OCSA: Research in reverse mathematics

Jeff Hirst
Appalachian State University
Boone, NC USA

with many collaborators

January 25, 2019

Mathematical Sciences Colloquium
Appalachian State University

OCSA activities

Invited talks

Leaf management, September 2018
Dagstuhl Seminar 18361
Leibniz-Zentrum für Informatik.

Hindman's theorem and ultrafilters, July 2018
RaTLoCC 2018
Bertinoro International Center for Informatics.

A weak coloring principle, July 2018
Workshop on Ramsey Theory and Computability
Rome Global Gateway of Notre Dame University.

Papers

Combinatorial principles equivalent to weak induction, with C. Davis, D. Hirschfeldt, J. Pardo, A. Pauly, and K. Yokoyama, submitted. [1]

Leaf management, submitted. [5]

Using Ramsey's theorem once, with C. Mummert, resubmitted. [7]

Reverse mathematics and colorings of hypergraphs with C. Davis, J. Pardo, and T. Ransom, Archive for Mathematical Logic. [2]

International contacts support student research

During his senior year, alumnus Noah Hughes gave a talk on his senior honors thesis in the logic seminar at the University of Ghent. Paul Shafer was our contact in Ghent.

Contact bait: An example of student work

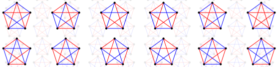
How many 2-colorings of K5 have no 1-colored K3 ?

Ramsey Interest Group: Anthony Hengst, Sergei Miles, Isaac Medina Silva, Allison Staley **Faculty Mentor: Jeff Hirst**

Appalachian State University, Department of Mathematical Sciences, Boone, North Carolina 28608

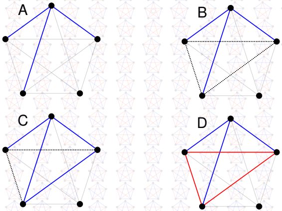
Introduction

Of the 1024 possible 2-colorings of K5, only 12 have no 1-colored triangles.



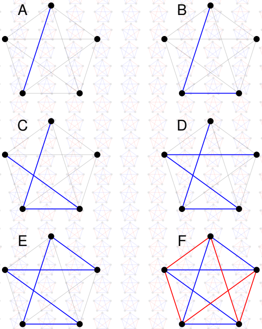
Claim 1

If any 3 edges match, then there is a 1-colored triangle.



Claim 2

If G has no 1-colored triangles, then G has a 1-colored 5-cycle.

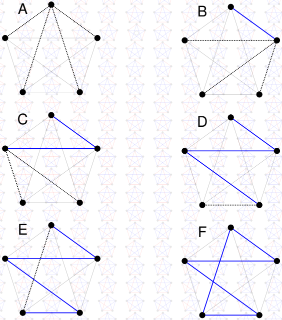


E: 1-colored 5-cycle

F: Remaining edges form a 5-cycle

Claim 3

There are 12 ways to construct a 1-colored 5-cycle.



$$\frac{4 \cdot 3 \cdot 2 \cdot 1 \cdot 1}{2} = 12$$

Making contacts, groundwork

International workshops provide opportunities to create new contacts.

- Smaller than conferences
  - greater interaction
  - disciplinary focus

- More international participants

- Travel tips

Organization of the workshop may or may not be international.

Workshop example 1: Rome

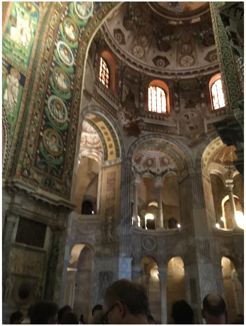
Workshop on Ramsey Theory and Computability
Rome Global Gateway of Notre Dame University
July 9-13, 2018

Participants from:

- Leeds University
- University of Bern
- Central South University of China
- Dartmouth College
- Japan Advanced Institute of Science and Technology
- Università di Roma Sapienza
- Appalachian State
- Cornell University
- Università di Pisa
- National University of Singapore
- University of Vienna
- Swansea University
- University of Pennsylvania
- Università degli Studi di Udine

Workshop example 2: Bertinoro, Italy

RaTLoCC18:
Ramsey Theory in Logic, Combinatorics, and Complexity
Bertinoro International Center for Informatics
July 15-20, 2018
37 participants from Spain, Germany, USA, England, Greece, Czech Republic, Russia, Poland, Italy, Austria, France, and Canada



Basilica of San Vitale in Ravenna

Workshop example 3: Wadern, Germany

Dagstuhl Seminar 18361:
Measuring the Complexity of Computational Content:
From Combinatorial Problems to Analysis
Leibniz-Zentrum für Informatik
September 2-7, 2018
43 participants from Spain, France, USA, Germany, Austria, England, Japan, New Zealand, Italy, Singapore, Chile, and Russia


Faculty from Appalachian can pursue funding from multiple sources:

- Office of International Education and Development
- Board of Trustees International Research Grants
- Student And Faculty Excellence (SAFE) Fund, College of Arts and Sciences

Part II: Reverse mathematics

Reverse mathematics uses a hierarchy of axioms of second order arithmetic to measure the strength of theorems.

The language has variables for natural numbers and sets of naturals numbers.

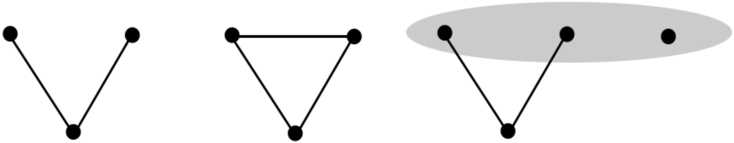
The base system, RCA_0, includes

- arithmetic facts (e.g. $n + 0 = n$),
- an induction scheme (restricted to Σ^0_1 formulas), and
- recursive comprehension
  (computable sets exist, i.e. sets with programmable characteristic functions exist).

Adding stronger comprehension axioms creates stronger axiom systems.

Proper colorings of hypergraphs

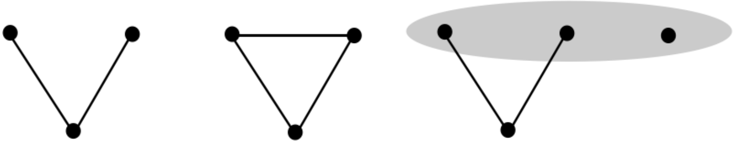
A *hypergraph* consists of vertices and edges. Edges may contain any number of vertices.



