BILL, RECORD LECTURE!!!!

BILL RECORD LECTURE!!!

Computable Ramsey Theory

Exposition by William Gasarch

February 18, 2025

Is The Proof of Ramsey's Thm Constructive?

Thm $\forall \mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ $\exists$ inf homog set.

Is The Proof of Ramsey's Thm Constructive?

Thm $\forall \mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]\ \exists$ inf homog set.

If someone **gave you** a coloring $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ could you **give them** the set H?

Is The Proof of Ramsey's Thm Constructive?

Thm $\forall \mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ $\exists$ inf homog set.

If someone **gave you** a coloring $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ could you **give them** the set H?

Using the proof I showed you, no. You need to ask questions like $\exists^\infty x \mathrm{COL}(x) = \mathbf{R}$. These are hard to ask.

Is The Proof of Ramsey's Thm Constructive?

Thm $\forall \mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ $\exists$ inf homog set.

If someone **gave you** a coloring $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ could you **give them** the set H?

Using the proof I showed you, no. You need to ask questions like $\exists^{\infty} x \mathrm{COL}(x) = \mathbf{R}$. These are hard to ask.

Furthore, the question is not rigorous. To make this question rigorous we need a notion of **give**.

Is The Proof of Ramsey's Thm Constructive?

Thm $\forall \mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ $\exists$ inf homog set.

If someone **gave you** a coloring $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ could you **give them** the set H?

Using the proof I showed you, no. You need to ask questions like $\exists^{\infty} x \mathrm{COL}(x) = \mathbf{R}$. These are hard to ask.

Furthore, the question is not rigorous. To make this question rigorous we need a notion of **give**.

Is there a program with the following behaviour:

Is The Proof of Ramsey's Thm Constructive?

Thm $\forall \mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ $\exists$ inf homog set.

If someone **gave you** a coloring $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ could you **give them** the set H?

Using the proof I showed you, no. You need to ask questions like $\exists^{\infty} x \mathrm{COL}(x) = \mathbf{R}$. These are hard to ask.

Furthore, the question is not rigorous. To make this question rigorous we need a notion of **give**.

Is there a program with the following behaviour:
Input A program M that computes a 0-1-valued function on $\binom{\mathbb{N}}{2}$.
(so a 2-coloring).

Is The Proof of Ramsey's Thm Constructive?

Thm $\forall \mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ $\exists$ inf homog set.

If someone **gave you** a coloring $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ could you **give them** the set H?

Using the proof I showed you, no. You need to ask questions like $\exists^{\infty} x \mathrm{COL}(x) = \mathbf{R}$. These are hard to ask.

Furthore, the question is not rigorous. To make this question rigorous we need a notion of **give**.

Is there a program with the following behaviour:
Input A program M that computes a 0-1-valued function on $\binom{\mathbb{N}}{2}$. (so a 2-coloring).
Output A program N that computes a 0-1-valued function on $\mathbb{N}$ (so a set) such that the set $\{x\colon N(x) = 1\}$ is homog.

Is The Proof of Ramsey's Thm Constructive?

Thm $\forall \mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ $\exists$ inf homog set.

If someone **gave you** a coloring $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ could you **give them** the set H?

Using the proof I showed you, no. You need to ask questions like $\exists^{\infty} x \mathrm{COL}(x) = \mathbf{R}$. These are hard to ask.

Furthore, the question is not rigorous. To make this question rigorous we need a notion of **give**.

Is there a program with the following behaviour:
Input A program M that computes a 0-1-valued function on $\binom{\mathbb{N}}{2}$. (so a 2-coloring).
Output A program N that computes a 0-1-valued function on $\mathbb{N}$ (so a set) such that the set $\{x\colon N(x) = 1\}$ is homog.
We will refine and ask this question later.

And The Answer Is

And The Answer Is

No.

And The Answer Is

No.
We will define terms and see what we can say.

Concepts From Computability Theory: Turing Machines

Concepts From Computability Theory: Turing Machines

$M_1, M_2, \ldots$ is a standard list of Turing Machines (TMs). You can think of them as all Java programs.

Concepts From Computability Theory: Turing Machines

$M_1, M_2, \ldots$ is a standard list of Turing Machines (TMs). You can think of them as all Java programs.

