



HAL
open science

A characterization of closed under intersection binary hypergraphs by a sequence of trees

Célia Châtel, François Brucker, Pascal Prea

► **To cite this version:**

Célia Châtel, François Brucker, Pascal Prea. A characterization of closed under intersection binary hypergraphs by a sequence of trees. Journées nationales GDR Informatique Mathématique, Apr 2018, Palaiseau, France. hal-01790946

HAL Id: hal-01790946

<https://hal.science/hal-01790946>

Submitted on 14 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A CHARACTERIZATION OF CLOSED UNDER INTERSECTION BINARY HYPERGRAPHS BY A SEQUENCE OF TREES

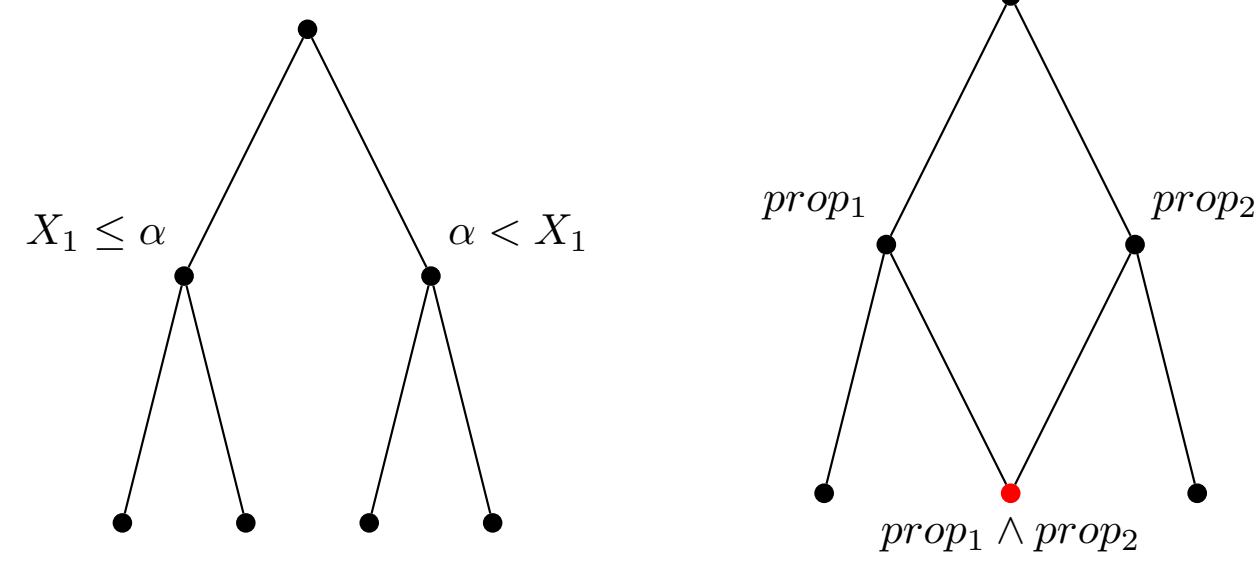
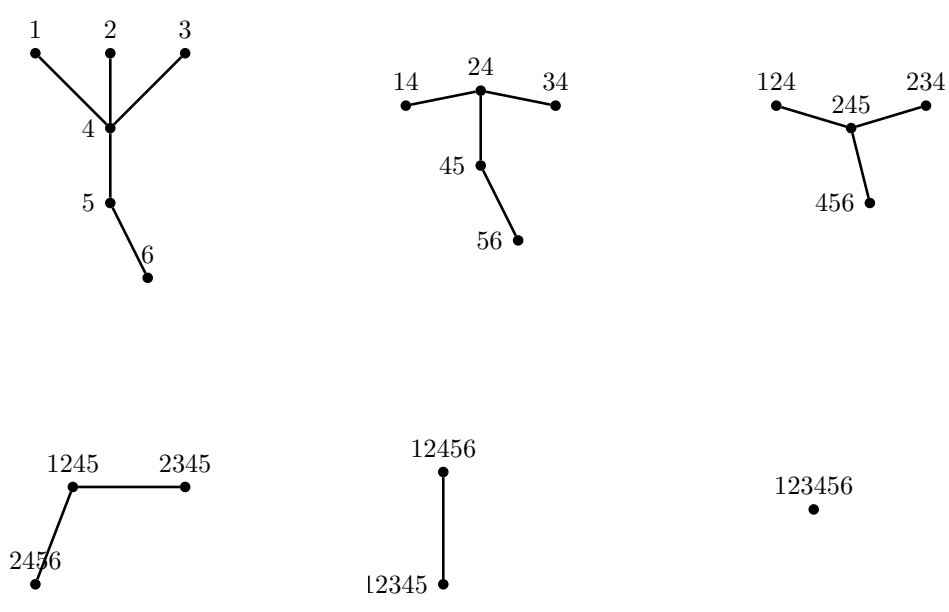
Célia Châtel, François Brucker & Pascal Pr ea

Laboratoire d'Informatique et Syst emes (UMR 7020), Marseille

Context

In 1983, Lehel proposed a characterization of totally balanced hypergraphs by a sequence of trees based on the contraction of all the edges of each tree.