Proper colorings of hypergraphs

A *hypergraph* consists of vertices and edges. Edges may contain any number of vertices.

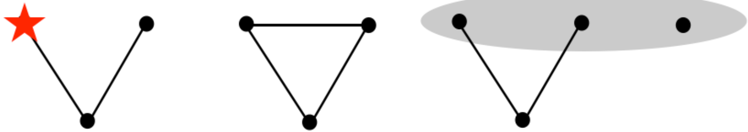
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

A *hypergraph* consists of vertices and edges. Edges may contain any number of vertices.

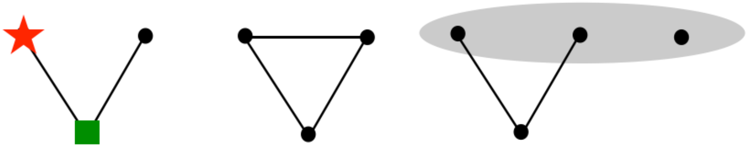
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

A *hypergraph* consists of vertices and edges. Edges may contain any number of vertices.

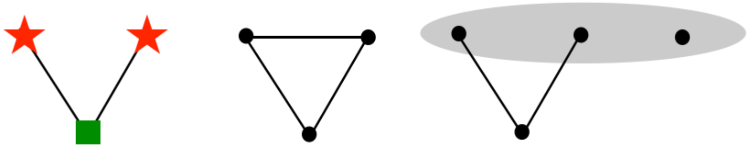
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

A *hypergraph* consists of vertices and edges. Edges may contain any number of vertices.

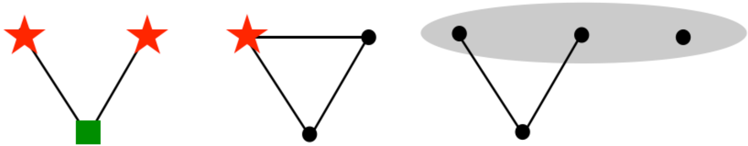
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

A *hypergraph* consists of vertices and edges. Edges may contain any number of vertices.

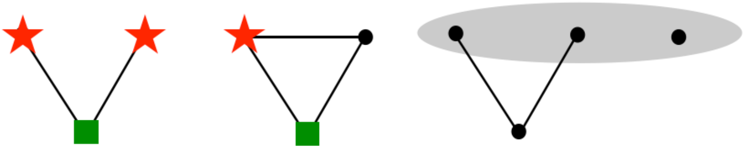
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

A *hypergraph* consists of vertices and edges. Edges may contain any number of vertices.

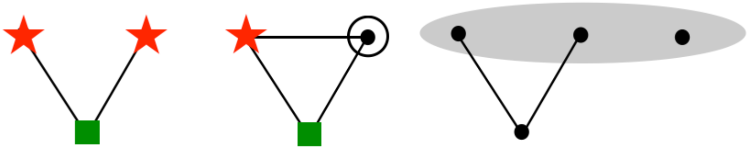
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

A *hypergraph* consists of vertices and edges. Edges may contain any number of vertices.

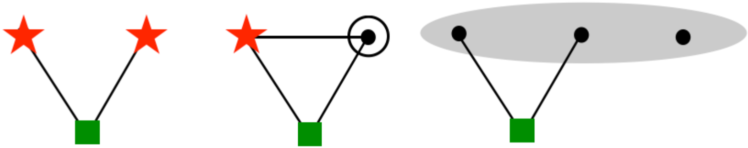
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

A *hypergraph* consists of vertices and edges. Edges may contain any number of vertices.

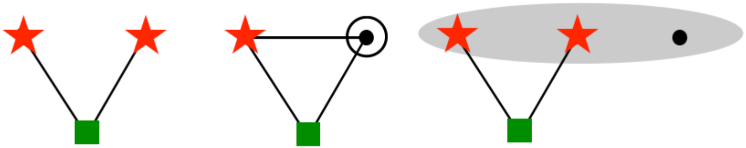
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

A *hypergraph* consists of vertices and edges. Edges may contain any number of vertices.

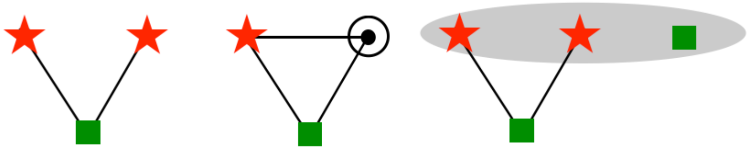
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

A *hypergraph* consists of vertices and edges. Edges may contain any number of vertices.

A 2-coloring is *proper* if no edge is monochromatic.



Hypergraphs with finite edges

The system ACA_0 adds arithmetical comprehension to RCA_0 (sets with arithmetically definable characteristic functions exist).

A theorem of reverse mathematics:

Theorem: Over RCA_0, the following are provably equivalent:

1. ACA_0.
2. Every injection has a range. (Friedman [4], Simpson [8]).
3. Suppose H is a hypergraph with finite edges presented as a sequence of characteristic functions. If every finite partial hypergraph of H has a proper 2-coloring, then H has a proper 2-coloring.

Hypergraphs with finite edges: Additional observations

Hypergraphs are different from graphs.

Theorem: RCA_0 proves the following are equivalent:

(1) ACA_0.

(2) Suppose H is a hypergraph with finite edges presented as a sequence of characteristic functions. If every finite partial hypergraph of H has a proper 2-coloring, then H has a proper 2-coloring.

Theorem: RCA_0 proves the following are equivalent:

(1) WKL_0.

(2) Suppose H is a graph with finite edges presented as a sequence of characteristic functions. If every finite partial graph of H has a proper 2-coloring, then H has a proper 2-coloring.

Hypergraphs with infinite edges

For hypergraphs with infinite edges, there is no arithmetical characterization of hypergraphs with proper 2-colorings. This is a corollary of:

Theorem: RCA_0 proves the following are equivalent:

(1) Π^1_1-CA_0, the comprehension scheme for Π^1_1 definable sets.

(2) $\widehat{\mathsf{HC}}$: If $\langle H_i \rangle_{i \in \mathbb{N}}$ is a sequence of hypergraphs, then there is a function $f : \mathbb{N} \to 2$ such that $f(i) = 1$ if and only if H_i has a proper 2-coloring.

