Advanced Graph Algorithms

Conf. dr. ing. Guillaume Ducoffe

guillaume.ducoffe@fmi.unibuc.ro

Sets in Graphs

- The families of all modules in a graph G admits a representation as a tree wth $\mathcal{O}(n)$ nodes.

- Since there are 2^n possible vertex subsets, there are 2^{2^n} possible families of sets on a graph G. In particulsr, not all such families can be encoded using only $\mathcal{O}(n)$ bits (nor even $\mathcal{O}(n^c)$ bits, for some $c > 1$).

- Objective: identify the families of sets which admit an efficient (tree) encoding. Deduce from the latter new graph decompositions.

Partitive families

A family $\mathcal{F}$ of subsets of V is called **partitive** if:

- $\emptyset, V \in \mathcal{F}$;
- for all $v \in V$, $\{v\} \in \mathcal{F}$;
- for all $X, Y \in \mathcal{F}$ such that X, Y overlap:
  - $X \cap Y \in \mathcal{F}$;
  - $X \cup Y \in \mathcal{F}$;
  - $X \setminus Y \in \mathcal{F}$;
  - $Y \setminus X \in \mathcal{F}$;
  - $X \Delta Y \in \mathcal{F}$.

Reminder: Modules form a partitive family.

Rooted tree-like families

A family $\mathcal{F}$ of subsets of V is called **rooted tree-like** if:

- $\emptyset, V \in \mathcal{F}$;
- for all $v \in V$, $\{v\} \in \mathcal{F}$;
- for all $X, Y \in \mathcal{F}$, X and Y do not overlap (*i.e.*, they are disjoint or comparable for inclusion).

Rooted tree-like families are partitive, but the converse is false in general.

Proposition: every rooted tree-like family has an inclusion tree, where the nodes represent sets of $\mathcal{F}$. In particular, V is the root, the leaves are $\{v\}$ for every $v \in V$, and for every $X \neq V$, its father is the smallest Y such that $X \subset Y$.

Strong members

Definition

A strong member X of a partitive family $\mathcal{F}$ is a set which does not overlap any other set $Y \in \mathcal{F}$.

Remark: the sets $\emptyset, V$ and $\{v\},\ v \in V$ are strong.

The notion of strong members generalises that of strong modules.

Proposition: the family of strong members of $\mathcal{F}$ is rooted tree-like (and so, it admits an inclusion tree).

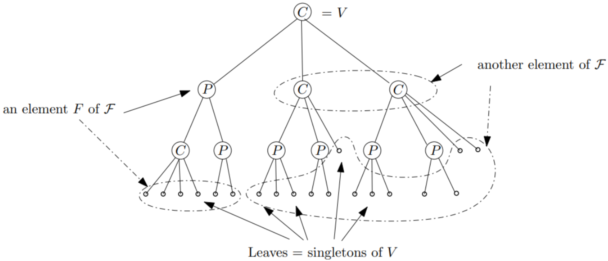
Decomposition theorem

Theorem

For every partitive family $\mathcal{F}$, the nodes in the inclusion tree T of its strong members can be labelled either complete or prime, in such a way that:

- *The union of any children of a complete node must be in $\mathcal{F}$.*
- *Every not strong member of $\mathcal{F}$ is the union of some children of a complete node.*

Consequence: the modular decomposition tree can be generalized to any partitive family $\mathcal{F}$. It is sometimes called the representative tree of $\mathcal{F}$.



Computation of the representative tree

Theorem

For every rooted tree-like family $\mathcal{F}$, its inclusion tree can be computed in $\mathcal{O}\left(\sum_{X \in \mathcal{F}} |X|\right) =^{def} \mathcal{O}(\|\mathcal{F}\|)$ time.

1) We compute $|X|$ for every $X \in \mathcal{F}$.

2) We sort the sets of $\mathcal{F}$ by nondecreasing size. Using counting sort, it can be done in $\mathcal{O}(|V| + |\mathcal{F}|)$ time.

