Graph Labelings and Computability Theory

Oscar Levin

University of Northern Colorado

Joint Mathematics Meeting

Atlanta, GA

January 7, 2017

Joint work with Taylor McMillan
Classical graph theory:

*Which graphs have which features?*

Computable graph theory:

*Which computable graphs have which computable features*

Often:

*Is there a computable graph with this feature, but no computable such feature.*
Some examples

- There is a computable graph with chromatic number 3 but no computable 3-coloring (Bean).

- There is a computable graph with an Euler path but no computable Euler path (Bean).

- There is a computable graph with domatic number 3 but no computable domatic 3-partition (Jura-L-Markkanen).

- There is a computable planar graph with no computable planar embedding (L-McMillan).
Given a graph $G = (V, E)$ define a function $f : V \rightarrow \mathbb{N}$ given some constraints.

**Example**

If $uv \in E$, then $f(u) \neq f(v)$.

Proper vertex coloring.

**Example**

If $uv \in E$ then $\gcd(f(u), f(v)) = 1$.

Prime labeling.

**Example**

If $uv \in E$ and $wx \in E$ then $|f(u) - f(v)| \neq |f(w) - f(x)|$.

Graceful labeling.
Labelings for Finite Graphs

Given a graph $G = (V, E)$:

A graceful labeling is an injection $f : V \rightarrow \{0, 1, \ldots, |E|\}$ such that the edge labels given by $|f(v_i) - f(v_j)|$ for $\{v_i, v_j\} \in E$ are distinct over all edges in $G$. 

![Diagram of graceful labeling example]

![Diagram of non-graceful labeling example]
Consider (countably) infinite graphs \((V = \mathbb{N})\).

Graceful: \(f : V \rightarrow \mathbb{N}\) must be a bijection. The induced edge labels must define a bijection between \(E\) and \(\mathbb{Z}^+\) (usually).
**Conjecture (Kotzig, Ringel and Rosa)**

*Every (finite) tree has a graceful labeling.*

**Theorem (Slater (1985); Chan, Cheung, Ng (2009))**

*Every infinite tree has a graceful labeling unless it has more than one but finitely many infinite degree vertices and does not contain an infinite path or once-subdivided infinite star.*

Proof: use the infinite path (or other suitable feature) to ensure surjectivity, use large labels for other vertices.
Theorem

Every infinite, locally finite graph has a graceful labeling.

Need:
1. Each natural number assigned to a vertex.
2. Each positive integer assigned to an edge.
3. No repeats.
**Proposition**

*Every computable locally finite graph has a computable graceful labeling.*

**Conjecture**

*Every computable graph with a graceful labeling has a computable graceful labeling.*
Other Questions

Basic question: Does every computable graph with a ____ labeling have a computable ____ labeling?

If no: replace “computable” with “highly computable.”

If yes: replace “computable” with “computably enumerable.”

Or: is the labeling uniformly computable?
Lots of questions

A few more types of labelings:

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge-graceful</td>
<td>Felicitous</td>
<td>Radio</td>
</tr>
<tr>
<td>Odd-graceful</td>
<td>Odd-harmonious</td>
<td>Line-graceful</td>
</tr>
<tr>
<td>$k$-graceful</td>
<td>Even-harmonious</td>
<td>$k$-sequential</td>
</tr>
<tr>
<td>Skolem-graceful</td>
<td>Vicious</td>
<td>Product cordial</td>
</tr>
<tr>
<td>Cordial</td>
<td>Magic</td>
<td>Edge product</td>
</tr>
<tr>
<td>$k$-equitable</td>
<td>Super edge-magic</td>
<td>cordial</td>
</tr>
<tr>
<td>Hamming-graceful</td>
<td>Vertex-magic</td>
<td>Difference cordial</td>
</tr>
<tr>
<td>Harmonious</td>
<td>$H$-magic</td>
<td>Prime cordial</td>
</tr>
<tr>
<td>Sequential</td>
<td>Distance magic</td>
<td>Geometric</td>
</tr>
<tr>
<td>$c$-harmonious</td>
<td>Antimagic</td>
<td>Mean</td>
</tr>
<tr>
<td>Strongly</td>
<td>Face antimagic</td>
<td>Irregular total</td>
</tr>
<tr>
<td>$c$-harmonious</td>
<td>Product antimagic</td>
<td>Set graceful</td>
</tr>
<tr>
<td>($k, d$)-arithmetic</td>
<td>Sum graph</td>
<td>Set sequential</td>
</tr>
<tr>
<td>($k, d$)-indexable</td>
<td>Prime</td>
<td>Square sum</td>
</tr>
<tr>
<td>Elegant</td>
<td>Vertex-prime</td>
<td>Square difference</td>
</tr>
</tbody>
</table>
Definition

A graph is a set $E$ of 2-element subsets of $\mathbb{N}$.
A computable graph is a computable set $E$ of 2-elements subsets of $\mathbb{N}$.
A c.e. graph is a c.e. set $E$ of 2-element subsets of $\mathbb{N}$.

A computable presentations of a particular graph is a computable graph isomorphic (as a graph) to that particular graph.

Definition

A computable (or c.e.) graph is graceful if $f : E \rightarrow \mathbb{Z}^+$ given by $f(\{x, y\}) = |x - y|$ is a bijection.
<table>
<thead>
<tr>
<th>Proposition</th>
<th>Every locally finite computable graph is <em>computably</em> isomorphic to a computable graceful graph.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conjecture</td>
<td>Every computable graph isomorphic to a graceful graph is <em>computably isomorphic</em> to a computable graceful graph.</td>
</tr>
</tbody>
</table>

However...
Proposition

There is a computable sequence $E_0, E_1, \ldots$ of computable graphs, each isomorphic to a graceful graph, but no computable function $\varphi_e$ is an isomorphism sending all graphs in the sequence to any graceful copy.

Idea: build each $E_i$ to be a graph consisting of an infinite star and a finite star, connected by an edge:
Decision problem

How hard is it to decide whether a particular computable graph is graceful?

Proposition

Deciding whether a computable graph is graceful is $\Pi^0_2$-complete

Idea: Build a sequence of graphs $E_0, E_1, \ldots$ such that $E_i$ is graceful iff $i \notin \text{INF}$.

Each graph will be a pair of infinite stars with either a finite or an infinite path.
There are c.e. graphs that are not computable.

In fact, there are locally finite c.e. graphs with no computable presentations.

**Proposition**

*Every graceful c.e. graph is computable.*

**Corollary**

*There is a c.e. graph isomorphic to a graceful graph but not isomorphic to any graceful c.e. graph.*
Definition
A computable graph is computably categorical provided any two computable isomorphic copies are *computably* isomorphic.

In general, it is difficult to compute an isomorphism between two isomorphic computable graphs.

Perhaps it is easier to build the isomorphism between two graceful computable graphs?
Conjecture

For any graceful graph, if any two graceful computable presentations of a graph are computably isomorphic, then any two computable presentations are computably isomorphic.
Question

Is there a “natural” labeling which does help compute isomorphisms?
Thanks

Slides: