Reverse mathematics and colorings of hypergraphs

Jeff Hirst
Appalachian State University
Boone, NC USA

in collaboration with Caleb Davis, Jake Pardo
and Tim Ransom

April 5, 2019

College of Charleston

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.

ACA_0

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

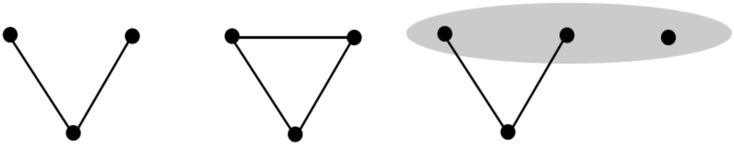
A theorem of reverse mathematics:

Theorem: Over RCA_0, the following are provably equivalent:

1. ACA_0.
2. Every injection has a range. (Lemma III.1.3, Simpson [5]).
3. Every countable sequence of reals in $[0, 1]$ has a convergent subsequence. (Friedman [3])

Proper colorings of hypergraphs

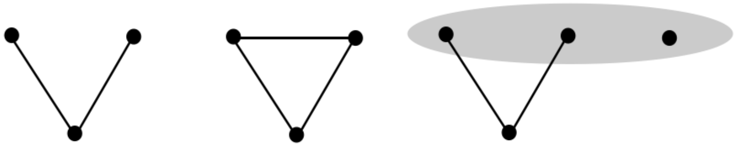
A *hypergraph* consists of vertices and edges. Edges may contain any number of vertices.



Proper colorings of hypergraphs

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

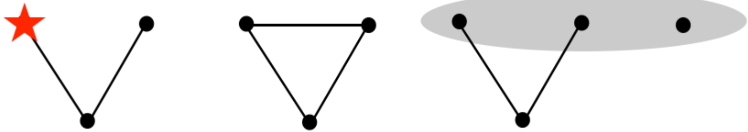
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

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

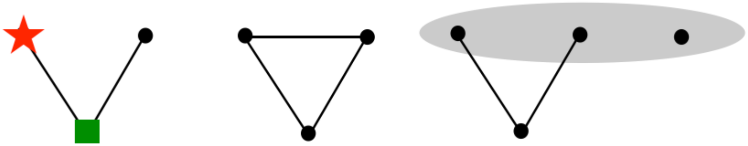
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

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

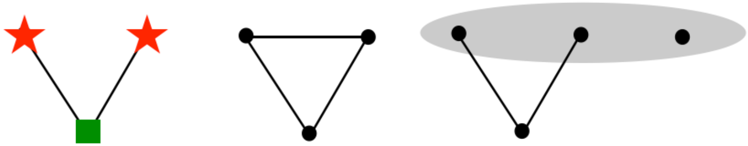
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

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

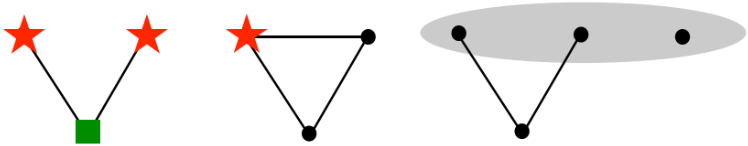
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

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

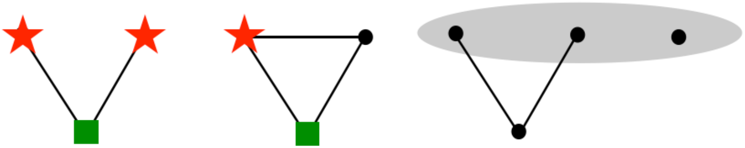
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

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

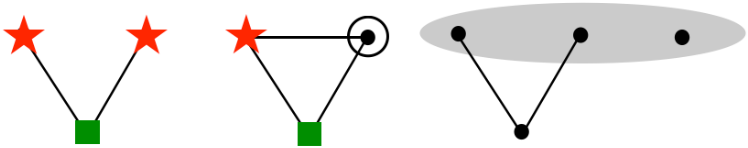
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

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

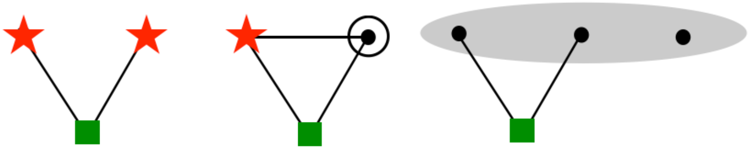
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

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

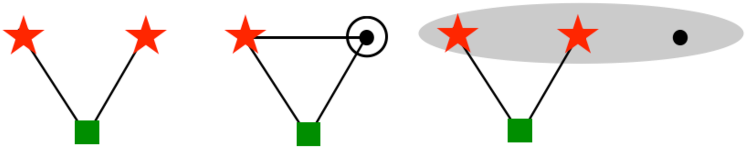
A 2-coloring is *proper* if no edge is monochromatic.



Proper colorings of hypergraphs

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

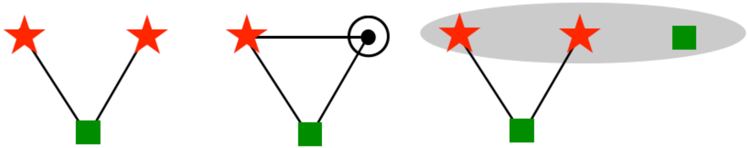
A 2-coloring is *proper* if no edge is monochromatic.



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

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.

Proof sketch:

(1)$\rightarrow$(2): For every m, there is a least 2-coloring of $v_0, \ldots, v_m$ that can be extended to a proper 2-coloring of every finite partial hypergraph. Nesting these least 2-colorings yields a 2-coloring of all of H that is arithmetically definable (in H).

The reversal: Proper 2-colorings $\to$ ACA_0

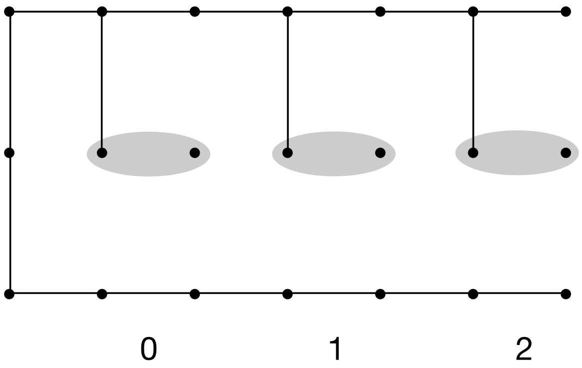
Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin \mathrm{Range}(f)$.

