A \not\geq_T C$.

Proof (Sketch)
It is easy to make A nonrecursive.

We use Cohen forcing to satisfy requirements $\Phi_e^A \neq C$ for all $e \in \omega$.

Given an initial segment p, ask if there are extensions q_1 and q_2 of p such that for some x,

$$\Phi_e^{q_1}(x) \downarrow \neq \Phi_e^{q_2}(x) \downarrow .$$

(I will loosely call such q_1 and q_2 an *e-split* or simply a split.)

Note the question is Σ_1^0.

Classical Cone Avoiding: Finding Splits

Proposition

Given C nonrecursive, there is a nonrecursive set $A \not\geq_T C$.

Proof (Sketch)
It is easy to make A nonrecursive.

We use Cohen forcing to satisfy requirements $\Phi_e^A \neq C$ for all $e \in \omega$.

Given an initial segment p, ask if there are extensions q_1 and q_2 of p such that for some x,

$$\Phi_e^{q_1}(x) \downarrow \neq \Phi_e^{q_2}(x) \downarrow .$$

(I will loosely call such q_1 and q_2 an *e-split* or simply a split.)

Note the question is Σ_1^0.

Classical Cone Avoiding: Finding Splits

Proposition

Given C nonrecursive, there is a nonrecursive set $A \not\geq_T C$.

Proof (Sketch)
It is easy to make A nonrecursive.

We use Cohen forcing to satisfy requirements $\Phi_e^A \neq C$ for all $e \in \omega$.

Given an initial segment p, ask if there are extensions q_1 and q_2 of p such that for some x,

$$\Phi_e^{q_1}(x) \downarrow \neq \Phi_e^{q_2}(x) \downarrow .$$

(I will loosely call such q_1 and q_2 an *e-split* or simply a split.)

Note the question is Σ_1^0.

Proof (conti.)

Case 1: There is a split.

Then one of the values $\Phi_e^{q_1}(x)$ and $\Phi_e^{q_2}(x)$ must disagree with $C(x)$, choose the extension which give the disagreement.

Case 2: There is no split.

Then if Φ_e^A is total and $p \subset A$ then Φ_e^A is recursive. Φ_e^A cannot compute the nonrecursive set C.

Proof (conti.)

Case 1: There is a split.

Then one of the values $\Phi_e^{q_1}(x)$ and $\Phi_e^{q_2}(x)$ must disagree with $C(x)$, choose the extension which give the disagreement.

Case 2: There is no split.

Then if Φ_e^A is total and $p \subset A$ then Φ_e^A is recursive. Φ_e^A cannot compute the nonrecursive set C.

A Useful Decomposition

Theorem (Cholak, Jockusch and Slaman, 2001)

$$\mathsf{RT}^2_2 = \mathsf{COH} + \mathsf{SRT}^2_2.$$

- This decomposition turns out to be extremely useful.
- Thus, we can (iteratively) add a solution to COH and then a solution to SRT^2_2, instead of adding a solution to RT^2_2.
- (Other combinatorial principles weaker than RT^2_2 can often be decomposed in a similar fashion.)

A Useful Decomposition

Theorem (Cholak, Jockusch and Slaman, 2001)

$$\mathsf{RT}^2_2 = \mathsf{COH} + \mathsf{SRT}^2_2.$$

- This decomposition turns out to be extremely useful.
- Thus, we can (iteratively) add a solution to COH and then a solution to SRT^2_2, instead of adding a solution to RT^2_2.
- (Other combinatorial principles weaker than RT^2_2 can often be decomposed in a similar fashion.)

A Useful Decomposition

Theorem (Cholak, Jockusch and Slaman, 2001)

$$\mathsf{RT}^2_2 = \mathsf{COH} + \mathsf{SRT}^2_2.$$

- This decomposition turns out to be extremely useful.
- Thus, we can (iteratively) add a solution to COH and then a solution to SRT^2_2, instead of adding a solution to RT^2_2.
- (Other combinatorial principles weaker than RT^2_2 can often be decomposed in a similar fashion.)

A Useful Decomposition

Theorem (Cholak, Jockusch and Slaman, 2001)

$$\mathsf{RT}^2_2 = \mathsf{COH} + \mathsf{SRT}^2_2.$$

- This decomposition turns out to be extremely useful.
- Thus, we can (iteratively) add a solution to COH and then a solution to SRT^2_2, instead of adding a solution to RT^2_2.
- (Other combinatorial principles weaker than RT^2_2 can often be decomposed in a similar fashion.)