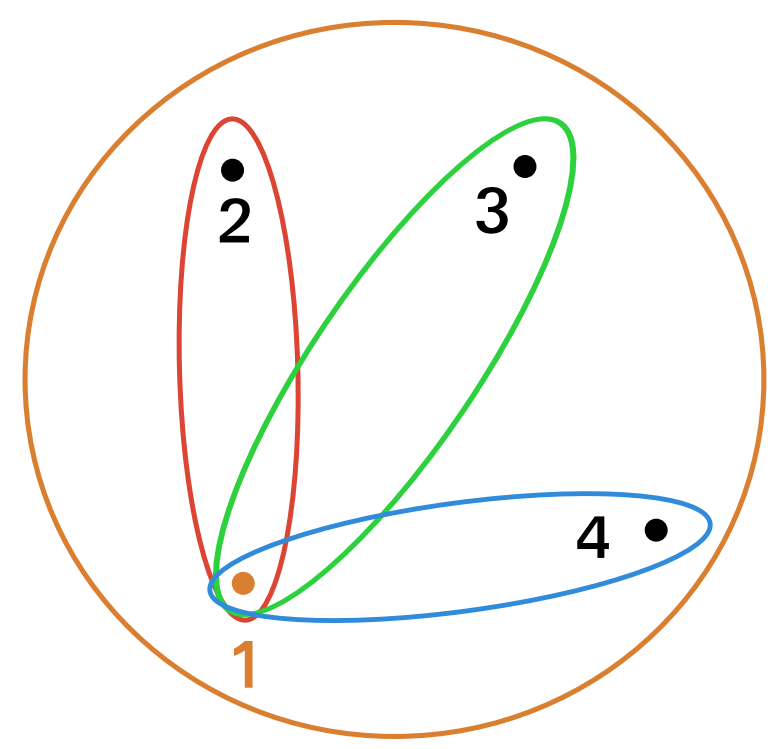
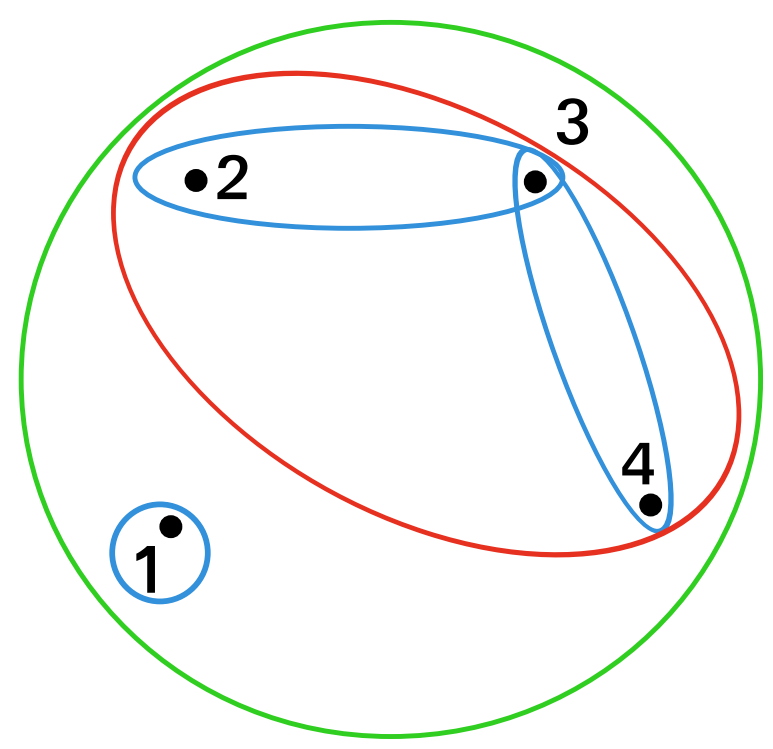
In machine learning, decision trees are widely used binary systems based on yes/no questions. Why not decision lattices?