$M_{e,s}(x)$ means that we run M_e for s steps.

Concepts From Computability Theory: Turing Machines

$M_1, M_2, \ldots$ is a standard list of Turing Machines (TMs). You can think of them as all Java programs.

$M_{e,s}(x)$ means that we run M_e for s steps.

$M(x) \downarrow = a$ means that $M(x)$ halts and outputs a.

Concepts From Computability Theory: Turing Machines

$M_1, M_2, \ldots$ is a standard list of Turing Machines (TMs). You can think of them as all Java programs.

$M_{e,s}(x)$ means that we run M_e for s steps.

$M(x) \downarrow = a$ means that $M(x)$ halts and outputs a.

$M(x) = a$ means that $M(x)$ halts and outputs a (we use the $\downarrow$ when we want to emphasize that $M(x)$ halts).

Concepts From Computability Theory: Turing Machines

$M_1, M_2, \ldots$ is a standard list of Turing Machines (TMs). You can think of them as all Java programs.

$M_{e,s}(x)$ means that we run M_e for s steps.

$M(x) \downarrow = a$ means that $M(x)$ halts and outputs a.

$M(x) = a$ means that $M(x)$ halts and outputs a (we use the $\downarrow$ when we want to emphasize that $M(x)$ halts).

$M(x) \uparrow$ means that $M(x)$ does not halt.

Computable and C.E. and . . .

Computable and C.E. and ...

A set A is *computable* if there is a TM M such that

$$\begin{aligned} x \in A &\implies M(x)\downarrow = 1 \\ x \notin A &\implies M(x)\downarrow = 0 \end{aligned}$$

Computable and C.E. and ...

A set A is *computable* if there is a TM M such that

$$\begin{aligned} x \in A &\implies M(x) \downarrow = 1 \\ x \notin A &\implies M(x) \downarrow = 0 \end{aligned}$$

(Older books use the term *recursive* instead of *computable*.)

Computable and C.E. and ...

A set A is *computable* if there is a TM M such that

$$\begin{aligned} x \in A &\implies M(x)\downarrow = 1 \\ x \notin A &\implies M(x)\downarrow = 0 \end{aligned}$$

(Older books use the term *recursive* instead of *computable*.)

If M is a TM such that on every input x, $M(x)\downarrow \in \{0,1\}$ (so M computes some set) then $L(M) = \{x \colon M(x) = 1\}$

Computable and C.E. and . . .

A set A is *computable* if there is a TM M such that

$$\begin{aligned} x \in A &\implies M(x)\downarrow = 1 \\ x \notin A &\implies M(x)\downarrow = 0 \end{aligned}$$

(Older books use the term *recursive* instead of *computable*.)

If M is a TM such that on every input x, $M(x)\downarrow \in \{0,1\}$ (so M computes some set) then $L(M) = \{x\colon M(x) = 1\}$

A set A is *computably enumerable (c.e.)* if there is a TM M:

$$\begin{aligned} x \in A &\implies M(x)\downarrow \\ x \notin A &\implies M(x)\uparrow \end{aligned}$$

Computable and C.E. and ...

A set A is *computable* if there is a TM M such that

$$\begin{aligned} x \in A &\implies M(x)\downarrow = 1 \\ x \notin A &\implies M(x)\downarrow = 0 \end{aligned}$$

(Older books use the term *recursive* instead of *computable*.)

If M is a TM such that on every input x, $M(x)\downarrow \in \{0,1\}$ (so M computes some set) then $L(M) = \{x\colon M(x) = 1\}$

A set A is *computably enumerable (c.e.)* if there is a TM M:

$$\begin{aligned} x \in A &\implies M(x)\downarrow \\ x \notin A &\implies M(x)\uparrow \end{aligned}$$

(Older books use the term *recursively enumerable (r.e.)* instead of *computably enumerable (c.e.)*.)

The Arithmetic Hierarchy (AH)

Sets are classified in the Arithmetic hierarchy.
comp means **computable**.

The Arithmetic Hierarchy (AH)

Sets are classified in the Arithmetic hierarchy.
comp means **computable**.
$A \in \Sigma_0$ if A is comp.

The Arithmetic Hierarchy (AH)

Sets are classified in the Arithmetic hierarchy.

comp means **computable**.

