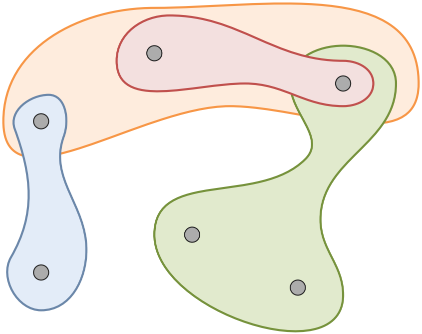
Computing the Union Join and Subset Graph of Acyclic Hypergraphs in Subquadratic Time

Arne Leitert

Hypergraph

Hypergraph

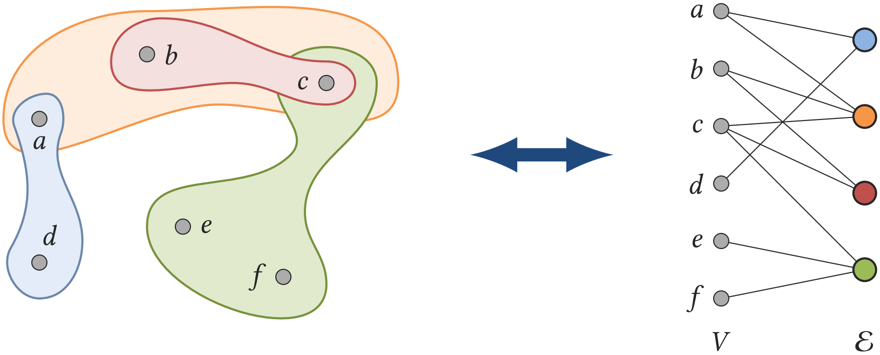
A *hypergraph* $H = (V, \mathcal{E})$ is a set of vertices V and a family $\mathcal{E}$ (called hyperedges) of subsets of V.



It is a generalisation of a graph; hyperedges can contain an arbitrary positive number of vertices.

Representation: Incidence Graph

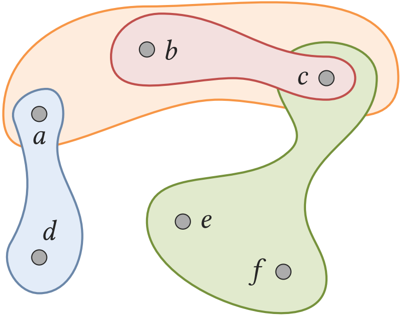
Representation as bipartite graph $\mathcal{I}(H)$, called *incidence graph*.



$n = |V|$ $\quad m = |\mathcal{E}|$ $\quad N = \sum_{E \in \mathcal{E}} |E|$ $\quad$ Total input size: $\mathcal{O}(N)$

Representation: Incidence Matrix

Representation as binary matrix $\mathcal{M}(H)$, called *incidence matrix*.

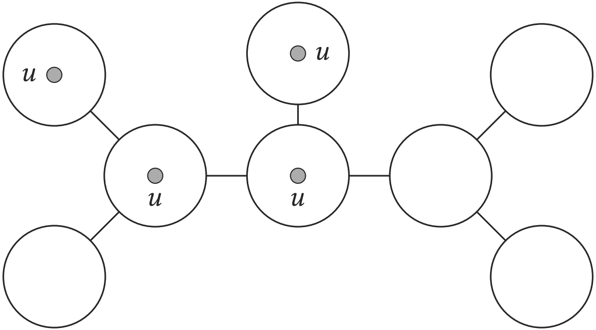


$$
\begin{array}{c}
a \\ b \\ c \\ d \\ e \\ f
\end{array}
\begin{bmatrix}
\bullet & \bullet & & \\
& \bullet & \bullet & \\
& \bullet & \bullet & \bullet \\
\bullet & & & \\
& & & \bullet \\
& & & \bullet
\end{bmatrix}
$$

Acyclic Hypergraphs

Acyclic Hypergraph

A hypergraph $H = (V, \mathcal{E})$ is *acyclic* if its hyperedges $\mathcal{E}$ form a tree T such that, for each vertex $v \in V$, the hyperedges containing v induce a subtree of T. T is called the *join tree* of H.



