Computable Combinatorics

Joe Mileti
Grinnell College

November 2, 2012

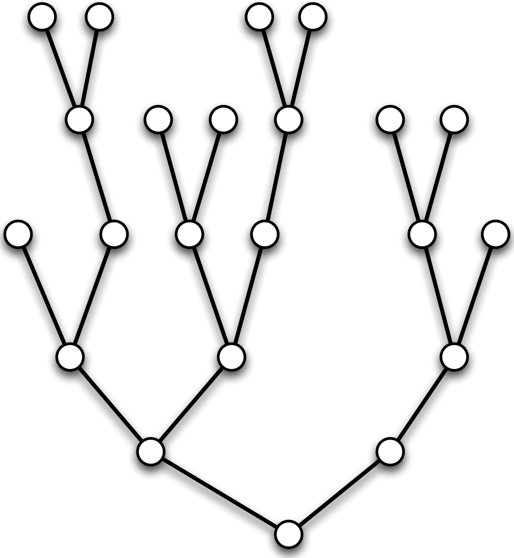
Trees

Definition
Let $2^{<\omega}$ be the set of all finite sequences of zeros and ones. We can view $2^{<\omega}$ as an infinite binary branching tree.

Definition
A *subtree* of $2^{<\omega}$ is a subset of $2^{<\omega}$ that is closed under initial segments.

Definition
Let T be a subtree of $2^{<\omega}$. A *branch* through T is an infinite sequence of zeros and ones such that every initial segment is an element of T.

Trees



König's Lemma

Theorem (König's Lemma)

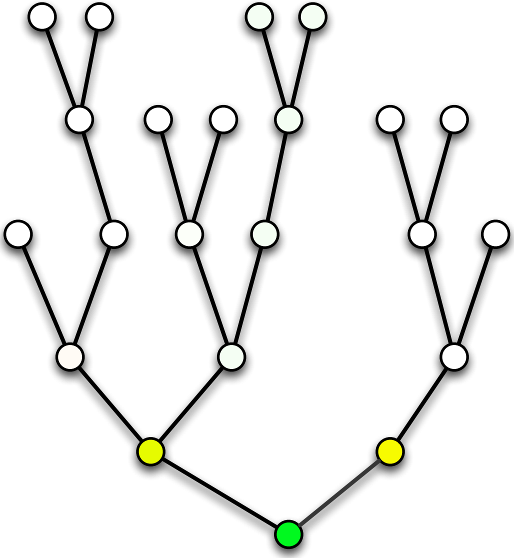
Every infinite subtree of $2^{<\omega}$ *has an infinite branch.*

König's Lemma is one way to express the compactness of 2^{ω}.

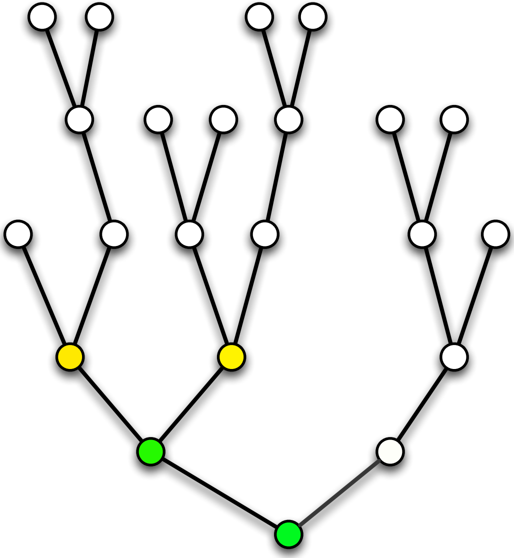
Corollary

If G *is a countable graph with the property that every finite subgraph of* G *is* k*-colorable, then* G *is* k*-colorable.*