COH and SRT^2_2

- Let R be an infinite set and $R^s = \{t | (s,t) \in R\}$. A set G is said to be *R-cohesive* if for all s, either $G \cap R^s$ is finite or $G \cap \overline{R^s}$ is finite.

- The cohesive principle COH states that for every R, there is an infinite G that is R-cohesive.

- We say that a coloring f for pairs is stable, if for all x,

$$\lim_{y \to \infty} f(x, y)$$

exists.

- SRT^2_2 states that every *stable* coloring of pairs has a solution.

COH and SRT^2_2

- Let R be an infinite set and $R^s = \{t|(s,t) \in R\}$. A set G is said to be R-*cohesive* if for all s, either $G \cap R^s$ is finite or $G \cap \overline{R^s}$ is finite.

- The cohesive principle COH states that for every R, there is an infinite G that is R-cohesive.

- We say that a coloring f for pairs is stable, if for all x,

$$\lim_{y \to \infty} f(x,y)$$

  exists.

- SRT^2_2 states that every *stable* coloring of pairs has a solution.

COH and SRT^2_2

- Let R be an infinite set and $R^s = \{t | (s,t) \in R\}$. A set G is said to be *R-cohesive* if for all s, either $G \cap R^s$ is finite or $G \cap \overline{R^s}$ is finite.

- The cohesive principle COH states that for every R, there is an infinite G that is R-cohesive.

- We say that a coloring f for pairs is stable, if for all x,

$$\lim_{y \to \infty} f(x,y)$$

  exists.

- SRT^2_2 states that every *stable* coloring of pairs has a solution.

COH and SRT^2_2

- Let R be an infinite set and $R^s = \{t | (s,t) \in R\}$. A set G is said to be *R-cohesive* if for all s, either $G \cap R^s$ is finite or $G \cap \overline{R^s}$ is finite.

- The cohesive principle COH states that for every R, there is an infinite G that is R-cohesive.

- We say that a coloring f for pairs is stable, if for all x,

$$\lim_{y \to \infty} f(x, y)$$

  exists.

- SRT^2_2 states that every *stable* coloring of pairs has a solution.

An Equivalent Decomposition

- A stable coloring naturally induces a partition of ω into two Δ^0_2 sets.
- Let D^2_2 be the statement: Every Δ^0_2 set contains an infinite subsets either as a subset or as a subset of its complement.
- $\mathrm{RT}^2_2 = D^2_2 + \mathrm{COH}$.

An Equivalent Decomposition

- A stable coloring naturally induces a partition of ω into two Δ^0_2 sets.
- Let D^2_2 be the statement: Every Δ^0_2 set contains an infinite subsets either as a subset or as a subset of its complement.
- $\mathrm{RT}^2_2 = D^2_2 + \mathrm{COH}$.

An Equivalent Decomposition

- A stable coloring naturally induces a partition of ω into two Δ_2^0 sets.
- Let D_2^2 be the statement: Every Δ_2^0 set contains an infinite subsets either as a subset or as a subset of its complement.
- $\mathrm{RT}_2^2 = D_2^2 + \mathrm{COH}$.

COH and cone avoiding

The following is a weaker version of Jockusch and Stephan (1993)

Theorem

Let R be a recursive set and C nonrecursive. Then there is an R-cohesive set G with $C \not\leq_T G$.

We use (effective) Mathias forcing.

- The conditions are pairs (σ, X) where σ is a finite set, X is an infinite set and $\max \sigma < \min X$.
- $(\tau, Y) \leq (\sigma, X)$ if $\tau \supseteq \sigma$, $Y \subseteq X$ and $\tau \setminus \sigma \subset X$.

We say the forcing is recursive if the sets X in the definition are recursive.

COH and cone avoiding

The following is a weaker version of Jockusch and Stephan (1993)

Theorem

Let R be a recursive set and C nonrecursive. Then there is an R-cohesive set G with $C \not\leq_T G$.

We use (effective) Mathias forcing.

- The conditions are pairs (σ, X) where σ is a finite set, X is an infinite set and $\max \sigma < \min X$.
- $(\tau, Y) \leq (\sigma, X)$ if $\tau \supseteq \sigma$, $Y \subseteq X$ and $\tau \setminus \sigma \subset X$.

We say the forcing is recursive if the sets X in the definition are recursive.