Proof sketch for $(1) \to (2)$:
$f(i) = 0$ if and only if every 2-coloring fails to be proper for H_i.
“Fails to be proper” means that for some j, all the vertices of edge E_j of H_i match.

Hypergraphs with infinite edges: the reversal

For the reversal, we need a combinatorial version of Π^1_1-CA_0.

Theorem: RCA_0 proves the following are equivalent:

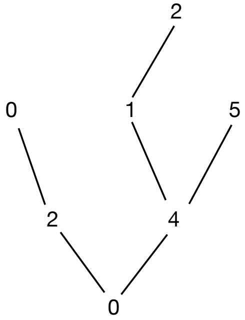
(1) Π^1_1-CA_0.

(2) $\widehat{\mathsf{WF}}$: If $\langle T_i \rangle_{i\in\mathbb{N}}$ is a sequence of trees with integer labeled nodes, then there is a function $f : \mathbb{N} \to 2$ such that $f(i) = 1$ if and only if T_i is well founded. (Lemma IV.1.1, Simpson [8])

(3) $\widehat{\mathsf{WF}}_L$: If $\langle T_i, L_i \rangle_{i\in\mathbb{N}}$ is a sequence of trees, each equipped with a leaf set L_i, then there is a function $f : \mathbb{N} \to 2$ such that $f(i) = 1$ if and only if T_i is well founded.

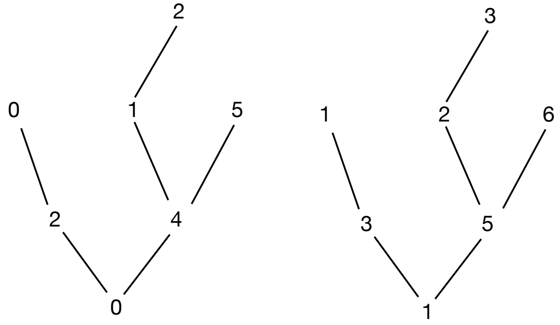
Leaf management

A tree can be converted to a tree with a leaf set by adding an extension with a new label to every existing nodes. The converted tree has the same infinite paths (and the same perfect subtrees).



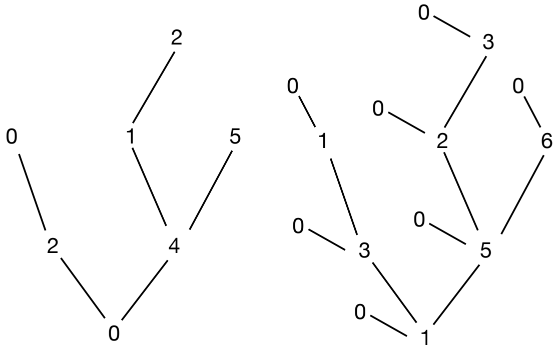
Leaf management

A tree can be converted to a tree with a leaf set by adding an extension with a new label to every existing nodes. The converted tree has the same infinite paths (and the same perfect subtrees).



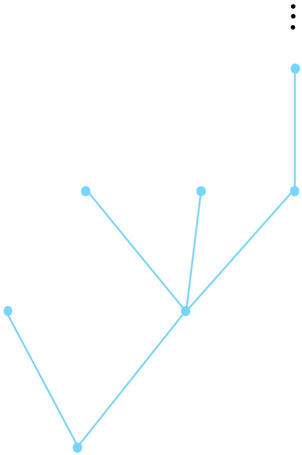
Leaf management

A tree can be converted to a tree with a leaf set by adding an extension with a new label to every existing nodes. The converted tree has the same infinite paths (and the same perfect subtrees).



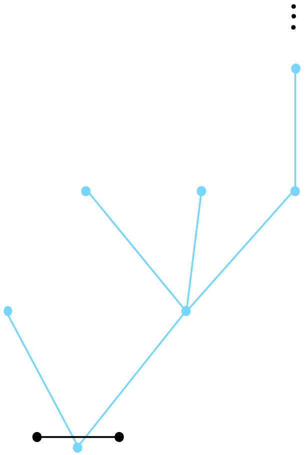
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



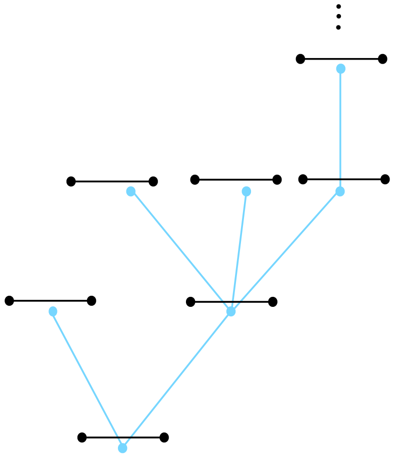
The reversal: $\widehat{HC} \to \widehat{WF}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



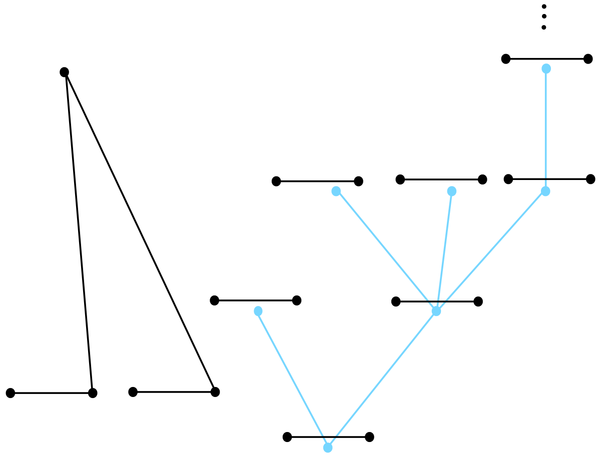
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



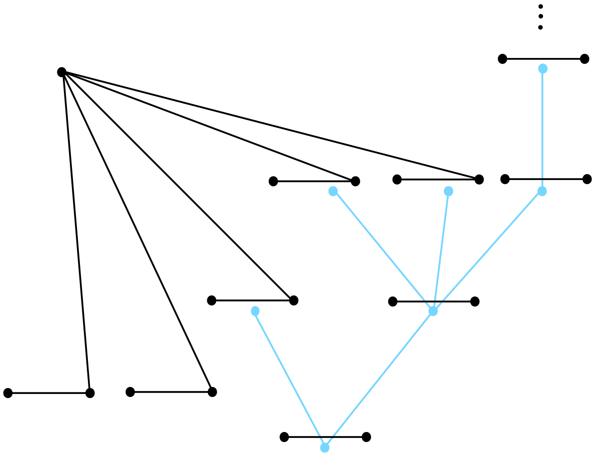
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



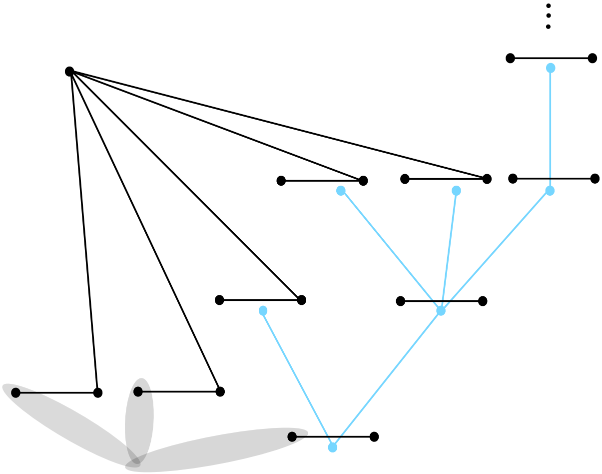
The reversal: $\widehat{HC} \to \widehat{WF}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



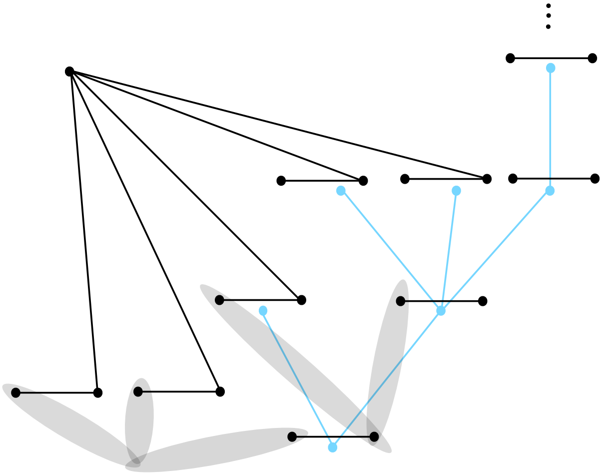
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



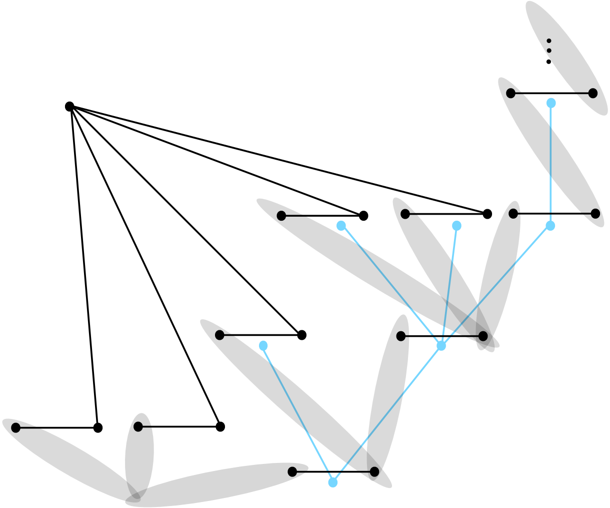
The reversal: $\widehat{HC} \to \widehat{WF}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



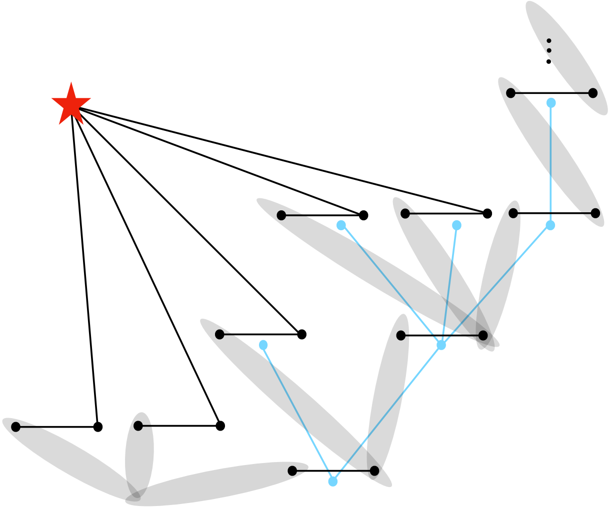
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



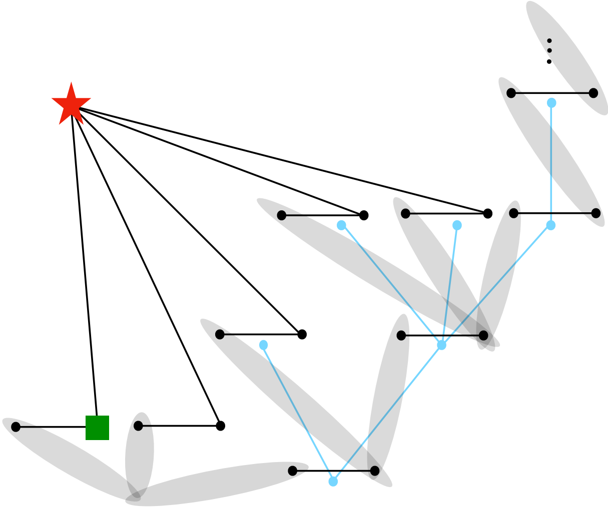
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



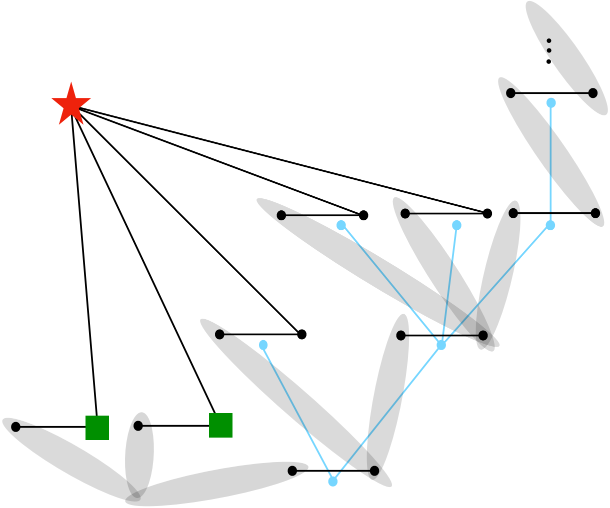
The reversal: $\widehat{\mathrm{HC}} \rightarrow \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



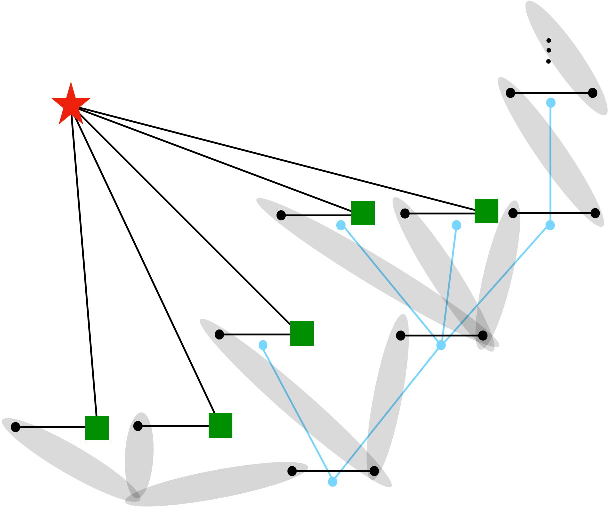
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



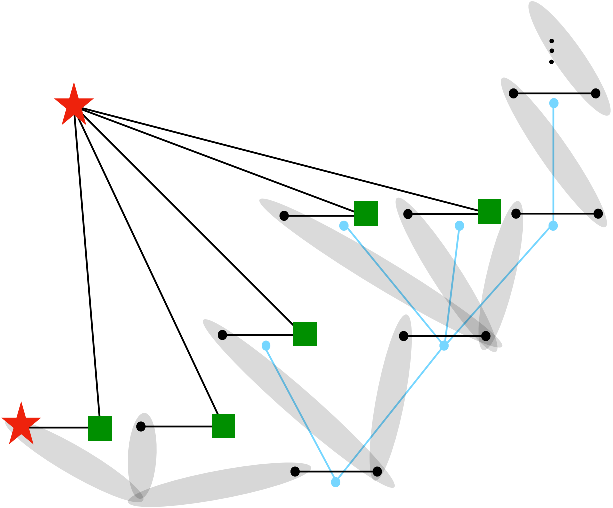
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



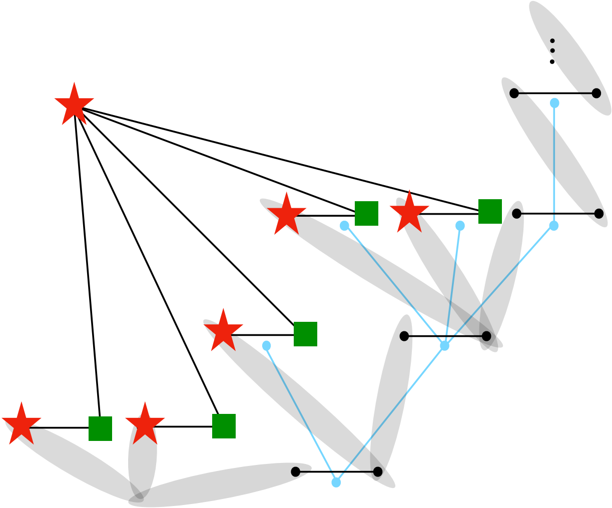
The reversal: $\widehat{\mathrm{HC}} \rightarrow \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



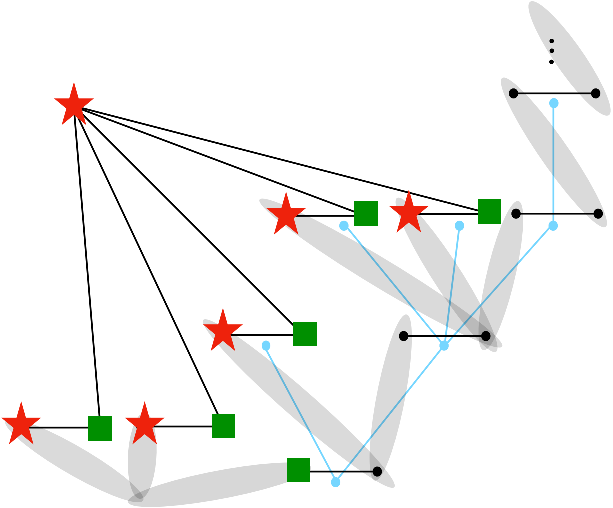
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



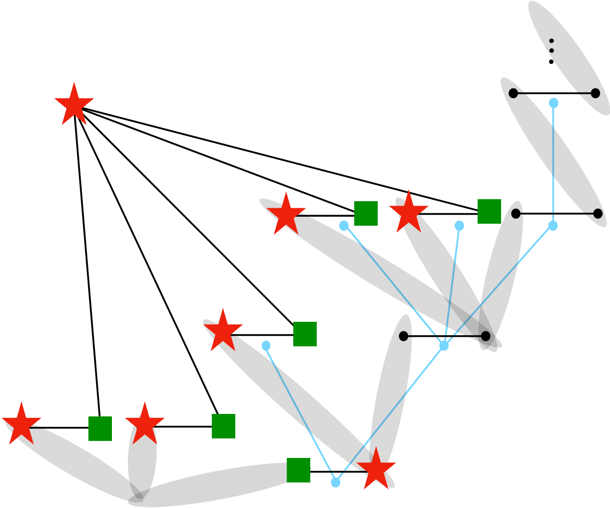
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



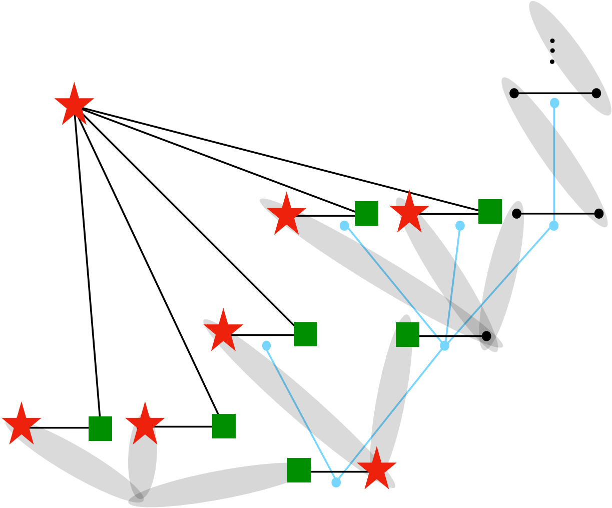
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