3) For every $v \in V$, we compute the ordered family $\mathcal{F}_v$ of all sets that contain v, ordered by decreasing size. Then, for every $X \in \mathcal{F}_v$ such that $X \neq V$, its father in the inclusion tree must be the set $Y \in \mathcal{F}_v$ that comes immediately before X.

Corollary

For every partitive family $\mathcal{F}$, its representative tree can be computed in $\mathcal{O}(\|\mathcal{F}\|^2)$ time.

Orthogonal family

Notation: $X \perp Y$ if X, Y do not overlap.

Definition

For any family $\mathcal{F}$, we define $\mathcal{F}^{\perp} = \{Y \mid \forall X \in \mathcal{F},\ X \perp Y\}$.

Remark: If $\mathcal{F}$ is partitive, then we always have $\mathcal{F} \cap \mathcal{F}^{\perp} \neq \emptyset$, because of the strong members of $\mathcal{F}$.

Theorem

For every $\mathcal{F}$ (not necessarily partitive), $\mathcal{F}^{\perp}$ is partitive.

Furthermore, if $\mathcal{F}$ is also partitive, then $(\mathcal{F}^{\perp})^{\perp} = \mathcal{F}$. The representation trees of $\mathcal{F}$ and $\mathcal{F}^{\perp}$ are the same, up to switching the complete and prime labels at every node.

Strong members of $\mathcal{F}^\perp$

- The overlap graph of $\mathcal{F}$ is the graph with vertex $\mathcal{S}$ and an edge XY for every overlapping sets X and Y.
- An overlap component is a connected component of the overlap graph.
- A block is a set of vertices which are exactly in the same sets X of some overlap component.

Theorem (McConnell, 2004)

The strong members of $\mathcal{F}^\perp$ are exactly:

- *V and $\{v\}$ for all $v \in V$;*
- *$\bigcup \mathcal{C}$ for every overlap component $\mathcal{C}$;*
- *The blocks in $\bigcup \mathcal{C}$ for every overlap component $\mathcal{C}$.*

Computation of overlap components

- The overlap graph may have $\mathcal{O}(|\mathcal{F}|^2)$ edges. Nevertheless, we can compute a graph on $\mathcal{F}$ with much few edges and the same connected components!

- In what follows, let $\mathcal{F} = (X_1, X_2, \ldots, X_k)$ be totally ordered so that $|X_1| \leq |X_2| \leq \ldots \leq |X_k|$.

Definition

For every i, let $MAX(X_i) = X_j$ be such that:

- $j < i$;
- X_i, X_j overlap;
- for all $t < j$, $X_i \perp X_t$.

Proposition: For every Y such that $Y \cap X \neq \emptyset$ and $|X| \leq |Y| \leq MAX(X)$, Y must overlap either X or $MAX(X)$.

Computation of $MAX(X)$

1) We initialize a **partition refinement** data structure, with one group equal to V.

2) (**new**) all groups in the partition, past and present, are nodes of some rooted tree T. The root of the tree corresponds to V, the initial group. The leaves represent groups of the current partition.

3) We consider each set X_j, $j = k \ldots 1$ sequentially. We refine according to X_j. For every group V_r in the data structure, if V_r and X_j overlap, then we create new nodes $V_r \cap X_j$ and $V_r \setminus X_j$ as children of V_r. **We label the edges between V_r and its children by X_j**

Computation of $MAX(X)$

1) We initialize a **partition refinement** data structure, with one group equal to V.

2) (**new**) all groups in the partition, past and present, are nodes of some rooted tree T. The root of the tree corresponds to V, the initial group. The leaves represent groups of the current partition.

3) We consider each set $X_j,\ j = k \dots 1$ sequentially. We refine according to X_j. For every group V_r in the data structure, if V_r and X_j overlap, then we create new nodes $V_r \cap X_j$ and $V_r \setminus X_j$ as children of V_r. **We label the edges between V_r and its children by X_j**

Proposition: $MAX(X_j)$ exists if and only if, before we consider set X_j, not all the elements in X_j are contained in the same group of the partition.