Binary hypergraphs

A hypergraph $\mathcal{H} = (V, E)$ is said to be *binary* if:

$$\forall v \in E, v \neq \emptyset \begin{cases} |\{w \in E, v \prec w\}| \leq 2 \\ |\{u \in E, u \prec v\}| \leq 2 \end{cases}$$

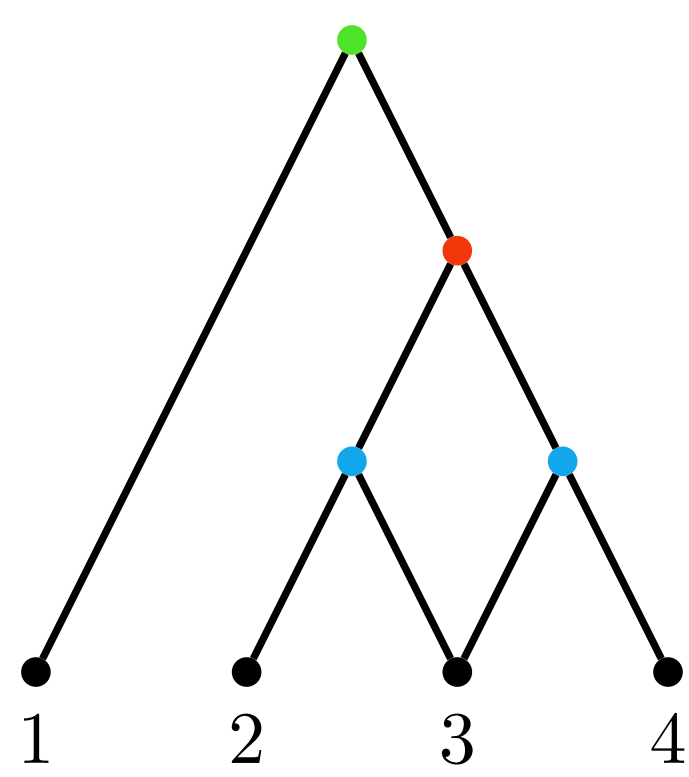


Binary hypergraph

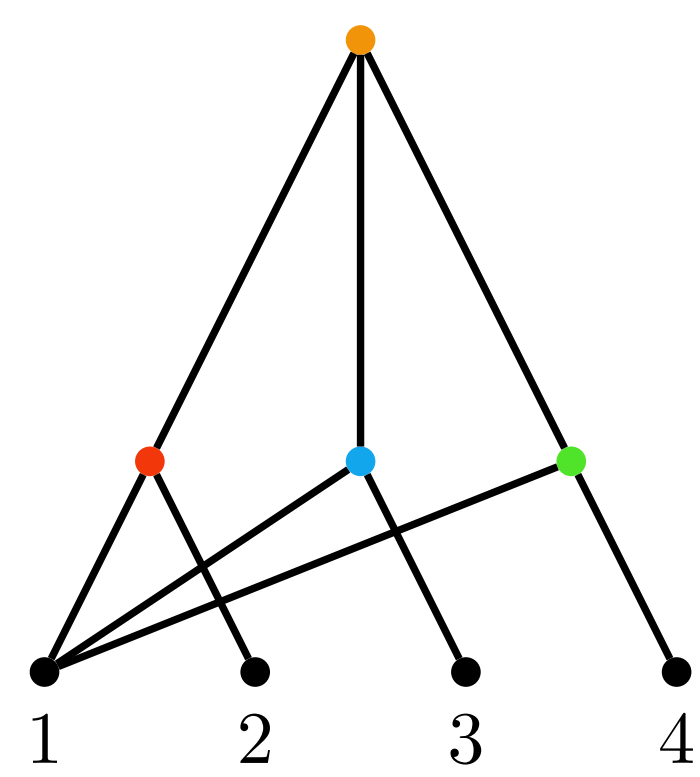
Non binary hypergraph

The *Hasse diagram* of the partially ordered set of the hypergraph's hyperedges is a well adapted representation. A Hasse diagram is such that :

- hyperedges of the hypergraph are vertices of the diagram
- an edge goes from x to y iff $x \prec y$ (i.e. $x \subsetneq y, \nexists z, x \subseteq z \subseteq y$)