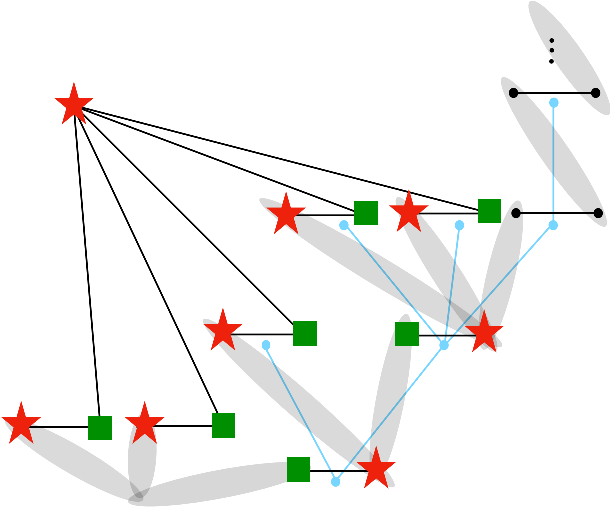
The reversal: $\widehat{\mathrm{HC}} \rightarrow \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



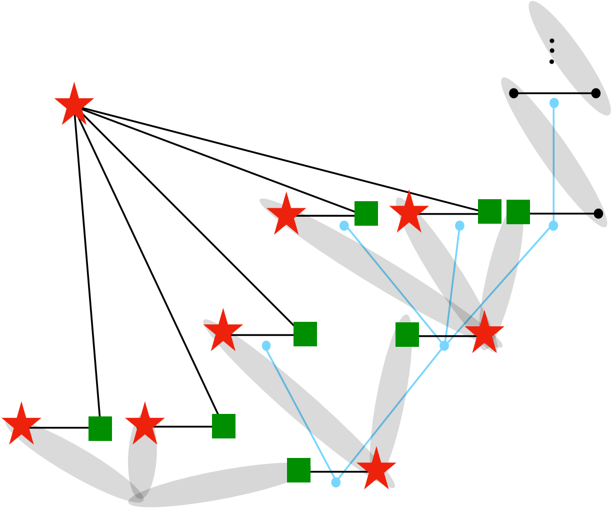
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



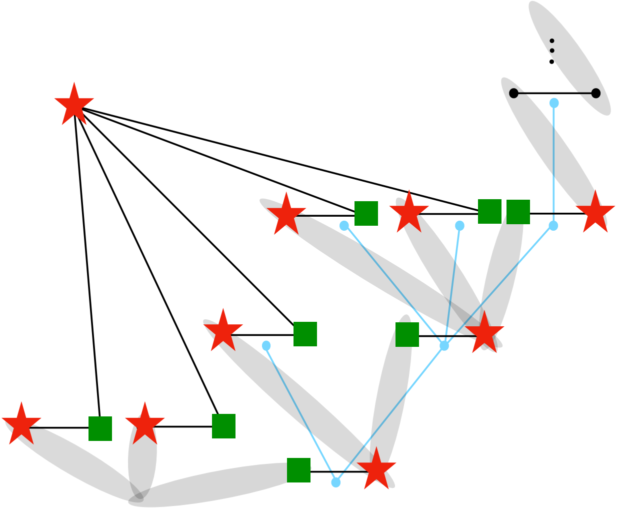
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



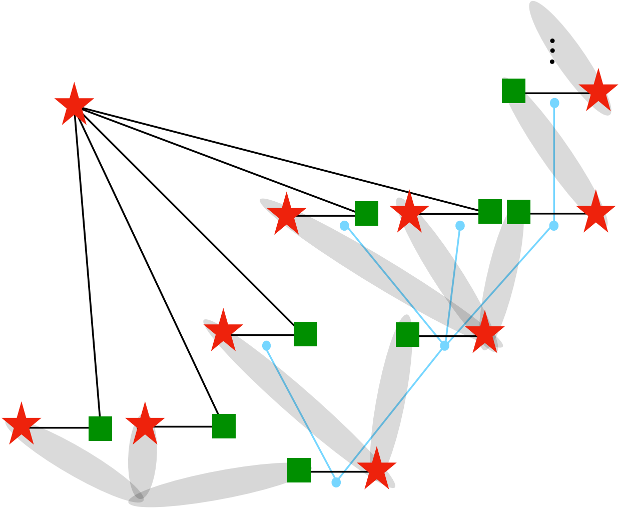
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



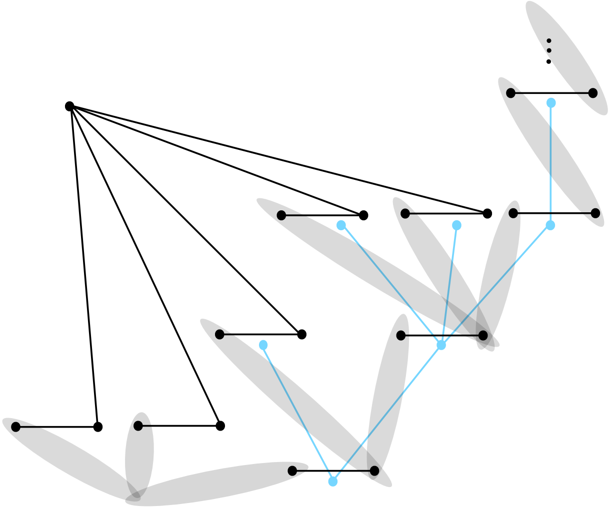
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



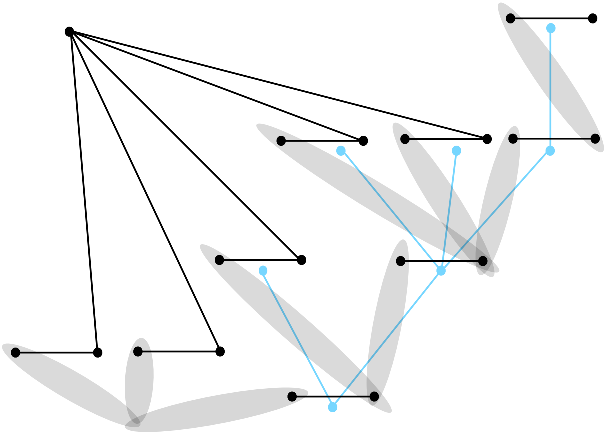
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



