Graphs between computable and highly computable

Oscar Levin

University of Northern Colorado

Joint Mathematics Meeting
Seattle, WA
January 8, 2016

Joint work with Matthew Jura and Tyler Markkanen

Computability and Graphs

A graph is computable provided there is an algorithm giving the edge relation.

$V = \mathbb{N}$. Is there an edge between vertices 7 and 253?

Compare: what vertices are adjacent to vertex 7?
This might be a harder question.

If a computable graph has a computable neighborhood relation, we say the graph is highly computable.

Computability and Graphs

A graph is computable provided there is an algorithm giving the edge relation.

$V = \mathbb{N}$. Is there an edge between vertices 7 and 253?

Compare: what vertices are adjacent to vertex 7?
This might be a harder question.

If a computable graph has a computable neighborhood relation, we say the graph is highly computable.

Computability and Graphs

A graph is computable provided there is an algorithm giving the edge relation.

$V = \mathbb{N}$. Is there an edge between vertices 7 and 253?

Compare: what vertices are adjacent to vertex 7?
This might be a harder question.

If a computable graph has a computable neighborhood relation, we say the graph is highly computable.

Computability and Graphs

A graph is computable provided there is an algorithm giving the edge relation.

$V = \mathbb{N}$. Is there an edge between vertices 7 and 253?

Compare: what vertices are adjacent to vertex 7?
This might be a harder question.

If a computable graph has a computable neighborhood relation, we say the graph is highly computable.

Computable Chromatic Number

The computable chromatic number $\chi^c(G)$ is the smallest n for which there is a computable proper vertex coloring using n colors.

Theorem (Bean)

There is a computable planar graph G with $\chi(G) = 3$ but $\chi^c(G) = \infty$.

Theorem (Schmerl)

Every highly computable graph has $\chi^c(G) \leq 2\chi(G) - 1$ (and this bound is tight).

Knowing the neighborhood of vertices helps color a graph.

Computable Chromatic Number

The computable chromatic number $\chi^c(G)$ is the smallest n for which there is a computable proper vertex coloring using n colors.

Theorem (Bean)

There is a computable planar graph G with $\chi(G) = 3$ but $\chi^c(G) = \infty$.

Theorem (Schmerl)

Every highly computable graph has $\chi^c(G) \leq 2\chi(G) - 1$ (and this bound is tight).

Knowing the neighborhood of vertices helps color a graph.

Computable Chromatic Number

The computable chromatic number $\chi^c(G)$ is the smallest n for which there is a computable proper vertex coloring using n colors.

Theorem (Bean)

There is a computable planar graph G with $\chi(G) = 3$ but $\chi^c(G) = \infty$.

Theorem (Schmerl)

Every highly computable graph has $\chi^c(G) \leq 2\chi(G) - 1$ (and this bound is tight).

Knowing the neighborhood of vertices helps color a graph.

Computable Sets

A subset of $\mathbb{N}$ is computable if there is some algorithm which determines membership.

Algorithm means... C++ program... or Turing machine. The algorithm is finite, but we have unlimited time and space.

$\{\varphi_e\}_{e \in \mathbb{N}}$ is a complete, effective list of all algorithms.

Non-Computable Sets

Not all sets are computable.

Example

$K = \{e \mid \varphi_e(e)\downarrow\}$ is not computable.

K is not computable, but it is computably enumerable (c.e.): there is an algorithm that lists all elements.

Non-Computable Sets

Not all sets are computable.

Example

$K = \{e \mid \varphi_e(e)\downarrow\}$ is not computable.

K is not computable, but it is computably enumerable (c.e.): there is an algorithm that lists all elements.

Between Computable and K

K is Turing complete for the c.e. sets.

That is, if A is a c.e. set, then we can compute A using K.

$A \leq_T B$ means there is an oracle Turing machine Φ_e which computes A using oracle B. So $A = \Phi_e^B$.

Post's Problem: is there a non-computable c.e. set $A <_T K$?
Yes.

Between Computable and K

K is Turing complete for the c.e. sets.

That is, if A is a c.e. set, then we can compute A using K.

$A \leq_T B$ means there is an oracle Turing machine Φ_e which computes A using oracle B. So $A = \Phi_e^B$.

Post's Problem: is there a non-computable c.e. set $A <_T K$?
Yes.

Between Computable and K

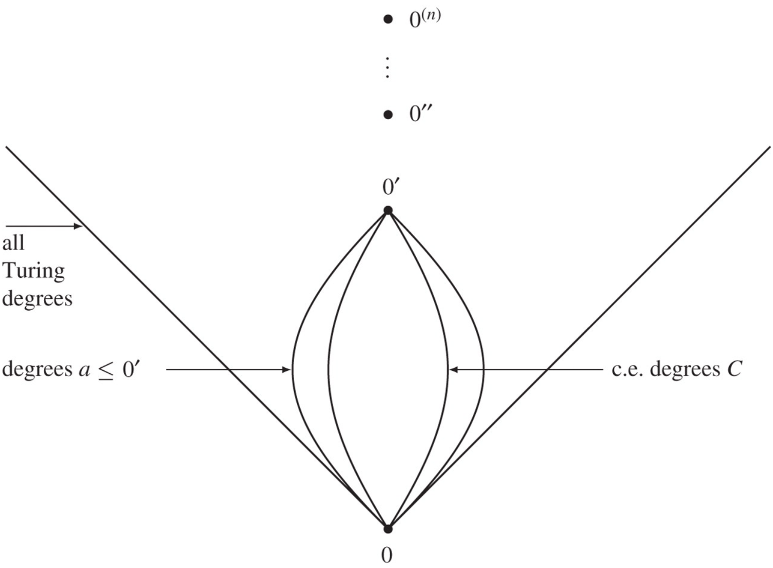
K is Turing complete for the c.e. sets.

