Coloring Hypergraphs is Harder than Coloring Graphs

Oscar Levin

University of Northern Colorado

Joint Mathematics Meeting, Baltimore
January 17, 2019

Joint work with Conner Hatton

Computable Graphs and Coloring

An infinite graph $G = (V, E)$ is *computable* provided E is a computable set.

Computable Graphs and Coloring

An infinite graph $G = (V, E)$ is *computable* provided E is a computable set.

Coloring *connected* graphs:

- There is a computable 3-colorable graph with no computable 3-coloring.

Computable Graphs and Coloring

An infinite graph $G = (V, E)$ is *computable* provided E is a computable set.

Coloring *connected* graphs:

- There is a computable 3-colorable graph with no computable 3-coloring.
- Every computable 2-colorable graph has a computable 2-coloring.

Hypergraphs and coloring

Hypergraphs generalize graphs in that edges can contain more than two vertices.

Hypergraphs and coloring

Hypergraphs generalize graphs in that edges can contain more than two vertices.

We consider infinite *3-uniform* hypergraphs: each hyperedge contains exactly three vertices.

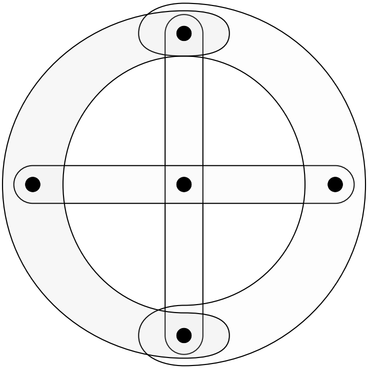
Hypergraphs and coloring

Hypergraphs generalize graphs in that edges can contain more than two vertices.

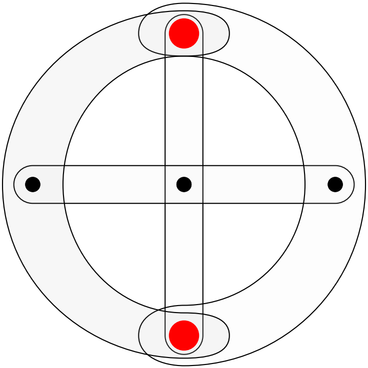
We consider infinite *3-uniform* hypergraphs: each hyperedge contains exactly three vertices.

A *(weak) k-coloring* of a hypergraph is an assignment of k colors to vertices such that no hyperedge is monochromatic.

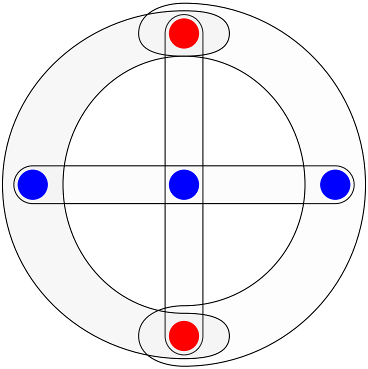
For example . . .



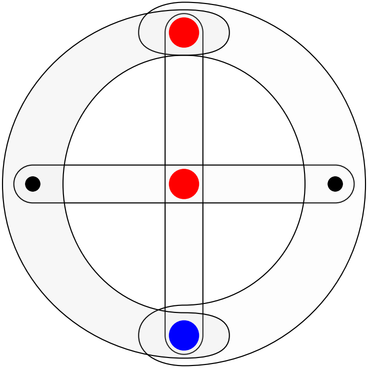
For example . . .



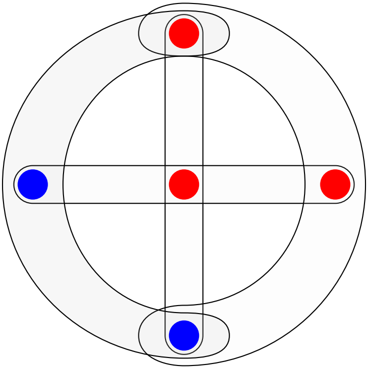
For example . . .



For example . . .



For example . . .



Computable Hypergraphs and Coloring

A hypergraph is computable if the set of hyperedges is computable. So there is an algorithm that correctly answers the question:

are these three vertices a hyperedge?

Computable Hypergraphs and Coloring

A hypergraph is computable if the set of hyperedges is computable. So there is an algorithm that correctly answers the question:

are these three vertices a hyperedge?

Theorem

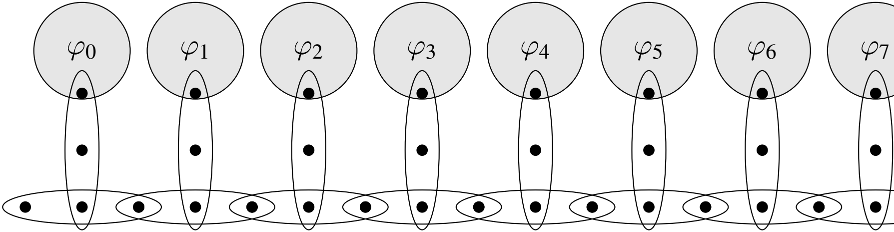
There exists a connected, computable 2-colorable hypergraph with no computable 2-coloring.

Construction

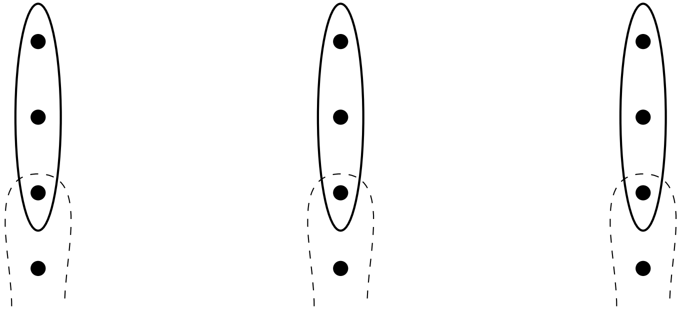
We build a computable hypergraph for which every potential 2-coloring is wrong.

Construction

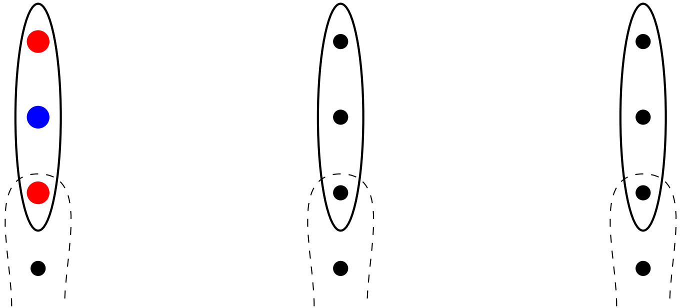
We build a computable hypergraph for which every potential 2-coloring is wrong.