The reversal: Proper 2-colorings $\rightarrow$ ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

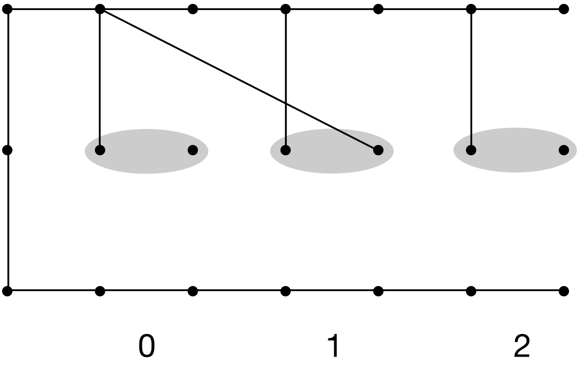
For example, suppose $f(0) = 1$, $f(2) = 0$, and 2 $\notin$Range(f).



The reversal: Proper 2-colorings → ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

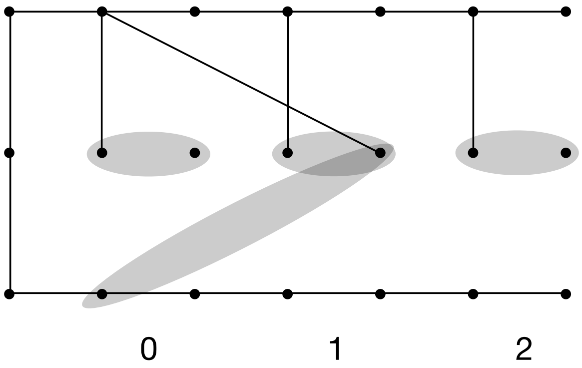
For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin \text{Range}(f)$.



The reversal: Proper 2-colorings → ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

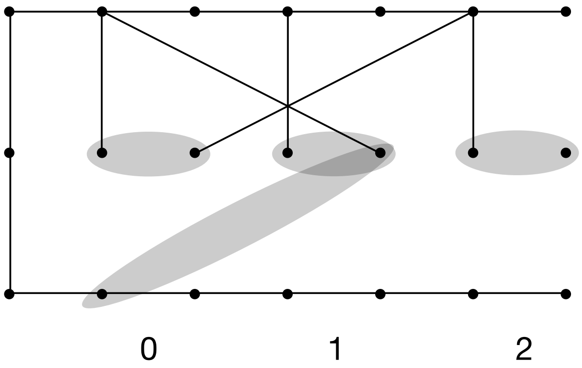
For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin \text{Range}(f)$.



The reversal: Proper 2-colorings → ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

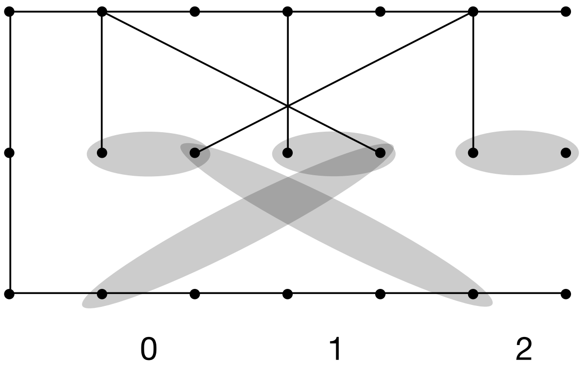
For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin \text{Range}(f)$.



The reversal: Proper 2-colorings → ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

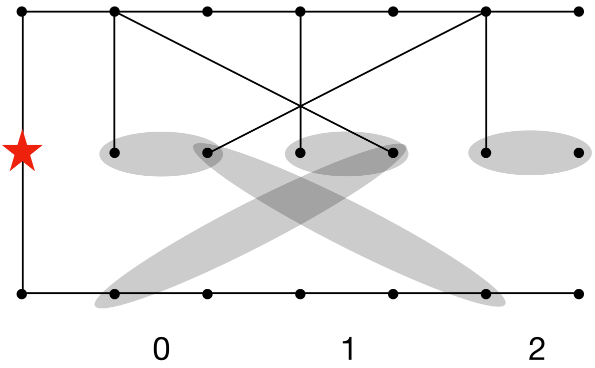
For example, suppose $f(0) = 1$, $f(2) = 0$, and 2 $\notin$Range(f).



The reversal: Proper 2-colorings → ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

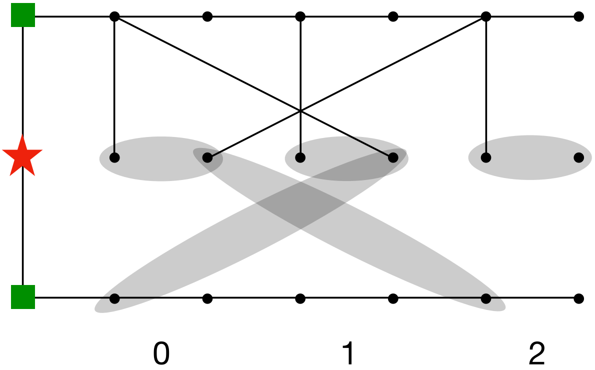
For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin \text{Range}(f)$.



The reversal: Proper 2-colorings → ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

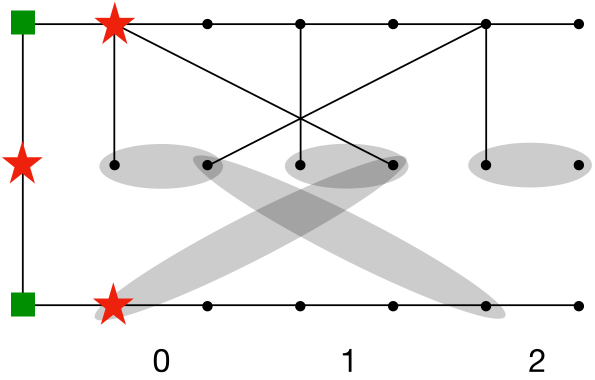
For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin \text{Range}(f)$.



