BILL, RECORD LECTURE!!!!

BILL RECORD LECTURE!!!

Computable Ramsey Theorem For Graphs

Exposition by William Gasarch

March 26, 2026

Queries

Queries

I want to classify A in the arithmetic hierarchy.

Queries

I want to classify A in the arithmetic hierarchy.

I have an algorithm for A but it makes **queries to Π_i**

Queries

I want to classify A in the arithmetic hierarchy.

I have an algorithm for A but it makes **queries to Π_i**

It turns out that such sets are in Σ_{i+1}

Queries

I want to classify A in the arithmetic hierarchy.

I have an algorithm for A but it makes **queries to Π_i**

It turns out that such sets are in Σ_{i+1}

I will not prove this, but I will use it.

The Infinite Computable Ramsey Theorem

Thm For all computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ there exists an **infinite** homog set in Σ_3.

The Infinite Computable Ramsey Theorem

Thm For all computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ there exists an **infinite** homog set in Σ_3.

We will, given $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$, produce a stream of ordered pairs

The Infinite Computable Ramsey Theorem

Thm For all computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ there exists an **infinite** homog set in Σ_3.

We will, given $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$, produce a stream of ordered pairs (x_1, c_1), (x_2, c_2), ... where $x_1 < x_2 < \cdots$.

The Infinite Computable Ramsey Theorem

Thm For all computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ there exists an **infinite** homog set in Σ_3.

We will, given $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$, produce a stream of ordered pairs (x_1, c_1), (x_2, c_2), ... where $x_1 < x_2 < \cdots$.

The process will ask Π_2 questions.

The Infinite Computable Ramsey Theorem

Thm For all computable $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ there exists an **infinite** homog set in Σ_3.

We will, given $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$, produce a stream of ordered pairs (x_1, c_1), (x_2, c_2), ... where $x_1 < x_2 < \cdots$.

The process will ask Π_2 questions.

We give the first few steps of the process.

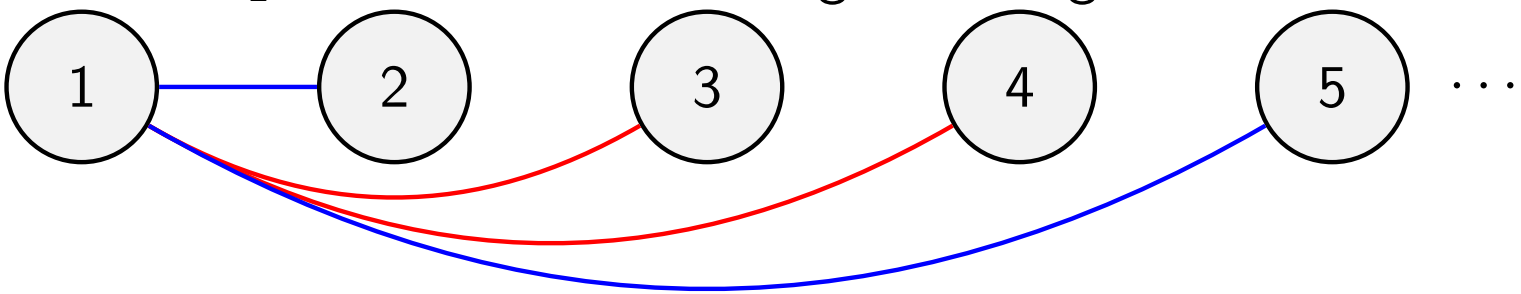
Examples of The First Few Steps of The Process

First Step of Our Process

Look at $x_1 = 1$ and all of the edges coming out of it:

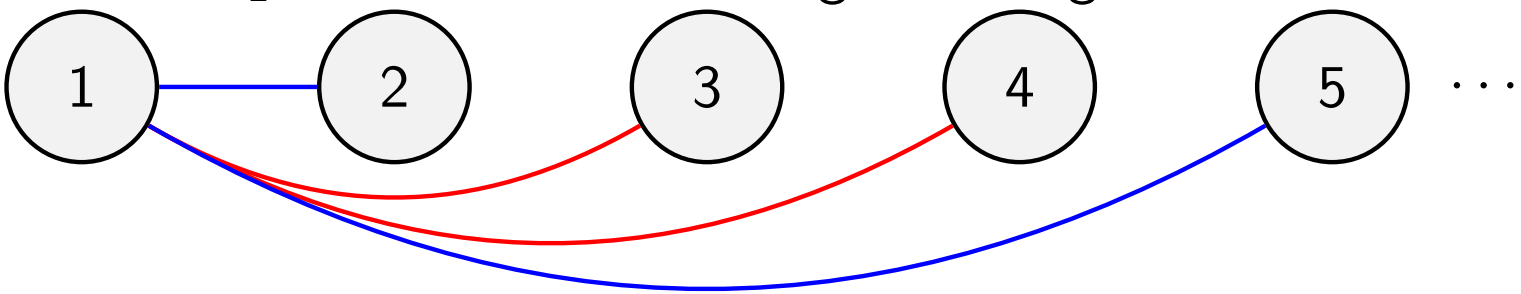
First Step of Our Process

Look at $x_1 = 1$ and all of the edges coming out of it:



First Step of Our Process

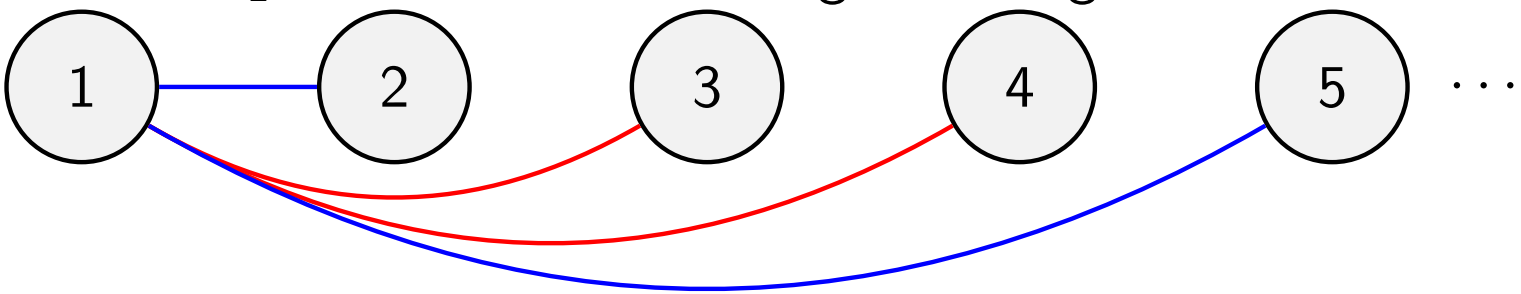
Look at $x_1 = 1$ and all of the edges coming out of it:



Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) = \mathbf{R}]$?

First Step of Our Process

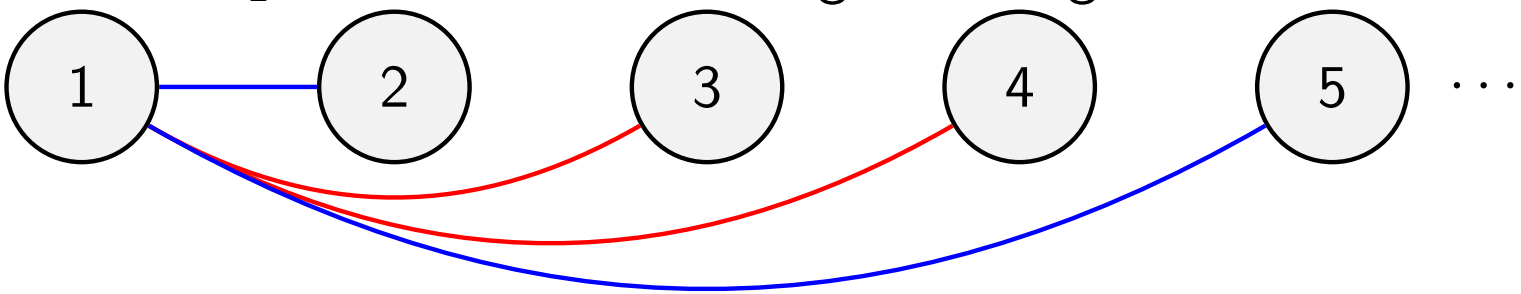
Look at $x_1 = 1$ and all of the edges coming out of it:



Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) = \mathbf{R}]$?
If YES then $\exists^{\infty}$ **R** coming out of x_1. Set $c_1 = \mathbf{R}$.

First Step of Our Process

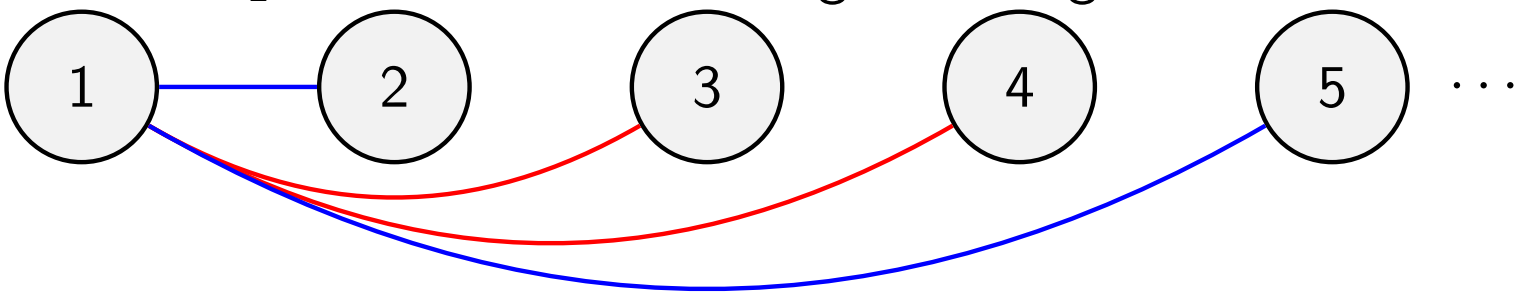
Look at $x_1 = 1$ and all of the edges coming out of it:



Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) =$ **R**$]$?
If YES then $\exists^\infty$ **R** coming out of x_1. Set $c_1 =$ **R**.
If NO then $\exists^\infty$ **B** coming out of x_1. Set $c_1 =$ **B**.
Output (x_1, c_1).

First Step of Our Process

Look at $x_1 = 1$ and all of the edges coming out of it:

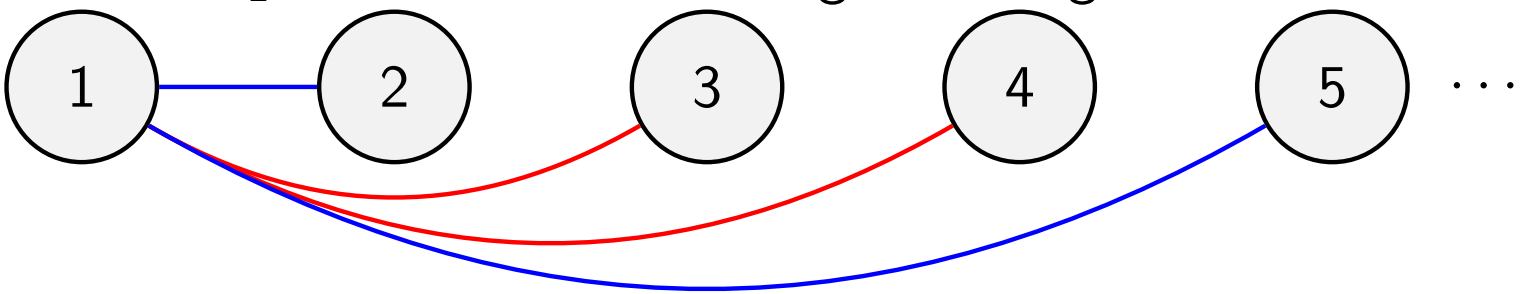


Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) = \mathbf{R}]$?
If YES then $\exists^\infty$ **R** coming out of x_1. Set $c_1 = \mathbf{R}$.
If NO then $\exists^\infty$ **B** coming out of x_1. Set $c_1 = \mathbf{B}$.
Output (x_1, c_1).

Killing?

First Step of Our Process

Look at $x_1 = 1$ and all of the edges coming out of it:

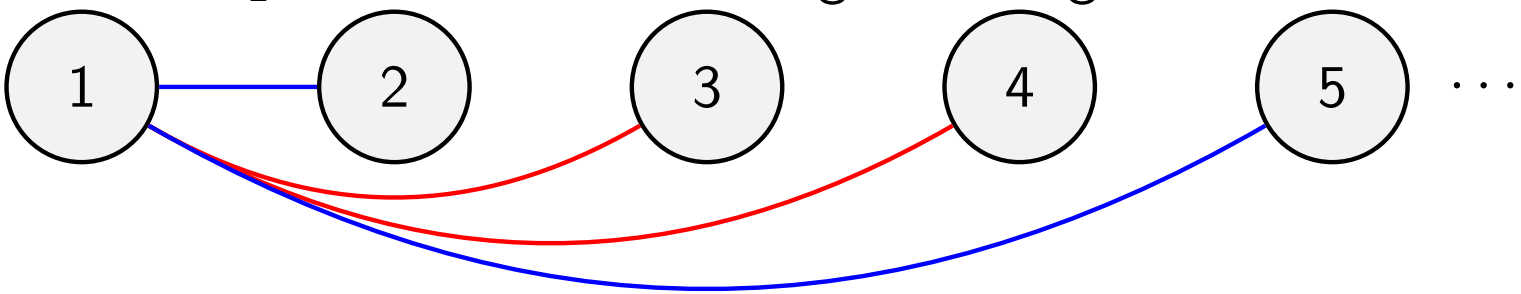


Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) = \mathbf{R}]$?
If YES then $\exists^\infty$ **R** coming out of x_1. Set $c_1 = \mathbf{R}$.
If NO then $\exists^\infty$ **B** coming out of x_1. Set $c_1 = \mathbf{B}$.
Output (x_1, c_1).

Killing? No. We ignore all those who disagree.

First Step of Our Process

Look at $x_1 = 1$ and all of the edges coming out of it:



Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) = \mathbf{R}]$?
If YES then $\exists^\infty$ **R** coming out of x_1. Set $c_1 = \mathbf{R}$.
If NO then $\exists^\infty$ **B** coming out of x_1. Set $c_1 = \mathbf{B}$.
Output (x_1, c_1).

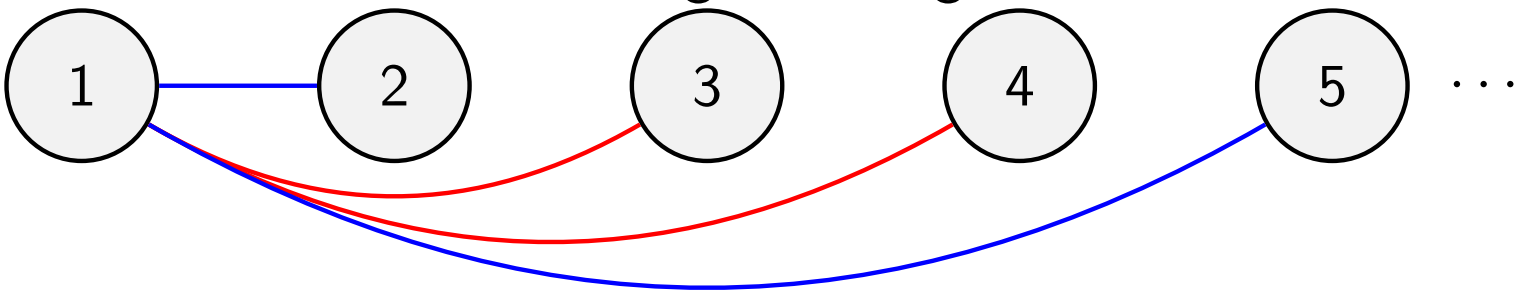
Killing? No. We ignore all those who disagree.
We have a picture of this on the next slide.

Node x_1 Has the Reds

Look at 1 and all of the edges coming out of it:

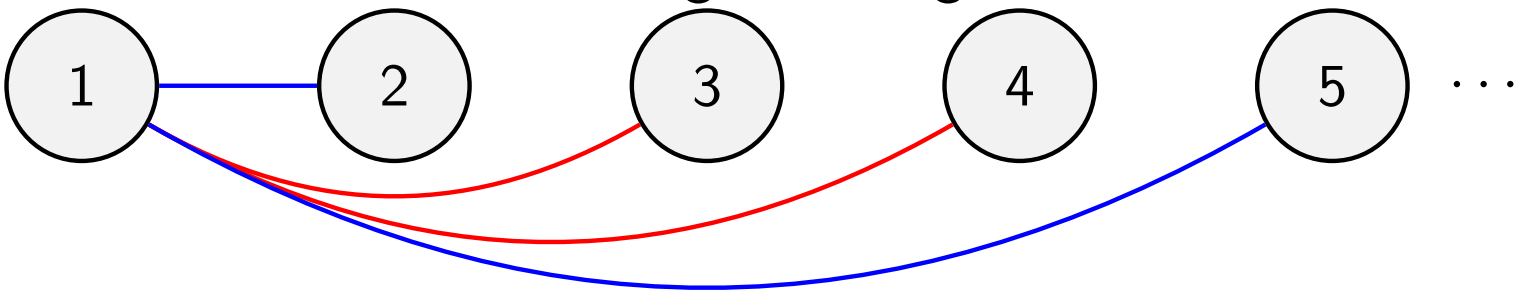
Node x_1 Has the Reds

Look at 1 and all of the edges coming out of it:



Node x_1 Has the Reds

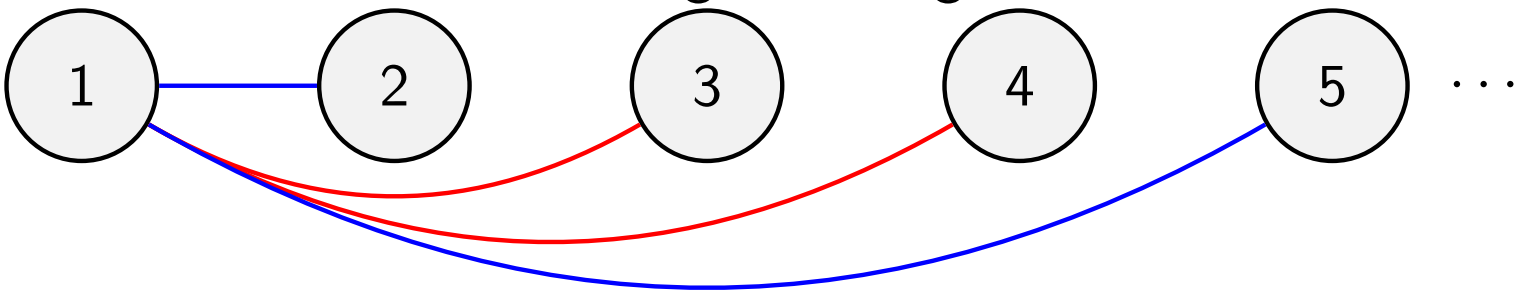
Look at 1 and all of the edges coming out of it:



Assume that 1 has an infinite number of **R** coming out of it.

Node x_1 Has the Reds

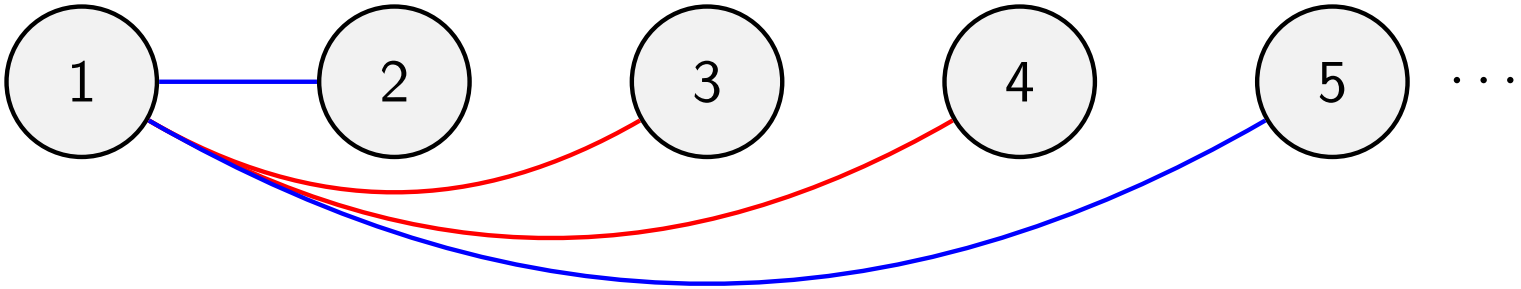
Look at 1 and all of the edges coming out of it:



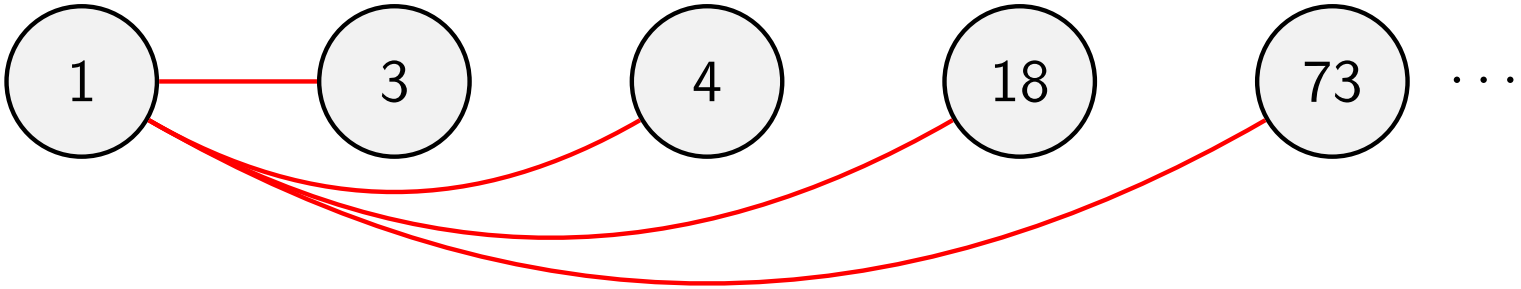
Assume that 1 has an infinite number of **R** coming out of it.
We view the graphs as follows:

Node x_1 Has the Reds

Look at 1 and all of the edges coming out of it:

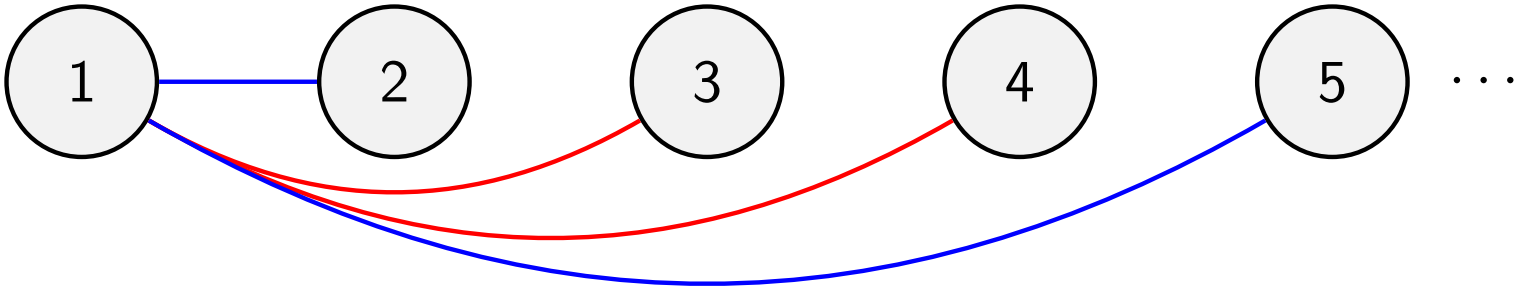


Assume that 1 has an infinite number of **R** coming out of it.
We view the graphs as follows:

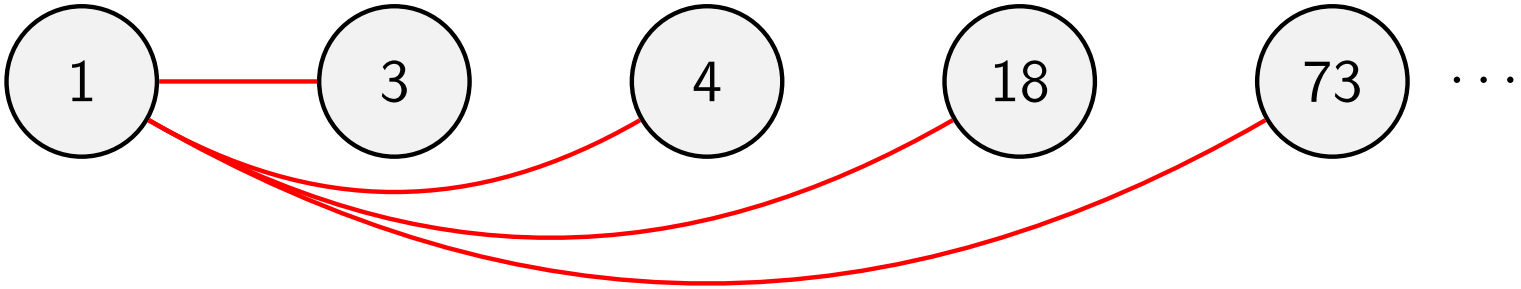


Node x_1 Has the Reds

Look at 1 and all of the edges coming out of it:



Assume that 1 has an infinite number of **R** coming out of it.
We view the graphs as follows:



We Omit 1 from future pictures but its **Still Alive and Well.**
`https://www.youtube.com/watch?v=8--jVqaU-G8`.

The Next Step: Finding x_2

The Next Step: Finding x_2

In our example $x_2 = 3$. But what do we do in general to **find** x_2.

The Next Step: Finding x_2

In our example $x_2 = 3$. But what do we do in general to **find** x_2.

Do we need to ask complicated questions?

The Next Step: Finding x_2

In our example $x_2 = 3$. But what do we do in general to **find** x_2.

Do we need to ask complicated questions?

No.

The Next Step: Finding x_2

In our example $x_2 = 3$. But what do we do in general to **find** x_2.

Do we need to ask complicated questions?

No.

We know that there are ∞ y such that $\mathrm{COL}(x_1, y) = c_1$.

The Next Step: Finding x_2

In our example $x_2 = 3$. But what do we do in general to **find** x_2.

Do we need to ask complicated questions?

No.

We know that there are ∞ y such that $\mathrm{COL}(x_1, y) = c_1$.

Compute $\mathrm{COL}(x_1, x_1 + 1)$, $\mathrm{COL}(x_1, x_1 + 2)$, ... until find y with $\mathrm{COL}(x_1, y) = c_1$.

The Next Step: Finding x_2

In our example $x_2 = 3$. But what do we do in general to **find** x_2.

Do we need to ask complicated questions?

No.

We know that there are ∞ y such that $\mathrm{COL}(x_1, y) = c_1$.

Compute $\mathrm{COL}(x_1, x_1 + 1)$, $\mathrm{COL}(x_1, x_1 + 2)$, ... until find y with $\mathrm{COL}(x_1, y) = c_1$.

This must happen since there are an infinite number of such y.

The Next Step: Finding x_2

In our example $x_2 = 3$. But what do we do in general to **find** x_2.

Do we need to ask complicated questions?

No.

We know that there are ∞ y such that $\mathrm{COL}(x_1, y) = c_1$.

Compute $\mathrm{COL}(x_1, x_1 + 1)$, $\mathrm{COL}(x_1, x_1 + 2)$, ... until find y with $\mathrm{COL}(x_1, y) = c_1$.

This must happen since there are an infinite number of such y.

Hence x_2 can be found easily.

The Next Step: Finding x_2

In our example $x_2 = 3$. But what do we do in general to **find** x_2.

Do we need to ask complicated questions?

No.

We know that there are ∞ y such that $\mathrm{COL}(x_1, y) = c_1$.

Compute $\mathrm{COL}(x_1, x_1 + 1)$, $\mathrm{COL}(x_1, x_1 + 2)$, ... until find y with $\mathrm{COL}(x_1, y) = c_1$.