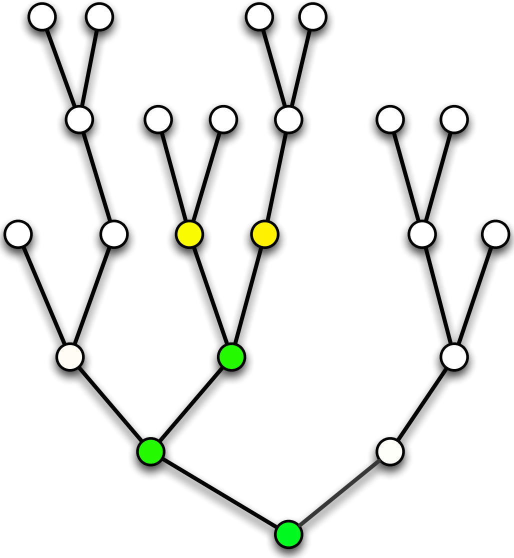
König's Lemma



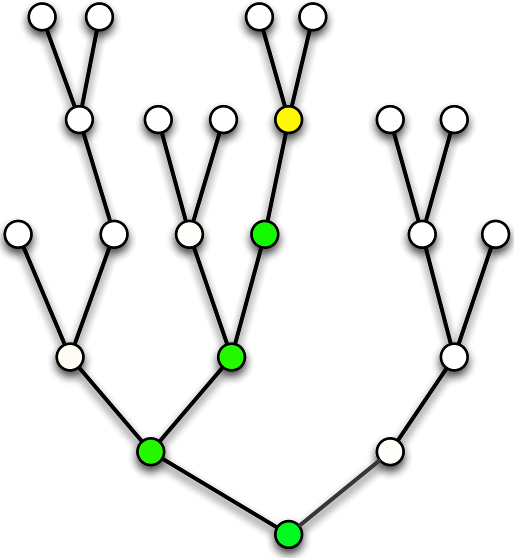
König's Lemma



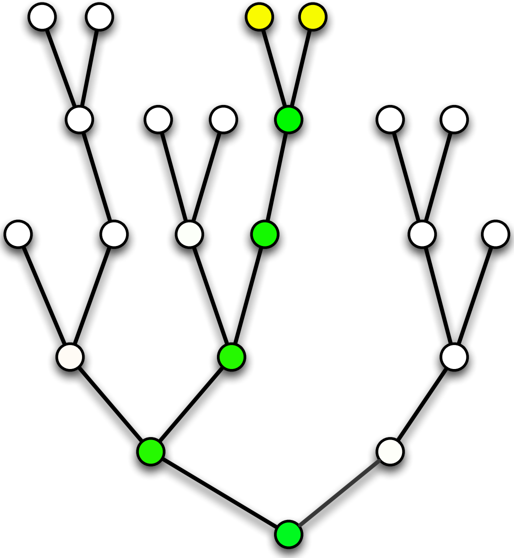
König's Lemma



König's Lemma



König's Lemma



Computable Trees

By a diagonalization argument, it is not hard to build an infinite computable subtree of $2^{<\omega}$ with no infinite computable branch.

Question

Let T be an infinite computable subtree of $2^{<\omega}$. How much noncomputable knowledge do we need to know in order to build a branch?

Looking at the above proof, we repeatedly ask whether certain computable sets are infinite, which is a Π^0_2 question, i.e. a question that can be written in the form $\forall x \exists y R(x, y)$ where $R(x, y)$ is computable.

Turing Reducibility

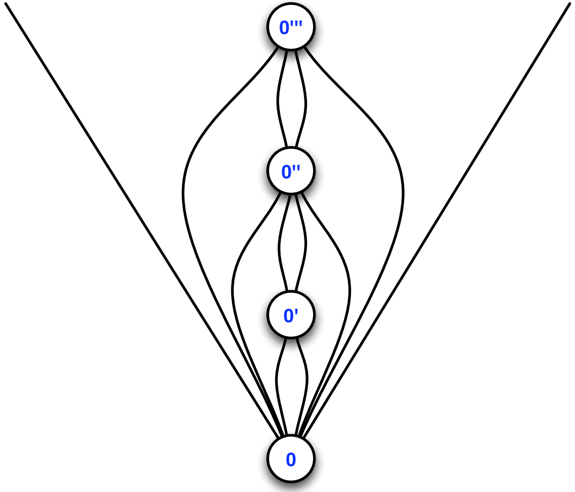
Definition

Let $A, B \subseteq \omega$. We define $A \leq_T B$ to mean that there exists a Turing machine that, when equipped with an oracle for B, computes A. We define $A \equiv_T B$ to mean both $A \leq_T B$ and $B \leq_T A$.

Equivalence classes of the relation $\equiv_T$ are called *Turing degrees*. There is minimal element, $\mathbf{0}$, the Turing degree of computable sets.