Proof Sketch

Fix R and C. The set $D_s = \{(\sigma, X) : X \subset R^s \vee X \subset (\overline{R}^s)\}$ is dense. That settles R-cohesiveness.

Use the same split finding trick to do cone avoiding (just extend σ as in the classical case).

Proof Sketch

Fix R and C. The set $D_s = \{(\sigma, X) : X \subset R^s \vee X \subset (\overline{R}^s)\}$ is dense. That settles R-cohesiveness.

Use the same split finding trick to do cone avoiding (just extend σ as in the classical case).

D_2^2 and cone avoiding

The following is a weaker version of Seetapun and Slaman (1995)

Theorem

Let D be a Δ_2^0 set and C be a nonrecursive set. Then there is an infinite set H such that either $H \subset D$ or $H \subset \overline{D}$ with $C \not\leq_T H$.

Main difficulty: When we find a split, it may involve both element in D and $\overline{D}$.

Q: Is there a way to find "D-safe" splits?

D^2_2 and cone avoiding

The following is a weaker version of Seetapun and Slaman (1995)

Theorem

Let D be a Δ^0_2 set and C be a nonrecursive set. Then there is an infinite set H such that either $H \subset D$ or $H \subset \overline{D}$ with $C \not\leq_T H$.

Main difficulty: When we find a split, it may involve both element in D and $\overline{D}$.

Q: Is there a way to find "D-safe" splits?

D^2_2 and cone avoiding

The following is a weaker version of Seetapun and Slaman (1995)

Theorem

Let D be a Δ^0_2 set and C be a nonrecursive set. Then there is an infinite set H such that either $H \subset D$ or $H \subset \overline{D}$ with $C \not\leq_T H$.

Main difficulty: When we find a split, it may involve both element in D and $\overline{D}$.

Q: Is there a way to find "D-safe" splits?

Blobs and Seetapun Trees

- A sequence of *blobs* is just a recursive sequence of finite sets $\vec{o}$ such that for each s less than the length of the sequence, $\max o_s < \min o_{s+1}$.
- Let $\vec{o}$ be a finite sequence of blobs, say of length h. Consider the set T of all choice functions σ with domain h such that $\sigma(s) \in o_s$. T can be viewed naturally as a tree, called the *Seetapun tree* associated with $\vec{o}$.
- For a Σ^0_1-formula $\psi(G)$, we will search for blobs o such that $\psi(o)$ holds.
- For example, for cone avoiding, we are looking for a finite set o having a split $q_1, q_2 \subset o$.

Blobs and Seetapun Trees

- A sequence of *blobs* is just a recursive sequence of finite sets $\vec{o}$ such that for each s less than the length of the sequence, $\max o_s < \min o_{s+1}$.
- Let $\vec{o}$ be a finite sequence of blobs, say of length h. Consider the set T of all choice functions σ with domain h such that $\sigma(s) \in o_s$. T can be viewed naturally as a tree, called the *Seetapun tree* associated with $\vec{o}$.
- For a Σ^0_1-formula $\psi(G)$, we will search for blobs o such that $\psi(o)$ holds.
- For example, for cone avoiding, we are looking for a finite set o having a split $q_1, q_2 \subset o$.

Blobs and Seetapun Trees

- A sequence of *blobs* is just a recursive sequence of finite sets $\vec{o}$ such that for each s less than the length of the sequence, $\max o_s < \min o_{s+1}$.
- Let $\vec{o}$ be a finite sequence of blobs, say of length h. Consider the set T of all choice functions σ with domain h such that $\sigma(s) \in o_s$. T can be viewed naturally as a tree, called the *Seetapun tree* associated with $\vec{o}$.
- For a Σ^0_1-formula $\psi(G)$, we will search for blobs o such that $\psi(o)$ holds.
- For example, for cone avoiding, we are looking for a finite set o having a split $q_1, q_2 \subset o$.

Blobs and Seetapun Trees

- A sequence of *blobs* is just a recursive sequence of finite sets $\vec{o}$ such that for each s less than the length of the sequence, $\max o_s < \min o_{s+1}$.

- Let $\vec{o}$ be a finite sequence of blobs, say of length h. Consider the set T of all choice functions σ with domain h such that $\sigma(s) \in o_s$. T can be viewed naturally as a tree, called the *Seetapun tree* associated with $\vec{o}$.

- For a Σ^0_1-formula $\psi(G)$, we will search for blobs o such that $\psi(o)$ holds.

- For example, for cone avoiding, we are looking for a finite set o having a split $q_1, q_2 \subset o$.

Seetapun Disjunctions

Definition

Given a Σ^0_1-formula $\psi(G)$, an *Seetapun disjunction* for ψ is a pair $(\vec{o}, S)$, where $\vec{o}$ is a sequence of blobs of length $h > 0$ and S is the Seetapun tree associated with $\vec{o}$, such that:

(i) For each $s < h$, $\psi(o_s)$ holds "independently".

(ii) For each maximal branch τ of S, there exists a finite subset $\iota \subseteq \tau$ such that $\psi(\iota)$ holds. We refer to the set ι as a *thread* (in τ).

Key observation: For an Seetapun disjunction, either there is a blob $o \subseteq D$ or there is a thread $\iota \subseteq \overline{D}$. Seetapun disjunction is D-save!

Seetapun Disjunctions

Definition

Given a Σ^0_1-formula $\psi(G)$, an *Seetapun disjunction* for ψ is a pair $(\vec{o}, S)$, where $\vec{o}$ is a sequence of blobs of length $h > 0$ and S is the Seetapun tree associated with $\vec{o}$, such that:

(i) For each $s < h$, $\psi(o_s)$ holds "independently".

(ii) For each maximal branch τ of S, there exists a finite subset $\iota \subseteq \tau$ such that $\psi(\iota)$ holds. We refer to the set ι as a *thread* (in τ).

Key observation: For an Seetapun disjunction, either there is a blob $o \subseteq D$ or there is a thread $\iota \subseteq \overline{D}$. Seetapun disjunction is D-save!

Proof Sketch

Fix Δ^0_2 set D and nonrecursive set C. We want to find and infinite $H \subset D$ or $\subset \overline{D}$ satisfying

$$R_{e,i} : \Phi_e^H \neq C \vee \Phi_i^H \neq C.$$

We recursively enumerate blobs containing an e-split.

Case 1. This sequence of blobs is finite, i.e., after $\langle o_i : i \leq s \rangle$ there is no more e-splits.

Then we simply move the construction "above the last blob". We refer this as *skipping*.

Proof Sketch

Fix Δ^0_2 set D and nonrecursive set C. We want to find and infinite $H \subset D$ or $\subset \overline{D}$ satisfying

$$R_{e,i} : \Phi_e^H \neq C \vee \Phi_i^H \neq C.$$

We recursively enumerate blobs containing an e-split.

Case 1. This sequence of blobs is finite, i.e., after $\langle o_i : i \leq s \rangle$ there is no more e-splits.

Then we simply move the construction “above the last blob”. We refer this as *skipping*.

Proof Sketch

Fix Δ^0_2 set D and nonrecursive set C. We want to find and infinite $H \subset D$ or $\subset \overline{D}$ satisfying

$$R_{e,i} : \Phi^H_e \neq C \vee \Phi^H_i \neq C.$$

We recursively enumerate blobs containing an e-split.

Case 1. This sequence of blobs is finite, i.e., after $\langle o_i : i \leq s \rangle$ there is no more e-splits.

Then we simply move the construction "above the last blob". We refer this as *skipping*.

Proof Sketch (conti.)

Case 2. The sequence of blobs is infinite. Then we form Seetapun tree along the way and check if every branch τ contains an i-split.

Subcase 2.1. Every branch τ contains an i-split, i.e., we found an Seetapun disjunction.

Then either D contains a blob o or $\overline{D}$ contains a thread ι. Say $D \supset o$. We choose the subset of o which gives us the value $\neq C$.

Subcase 2.2. No Seetapun disjunction occurs.

Then we get an infinite subtree T of the Seetapun tree. Any infinite branch will not see an i-split.

Proof Sketch (conti.)

Case 2. The sequence of blobs is infinite. Then we form Seetapun tree along the way and check if every branch τ contains an i-split.

Subcase 2.1. Every branch τ contains an i-split, i.e., we found an Seetapun disjunction.

Then either D contains a blob o or $\overline{D}$ contains a thread ι. Say $D \supset o$. We choose the subset of o which gives us the value $\neq C$.

Subcase 2.2. No Seetapun disjunction occurs.

Then we get an infinite subtree T of the Seetapun tree. Any infinite branch will not see an i-split.

Proof Sketch (conti.)

Case 2. The sequence of blobs is infinite. Then we form Seetapun tree along the way and check if every branch τ contains an *i*-split.