This must happen since there are an infinite number of such y.

Hence x_2 can be found easily.

What about c_2. This is the **Key**.

Finding c_2

Finding c_2

Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) = \mathbf{R} \wedge \mathrm{COL}(x_2, y) = \mathbf{R}]$?

Finding c_2

Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) = \mathbf{R} \wedge \mathrm{COL}(x_2, y) = \mathbf{R}]$?

If YES then $\exists^{\infty}$ **R** coming out of x_2. Set $c_1 = \mathbf{R}$.

Finding c_2

Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) = \mathbf{R} \wedge \mathrm{COL}(x_2, y) = \mathbf{R}]$?

If YES then $\exists^\infty$ **R** coming out of x_2. Set $c_1 = \mathbf{R}$.

If NO then $\exists^\infty$ **B** coming out of x_2. Set $c_1 = \mathbf{B}$.

Finding c_2

Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) = \mathbf{R} \wedge \mathrm{COL}(x_2, y) = \mathbf{R}]$?

If YES then $\exists^\infty$ **R** coming out of x_2. Set $c_1 = \mathbf{R}$.

If NO then $\exists^\infty$ **B** coming out of x_2. Set $c_1 = \mathbf{B}$.

Output (x_2, c_2).

Finding c_2

Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) = \mathbf{R} \wedge \mathrm{COL}(x_2, y) = \mathbf{R}]$?

If YES then $\exists^\infty$ **R** coming out of x_2. Set $c_1 = \mathbf{R}$.

If NO then $\exists^\infty$ **B** coming out of x_2. Set $c_1 = \mathbf{B}$.

Output (x_2, c_2).

This is how we ignore the elements who disagreed with x_1, c_1.

Finding c_2

Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) = \mathbf{R} \wedge \mathrm{COL}(x_2, y) = \mathbf{R}]$?

If YES then $\exists^\infty$ **R** coming out of x_2. Set $c_1 = \mathbf{R}$.

If NO then $\exists^\infty$ **B** coming out of x_2. Set $c_1 = \mathbf{B}$.

Output (x_2, c_2).

This is how we ignore the elements who disagreed with x_1, c_1.

We ask if an ∞ number of y that agreed with x_1, c_1 have $\mathrm{COL}(x_2, y) = \mathbf{R}$.

Finding c_2

Ask $(\forall x)(\exists y > x)[\mathrm{COL}(x_1, y) = \mathbf{R} \wedge \mathrm{COL}(x_2, y) = \mathbf{R}]$?

If YES then $\exists^\infty$ **R** coming out of x_2. Set $c_1 = \mathbf{R}$.

If NO then $\exists^\infty$ **B** coming out of x_2. Set $c_1 = \mathbf{B}$.

Output (x_2, c_2).

This is how we ignore the elements who disagreed with x_1, c_1.

We ask if an ∞ number of y that agreed with x_1, c_1 have $\mathrm{COL}(x_2, y) = \mathbf{R}$.

Output (x_2, c_2).

Keep Doing This

Keep Doing This

Keep doing this

We Have our Process. Now What?

We Have our Process. Now What?

We have a Turing Machine that asks Π_2 questions, runs forever, and outputs

We Have our Process. Now What?

We have a Turing Machine that asks Π_2 questions, runs forever, and outputs

(x_1, c_1), (x_2, c_2), $\cdots$

We Have our Process. Now What?

We have a Turing Machine that asks Π_2 questions, runs forever, and outputs

(x_1, c_1), (x_2, c_2), $\cdots$

In the Original Proof we said that either **R** or **B** appears ∞ often.

We Have our Process. Now What?

We have a Turing Machine that asks Π_2 questions, runs forever, and outputs

(x_1, c_1), (x_2, c_2), $\cdots$

In the Original Proof we said that either **R** or **B** appears ∞ often.

Thats True But we don't know which one.

We Have our Process. Now What?

We have a Turing Machine that asks Π_2 questions, runs forever, and outputs

(x_1, c_1), (x_2, c_2), $\cdots$

In the Original Proof we said that either **R** or **B** appears ∞ often.

Thats True But we don't know which one.

We are going to **two** programs.

We Have our Process. Now What?

We have a Turing Machine that asks Π_2 questions, runs forever, and outputs

(x_1, c_1), (x_2, c_2), $\cdots$

In the Original Proof we said that either **R** or **B** appears ∞ often.

Thats True But we don't know which one.

We are going to **two** programs.

At least one of them will be an algorithm for an ∞ homog set that makes queries Π_2 queries.

RED Algorithms

ALG-RED An algorithm that asks Π_2 questions.

RED Algorithms

ALG-RED An algorithm that asks Π_2 questions.
1) Input x

RED Algorithms

ALG-RED An algorithm that asks Π_2 questions.
1) Input x
2) Run the process until either:
    you see $(x, \mathbf{R})$. Output YES and stop.
    you see (x_i, c_i), (x_{i+1}, c_{i+1}) where $x_i < x < x_{i+1}$. Output NO and stop.

RED Algorithms

ALG-RED An algorithm that asks Π_2 questions.
1) Input x
2) Run the process until either:
    you see $(x, \mathbf{R})$. Output YES and stop.
    you see (x_i, c_i), (x_{i+1}, c_{i+1}) where $x_i < x < x_{i+1}$. Output NO and stop.

This is an algorithm with queries to Π_2 for a **R**-homog set.

RED Algorithms

ALG-RED An algorithm that asks Π_2 questions.
1) Input x
2) Run the process until either:
    you see $(x, \mathbf{R})$. Output YES and stop.
    you see (x_i, c_i), (x_{i+1}, c_{i+1}) where $x_i < x < x_{i+1}$. Output NO and stop.