Given any set $A \subseteq \omega$, the halting problem relative to A is denoted by A'. It turns out that $A <_T A'$ and this operation is well-defined on the Turing degrees.

Turing Degrees



Using Oracles

Intuition: Having $0'$ as an oracle allows you to answer any question that begins with a block of identical quantifiers (either $\exists$ or $\forall$) followed by a computable question.

Hilbert's Tenth Problem: Devise an algorithm that given a polynomial in many variables with integer coefficients, determines whether it has integer roots (i.e. a tuple of integers to plug into the polynomial which gives the value 0).

Let P be the set of all such polynomials which have integer roots.

Hilbert's Tenth Problem

Proposition

$P \leq_T 0'$.

Proof.

Suppose that you are given a polynomial $p(x_1, x_2, \ldots, x_n)$. Since $\mathbb{Z}^n$ is countable, we may list its elements. Create a computer program which runs through each tuple in $\mathbb{Z}^n$ in order and tests if you get zero when plugged in. If it finds such a tuple, the program halts. Otherwise it proceeds to the next tuple. Then $p(x_1, x_2, \ldots, x_n)$ has integer roots if and only the program halts. $\square$

Theorem (Davis, Matiyasevich, Putnam, Robinson)

$0' \leq_T P$, so $P \equiv_T 0'$.

Degrees of Branches

$0''$ lets you answer any Σ^0_2 or Π^0_2 question, and similarly $0^{(n)}$ has the ability to answer Σ^0_n or Π^0_n questions.

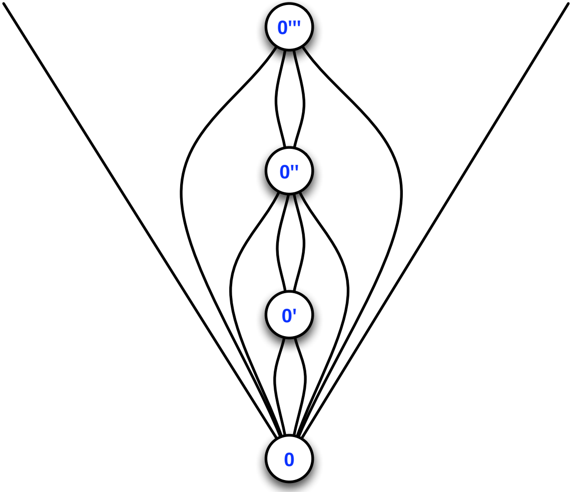
Proposition

Every infinite computable subtree of $2^{<\omega}$ has a $\mathbf{0}''$-computable infinite branch.

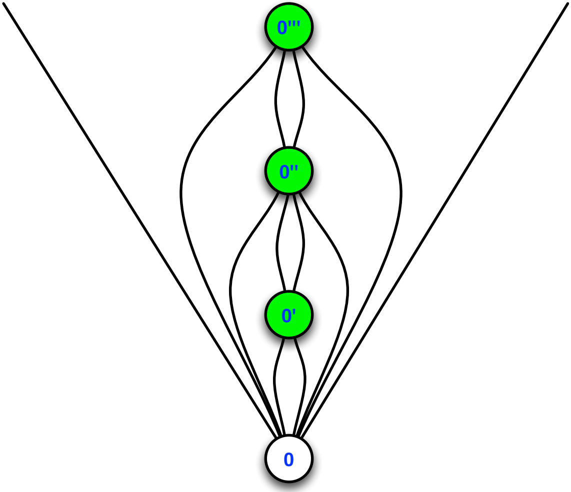
Proposition (Kreisel)

Every infinite computable subtree of $2^{<\omega}$ has a $\mathbf{0}'$-computable infinite branch.

Tree Degrees



Tree Degrees



Degrees of Branches

Definition

Given two degrees, we write $\mathbf{a} \gg \mathbf{b}$ to mean that every infinite $\mathbf{b}$-computable subtree of $2^{<\omega}$ has an infinite $\mathbf{a}$-computable branch.

- $\mathbf{a} \gg \mathbf{b} \Rightarrow \mathbf{a} > \mathbf{b}$
- $\mathbf{a}' \gg \mathbf{a}$

Trees correspond to closed subsets of 2^ω. Thus, computable trees can be thought of as coding "computably closed" sets. If you have a problem such that you can computably recognize at some finite stage when something is not a solution, then you can code solutions as branches through a computable tree.