The reversal: Proper 2-colorings → ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

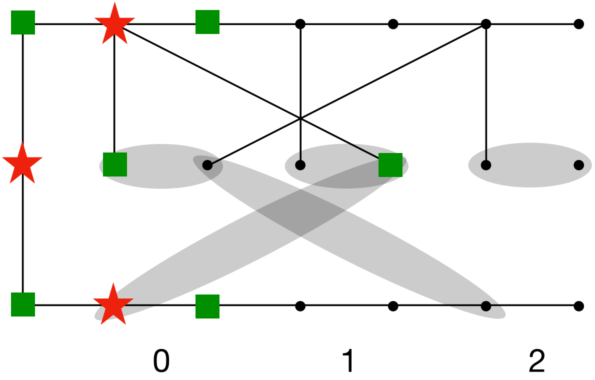
For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin$Range(f).



The reversal: Proper 2-colorings → ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

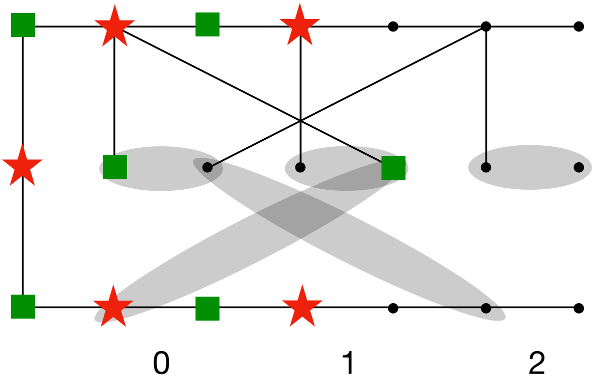
For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin \text{Range}(f)$.



The reversal: Proper 2-colorings → ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

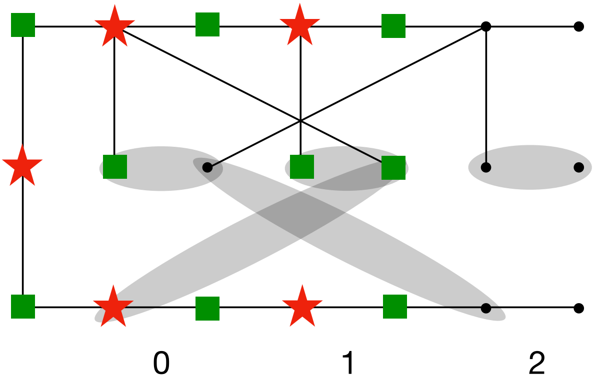
For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin \text{Range}(f)$.



The reversal: Proper 2-colorings → ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

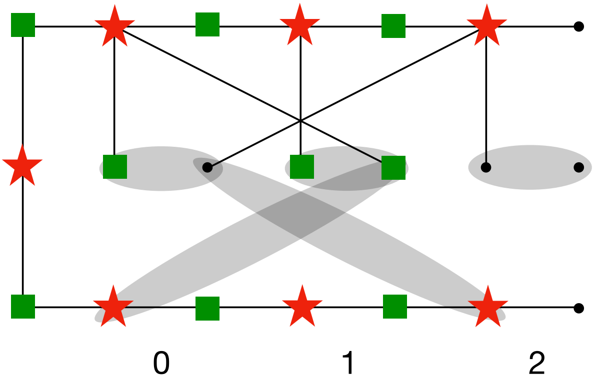
For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin$Range(f).



The reversal: Proper 2-colorings → ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

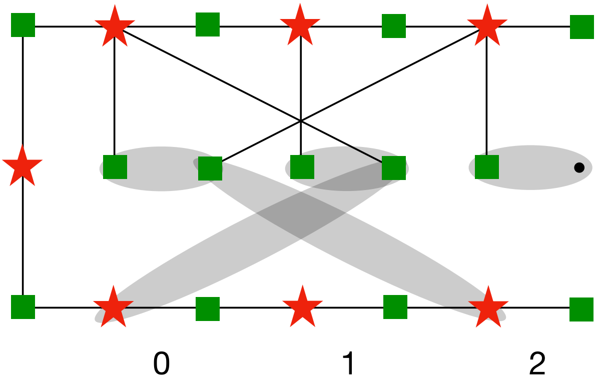
For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin \text{Range}(f)$.



The reversal: Proper 2-colorings $\rightarrow$ ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

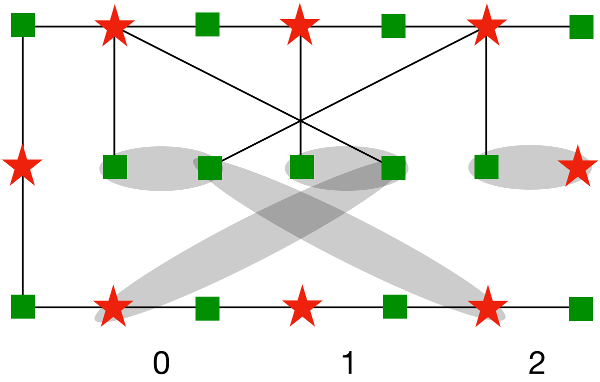
For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin \text{Range}(f)$.



The reversal: Proper 2-colorings → ACA_0

Given an injection f, we want to build H so that the range of f can be computed from any 2-coloring of H.

For example, suppose $f(0) = 1$, $f(2) = 0$, and $2 \notin \text{Range}(f)$.



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) $\widehat{\text{HC}}$: 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:

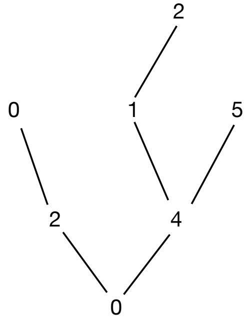
(1) Π^1_1-CA_0.

(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 [5])

(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.

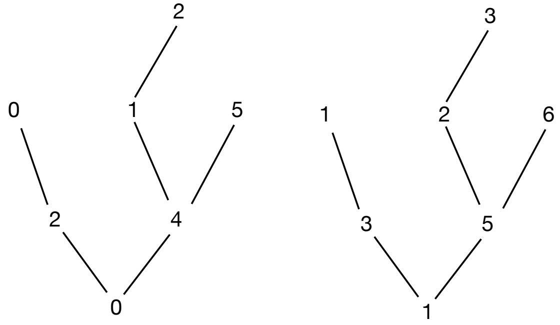
Leaf management

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).



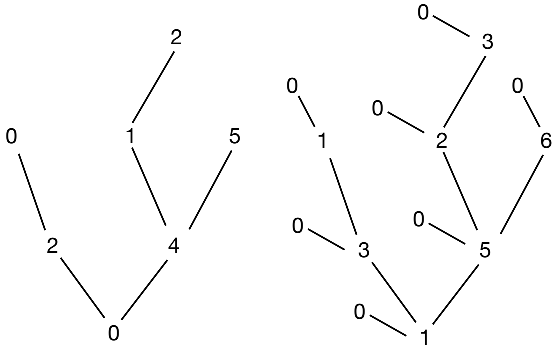
Leaf management

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).



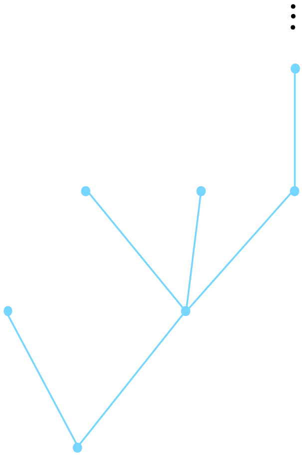
Leaf management

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).



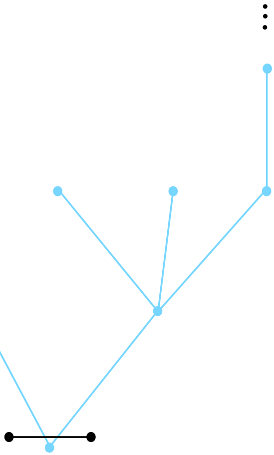
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



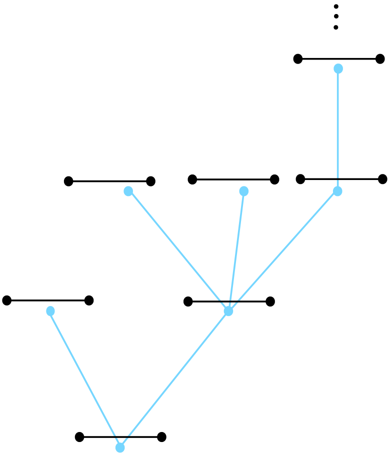
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



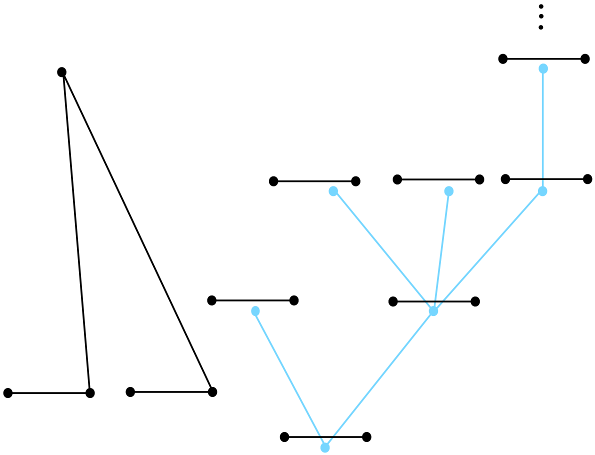
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

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



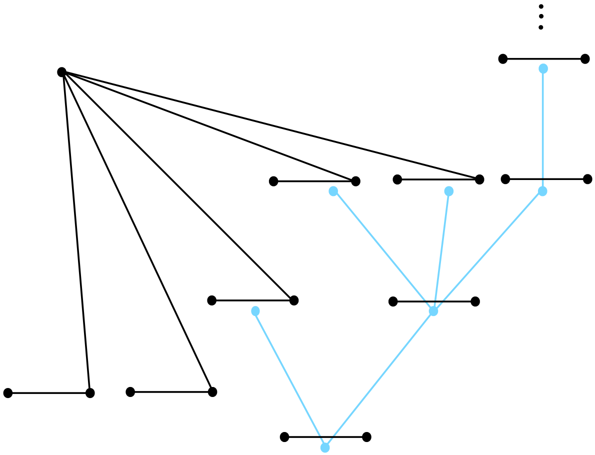
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

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



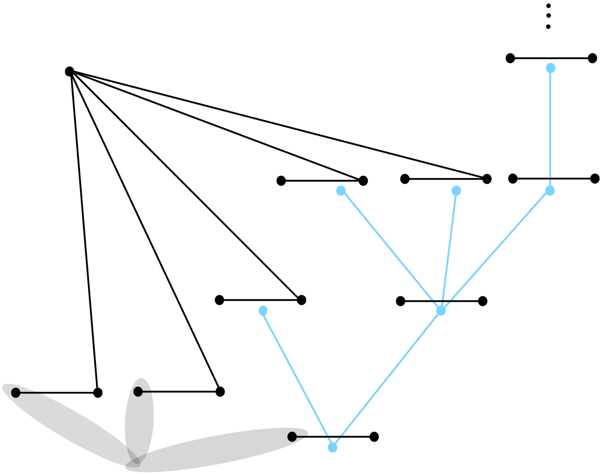
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



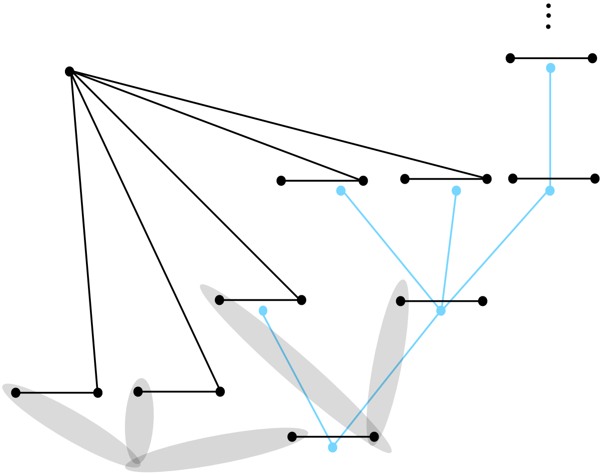
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

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



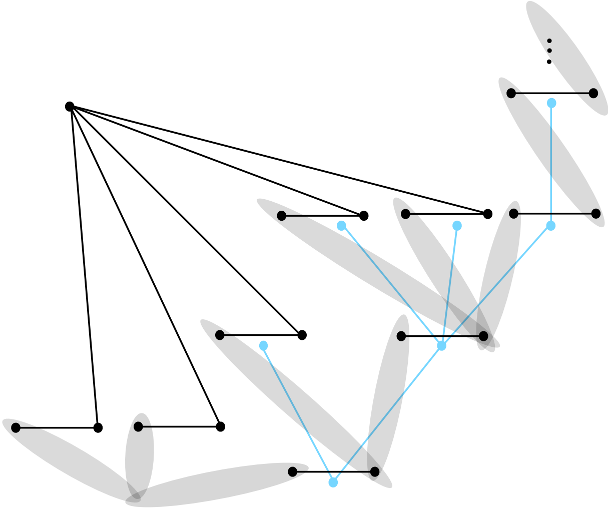
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

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



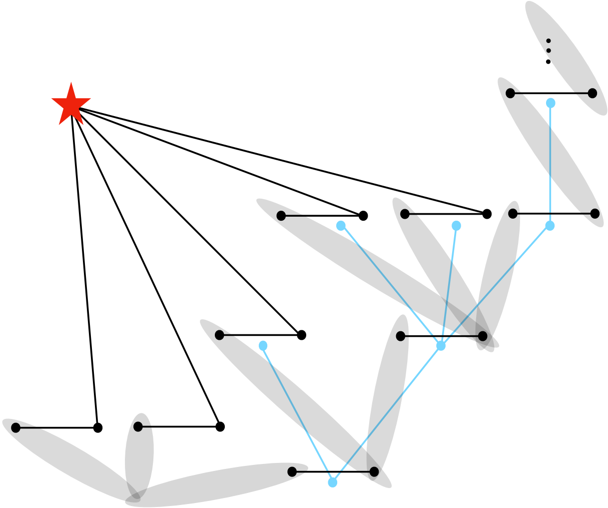
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



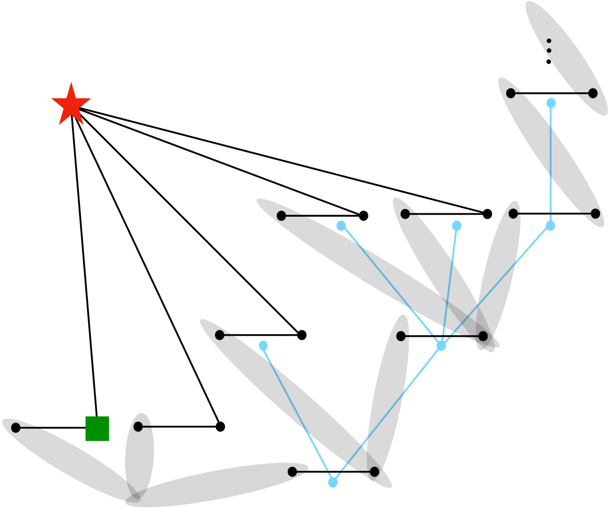
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



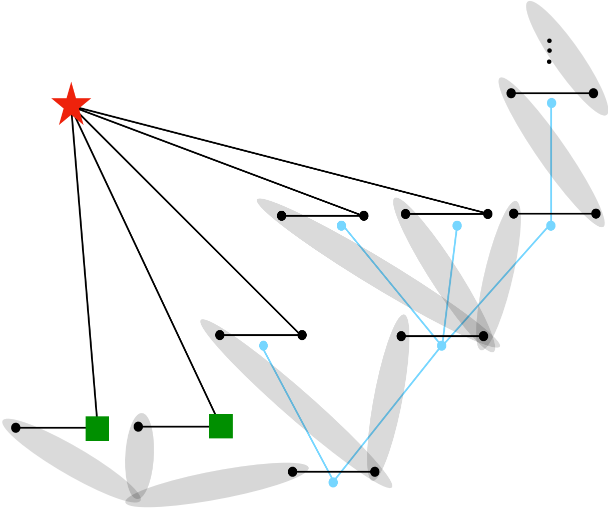
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



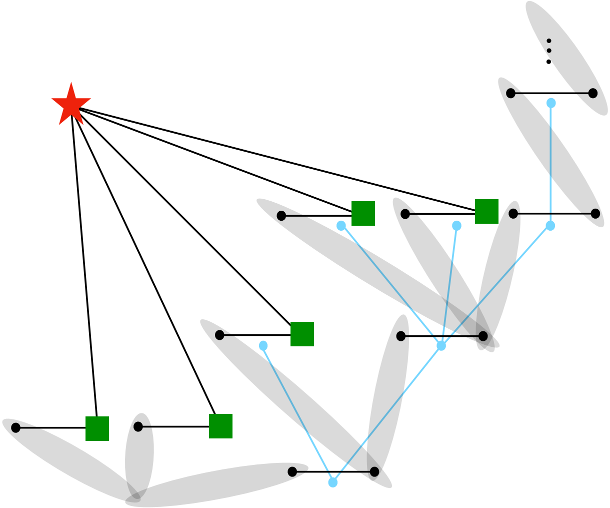
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



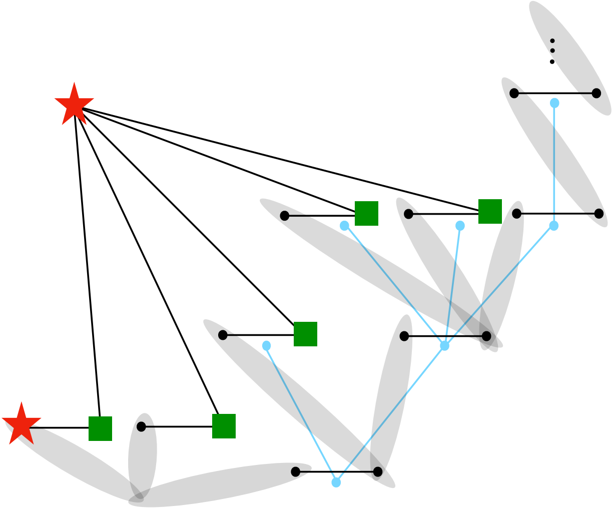
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

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



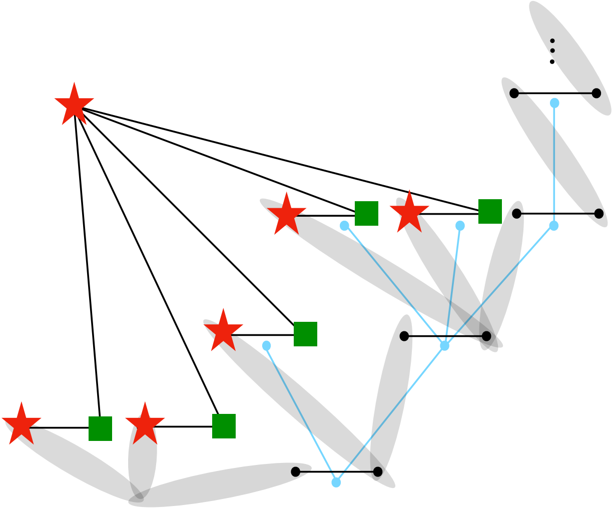
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



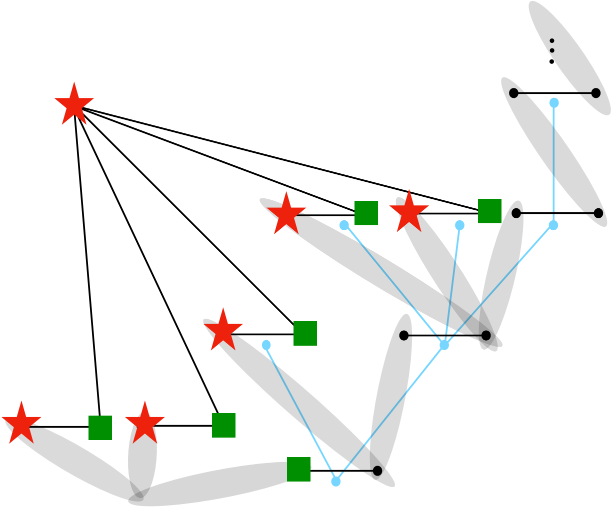
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

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



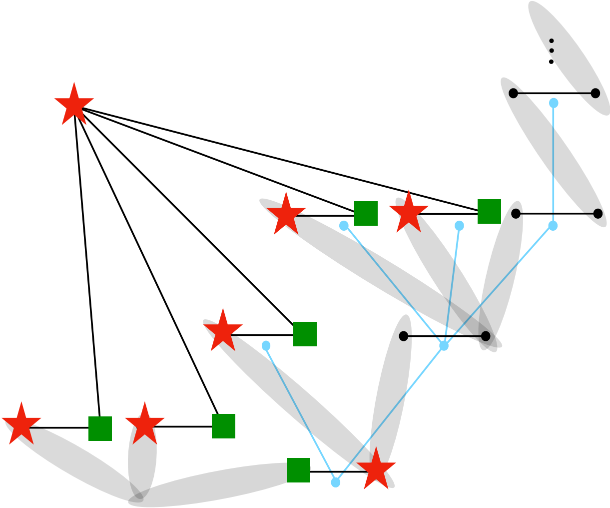
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



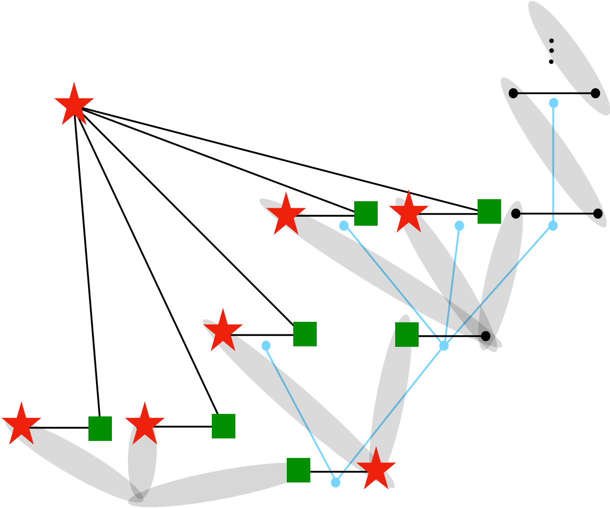
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



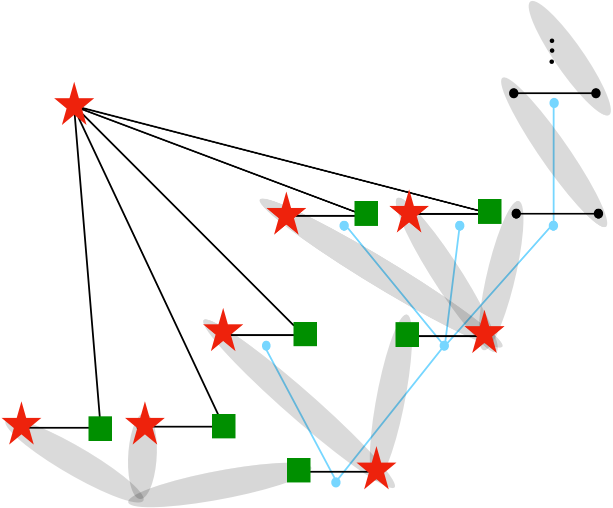
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



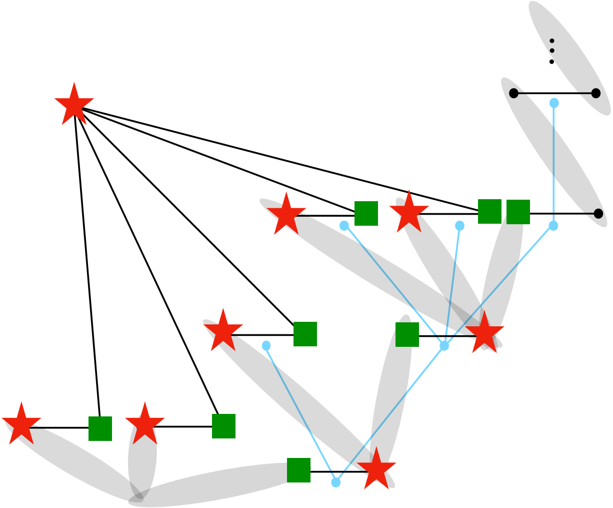
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



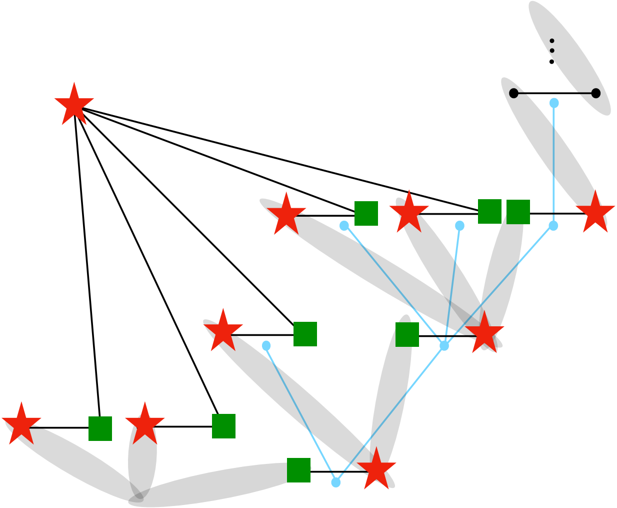
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

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



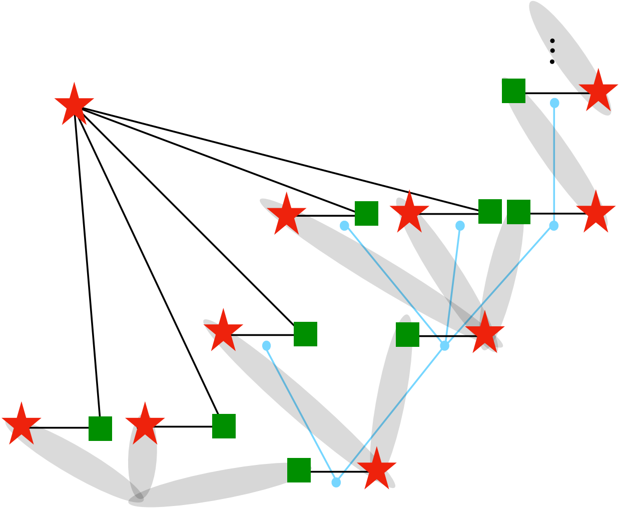
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



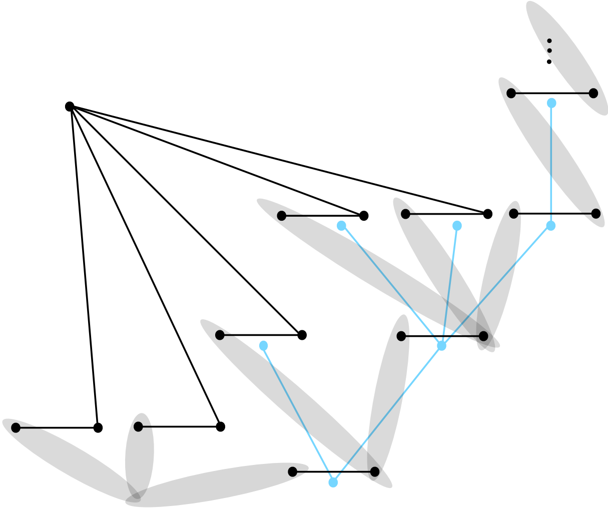
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

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



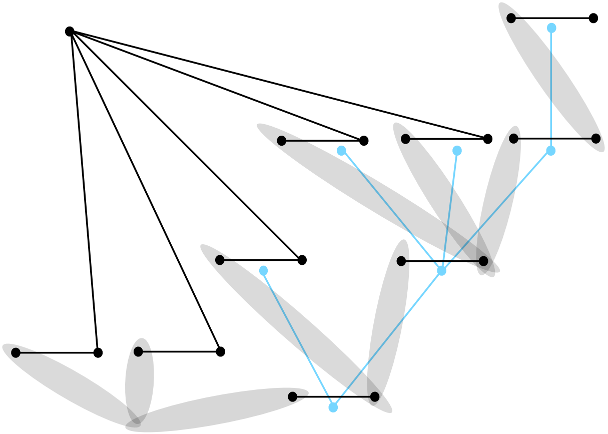
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

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



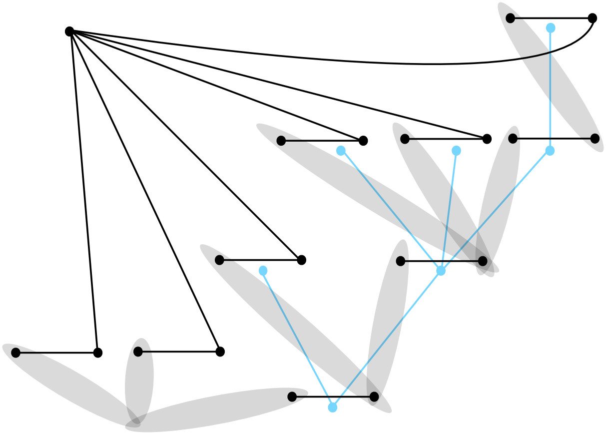
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



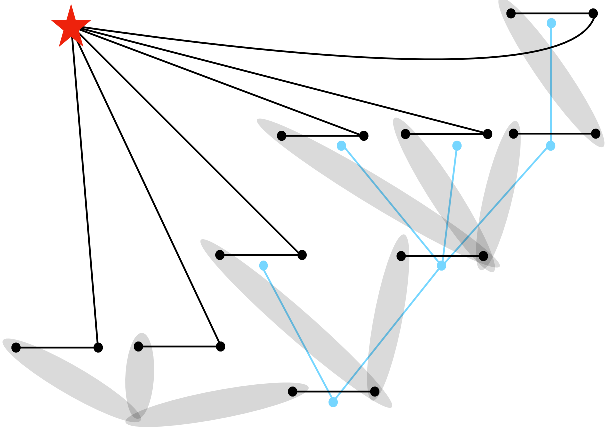
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



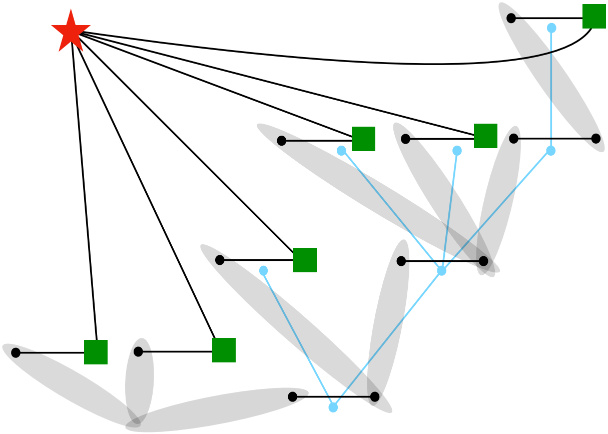
The reversal: $\widehat{\mathrm{HC}} \rightarrow \widehat{\mathrm{WF}}$

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



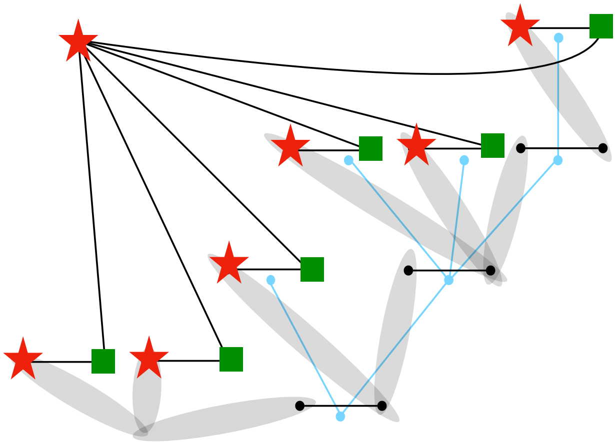
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