$\rightarrow$ If we compute the LCA of all groups that intersect X_j, then $MAX(X_j)$ must label the edges toward its two children.

Definition of an auxiliary graph

1) For every $v \in V$, let $\mathcal{F}_v = (X_1^v, X_2^v, \ldots, X_q^v)$ be the family of all sets in $\mathcal{F}$ containing v (ordered by nondecreasing size).

2) Compute by dynamic programming
$m_i^v = \max\{|MAX(X_j^v)| \mid 1 \leq j \leq i\}$.

3) If $|X_{i+1}^v| \leq m_i^v$, then add an edge $X_i^v X_{i+1}^v$.

Proposition: the connected components of the resulting graph are exactly the overlap components.

Definition of an auxiliary graph

1) For every $v \in V$, let $\mathcal{F}_v = (X_1^v, X_2^v, \dots, X_q^v)$ be the family of all sets in $\mathcal{F}$ containing v (ordered by nondecreasing size).

2) Compute by dynamic programming
$m_i^v = \max\{|MAX(X_j^v)| \mid 1 \le j \le i\}$.

3) If $|X_{i+1}^v| \le m_i^v$, then add an edge $X_i^v X_{i+1}^v$.

Proposition: the connected components of the resulting graph are exactly the overlap components.

Theorem (Dahlaus, 2000)

For every $\mathcal{F}$, the overlap components can be computed in $\mathcal{O}(\|\mathcal{F}\|)$ time.

Optimal computation of the representative tree

Proposition: Being given an overlap component of $\mathcal{C}$, we can compute all its blocks in $\mathcal{O}(\|\mathcal{C}\|)$ time.

Proof: we compute twins in the incidence graph, with respective partite sets $\mathcal{C}$ and $\bigcup \mathcal{C}$.

Consequence: By McConnell's theorem, all strong elements of $\mathcal{F}^\perp$ can be computed in $\mathcal{O}(\|\mathcal{F}\|)$ time.

Reminder: the inclusion tree of a rooted tree-like family can be computed in linear time.

Theorem

We can compute the representative tree of $\mathcal{F}^\perp$ in $\mathcal{O}(\|\mathcal{F}\|)$ time.

Weakly partitive families

A family $\mathcal{F}$ of subsets of V is called **weakly partitive** if:

- $\emptyset, V \in \mathcal{F}$;
- for all $v \in V$, $\{v\} \in \mathcal{F}$;
- for all $X, Y \in \mathcal{F}$ such that X, Y overlap:
  - $X \cap Y \in \mathcal{F}$;
  - $X \cup Y \in \mathcal{F}$;
  - $X \setminus Y \in \mathcal{F}$;
  - $Y \setminus X \in \mathcal{F}$;
  - $\cancel{X \Delta Y \in \mathcal{F}}$.

<u>Remark</u>: partitive = weakly partitive + closed under symmetric difference

Decomposition theorem

Theorem (Chein et al.)

If $\mathcal{F}$ is weakly partitive, then the nodes in the inclusion tree T of its strong members can be labelled either complete, prime or linear, so that:

- *Any union of children of a complete node is in $\mathcal{F}$;*
- *Any union of consecutive children of a linear node is also in $\mathcal{F}$;*
- *Every non strong member of $\mathcal{F}$ can be represented as such a union of (consecutive) children.*

A representative tree may still exist under fewer properties (*e.g.*, union-difference), but its number of nodes becomes $\mathcal{O}(n^2)$.

From sets to bipartitions

- The theory goes on for the families of **bipartitions** $(X, V \setminus X)$ on V.

- Two bipartitions $(X, V \setminus X)$ and $(Y, V \setminus Y)$ overlap if the four subsets $X \cap Y, X \setminus Y, Y \setminus X$ and $V \setminus (X \cup Y)$ are nonempty.