This is an algorithm with queries to Π_2 for a **R**-homog set.

One can show that any set computable with queries to Π_2 is in Σ_3.

RED Algorithms

ALG-RED An algorithm that asks Π_2 questions.
1) Input x
2) Run the process until either:
    you see $(x, \mathbf{R})$. Output YES and stop.
    you see (x_i, c_i), (x_{i+1}, c_{i+1}) where $x_i < x < x_{i+1}$. Output NO and stop.

This is an algorithm with queries to Π_2 for a **R**-homog set.

One can show that any set computable with queries to Π_2 is in Σ_3.

Hence there is a Σ_3 **R**-homog set.

RED Algorithms

ALG-RED An algorithm that asks Π_2 questions.
1) Input x
2) Run the process until either:
    you see $(x, \mathbf{R})$. Output YES and stop.
    you see (x_i, c_i), (x_{i+1}, c_{i+1}) where $x_i < x < x_{i+1}$. Output NO and stop.

This is an algorithm with queries to Π_2 for a **R**-homog set.

One can show that any set computable with queries to Π_2 is in Σ_3.

Hence there is a Σ_3 **R**-homog set.

Are We Done? No, the set could be finite.

BLUE Algorithms

ALG-BLUE An algorithm that asks Π_2 questions.

BLUE Algorithms

ALG-BLUE An algorithm that asks Π_2 questions.
1) Input x

BLUE Algorithms

ALG-BLUE An algorithm that asks Π_2 questions.

1) Input x

2) Run the process until either:

you see $(x, \mathbf{B})$. Output YES and stop.

you see (x_i, c_i), (x_{i+1}, c_{i+1}) where $x_i < x < x_{i+1}$. Output NO and stop.

BLUE Algorithms

ALG-BLUE An algorithm that asks Π_2 questions.
1) Input x
2) Run the process until either:
    you see $(x, \mathbf{B})$. Output YES and stop.
    you see (x_i, c_i), (x_{i+1}, c_{i+1}) where $x_i < x < x_{i+1}$. Output NO and stop.

This is an algorithm with queries to Π_2 for a **B**-homog set.

BLUE Algorithms

ALG-BLUE An algorithm that asks Π_2 questions.
1) Input x
2) Run the process until either:
    you see $(x, \mathbf{B})$. Output YES and stop.
    you see (x_i, c_i), (x_{i+1}, c_{i+1}) where $x_i < x < x_{i+1}$. Output NO and stop.

This is an algorithm with queries to Π_2 for a **B**-homog set.

One can show that any set computable with queries to Π_2 is in Σ_3.

BLUE Algorithms

ALG-BLUE An algorithm that asks Π_2 questions.
1) Input x
2) Run the process until either:
    you see $(x, \mathbf{B})$. Output YES and stop.
    you see (x_i, c_i), (x_{i+1}, c_{i+1}) where $x_i < x < x_{i+1}$. Output NO and stop.

This is an algorithm with queries to Π_2 for a **B**-homog set.

One can show that any set computable with queries to Π_2 is in Σ_3.

Hence there is a Σ_3 **B**-homog set.

BLUE Algorithms

ALG-BLUE An algorithm that asks Π_2 questions.
1) Input x
2) Run the process until either:
   you see $(x, \mathbf{B})$. Output YES and stop.
   you see (x_i, c_i), (x_{i+1}, c_{i+1}) where $x_i < x < x_{i+1}$. Output NO and stop.

This is an algorithm with queries to Π_2 for a **B**-homog set.

One can show that any set computable with queries to Π_2 is in Σ_3.

Hence there is a Σ_3 **B**-homog set.

Are We Done? No, the set could be finite.

Now What

Now What

If there are an ∞ number of **R** points then **ALG-RED** shows there is an ∞ Σ_3- **R**-homog set

Now What

If there are an ∞ number of **R** points then **ALG-RED** shows there is an ∞ Σ_3- **R**-homog set

If there are an ∞ number of **B** points then **ALG-BLUE** shows there is an ∞ Σ_3- **B**-homog set

Now What

If there are an ∞ number of **R** points then **ALG-RED** shows there is an ∞ Σ_3- **R**-homog set

If there are an ∞ number of **B** points then **ALG-BLUE** shows there is an ∞ Σ_3- **B**-homog set

In either case there is an ∞ Σ_3-homog set.

Now What

If there are an ∞ number of **R** points then **ALG-RED** shows there is an ∞ Σ_3- **R**-homog set

If there are an ∞ number of **B** points then **ALG-BLUE** shows there is an ∞ Σ_3- **B**-homog set

In either case there is an ∞ Σ_3-homog set.

The proof is nonconstructive.

Now What

If there are an ∞ number of **R** points then **ALG-RED** shows there is an ∞ Σ_3- **R**-homog set

If there are an ∞ number of **B** points then **ALG-BLUE** shows there is an ∞ Σ_3- **B**-homog set

In either case there is an ∞ Σ_3-homog set.

The proof is nonconstructive.

It cannot be made constructive.

Now What

If there are an ∞ number of **R** points then **ALG-RED** shows there is an ∞ Σ_3- **R**-homog set

If there are an ∞ number of **B** points then **ALG-BLUE** shows there is an ∞ Σ_3- **B**-homog set

In either case there is an ∞ Σ_3-homog set.

The proof is nonconstructive.

It cannot be made constructive. I probably proved that.

Now What

If there are an ∞ number of **R** points then **ALG-RED** shows there is an ∞ Σ_3- **R**-homog set

If there are an ∞ number of **B** points then **ALG-BLUE** shows there is an ∞ Σ_3- **B**-homog set

In either case there is an ∞ Σ_3-homog set.

The proof is nonconstructive.

It cannot be made constructive. I probably proved that.

Does this limit the applicability of the theorem?