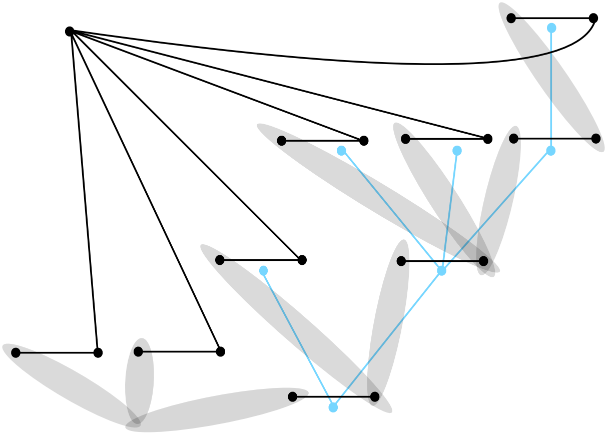
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



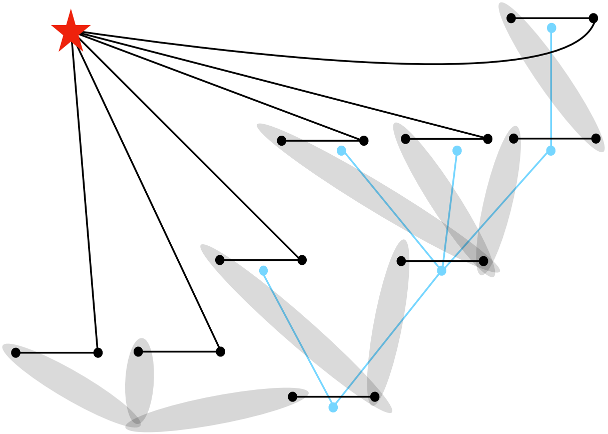
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



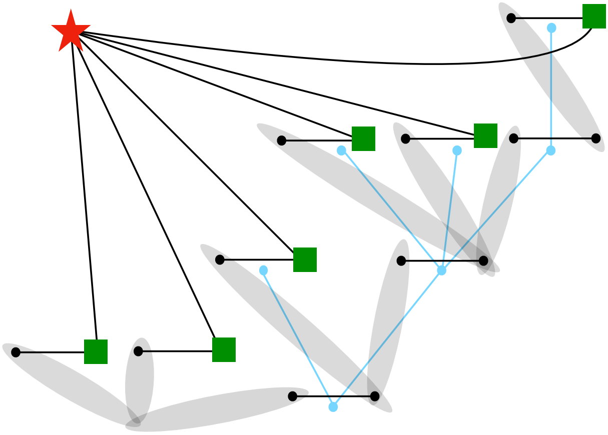
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



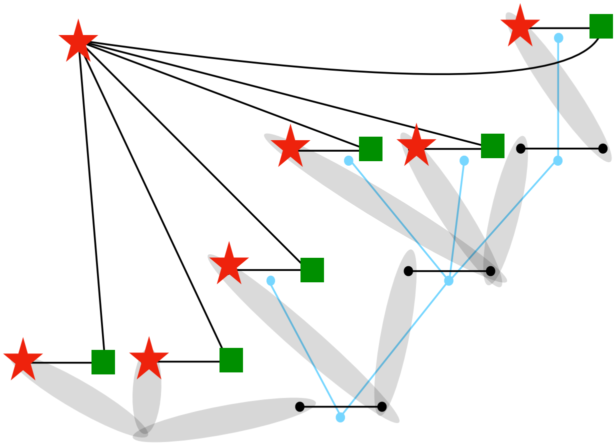
The reversal: $\widehat{\mathrm{HC}} \rightarrow \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



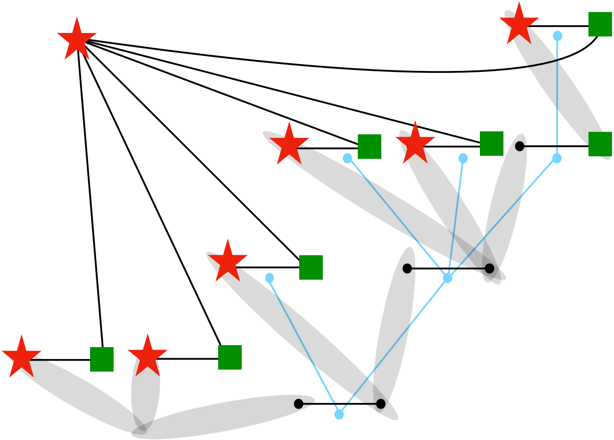
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



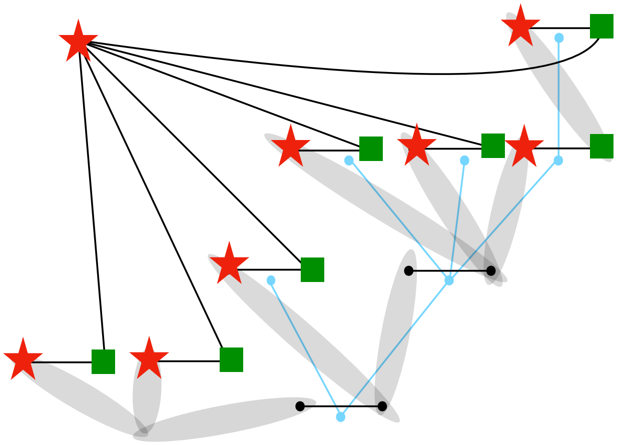
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



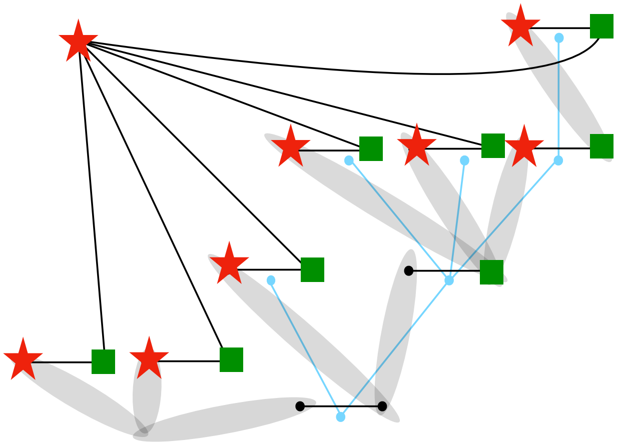
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