Throughout Mathematics

Suppose that $\mathbf{a} \gg \mathbf{0}$. We have the following.

- Every computable k-colorable graph has an $\mathbf{a}$-computable k-coloring.
- Every consistent axiomatizable theory has an $\mathbf{a}$-computable complete extension.
- Every computable commutative ring with identity has an $\mathbf{a}$-computable prime ideal.
- There exists an $\mathbf{a}$-computable sequence such that no initial segment is compressible beyond a fixed finite amount.

Degrees of Branches

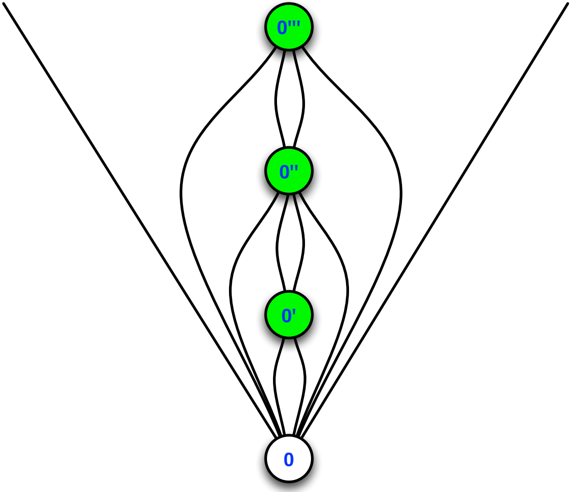
Although $\mathbf{0}'$ suffices to find branches, there is some potential wiggle room to lower complexity. We need only be able to do the following: Given two Π^0_1 statements, at least one of which is true, pick a true one.

Notice that if you're give two Σ^0_1 statements, at least one of which is true, you can *computably* pick a true one.

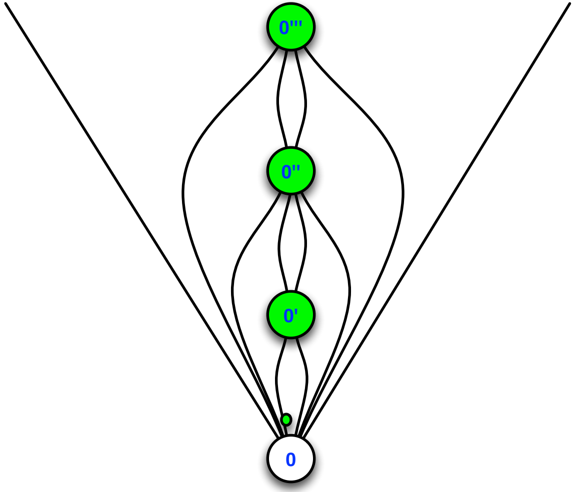
Theorem (Low Basis Theorem - Jocksuch, Soare)

There exists $\mathbf{a} \gg \mathbf{0}$ *such that* $\mathbf{a}' = \mathbf{0}'$.

Tree Degrees



Tree Degrees



Homogeneous Sets

Definition

Given a set A and an $n \in \omega$, we let $[A]^n = \{x \subseteq A : |x| = n\}$.

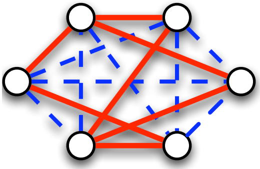
Definition

A function $f : [m]^n \to k$ is called a k-coloring of $[m]^n$. A set $H \subseteq m$ is said to be *homogeneous for f* if f is constant on $[H]^n$.

Finding a Homogeneous Triangle

Proposition

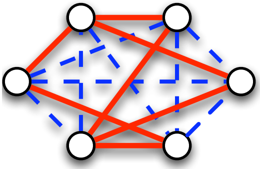
For every $f : [6]^2 \to 2$, there exists a set H homogeneous for f with $|H| = 3$.

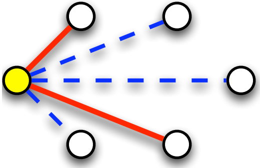


Finding a Homogeneous Triangle

Proposition

For every $f : [6]^2 \to 2$*, there exists a set* H *homogeneous for* f *with* $|H| = 3$.