Subcase 2.1. Every branch τ contains an *i*-split, i.e., we found an Seetapun disjunction.

Then either D contains a blob o or $\overline{D}$ contains a thread ι. Say $D \supset o$. We choose the subset of o which gives us the value $\neq C$.

Subcase 2.2. No Seetapun disjunction occurs.

Then we get an infinite subtree T of the Seetapun tree. Any infinite branch will not see an *i*-split.

Proof Sketch (conti.)

Case 2. The sequence of blobs is infinite. Then we form Seetapun tree along the way and check if every branch τ contains an i-split.

Subcase 2.1. Every branch τ contains an i-split, i.e., we found an Seetapun disjunction.

Then either D contains a blob o or $\overline{D}$ contains a thread ι. Say $D \supset o$. We choose the subset of o which gives us the value $\neq C$.

Subcase 2.2. No Seetapun disjunction occurs.

Then we get an infinite subtree T of the Seetapun tree. Any infinite branch will not see an i-split.

Proof Sketch (conti.)

Case 2. The sequence of blobs is infinite. Then we form Seetapun tree along the way and check if every branch τ contains an i-split.

Subcase 2.1. Every branch τ contains an i-split, i.e., we found an Seetapun disjunction.

Then either D contains a blob o or $\overline{D}$ contains a thread ι. Say $D \supset o$. We choose the subset of o which gives us the value $\neq C$.

Subcase 2.2. No Seetapun disjunction occurs.

Then we get an infinite subtree T of the Seetapun tree. Any infinite branch will not see an i-split.

This is only a sketch

The above sketch overlooked the interaction between requirements; and selection of the infinite set.

For a complete proof (not mentioning Seetapun disjunctions), reader can refer to Hirschfeldt *Slicing the Truth*, World Scientific, 2015.

This is only a sketch

The above sketch overlooked the interaction between requirements; and selection of the infinite set.

For a complete proof (not mentioning Seetapun disjunctions), reader can refer to Hirschfeldt *Slicing the Truth*, World Scientific, 2015.

An application

- (Chong, Slaman and Yang 2012) Stable Ramsey's Theorem for pairs does not imply Ramsey's Theorem for Pairs.
- The idea of Seetapun disjunction is important in the proof.
- To distinguish Case 1 and Case 2, one needs $0''$.

An application

- (Chong, Slaman and Yang 2012) Stable Ramsey's Theorem for pairs does not imply Ramsey's Theorem for Pairs.
- The idea of Seetapun disjunction is important in the proof.
- To distinguish Case 1 and Case 2, one needs $0''$.

An application

- (Chong, Slaman and Yang 2012) Stable Ramsey's Theorem for pairs does not imply Ramsey's Theorem for Pairs.
- The idea of Seetapun disjunction is important in the proof.
- To distinguish Case 1 and Case 2, one needs $0''$.

Nonstandard model

- Nonstandard models are crucial in the proof.
  - We divide M many requirements into ω many blocks.
  - The case 1/case 2 switch for each block needs one bit information.
  - In some nonstandard model, we can code ω many bits by a single number a.
  - Using a, $0'$ is sufficient to carry out the construction, not $0''$.
- It is known that the method does not apply to ω.
- Question: What happens in ω-models?

Nonstandard model

- Nonstandard models are crucial in the proof.
  - We divide M many requirements into ω many blocks.
  - The case 1/case 2 switch for each block needs one bit information.
  - In some nonstandard model, we can code ω many bits by a single number a.
  - Using a, $0'$ is sufficient to carry out the construction, not $0''$.
- It is known that the method does not apply to ω.
- Question: What happens in ω-models?

Nonstandard model

- Nonstandard models are crucial in the proof.
  - We divide M many requirements into ω many blocks.
  - The case 1/case 2 switch for each block needs one bit information.
  - In some nonstandard model, we can code ω many bits by a single number a.
  - Using a, $0'$ is sufficient to carry out the construction, not $0''$.
- It is known that the method does not apply to ω.
- Question: What happens in ω-models?

Nonstandard model

- Nonstandard models are crucial in the proof.
  - We divide M many requirements into ω many blocks.
  - The case 1/case 2 switch for each block needs one bit information.
  - In some nonstandard model, we can code ω many bits by a single number a.
  - Using a, $0'$ is sufficient to carry out the construction, not $0''$.
- It is known that the method does not apply to ω.
- Question: What happens in ω-models?