Hasse diagram of a binary hypergraph

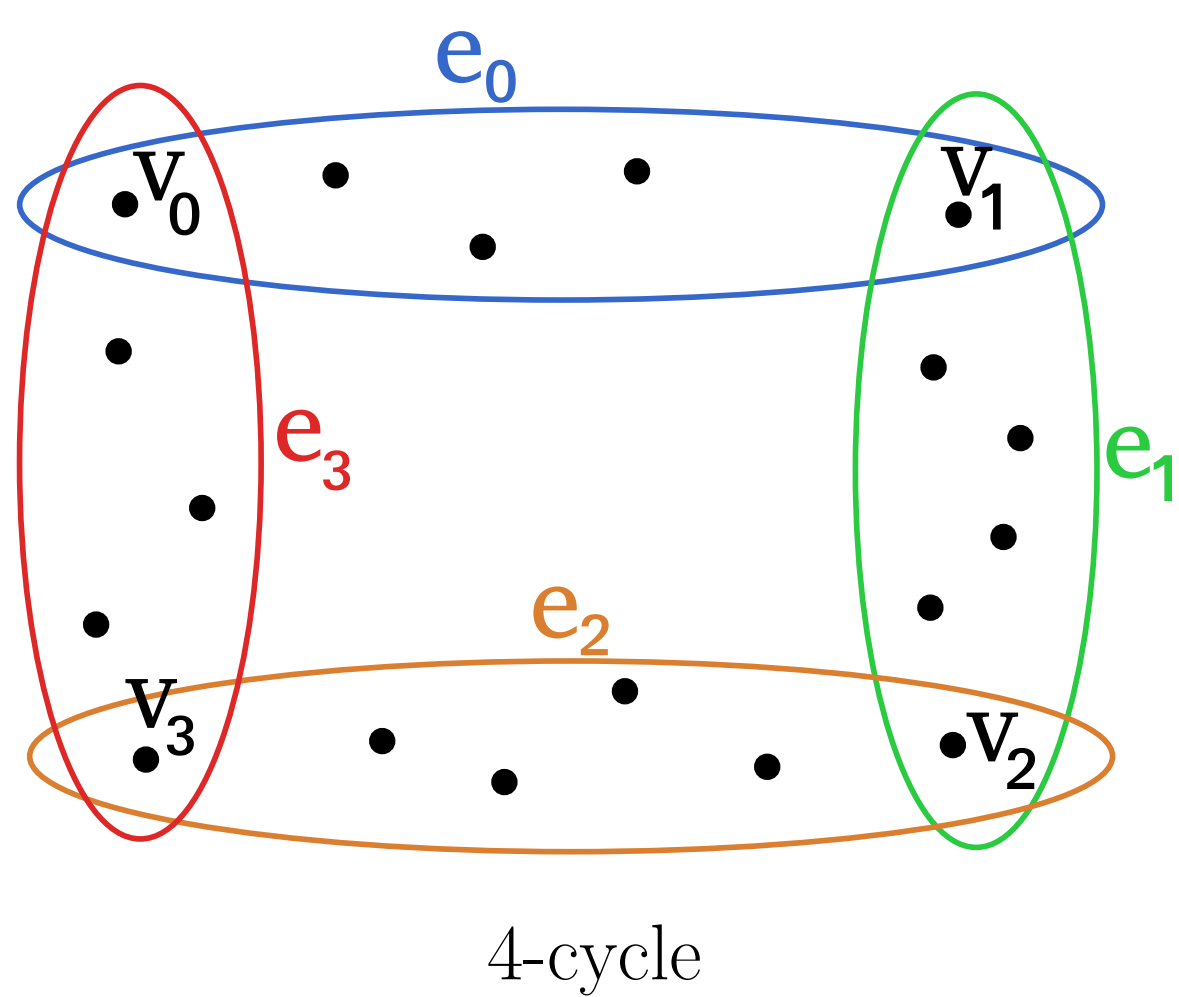


Hasse diagram of a non binary hypergraph

Totally balanced and binarizable hypergraphs

$\mathcal{H} = (V, E)$ *totally balanced* iff it has no special cycle i.e. $(v_0, e_0, \dots, v_n, e_n)$ such that,

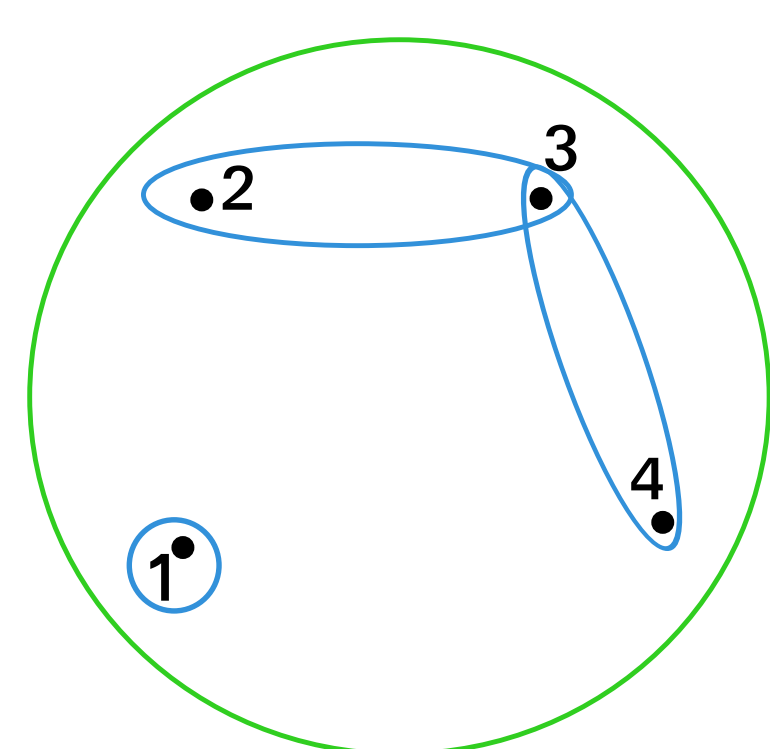
- $\forall i, v_i \in e_i$
- $\forall i, v_i \in e_{i-1 \bmod n}$
- $\forall j \neq i, j \neq i-1 \bmod n, v_i \notin e_j$