Now What

If there are an ∞ number of **R** points then **ALG-RED** shows there is an ∞ Σ_3- **R**-homog set

If there are an ∞ number of **B** points then **ALG-BLUE** shows there is an ∞ Σ_3- **B**-homog set

In either case there is an ∞ Σ_3-homog set.

The proof is nonconstructive.

It cannot be made constructive. I probably proved that.

Does this limit the applicability of the theorem?
Not even a little!

Better/More Is Known for Comp. Inf Ramsey

Better/More Is Known for Comp. Inf Ramsey

$\forall$ comp. $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ $\exists$ an ∞ homog Π_2 set.

Better/More Is Known for Comp. Inf Ramsey

$\forall$ comp. $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ $\exists$ an ∞ homog Π_2 set.

$\exists$ comp. $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ with no ∞ homog Σ_2 homog set.

Better/More Is Known for Comp. Inf Ramsey

$\forall$ comp. $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ $\exists$ an ∞ homog Π_2 set.

$\exists$ comp. $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ with no ∞ homog Σ_2 homog set.

$\forall$ comp $\mathrm{COL}\colon \binom{\mathbb{N}}{a} \to [2]$ $\exists$ an ∞ homog Π_a set.

Better/More Is Known for Comp. Inf Ramsey

$\forall$ comp. $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ $\exists$ an ∞ homog Π_2 set.

$\exists$ comp. $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ with no ∞ homog Σ_2 homog set.

$\forall$ comp $\mathrm{COL}\colon \binom{\mathbb{N}}{a} \to [2]$ $\exists$ an ∞ homog Π_a set.

$\exists$ comp. $\mathrm{COL}\colon \binom{\mathbb{N}}{2} \to [2]$ with no ∞ homog Σ_a homog set.

What Is Known for Inf Can Ramsey?

What Is Known for Inf Can Ramsey?

∞ a-ary Can Ramsey can be proven from ∞ $2a-1$-ary Ramsey.

What Is Known for Inf Can Ramsey?

∞ a-ary Can Ramsey can be proven from ∞ $2a-1$-ary Ramsey. Hence

What Is Known for Inf Can Ramsey?

∞ a-ary Can Ramsey can be proven from ∞ $2a-1$-ary Ramsey. Hence

Thm $\forall$ COL: $\binom{N}{a} \to [2]$ there is an ∞ Π_{2a-1} set that is A-homog for some $A \subseteq \{1, \ldots, a\}$.

What Is Known for Inf Can Ramsey?

∞ a-ary Can Ramsey can be proven from ∞ $2a-1$-ary Ramsey. Hence

Thm $\forall$ COL: $\binom{N}{a} \to [2]$ there is an ∞ Π_{2a-1} set that is A-homog for some $A \subseteq \{1, \ldots, a\}$.

Mileti's motivation was to prove ∞ a-ary Can Ramsey a diff way to reduce the complexity.

What Is Known for Inf Can Ramsey?

∞ a-ary Can Ramsey can be proven from ∞ $2a-1$-ary Ramsey. Hence

Thm $\forall$ COL: $\binom{N}{a} \to [2]$ there is an ∞ Π_{2a-1} set that is A-homog for some $A \subseteq \{1, \ldots, a\}$.

Mileti's motivation was to prove ∞ a-ary Can Ramsey a diff way to reduce the complexity.

His construction if looked at carefully yields:

What Is Known for Inf Can Ramsey?

∞ a-ary Can Ramsey can be proven from ∞ $2a-1$-ary Ramsey. Hence

Thm $\forall$ COL: $\binom{N}{a} \to [2]$ there is an ∞ Π_{2a-1} set that is A-homog for some $A \subseteq \{1, \ldots, a\}$.

Mileti's motivation was to prove ∞ a-ary Can Ramsey a diff way to reduce the complexity.

His construction if looked at carefully yields:

Thm $\forall$ COL: $\binom{N}{a} \to [2]$ there is an ∞ Π_{2a-2} set that is A-homog for some $A \subseteq \{1, \ldots, a\}$.

What Is Known for Inf Can Ramsey?

∞ a-ary Can Ramsey can be proven from ∞ $2a-1$-ary Ramsey. Hence

Thm $\forall$ COL: $\binom{N}{a} \to [2]$ there is an ∞ Π_{2a-1} set that is A-homog for some $A \subseteq \{1, \ldots, a\}$.

Mileti's motivation was to prove ∞ a-ary Can Ramsey a diff way to reduce the complexity.

His construction if looked at carefully yields:

Thm $\forall$ COL: $\binom{N}{a} \to [2]$ there is an ∞ Π_{2a-2} set that is A-homog for some $A \subseteq \{1, \ldots, a\}$.

Thm $\exists$ COL: $\binom{N}{a} \to [2]$ with no ∞ Σ_a set that is A-homog for some $A \subseteq \{1, \ldots, a\}$.

What Is Known for Inf Can Ramsey?

∞ a-ary Can Ramsey can be proven from ∞ $2a-1$-ary Ramsey. Hence

Thm $\forall$ COL: $\binom{N}{a} \to [2]$ there is an ∞ Π_{2a-1} set that is A-homog for some $A \subseteq \{1, \ldots, a\}$.

Mileti's motivation was to prove ∞ a-ary Can Ramsey a diff way to reduce the complexity.

His construction if looked at carefully yields:

Thm $\forall$ COL: $\binom{N}{a} \to [2]$ there is an ∞ Π_{2a-2} set that is A-homog for some $A \subseteq \{1, \ldots, a\}$.

Thm $\exists$ COL: $\binom{N}{a} \to [2]$ with no ∞ Σ_a set that is A-homog for some $A \subseteq \{1, \ldots, a\}$.

Open Narrow the gap between Π_{2a-1} and Σ_a.