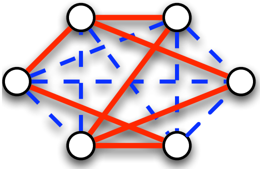


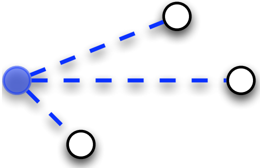


Finding a Homogeneous Triangle

Proposition

For every $f : [6]^2 \to 2$, *there exists a set* H *homogeneous for* f *with* $|H| = 3$.

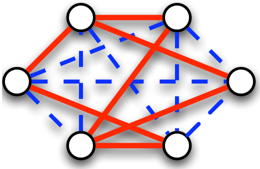


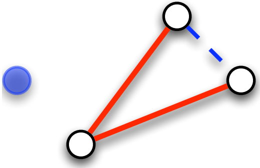


Finding a Homogeneous Triangle

Proposition

For every $f : [6]^2 \to 2$, *there exists a set* H *homogeneous for* f *with* $|H| = 3$.

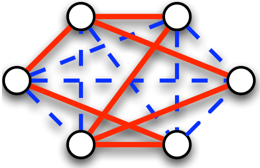


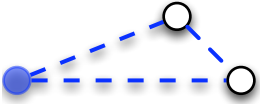


Finding a Homogeneous Triangle

Proposition

For every $f : [6]^2 \to 2$*, there exists a set* H *homogeneous for* f *with* $|H| = 3$*.*





Ramsey's Theorem

Theorem (Finite Version)

For every $n, k, \ell \in \omega$, there exists $m \in \omega$ such that every $f : [m]^n \to k$ has a homogeneous set H with $|H| = \ell$.

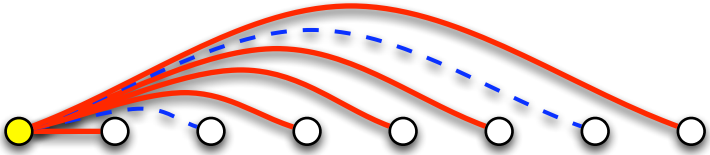
Theorem (Infinite Version)

For every $n, k \in \omega$, every $f : [\omega]^n \to k$ has an infinite homogeneous set.

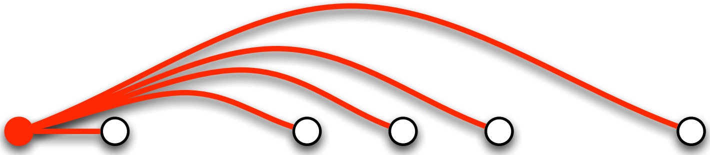
Corollaries of Ramsey's Theorem

- Every sequence of real numbers has either an infinite ascending subsequence or an infinite descending subsequence.
- Every infinite linear ordering has either an infinite ascending sequence or an infinite descending sequence.
- Every infinite partial ordering has either an infinite chain or an infinite antichain.

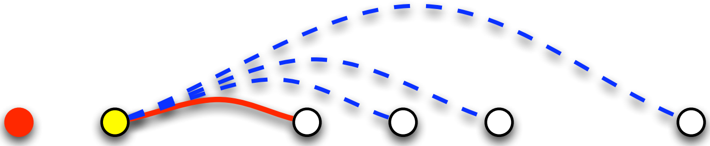
Proving the Infinite Version



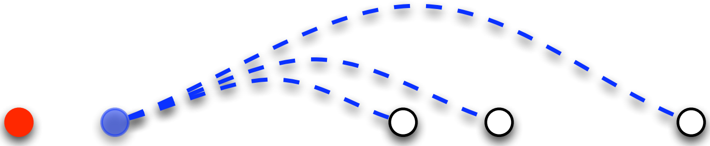
Proving the Infinite Version



Proving the Infinite Version



Proving the Infinite Version



Proving the Infinite Version

Proving the Infinite Version

Proving the Infinite Version

Analyzing the Proof

Analyzing the proof, it follows that every computable $f : [\omega]^2 \to 2$ has an infinite $\mathbf{0}''$-computable homogeneous set.

The naive attempt to code homogenous sets as branches of a tree fails. You can detect violation of homogeneity at a finite stage, but branches of the corresponding tree might correspond to finite sets. Requiring an infinite homogeneous set seems to introduce additional complexity.

From the other direction, every subset of a homogeneous set is homogeneous, so it is challenging to code things into all homogeneous sets.