$A \in \Sigma_0$ if A is comp.

$A \in \Pi_0$ if A is comp.

The Arithmetic Hierarchy (AH)

Sets are classified in the Arithmetic hierarchy.
comp means **computable**.
$A \in \Sigma_0$ if A is comp.
$A \in \Pi_0$ if A is comp.

$A \in \Sigma_1$ is there exists comp B such that
$A = \{x \colon (\exists y)[(x, y) \in B]\}$.

The Arithmetic Hierarchy (AH)

Sets are classified in the Arithmetic hierarchy.
comp means **computable**.
$A \in \Sigma_0$ if A is comp.
$A \in \Pi_0$ if A is comp.

$A \in \Sigma_1$ is there exists comp B such that
$A = \{x \colon (\exists y)[(x, y) \in B]\}$.
$A \in \Pi_1$ is there exists comp B such that
$A = \{x \colon (\forall y)[(x, y) \in B]\}$.

The Arithmetic Hierarchy (AH)

Sets are classified in the Arithmetic hierarchy.
comp means **computable**.
$A \in \Sigma_0$ if A is comp.
$A \in \Pi_0$ if A is comp.

$A \in \Sigma_1$ is there exists comp B such that
$A = \{x \colon (\exists y)[(x,y) \in B]\}$.
$A \in \Pi_1$ is there exists comp B such that
$A = \{x \colon (\forall y)[(x,y) \in B]\}$.

$A \in \Sigma_2$ is there exists comp B such that
$A = \{x \colon (\exists y)(\forall z)[(x,y,z) \in B]\}$.

The Arithmetic Hierarchy (AH)

Sets are classified in the Arithmetic hierarchy.
comp means **computable**.
$A \in \Sigma_0$ if A is comp.
$A \in \Pi_0$ if A is comp.

$A \in \Sigma_1$ is there exists comp B such that
$A = \{x \colon (\exists y)[(x, y) \in B]\}$.
$A \in \Pi_1$ is there exists comp B such that
$A = \{x \colon (\forall y)[(x, y) \in B]\}$.

$A \in \Sigma_2$ is there exists comp B such that
$A = \{x \colon (\exists y)(\forall z)[(x, y, z) \in B]\}$.
$A \in \Pi_2$ is there exists comp B such that
$A = \{x \colon (\forall y)(\exists z)[(x, y, z) \in B]\}$.

The Arithmetic Hierarchy (AH)

Sets are classified in the Arithmetic hierarchy.
comp means **computable**.
$A \in \Sigma_0$ if A is comp.
$A \in \Pi_0$ if A is comp.

$A \in \Sigma_1$ is there exists comp B such that
$A = \{x\colon (\exists y)[(x,y) \in B]\}$.
$A \in \Pi_1$ is there exists comp B such that
$A = \{x\colon (\forall y)[(x,y) \in B]\}$.

$A \in \Sigma_2$ is there exists comp B such that
$A = \{x\colon (\exists y)(\forall z)[(x,y,z) \in B]\}$.
$A \in \Pi_2$ is there exists comp B such that
$A = \{x\colon (\forall y)(\exists z)[(x,y,z) \in B]\}$.

One can define Σ_i, Π_i.

Examples of Sets In the Arithmetic Hierarchy

$$\text{HALT} = \{(e, x) : (\exists s)[M_{e,s}(x) \downarrow\} \in \Sigma_1 - \Sigma_0$$

Examples of Sets In the Arithmetic Hierarchy

$\mathrm{HALT} = \{(e, x) \colon (\exists s)[M_{e,s}(x) \downarrow\} \in \Sigma_1 - \Sigma_0$
FIN is all e such that M_e halts on finite numb of inputs.

Examples of Sets In the Arithmetic Hierarchy

$\mathrm{HALT} = \{(e, x) \colon (\exists s)[M_{e,s}(x) \downarrow\} \in \Sigma_1 - \Sigma_0$
FIN is all e such that M_e halts on finite numb of inputs.
$\mathrm{FIN} = \{e \colon (\exists x)(\forall y, s)[y > x \implies M_{e,s}(y) \uparrow\} \in \Sigma_2 - \Pi_2$.
(Proof that $\mathrm{FIN} \notin \Pi_2$ is hard. So proof that FIN is hard, is hrd.)

Examples of Sets In the Arithmetic Hierarchy

$\mathrm{HALT} = \{(e, x) \colon (\exists s)[M_{e,s}(x)\downarrow\} \in \Sigma_1 - \Sigma_0$
FIN is all e such that M_e halts on finite numb of inputs.
$\mathrm{FIN} = \{e \colon (\exists x)(\forall y, s)[y > x \implies M_{e,s}(y)\uparrow\} \in \Sigma_2 - \Pi_2$.
(Proof that $\mathrm{FIN} \notin \Pi_2$ is hard. So proof that FIN is hard, is hrd.)
INF is all e such that M_e halts on infinite numb of inputs.
$\mathrm{INF} \in \Pi_2 - \Sigma_2$.
(The proof that $\mathrm{INF} \notin \Sigma_2$ is hard.)

Examples of Sets In the Arithmetic Hierarchy

$\text{HALT} = \{(e, x) \colon (\exists s)[M_{e,s}(x) \downarrow\} \in \Sigma_1 - \Sigma_0$
FIN is all e such that M_e halts on finite numb of inputs.
$\text{FIN} = \{e \colon (\exists x)(\forall y, s)[y > x \implies M_{e,s}(y) \uparrow\} \in \Sigma_2 - \Pi_2$.
(Proof that $\text{FIN} \notin \Pi_2$ is hard. So proof that FIN is hard, is hrd.)
INF is all e such that M_e halts on infinite numb of inputs.
$\text{INF} \in \Pi_2 - \Sigma_2$.
(The proof that $\text{INF} \notin \Sigma_2$ is hard.)
COF is all e such that M_e halts on almost all inputs.
$\text{COF} \in \Sigma_3 - \Pi_3$. (The proof that $\text{COF} \notin \Pi_3$ is not easy.)

Facts About the Arithmetic Hierarchy

Facts About the Arithmetic Hierarchy

$$\Sigma_0 \subset \Sigma_1 \subset \Sigma_2 \subset \cdots.$$

Facts About the Arithmetic Hierarchy

$$\Sigma_0 \subset \Sigma_1 \subset \Sigma_2 \subset \cdots.$$

$$\Pi_0 \subset \Pi_1 \subset \Pi_2 \subset \cdots.$$

Facts About the Arithmetic Hierarchy

$\Sigma_0 \subset \Sigma_1 \subset \Sigma_2 \subset \cdots$.

$\Pi_0 \subset \Pi_1 \subset \Pi_2 \subset \cdots$.

For all $i \geq 1$, Σ_i and Π_i are incomparable.

Back to Ramsey Theory

Back to Ramsey Theory

Vote on how to finish the theorem.

Back to Ramsey Theory

Vote on how to finish the theorem.

Thm Let COL be computable 2-coloring of $\binom{\mathbb{N}}{2}$. Then there exists an infinite homog H such that

Back to Ramsey Theory

Vote on how to finish the theorem.

Thm Let COL be computable 2-coloring of $\binom{\mathbb{N}}{2}$. Then there exists an infinite homog H such that
1) $H \in \Sigma_1$.

Back to Ramsey Theory

Vote on how to finish the theorem.

Thm Let COL be computable 2-coloring of $\binom{\mathbb{N}}{2}$. Then there exists an infinite homog H such that
1) $H \in \Sigma_1$.
2) $H \in \Pi_1$.

Back to Ramsey Theory

Vote on how to finish the theorem.

Thm Let COL be computable 2-coloring of $\binom{\mathbb{N}}{2}$. Then there exists an infinite homog H such that
1) $H \in \Sigma_1$.
2) $H \in \Pi_1$.
3) $H \in \Sigma_2$.

Back to Ramsey Theory

Vote on how to finish the theorem.

Thm Let COL be computable 2-coloring of $\binom{\mathbb{N}}{2}$. Then there exists an infinite homog H such that
1) $H \in \Sigma_1$.
2) $H \in \Pi_1$.
3) $H \in \Sigma_2$.
4) $H \in \Pi_2$.

Back to Ramsey Theory

Vote on how to finish the theorem.