Acyclicity can be checked in linear time (also computes T) [Tarjan, Yannak 1984].

Acyclic Hypergraphs

Hierarchy (all subsets are proper subsets)

acyclic = α-acyclic

β-acyclic

γ-acyclic   interval

Applications

- Relational Databases
- Tree-Decompositions (e. g. Tree-Breadth, Tree-Width)
- Atoms of graphs
- Closely related to Chordal graphs and Dually Chordal graphs

Subset Graph and Union Join Graph

Sperner Family Problem

Input: A family $\mathcal{F}$ of sets.

Question: Does $\mathcal{F}$ contains two distinct sets S_i and S_j such that $S_i \subseteq S_j$?

Hardenss of Sperner Family Problem

Strong Exponential Time Hypothesis (SETH)

There is no algorithm that solves the Boolean satisfiability problem (without limiting clause sizes) in $\mathcal{O}(2^{n-\varepsilon})$ time for some $\varepsilon > 0$.

Hardenss of Sperner Family Problem

Strong Exponential Time Hypothesis (SETH)

There is no algorithm that solves the Boolean satisfiability problem (without limiting clause sizes) in $\mathcal{O}\left(2^{n-\varepsilon}\right)$ time for some $\varepsilon > 0$.

A chain of reductions then allows to state the following:

Theorem [Borassi, Crescenzi, Habib 2016]

If SETH is true, then there is no algorithm that solves the Sperner Family problem for an arbitrary family $\mathcal{F}$ (i. e., for an arbitrary hypergraph) in $\mathcal{O}\left(N^{2-\varepsilon}\right)$ time.

Note that all subfamilies $\mathcal{F}_S = \left\{ S' \mid S = S' \right\}$ can be determined in linear time.

Subset Graph Problem

Subset Graph Problem

Input: A family $\mathcal{F} = \{S_1, S_2, \ldots, S_m\}$ of sets.

Output: The *subset graph* $G = (\mathcal{F}, E)$ with $S_i S_j \in E$ if and only if $S_i \subseteq S_j$ and $i \neq j$.

Hardenss of Subset Graph Problem

Theorem [Borassi, Crescenzi, Habib 2016]

If SETH is true, then there is no algorithm to compute the subset graph of an arbitrary hypergraph in $\mathcal{O}(N^{2-\varepsilon})$ time, even if the output is sparse.

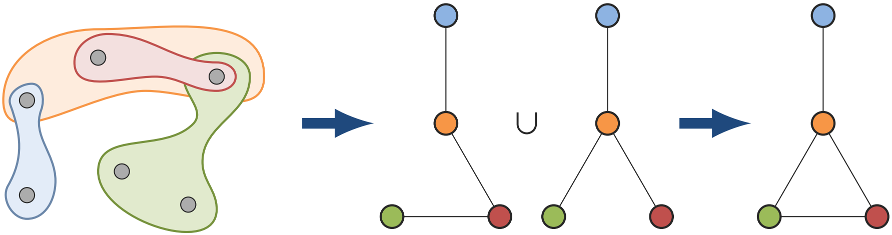
Follows directly from hardness of Sperner Family Problem.

Theorem [Pritchard 1999]

There is an $\mathcal{O}(N^2/\log N)$-time algorithm which computes the subset graph for a given family of sets.

Union Join Graph

The *union join graph* of an acyclic hypergraph is the union of all its join trees. That is, for an acyclic hypergraph $H = (V, \mathcal{E})$, the *union join graph* is a the graph $G = (\mathcal{E}, X)$ with $X = \left\{ E_i E_j \;\middle|\; \text{There exist a join tree for } H \text{ with the edge } E_i E_j. \right\}$.



[Berry, Simonet 2016] gave $\mathcal{O}(Nm)$-time algorithm for acyclic hypergraphs.

Hardness of Union Join Graph

Is Join Tree Unique?

Join Tree

- Can be computed in linear time.
- Question: Is it unique?

Is Join Tree Unique?

Join Tree

- Can be computed in linear time.
- Question: Is it unique?

Lemma

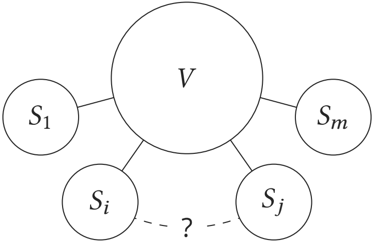
If SETH is true, then there is no algorithm that determines in $O\left(N^{2-\varepsilon}\right)$ time whether a given acyclic hypergraph has a unique join tree.

Proof

- Linear-time reduction from Sperner Family Problem.

Reduction

- Given family $\mathcal{F} = \{S_1, S_2, \ldots, S_m\}$.
- Create hypergraph $H = (V, \mathcal{E})$ with $V = \bigcup_{S_i \in \mathcal{F}} S_i$ and $\mathcal{E} = \mathcal{F} \cup \{V\}$.



Observation

- There is no pair $S_i \subseteq S_j$ if and only if the join tree for H is unique. $\square$

Implication for Union Join Graph

Observation

- Join tree for H is unique if and only if union join graph is a tree.

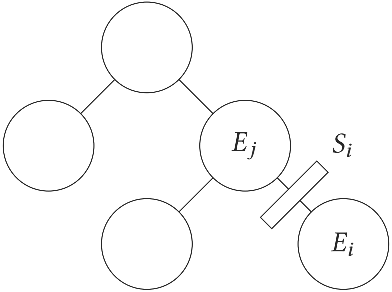
Theorem

If SETH is true, then there is no algorithm that computes the union join graph of an α-acyclic hypergraph in $\mathcal{O}\left(N^{2-\varepsilon}\right)$ time.

Union Join Graph via Subset Graph

Separator Hypergraph

Let T be join tree of $H = (V, \mathcal{E})$ rooted in some hyperedge R.



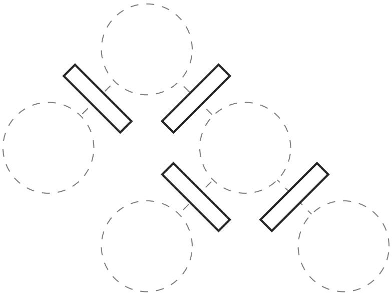
Up-Separator $\boldsymbol{S^{\uparrow}(E_i)}$ of hyperedge E_i is intersection with parent E_j, i. e.,

$$S^{\uparrow}(E_i) := E_i \cap E_j.$$

Separator Hypergraph

Separator Hypergraph $\mathcal{S}(H)$ for H is the hypergraph formed from the set

$$\mathcal{E}_S = \left\{ S^{\uparrow}(E_i) \,\middle|\, E_i \in \mathcal{E}, E_i \neq R \right\}.$$

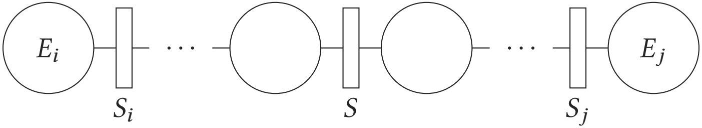


Separators and Union Join Graph

Lemma

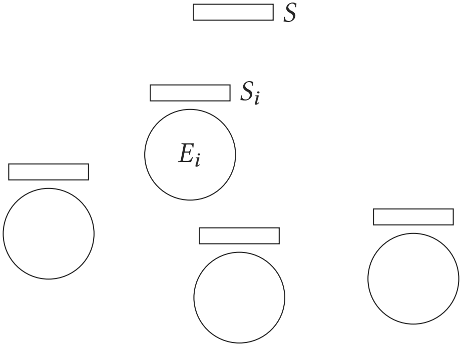
For any distinct $E_i, E_j \in \mathcal{E}$, the following are equivalent.