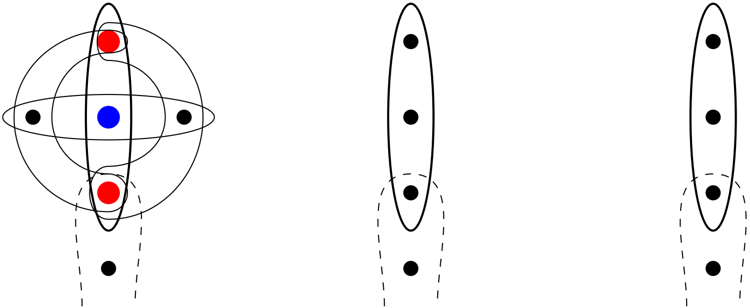
The gadgets



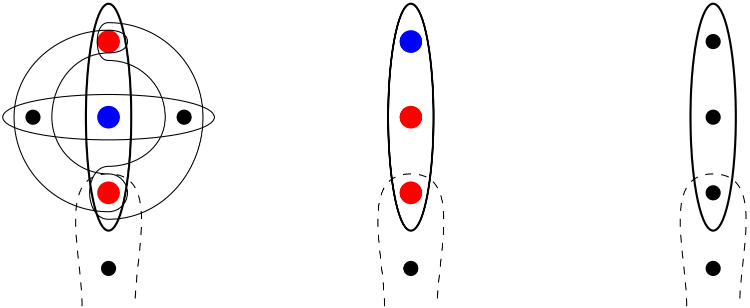
The gadgets



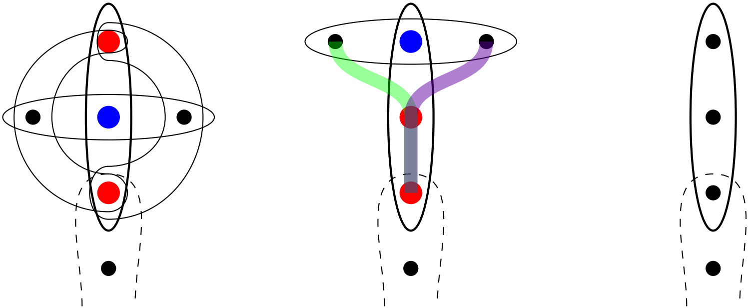
The gadgets



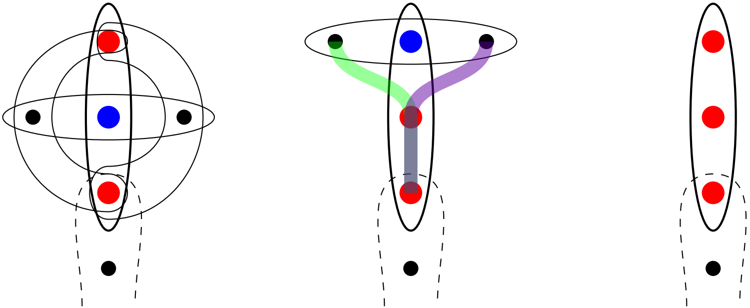
The gadgets



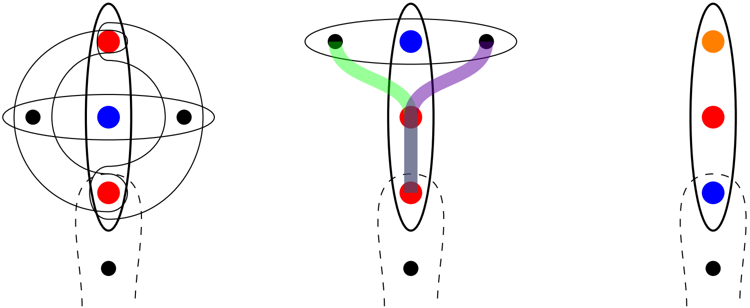
The gadgets



The gadgets



The gadgets

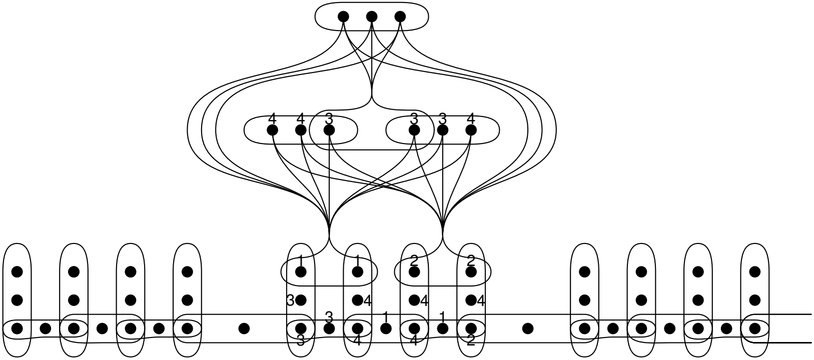


Increasing the computable chromatic number

There is a connected, computable 2-colorable hypergraphs with no computable 4 coloring.

Increasing the computable chromatic number

There is a connected, computable 2-colorable hypergraphs with no computable 4 coloring.



If we know more about the hypergraph . . .

A hypergraph is *highly computable* provided it is computable and the degree function is computable.

If we know more about the hypergraph . . .

A hypergraph is *highly computable* provided it is computable and the degree function is computable.

This means that given a vertex, we can effectively get a list of all hyperedges it belongs to.

If we know more about the hypergraph . . .

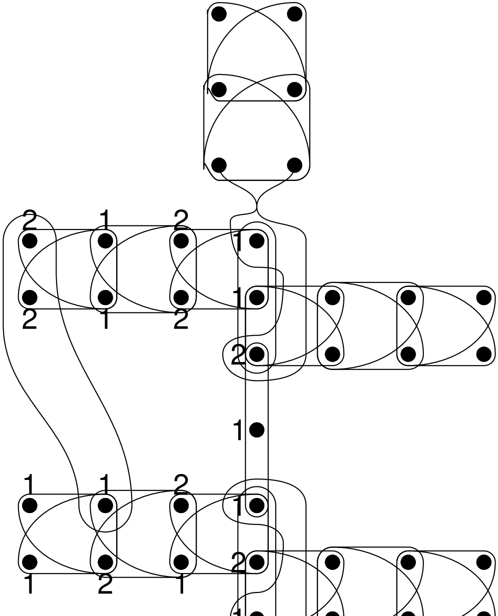
A hypergraph is *highly computable* provided it is computable and the degree function is computable.

This means that given a vertex, we can effectively get a list of all hyperedges it belongs to.

Theorem

There is a connected, highly computable 2-colorable hypergraph with no computable 2-coloring.

The idea of a construction



Where is the complexity?

Consider not-necessarily connected graphs:

For 2-colorable graphs, you can compute a 2-coloring from the relation that says whether a pair of vertices is in the same component.

Where is the complexity?

Consider not-necessarily connected graphs:

For 2-colorable graphs, you can compute a 2-coloring from the relation that says whether a pair of vertices is in the same component.

For 2-colorable hypergraphs, even with this relation, you cannot compute a 2-coloring.

Where is the complexity?

Consider not-necessarily connected graphs:

For 2-colorable graphs, you can compute a 2-coloring from the relation that says whether a pair of vertices is in the same component.

For 2-colorable hypergraphs, even with this relation, you cannot compute a 2-coloring.

What should we color v_{207}? We don't know whether we have freedom to decide for ourselves!

Thanks!

Slides by email:

oscar.levin@unco.edu