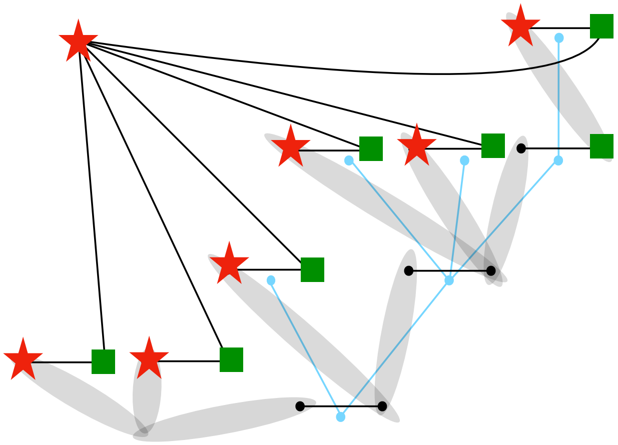
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

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



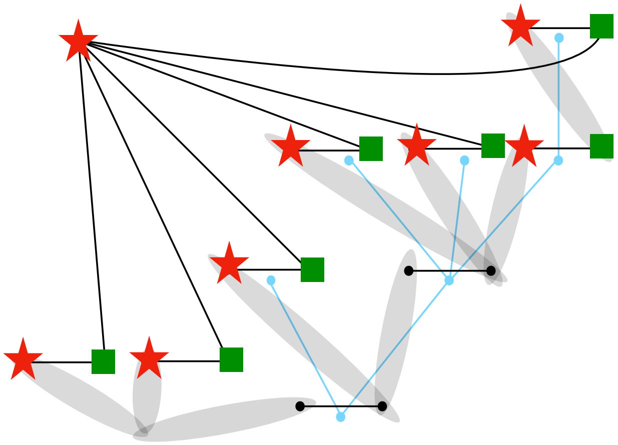
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



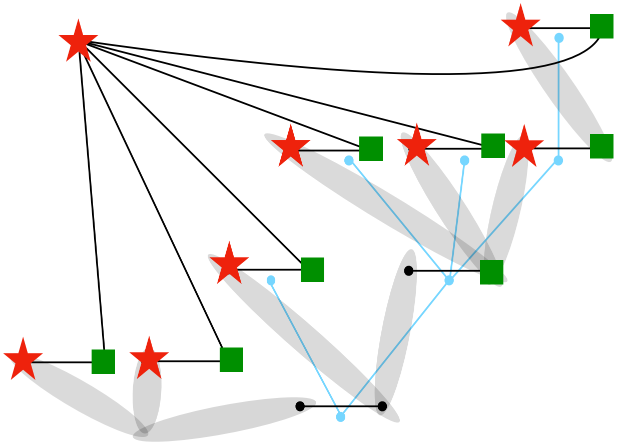
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



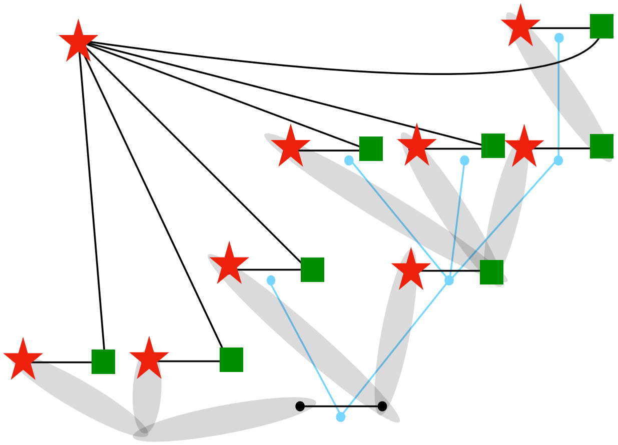
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

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



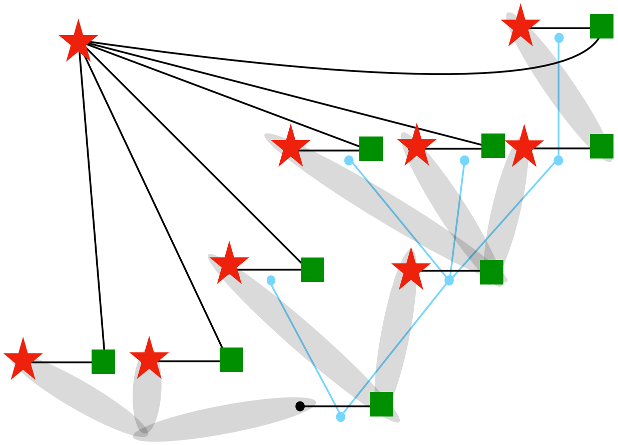
The reversal: $\widehat{\mathrm{HC}} \to \widehat{\mathrm{WF}}$

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



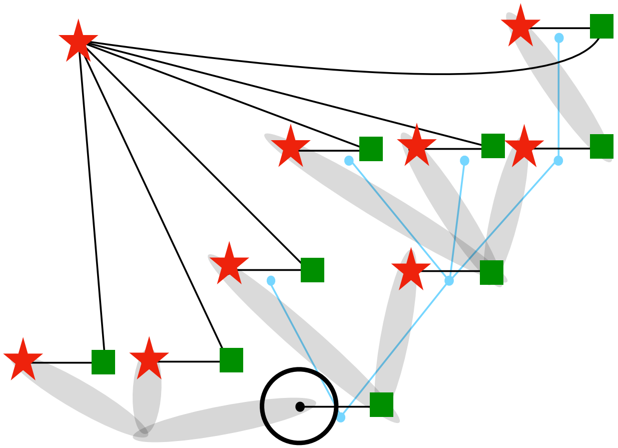
The reversal: $\widehat{\text{HC}} \to \widehat{\text{WF}}$

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



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

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



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

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

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* [4]

References

[1] 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.

[2] 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.

[3] 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.

[4] Jeffry L. Hirst, *Leaf management*. To appear in Computability. Available at arxiv.org/abs/1812.09762.

[5] 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.