That is, if A is a c.e. set, then we can compute A using K.

$A \leq_T B$ means there is an oracle Turing machine Φ_e which computes A using oracle B. So $A = \Phi_e^B$.

Post's Problem: is there a non-computable c.e. set $A <_T K$?
Yes.

(A small part of) The Big Picture



A-computable Graphs

How hard is it to compute the neighborhood relation, N_G?

No matter what $N_G \leq_T K$. If G is highly computable, $N_G \leq_T \emptyset$.

But there are sets between $\emptyset$ and K.

Definition (Gasarch and Lee)

A computable graph G is A-computable provided $N_G \leq_T A$.

A-computable Graphs

How hard is it to compute the neighborhood relation, N_G?

No matter what $N_G \leq_T K$. If G is highly computable, $N_G \leq_T \emptyset$.

But there are sets between $\emptyset$ and K.

Definition (Gasarch and Lee)

A computable graph G is A-computable provided $N_G \leq_T A$.

A-computable Graphs

How hard is it to compute the neighborhood relation, N_G?

No matter what $N_G \leq_T K$. If G is highly computable, $N_G \leq_T \emptyset$.

But there are sets between $\emptyset$ and K.

Definition (Gasarch and Lee)

A computable graph G is A-computable provided $N_G \leq_T A$.

Chromatic Number for c.e. A

Theorem (Gasarch and Lee)

For any non-computable c.e. set A, there exists an A-computable graph G with $\chi(G) = 3$ but $\chi^c(G) = \infty$

In other words: having more (but not complete) information about the neighborhood relation doesn't help.

The theorem generalizes to many (every?) graph property with different results for computable and highly computable graphs.

Chromatic Number for c.e. A

Theorem (Gasarch and Lee)

For any non-computable c.e. set A, there exists an A-computable graph G with $\chi(G) = 3$ but $\chi^c(G) = \infty$

In other words: having more (but not complete) information about the neighborhood relation doesn't help.

The theorem generalizes to many (every?) graph property with different results for computable and highly computable graphs.

Chromatic Number for c.e. A

Theorem (Gasarch and Lee)

For any non-computable c.e. set A, there exists an A-computable graph G with $\chi(G) = 3$ but $\chi^c(G) = \infty$

In other words: having more (but not complete) information about the neighborhood relation doesn't help.

The theorem generalizes to many (every?) graph property with different results for computable and highly computable graphs.

What if A is not c.e.?

There are non-c.e. sets A between $\emptyset$ and K. What about them?

Theorem (JLM)

There exists a non-computable set $A \leq_T K$ such that every A-computable graph with chromatic number 3 has finite computable chromatic number.

What if A is not c.e.?

There are non-c.e. sets A between $\emptyset$ and K. What about them?

Theorem (JLM)

There exists a non-computable set $A \leq_T K$ such that every A-computable graph with chromatic number 3 has finite computable chromatic number.

Proof Idea

Build A to be limit computable using a finite injury priority construction.

Essentially: for each potential A-computable graph, attempt to color (using the highly computable coloring algorithm). If something goes wrong, either use (a few) extra colors or change A to make the graph not A-computable.

Proof Idea

Build A to be limit computable using a finite injury priority construction.

Essentially: for each potential A-computable graph, attempt to color (using the highly computable coloring algorithm). If something goes wrong, either use (a few) extra colors or change A to make the graph not A-computable.

Using graphs to classify sets

A-computable graphs act like computable graphs	A-computable graphs act like highly computable graphs
$A = K$	$A = \emptyset$
All non-computable c.e. sets	Some non-computable non-c.e. sets
Some non-computable non-c.e. sets	A is low for graph neighborhood: every A-computable graph is highly computable.

Using graphs to classify sets

A-computable graphs act like computable graphs	A-computable graphs act like highly computable graphs
$A = K$	$A = \emptyset$
All non-computable c.e. sets	Some non-computable non-c.e. sets
Some non-computable non-c.e. sets	A is low for graph neighborhood: every A-computable graph is highly computable.

Using graphs to classify sets

A-computable graphs act like computable graphs	A-computable graphs act like highly computable graphs
$A = K$	$A = \emptyset$
All non-computable c.e. sets	Some non-computable non-c.e. sets
Some non-computable non-c.e. sets	A is low for graph neighborhood: every A-computable graph is highly computable.

Using graphs to classify sets

A-computable graphs act like computable graphs	A-computable graphs act like highly computable graphs
$A = K$	$A = \emptyset$
All non-computable c.e. sets	Some non-computable non-c.e. sets
Some non-computable non-c.e. sets	A is <u>low for graph neighborhood</u>: every A-computable graph is highly computable.

Classification of sets low for graph neighborhood

Theorem (JLM)

Let $A \leq_T K$ be a non-computable set. The following are equivalent.

1. *A is low for graph neighborhood.*
2. *Every c.e. set $B \leq_T A$ is computable.*
3. *Every A-computable graph with finite chromatic number has finite computable chromatic number.*
4. *Every A-computable graph with an Euler path has a computable Euler path.*

Thanks

Slides and paper:

`math.oscarlevin.com/research.php`