- H has a join tree T with the edge E_iE_j.
- T has a separator S on the path from E_i to E_j with $S \subseteq S_i$ and $S \subseteq S_j$.



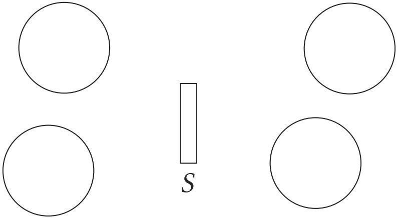
Algorithm

1. Compute subset graph $G_{\mathcal{S}}$ of separator hypergraph $\mathcal{S}(H)$
2. For some $S \in \mathcal{S}(H)$, use $G_{\mathcal{S}}$ to find all hyperedges E_i with $S \subseteq S_i$.



Algorithm

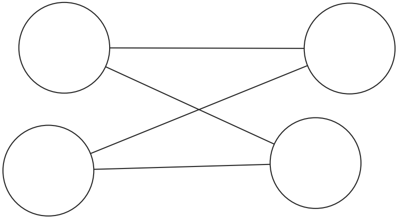
3. Partition hyperedges based on which side of S in T they are.
   (Determine if E is descendants of S or not in rooted tree, constant time after pre- and post-order on tree.)



Finding Union Join Graph

Algorithm

4. Make all “left” hyperedges adjacent to all “right” hyperedges.
5. Repeat for all $S \in \mathcal{S}(H)$.



Finding Union Join Graph

Theorem

The algorithm computes the union join graph G of a given acyclic hypergraph H in $\mathcal{O}\big(T_{\mathcal{A}}(H) + N + |G|\big)$ time where $T_{\mathcal{A}}(H)$ is the runtime of a given algorithm $\mathcal{A}$ with the separator hypergraph of H as input.

Finding Union Join Graph

Recall, subset graph can be computed in $\mathcal{O}(N^2/\log N)$ time. Therefore:

Theorem

There is an algorithm that computes the union join graph G of an acyclic hypergraph in $\mathcal{O}\left(N^2/\log N + |G|\right)$ time.

β-Acyclic Hypergraphs

β-Acyclic Hypergraphs

β-Acyclic Hypergraph

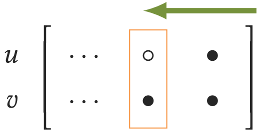
A hypergraph $H = (V, \mathcal{E})$ is *β-acyclic* if each $\mathcal{E}' \subseteq \mathcal{E}$ forms an acyclic hypergraph.

β-Acyclic Hypergraphs

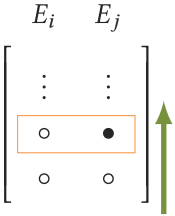
- Closely related to Strongly Chordal graphs and Chordal Bipartite graphs.
- Also called *Totally Balanced* hypergraphs.

Doubly Lexically Ordered Matrix

A matrix is *doubly lexically ordered* if rows and columns are permuted in such a way that row- and column-vectors are in non-decreasing lexicographic order. Priorities in rows decrease from right to left and in columns from bottom to top.



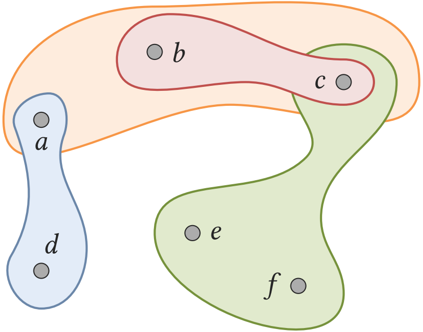
$u \prec v$



$E_i \prec E_j$

A doubly lexically ordering of vertices and hyperedges can be found in $O\left(N \log(n+m)\right)$ time (full matrix is not required) [Piage, Tarjan 1987].