Computable Homogeneous Sets?

Theorem (Specker)

There exists a computable $f : [\omega]^2 \to 2$ with no infinite computable homogeneous set.

We need to build our coloring by defeating each possible computer program. Suppose that you have one computer program written by an adversary and you want to defeat it.

Computable Homogeneous Sets?

Idea 1: Sit around and wait for the program to output "1" (green) on three distinct vertices. Make one pair red and another blue.

Computable Homogeneous Sets?

Idea 1: Sit around and wait for the program to output “1” (green) on three distinct vertices. Make one pair red and another blue.

Computable Homogeneous Sets?

Idea 1: Sit around and wait for the program to output "1" (green) on three distinct vertices. Make one pair red and another blue.

Computable Homogeneous Sets?

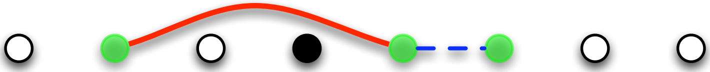
Idea 1: Sit around and wait for the program to output “1” (green) on three distinct vertices. Make one pair red and another blue.

Computable Homogeneous Sets?

Idea 1: Sit around and wait for the program to output "1" (green) on three distinct vertices. Make one pair red and another blue.

Computable Homogeneous Sets?

Idea 1: Sit around and wait for the program to output "1" (green) on three distinct vertices. Make one pair red and another blue.



Computable Homogeneous Sets?

Idea 2: Sit around and wait for the program to output "1" (green) on two distinct vertices. From this point onward, make these two vertices have different colors with all others.

Computable Homogeneous Sets?

Idea 2: Sit around and wait for the program to output "1" (green) on two distinct vertices. From this point onward, make these two vertices have different colors with all others.

Computable Homogeneous Sets?

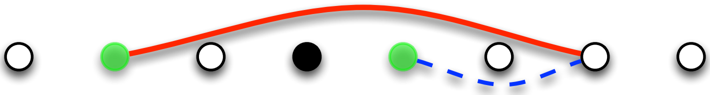
Idea 2: Sit around and wait for the program to output “1” (green) on two distinct vertices. From this point onward, make these two vertices have different colors with all others.

Computable Homogeneous Sets?

Idea 2: Sit around and wait for the program to output "1" (green) on two distinct vertices. From this point onward, make these two vertices have different colors with all others.

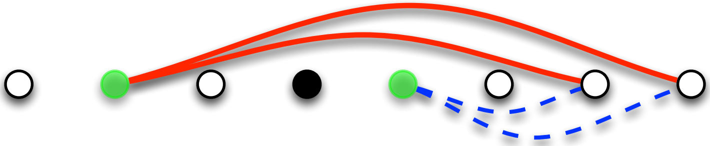
Computable Homogeneous Sets?

Idea 2: Sit around and wait for the program to output "1" (green) on two distinct vertices. From this point onward, make these two vertices have different colors with all others.



Computable Homogeneous Sets?

Idea 2: Sit around and wait for the program to output "1" (green) on two distinct vertices. From this point onward, make these two vertices have different colors with all others.



Homogenous Sets

Theorem (Jockusch)

There exists a computable $f : [\omega]^2 \to 2$ with no infinite $\mathbf{0}'$-computable homogeneous set.

Similar to König's Lemma but at a higher level, we can get by with the ability to determine which of two Π^0_2 statements is true.

Theorem (Jockusch)

Let $\mathbf{a} \gg \mathbf{0}'$. Every computable $f : [\omega]^2 \to 2$ has an infinite $\mathbf{a}$-computable homogeneous set.

Ramsey's Theorem

Theorem (Cholak, Jockusch, Slaman)

Let $\mathbf{a} \gg \mathbf{0}'$*. Every computable* $f : [\omega]^2 \to 2$ *has an infinite homogeneous set* H *with the property that* $\mathbf{H}' \leq \mathbf{a}$*.*