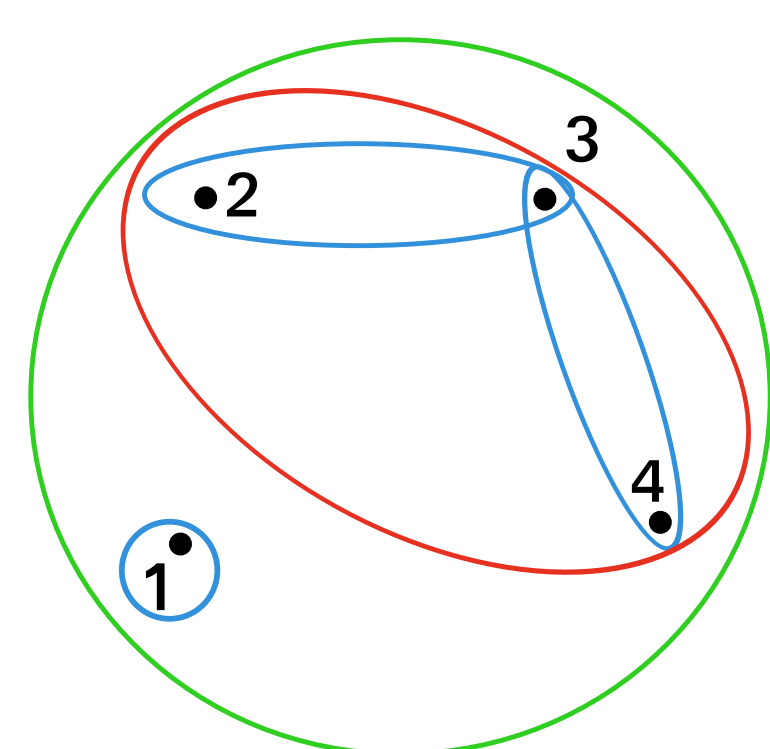
4-cycle

$\mathcal{H} = (V, E)$ is *binarizable* iff

$$\exists \mathcal{H}' = (V', E') \text{ binary hypergraph such that } V \subseteq V' \text{ and } E \subseteq E'$$