Thm Let COL be computable 2-coloring of $\binom{\mathbb{N}}{2}$. Then there exists an infinite homog H such that
1) $H \in \Sigma_1$.
2) $H \in \Pi_1$.
3) $H \in \Sigma_2$.
4) $H \in \Pi_2$.
5) H is not in the arithmetic hierarchy.

Known

Known

Thm

Known

Thm

1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ $\exists$ a Π_2 homog set.

Known

Thm

1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ $\exists$ a Π_2 homog set.
2) $\exists$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ with no Σ_2 homog set.

Known

Thm

1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ $\exists$ a Π_2 homog set.
2) $\exists$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ with no Σ_2 homog set.

What about 3-ary Ramsey?

Known

Thm

1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ $\exists$ a Π_2 homog set.
2) $\exists$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ with no Σ_2 homog set.

What about 3-ary Ramsey?
1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{3} \to [2]$ $\exists$ a Π_3 homog set.

Known

Thm

1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ $\exists$ a Π_2 homog set.

2) $\exists$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \rightarrow [2]$ with no Σ_2 homog set.

What about 3-ary Ramsey?

1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{3} \rightarrow [2]$ $\exists$ a Π_3 homog set.

2) $\exists$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{3} \rightarrow [2]$ with no Σ_3 homog set.

Known

Thm

1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ $\exists$ a Π_2 homog set.
2) $\exists$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ with no Σ_2 homog set.

What about 3-ary Ramsey?
1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{3} \to [2]$ $\exists$ a Π_3 homog set.
2) $\exists$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{3} \to [2]$ with no Σ_3 homog set.

What about *a*-ary Ramsey?

Known

Thm

1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ $\exists$ a Π_2 homog set.
2) $\exists$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ with no Σ_2 homog set.

What about 3-ary Ramsey?
1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{3} \to [2]$ $\exists$ a Π_3 homog set.
2) $\exists$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{3} \to [2]$ with no Σ_3 homog set.

What about *a*-ary Ramsey?
1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{a} \to [2]$ $\exists$ a Π_a homog set.

Known

Thm

1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ $\exists$ a Π_2 homog set.
2) $\exists$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ with no Σ_2 homog set.

What about 3-ary Ramsey?
1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{3} \to [2]$ $\exists$ a Π_3 homog set.
2) $\exists$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{3} \to [2]$ with no Σ_3 homog set.

What about *a*-ary Ramsey?
1) $\forall$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{a} \to [2]$ $\exists$ a Π_a homog set.
2) $\exists$ computable $\mathrm{COL}\colon \binom{\mathbb{N}}{a} \to [2]$ with no Σ_a homog set.

Computable Combinatorics

Computable Combinatorics

Question Given a proof that is non-constructive one can ask the following:

Computable Combinatorics

Question Given a proof that is non-constructive one can ask the following:
Is there an alternative proof that is constructive? (usually not)

Computable Combinatorics

Question Given a proof that is non-constructive one can ask the following:
Is there an alternative proof that is constructive? (usually not)
If the input is computable then is the output on some level of AH?

Computable Combinatorics

Question Given a proof that is non-constructive one can ask the following:
Is there an alternative proof that is constructive? (usually not)
If the input is computable then is the output on some level of AH?
Is there a weaker version that is constructive?

Computable Combinatorics

Question Given a proof that is non-constructive one can ask the following:
Is there an alternative proof that is constructive? (usually not)
If the input is computable then is the output on some level of AH?
Is there a weaker version that is constructive?

The study of these questions in field X is called **Computable X**.

Computable Combinatorics

Question Given a proof that is non-constructive one can ask the following:
Is there an alternative proof that is constructive? (usually not)
If the input is computable then is the output on some level of AH?
Is there a weaker version that is constructive?

The study of these questions in field X is called **Computable X**.

There is a 131 page survey of computable combinatorics here:
`https://www.cs.umd.edu/~gasarch/papers/rcombsur.pdf`

Computable Combinatorics

Question Given a proof that is non-constructive one can ask the following:
Is there an alternative proof that is constructive? (usually not)
If the input is computable then is the output on some level of AH?
Is there a weaker version that is constructive?

The study of these questions in field X is called **Computable X**.

There is a 131 page survey of computable combinatorics here:
`https://www.cs.umd.edu/~gasarch/papers/rcombsur.pdf`

I had more energy back then.