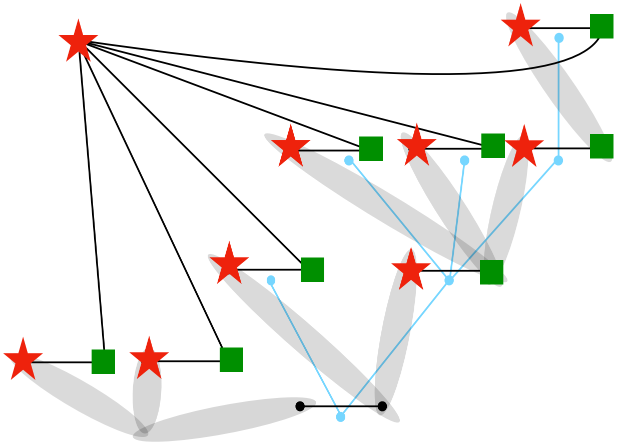
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



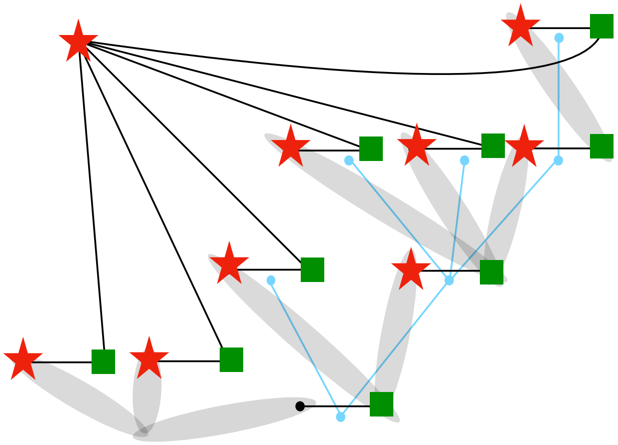
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



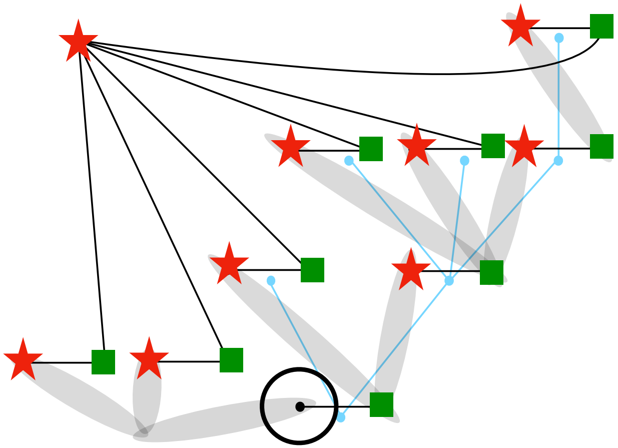
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

We want to convert a tree into a hypergraph that has a proper 2-coloring iff the tree has a path.



Weihrauch reductions

Sample problems

WF: input a tree T; output 1 iff T is well-founded.

HC: input a hypergraph H; output 1 iff H has a proper 2-coloring.

Parallelization

$\widehat{\mathrm{HC}}$: input an infinite sequence of hypergraphs; output list of indices of hypergraphs with proper 2-colorings.

Reductions

$\mathrm{P} \leqslant_{\mathrm{sW}} \mathrm{Q}$ if there are uniformly computable procedures φ and ψ such that

$$\begin{array}{ccc} \mathrm{P}_{\mathrm{input}} & \rightarrow_{\varphi} & \mathrm{Q}_{\mathrm{input}} \\ \downarrow & & \downarrow \\ \mathrm{P}_{\mathrm{output}} & \leftarrow_{\psi} & \mathrm{Q}_{\mathrm{output}} \end{array}$$

Equivalences

$\mathrm{P} \equiv_{\mathrm{sW}} \mathrm{Q}$ iff $\mathrm{P} \leqslant_{\mathrm{sW}} \mathrm{Q}$ and $\mathrm{Q} \leqslant_{\mathrm{sW}} \mathrm{P}$

Weihrauch equivalences

$$\mathsf{WF} \equiv_{sW} \mathsf{WF}_L \equiv_{sW} \mathsf{HC}$$

$$\widehat{\mathsf{WF}} \equiv_{sW} \widehat{\mathsf{WF}}_L \equiv_{sW} \widehat{\mathsf{HC}}$$

Another problem

PK: input a tree T; output the perfect kernel of T.

$$\widehat{\mathsf{WF}} \equiv_{sW} \mathsf{PK}$$

These results appear in *Leaf management* [5]

References

[1] Caleb Davis, Jeffry L. Hirst, Denis Hirschfeldt, Jake Pardo, Arno Pauly, and Keita Yokoyama, *Combinatorial principles equivalent to weak induction*. Submitted. Available at arxiv.org/abs/1812.09943.

[2] Caleb Davis, Jeffry L. Hirst, Jake Pardo, and Tim Ransom, *Reverse mathematics and colorings of hypergraphs*, Archive for Mathematical Logic (2018). DOI 10.1007/s00153-018-0654-z.

[3] Harvey Friedman, *Some systems of second order arithmetic and their use*, Proceedings of the International Congress of Mathematicians (Vancouver, B. C., 1974), Vol. 1, 1975, pp. 235–242. http://www.mathunion.org MR0429508.

[4] Harvey Friedman, *Abstracts: Systems of second order arithmetic with restricted induction, I and II*, J. Symbolic Logic **41** (1976), 557–559. http://www.jstor.org/stable/2272259.

[5] Jeffry L. Hirst, *Leaf management*. Submitted. Available at arxiv.org/abs/1812.09762.

[6] Jeffry L. Hirst and Noah A. Hughes, *Reverse mathematics and marriage problems with unique solutions*, Arch. Math. Logic **54** (2015), no. 1-2, 49–57. 10.1007/s00153-014-0401-z MR3304736.

[7] Jeffry L. Hirst and Carl Mummert, *Using Ramsey's theorem once*. Submitted. Available at arxiv.org/abs/1611.03134.

[8] Stephen G. Simpson, *Subsystems of second order arithmetic*, 2nd ed., Perspectives in Logic, Cambridge University Press, Cambridge; Association for Symbolic Logic, Poughkeepsie, NY, 2009. 10.1017/CBO9780511581007 MR2517689.