Non binary hypergraph

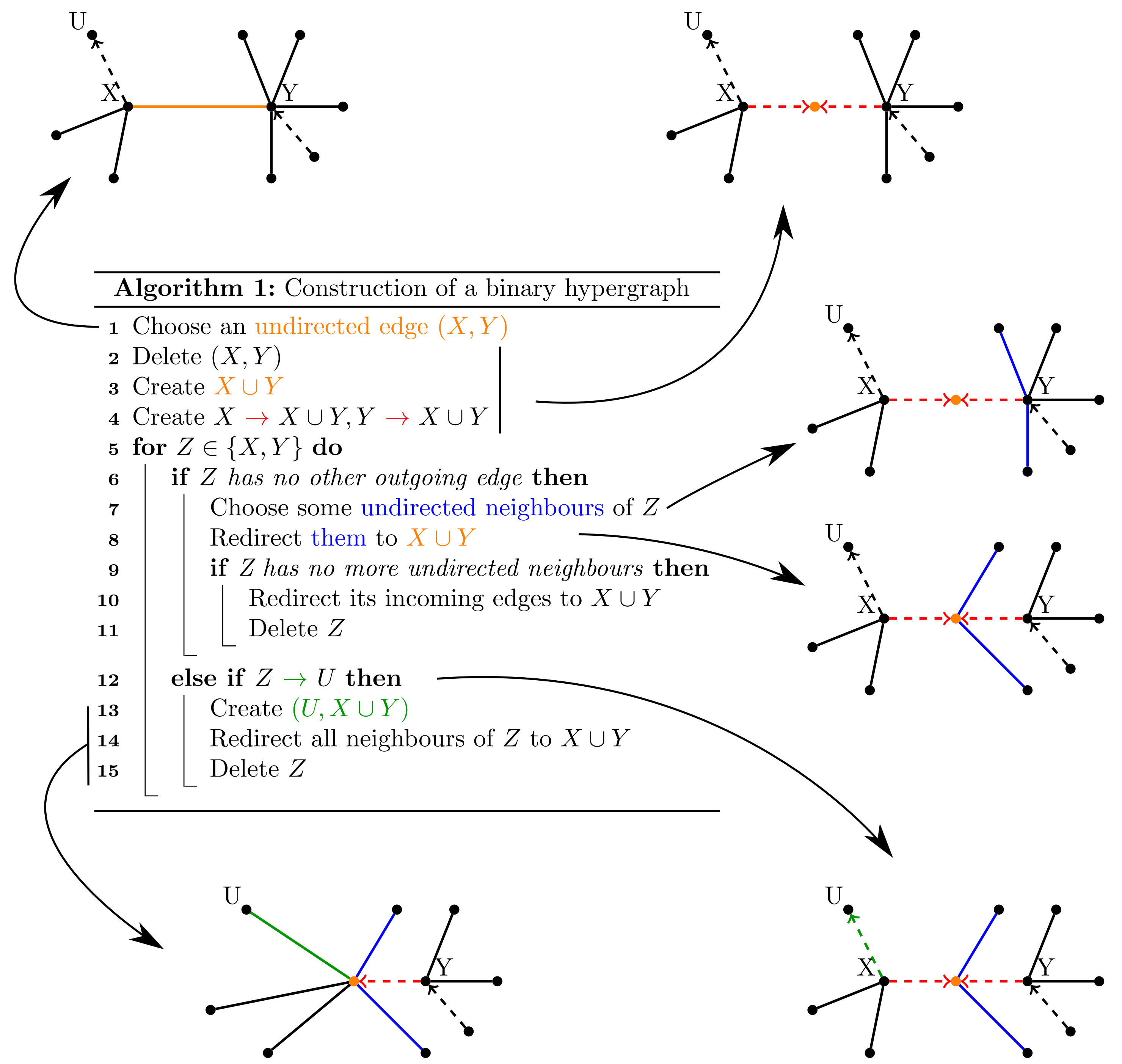


Binarized hypergraph

Theorem

A closed under intersection hypergraph is