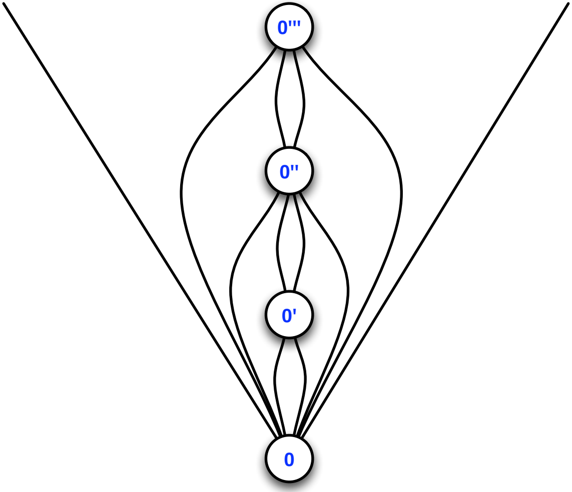
Corollary

Every computable $f : [\omega]^2 \to 2$ *has an infinite homogeneous set* H *with the property that* $\mathbf{H}'' \leq \mathbf{0}''$*.*

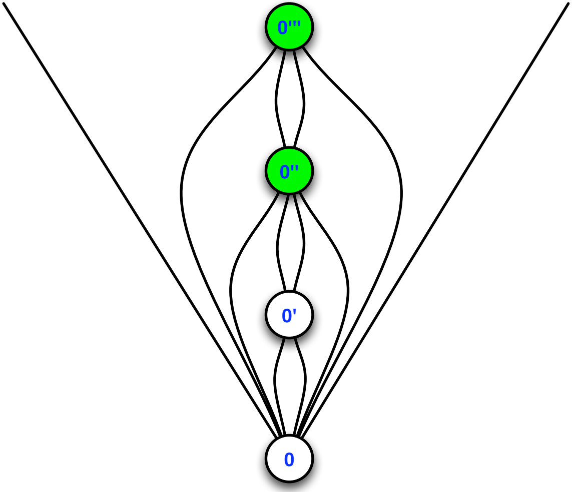
Theorem (Cholak, Jockusch, Slaman)

There exists a computable $f : [\omega]^2 \to 2$ *such that* $\mathbf{H}' \gg \mathbf{0}'$ *for all infinite sets* H *homogeneous for* f*.*

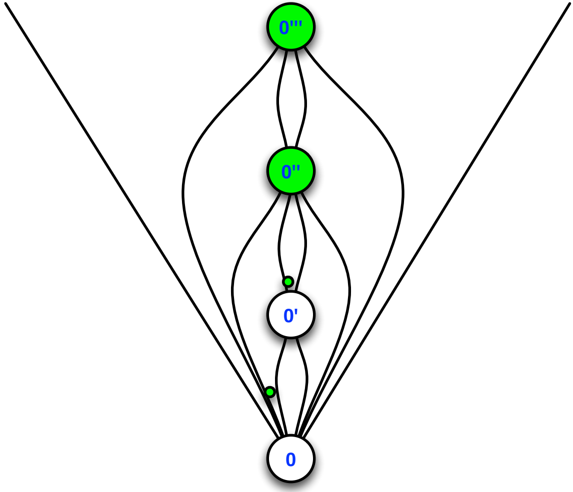
Locating Homogeneous Sets



Locating Homogeneous Sets



Locating Homogeneous Sets



Locating Homogeneous Sets

Theorem (Mileti)

For any degree $\mathbf{b}$ *with* $\mathbf{b}'' \leq \mathbf{0}''$*, there exists a computable* $f : [\omega]^2 \to 2$ *with no infinite* $\mathbf{b}$*-computable homogeneous set.*

Theorem (Mileti)

There exists a computable $f : [\omega]^2 \to 2$ *such that*

$$\mu(\{X \in 2^\omega : X \textit{ computes a infinite homogeneous set for } f\}) = 0$$

Rainbow Ramsey Theorem

Definition

A function $g\colon [\omega]^n \to \omega$ is called *k-bounded* if $|g^{-1}(c)| \leq k$ for all $c \in \omega$. A set R is called a *rainbow for g* if g is injective on $[R]^n$.

Theorem (Galvin)

For every $n, k \in \omega$, every k-bounded $g\colon [\omega]^n \to \omega$ has an infinite rainbow.

Rainbow from Ramsey

- Suppose that $g\colon [\omega]^n \to \omega$ is k-bounded.
- Fix a linear ordering $\prec$ of $[\omega]^n$.
- Define $f\colon [\omega]^n \to k$ by letting

  $$f(x) = |\{y \in [\omega]^n : y \prec x \text{ and } g(y) = g(x)\}|$$

  In words, saying $f(x) = 2$ means that there are exactly 2 tuples before x with the same color as x, so x is the third tuple of its color.
- Fix a set H homogeneous for f.
- Notice that H is a rainbow for g.