Remark: this is a stronger condition than just having X, Y overlapping.

- Let $\mathcal{F}$ be a family of bipartitions. A **bipartition tree** is a tree T whose leaves are labelled by V and such that, for every $(X, V \setminus X) \in \mathcal{F}$, there is an edge $e \in E(T)$ such that the leaf-sets in the two components of $T - e$ are exactly $X, V \setminus X$. Conversely, every edge of T can be associated to some bipartition of $\mathcal{F}$.

$\rightarrow$ equivalent of the inclusion tree.

Unrooted tree-like families

Definition

A family $\mathcal{F}$ of bipartitions on V is called unrooted tree-like if the following conditions hold:

- $(\emptyset, V) \notin \mathcal{F}$;
- for all $v \in V$, $(\{v\}, V \setminus \{v\}) \in \mathcal{F}$;
- Two bipartitions of $\mathcal{F}$ do not overlap.

Proposition: every unrooted tree-like family admits a bipartition tree.

Bipartitive families

Definition

A family $\mathcal{F}$ of bipartitions on V is called bipartitive if the following conditions hold:

- $(\emptyset, V) \notin \mathcal{F}$;
- for all $v \in V$, $(\{v\}, V \setminus \{v\}) \in \mathcal{F}$;
- for all overlapping $(X, V \setminus X), (Y, V \setminus Y) \in \mathcal{F}$:
  - $(X \cap Y, V \setminus (X \cap Y)) \in \mathcal{F}$;
  - $(X \cup Y, V \setminus (X \cup Y)) \in \mathcal{F}$;
  - $(X \setminus Y, (V \setminus X) \cup Y) \in \mathcal{F}$;
  - $(Y \setminus X, (V \setminus Y) \cup X) \in \mathcal{F}$;
  - $(X \Delta Y, V \setminus (X \Delta Y)) \in \mathcal{F}$.

The strong members of $\mathcal{F}$ are unrooted tree-like, and we can label nodes in their bipartition tree either complete or prime, with similar meaning as for partitive families.

Splits

Definition

A **split** is a bipartition $(X, V \setminus X)$ such that the cut $E(X, V \setminus X)$ induces a complete bipartite subgraph.

Remark: If M is a module, then $(M, V \setminus M)$ is a split.

Split decomposition

Theorem (Cunningham, 1982)

The family of all splits in a graph is bipartitive.

The representative bipartition tree of strong splits is called the **split decomposition tree**.

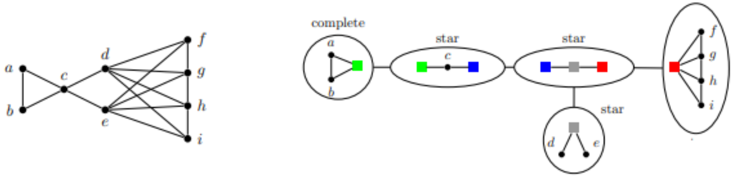


Figure 3: A graph and its split decomposition.

To every node, we associate a *split component*: where we add new nodes to represent incident strong splits. A split component is either a clique, a star, or prime for split decomposition.

Computation of a split

- For an edge xy, we try to compute a split $(X, V \setminus X)$ such that $x \in X, y \notin X$.

- The edges of the split (if it exists) must have their respective ends in $N(x), N(y)$. By removing all edges of $E \cap (N(x) \times N(y))$, we obtain connected components $C_1, \ldots, C_q$. Wlog $x \in C_1,\ y \in C_q$.

- We must have $C_1 \subseteq X, C_q \cap X = \emptyset$. In particular, we check whether $E(C_1, C_q)$ induces a complete bipartite subgraph.

- We construct a graph H whose vertices are $c_1, c_2, \ldots, c_q$. For every $1 \leq i < j \leq q$, if $E(C_i, C_j)$ does not incude a complete bipartite graph, then C_i, C_j must be on the same partite set. We add an edge $c_i c_j$.