binarizable iff it is totally balanced.

Algorithm



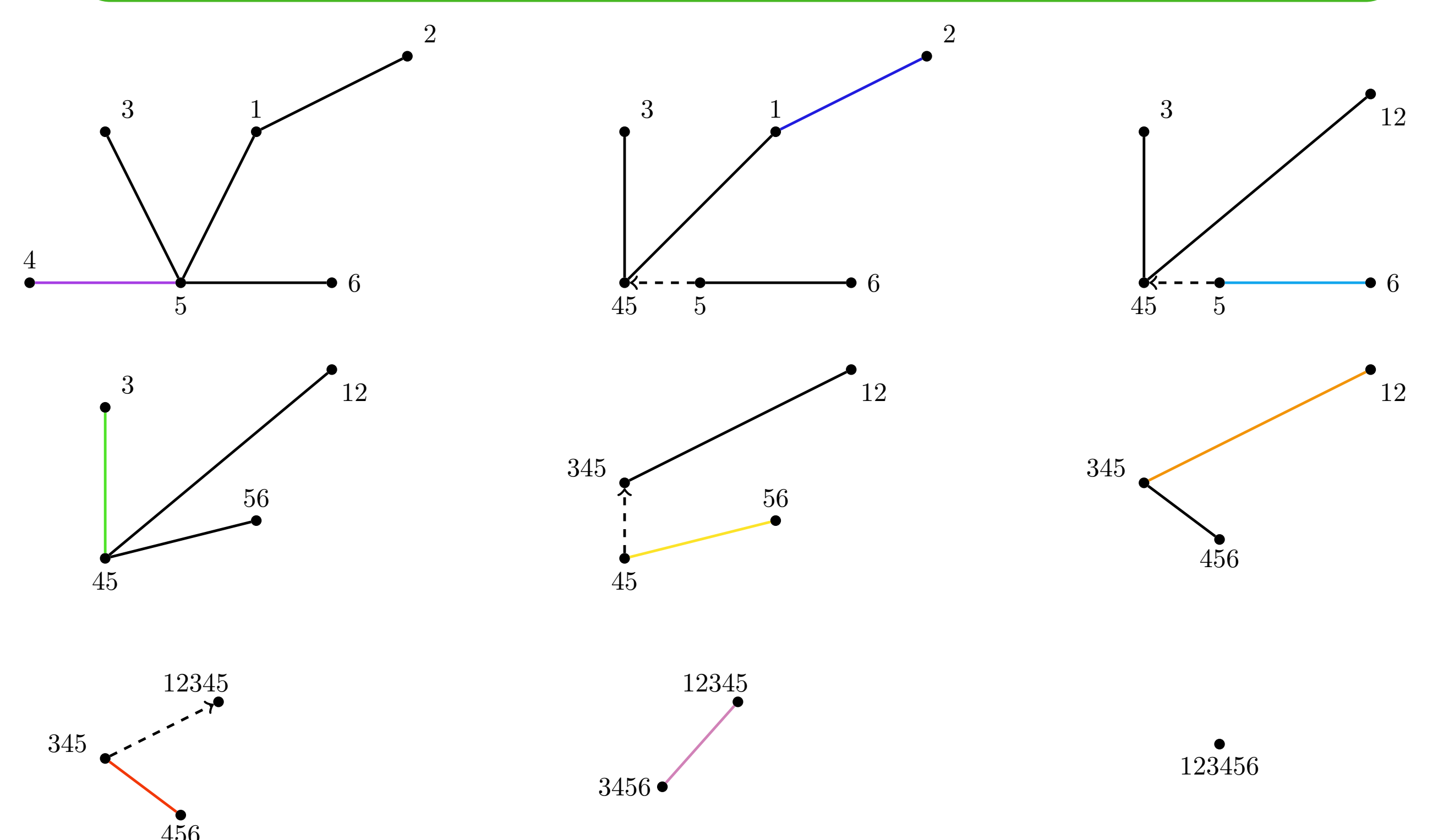
Algorithm 1: Construction of a binary hypergraph