Homogeneous Sets vs. Rainbows

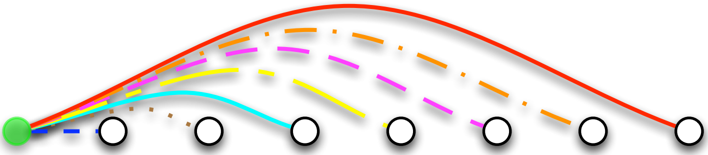
Proposition

If $\mathbf{a}$ *is a Turing degree such that every computable* $f\colon [\omega]^2 \to 2$ *has an* $\mathbf{a}$*-computable infinite homogeneous set, then every computable* 2*-bounded* $g\colon [\omega]^2 \to \omega$ *has an* $\mathbf{a}$*-computable infinite rainbow.*

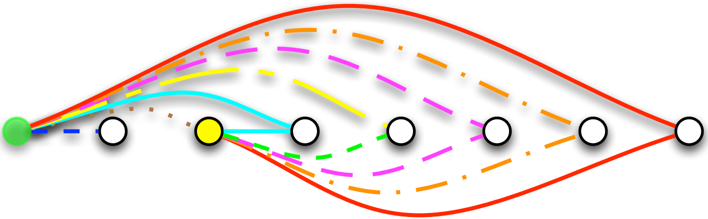
Question

Suppose that $\mathbf{a}$ *is a Turing degree such that every computable* 2*-bounded* $g\colon [\omega]^2 \to \omega$ *has an* $\mathbf{a}$*-computable infinite rainbow. Must every computable* $f\colon [\omega]^2 \to 2$ *have an* $\mathbf{a}$*-computable infinite homogeneous set?*

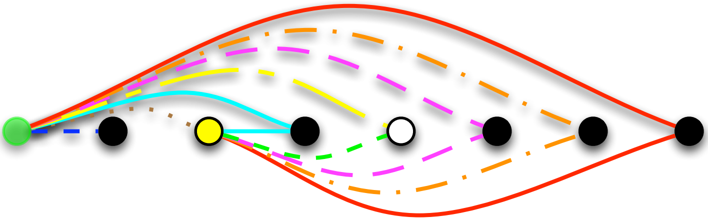
Constructing Rainbows



Constructing Rainbows



Constructing Rainbows



Randomly Building Rainbows

Suppose that $f\colon [\omega]^2 \to \omega$ is computable and 2-bounded. One way to build a rainbow is as follows:

- Suppose you have committed yourself to a finite set of vertices which leaves an infinite stock in play.
- Argue that only finitely many of the remaining vertices destroy this "infinite stock" property when included in your finite set. In fact, there is a (small) computable bound on how many such elements there can be.
- Pick one of the remaining vertices "at random". Unless you are really unlucky, then you will have infinitely many vertices still around to work with and you can continue.

Randomness and Rainbows

Theorem (Csima, Mileti)

Suppose that X is 2-random. Every computable 2-bounded $g\colon [\omega]^2 \to \omega$ has an X-computable infinite rainbow.

Corollary

There exists a Turing degree $\mathbf{a}$ with the following properties:

- *Every computable 2-bounded $g\colon [\omega]^2 \to \omega$ has an $\mathbf{a}$-computable infinite rainbow.*
- *There exists a computable $f : [\omega]^2 \to 2$ which has no $\mathbf{a}$-computable infinite homogeneous set.*

Changing the Number of Colors

All of the above results for Ramsey's Theorem do not depend on the number of colors (or the bound), and apply equally well to any $k \geq 2$. There are no known degree-theoretic differences between homogeneous sets for computable j-colorings and homogeneous sets for computable k-colorings.

Theorem (Dorais, Dzhafarov, Hirst, Mileti, Shafer)

Let $2 \leq j < k$. There do not exist computable transformations Φ and Ψ with the following properties.

- *If f is a k-coloring of $[\omega]^2$, then $\Phi(f)$ is a j-coloring of $[\omega]^2$.*
- *If f is a k-coloring of $[\omega]^2$ and H is an infinite homogeneous set for $\Phi(f)$, then $\Psi(H)$ is an infinite homogeneous set for f.*