Doubly Lexically Ordered Matrix



$$\begin{array}{c} d \\ e \\ f \\ a \\ b \\ c \end{array} \begin{bmatrix} \bullet & & & \\ & \bullet & & \\ & \bullet & & \\ \bullet & & & \bullet \\ & & \bullet & \bullet \\ & \bullet & \bullet & \bullet \end{bmatrix}$$

Preparation

- Find double lexicographical ordering of vertices and hyperedges.

Lemma

Let v be the vertex highest in E_i. Then, $E_i \subseteq E_j$ if and only if $E_i \preceq E_j$ and $v \in E_j$.

$$\begin{array}{cc} & \begin{array}{ccc} E_i & \overset{?}{\subseteq} & E_j \end{array} \\ \begin{array}{c} v \\ \\ \\ u \end{array} & \begin{bmatrix} \bullet & & \bullet \\ \vdots & & \vdots \\ \bullet & & \circ \end{bmatrix} \end{array}$$

Finding Subset Graph

Approach

- Iterate (backwards) over hyperedges of v until E_i is found. E_i is subset of all found hyperedges.
- Runtime is number of edges in output created.

$$
\begin{array}{c c}
 & \begin{array}{cc} E_i & E_j \end{array} \\
\begin{array}{c} v \\ \\ \\ u \end{array} & \left[\begin{array}{cc} \bullet & \bullet \\ \vdots & \vdots \\ \bullet & \bullet \end{array}\right]
\end{array}
$$

Finding Subset Graph

Theorem

There is an algorithm that computes the subset graph G of a given β-acyclic hypergraph in $\mathcal{O}\big(N \log(n+m) + |G|\big)$ time.

Finding Union Join Graph

Lemma

If a hypergraph is β-acyclic, then its separator hypergraph is β-acyclic, too.

Using the algorithm for general acyclic hypergraphs and for subset graphs of β-acyclic hypergaphs then gives the following:

Theorem

There is an algorithm that computes the union join graph G of a given β-acyclic hypergraph in $O\big(N \log(n+m) + |G|\big)$ time.

γ-Acyclic Hypergraphs

γ-Acyclic Hypergraphs

γ-Acyclic Hypergraph

A hypergraph is γ-*acyclic* if, for all distinct hyperedges E_i and E_j,

$$S = E_i \cap E_j \neq \emptyset \quad \text{implies} \quad S \text{ separates } E_i \setminus E_j \text{ from } E_j \setminus E_i.$$

Lemma

For an acyclic hypergraph with join tree T, the following are equivalent:

- H has a join tree T with the edge E_iE_j.
- $S = E_i \cap E_j \neq \emptyset$ and S separates $E_i \setminus E_j$ from $E_j \setminus E_i$.

Follows from definition of join trees.

Line Graph

The *line graph* $L(H) = (\mathcal{E}, \mathcal{E}_L)$ of a hypergraph $H = (V, \mathcal{E})$ is the intersection graph of its hyperedges. That is,

$$\mathcal{E}_L = \{\, E_iE_j \mid E_i, E_j \in \mathcal{E}; E_i \cap E_j \neq \emptyset \,\}.$$

Finding Union Join Graph

Theorem

An acyclic hypergraph is γ-acyclic if and only if its linegraph is isomorphic to its union join graph.

Theorem

There is an algorithm that computes the union join graph G of a given γ-acyclic hypergraph in $\mathcal{O}(N + |G|)$ time.

Bachman Diagram

For $H = (V, \mathcal{E})$, let $\mathcal{X}$ be the set of non-empty intersections $\mathfrak{X}$ of subsets of $\mathcal{E}$, i. e.,

$$\mathcal{X} = \bigcup_{\mathcal{E}' \subseteq \mathcal{E}} \left\{ \mathfrak{X} \,\middle|\, \mathfrak{X} = \bigcap_{E \in \mathcal{E}'} E, \mathfrak{X} \neq \emptyset \right\}$$