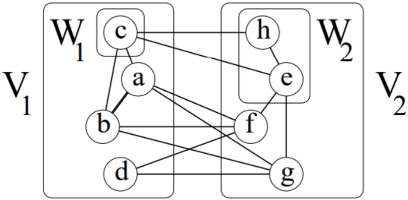
- We are done computing the connected components of H.

Bijoins

Observation: If $(X, V \setminus X)$ is a split of G, then in general it is not a split of its complement $\overline{G}$.

Definition

A **bijoin** is a bipartition $(X, V \setminus X)$ such that `(i)` $N(X) = V \setminus X$, and `(ii)` the cut $E(X, V \setminus X)$ induces the disjoint union of two complete bipartite subgraphs.



Proposition: $(X, V \setminus X)$ is a bijoin of G if and only if it is a bijoin of $\overline{G}$.

Bijoin decomposition

Theorem (de Montgolfier & Rao)

The family of all bijoins of a graph is bipartitive.

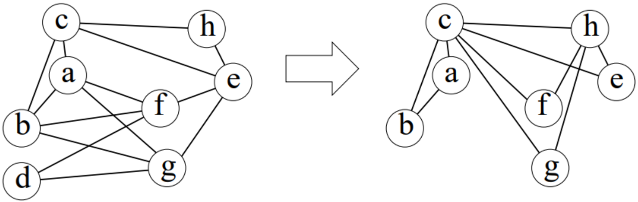
The representative bipartition tree of strong bijoins is called the **bijoin decomposition tree**.

Seidel switch

Definition

The Seidel switch of a graph G with respect to W is the graph obtained from G by removing all edges from the cut $E(W, V \setminus W)$, and replacing them with $\overline{E}(W, V \setminus W)$ (cut in the complement $\overline{G}$).

The <u>Seidel reduction</u> $\tilde{G}_v$ is the graph obtained from the Seidel switch with respect to $N(v)$ by removing vertex v.



Fundamental bijoin lemma

Lemma

Let $(X, V \setminus X)$ be an arbitrary bipartition and let $x \in X$. Then, $(X, V \setminus X)$ is a bijoin of G if and only if $V \setminus X$ is a module of $\tilde{G}_x$.

Apply the lemma to some x such that $d(x) \leq 2m/n$.

Theorem

The bijoin decomposition of any graph can be computed in $\mathcal{O}(n + m)$ time.

The same is true for split decompositon, but the algorithm is much more intricate:

Theorem

The split decomposition tree of any graph can be computed in $\mathcal{O}(n + m)$ time.

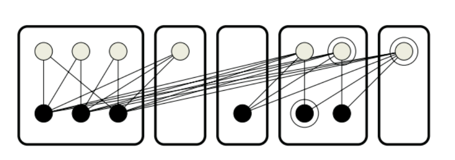
Bimodules

Observation: modules in a bipartite graphs are always twin classes...

Definition

Let $B = (V_0 \cup V_1, E)$ be a bipartite graph. A bimodule is a pair (M_0, M_1) such that the following holds for $i \in \{0, 1\}$:

- $M_i \subseteq V_i$;
- for every $x_i, y_i \in M_i$, $N(x_i) \setminus M_{1-i} = N(y_i) \setminus M_{1-i}$.



There are canonical bimodules and degenerate ones. The canonical bimodules form a weakly bipartitive family. $\Longrightarrow$ **bimodular decomposition**

2-Modules

Definition

M is a 2-module of G if there exists a bipartition (M_0, M_1) of M such that, for $i \in \{0, 1\}$, M_i is a module of $G \setminus M_{i-1}$.

Special cases of 2-modules:

- modules
- splits
- bijoins
- bimodules

Unfortunately, there is no more a nice tree-like structure for representing all 2-modules in a graph.

Questions