- 1 Choose an **undirected edge** (X, Y)
- 2 Delete (X, Y)
- 3 Create $X \cup Y$
- 4 Create $X \rightarrow X \cup Y, Y \rightarrow X \cup Y$
- 5 **for** $Z \in \{X, Y\}$ **do**
- 6 **if** Z has no other outgoing edge **then**
- 7 Choose some **undirected neighbours** of Z
- 8 Redirect **them** to $X \cup Y$
- 9 **if** Z has no more undirected neighbours **then**
- 10 Redirect its incoming edges to $X \cup Y$
- 11 Delete Z
- 12 **else if** $Z \rightarrow U$ **then**
- 13 Create $(U, X \cup Y)$
- 14 Redirect all neighbours of Z to $X \cup Y$
- 15 Delete Z

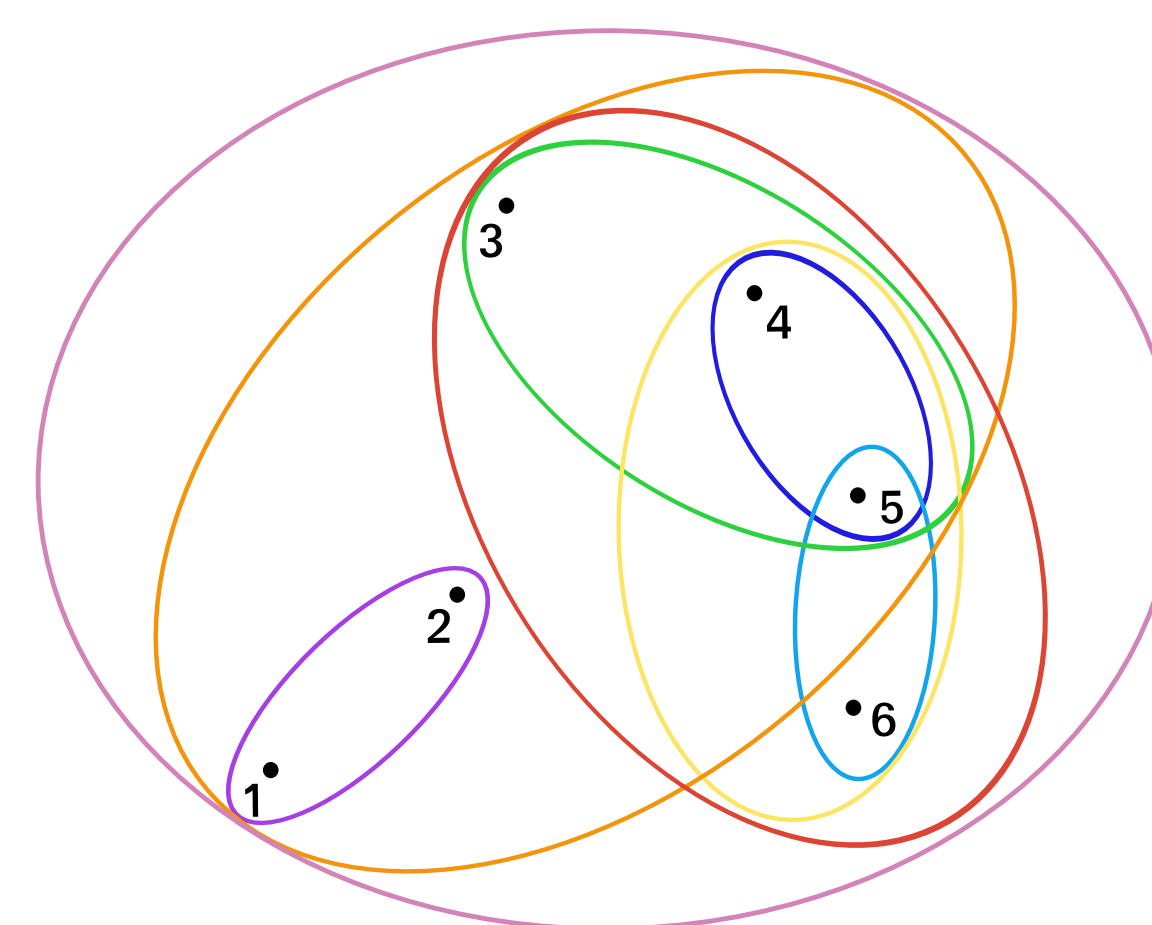
Theorem

A hypergraph closed under intersection is binary iff its hyperedges are the vertices of a sequence of trees constructed by the algorithm.

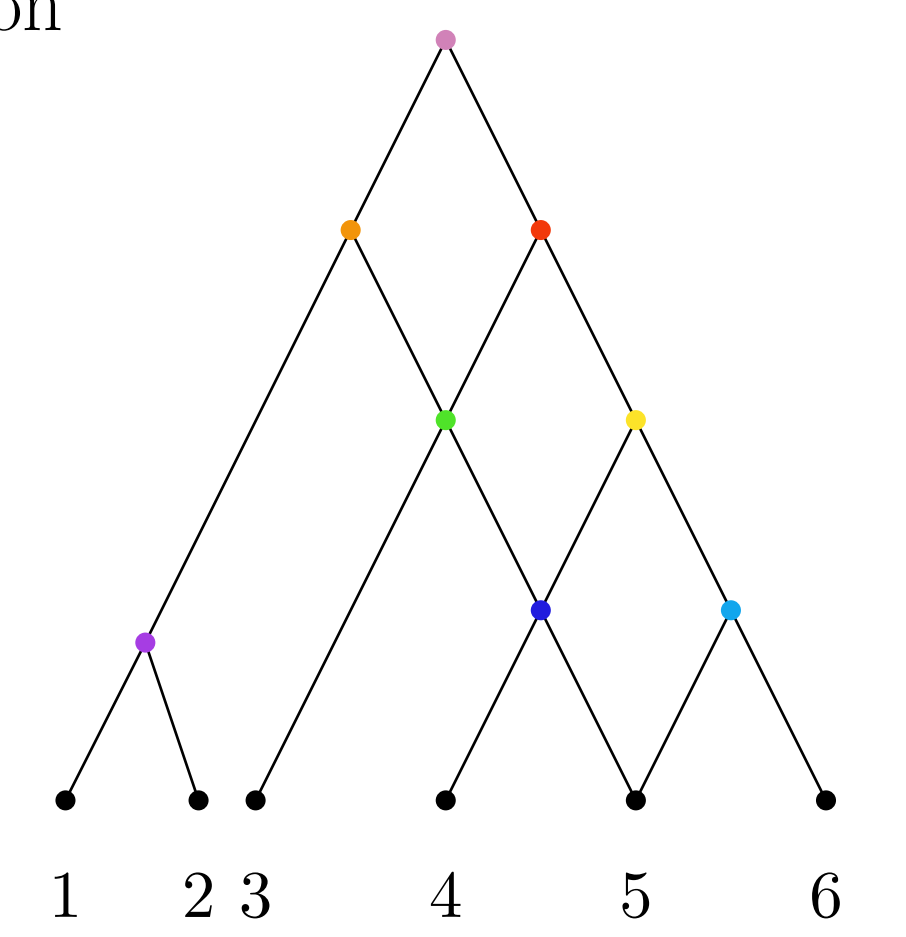
Example



Example of construction



Hypergraph obtained



Hasse diagram of the hypergraph

Conclusion

We give a characterization by a sequence of trees of binary closed under intersection hypergraphs similar to Lehel's characterization of totally balanced hypergraphs. This construction can also be extended to

- build a top down or bottom up binary hypergraph,
- binarize totally balanced non binary hypergraphs,
- approximate a non totally balanced hypergraph by a binary hypergraph.