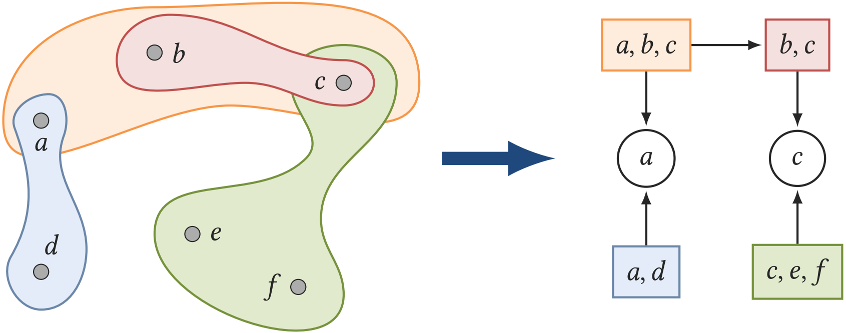
Bachman Diagram

The *Bachman diagram* $\mathcal{B}(H)$ of H is a directed graph with the node set $\mathcal{X}$ such that there is an edge from $\mathfrak{X}$ to $\mathfrak{Y}$ if $\mathfrak{X} \supset \mathfrak{Y}$ and there is no $\mathfrak{Z}$ with $\mathfrak{X} \supset \mathfrak{Z} \supset \mathfrak{Y}$.

Bachman Diagram

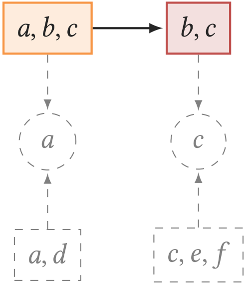
Lemma [Fagin 1983]

A hypergraph is γ-acyclic if and only if its Bachman diagram forms a tree.



Observation

- $E_i \subseteq E_j$ if and only if there is a path from E_j to E_i in $\mathcal{B}(H)$.



Algorithm Idea

- Compute a simplified Bachman diagram B for H in $\mathcal{O}(N)$ time.
- For each E_i, determine all distinct E_j such that there is path from E_j to E_i in B.
- For all such E_i, E_j, add edge (E_i, E_j) to subset graph G.

Theorem

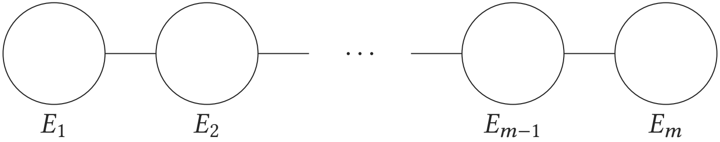
There is an algorithm that computes the subset graph G of a given γ-acyclic hypergraph in $\mathcal{O}\big(N + |G|\big)$ time.

Interval Hypergraphs

Interval Hypergraphs

Interval Hypergraphs

An acyclic hypergraph is an *interval hypergraph* if it admits a join tree that forms a path.

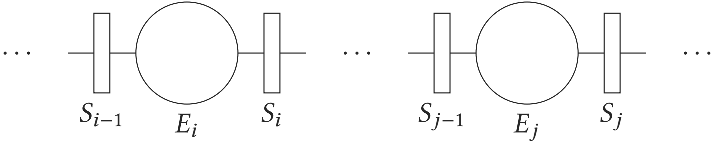


Recognition and computing order $\sigma = \langle E_1, E_2, \ldots, E_m \rangle$ in $\mathcal{O}(N)$ time [Habib et al. 2000].

Interval Hypergraphs

$\phi(v)$ Index of right-most hyperedge containing v.

$\phi(S_i) = \min_{v \in S_i} \phi(v)$ (index of right-most hyperedge containing all of S_i)



Lemma

For all E_i, E_j with $i < j$, $E_i \subseteq E_j$ if and only if $|E_i| = |S_i|$ and $\phi(S_i) \geq j$.

Symmetry allows to also determine all $E_i \supseteq E_j$.

Theorem

There is an algorithm that computes the subset graph G of a given interval hypergraph in $\mathcal{O}(N + |G|)$ time.

Using the algorithm from earlier:

Theorem

There is an algorithm that computes the union join graph G of a given interval hypergraph in $\mathcal{O}(N + |G|)$ time.

Thank You!