# Enumeration of Minimal Dominating Sets and Variants

Mamadou M. Kanté, V. Limouzy, A. Mary, L. Nourine

LIMOS - Université Blaise Pascal
GROW 2011

# Enumeration Algorithms

**Definition (Generation Problem)**

**Input :** A finite discrete structure $\mathfrak{A}$ and a predicate $P$ over $\mathfrak{A}$.
**Output :** The set $P(\mathfrak{A})$ of elements of $\mathfrak{A}$ which satisfy $P$.

## Enumeration Algorithms

**Definition (Generation Problem)**

**Input** : A finite discrete structure $\mathfrak{A}$ and a predicate $P$ over $\mathfrak{A}$.
**Output** : The set $P(\mathfrak{A})$ of elements of $\mathfrak{A}$ which satisfy $P$.

- Maximal cliques
- Maximal independent sets
- Bases of a matroid
- Minimal vertex-covers
- Triangulations
- ...

# Complexity

The set $P(\mathfrak{A})$ is usually large compared to $\mathfrak{A}$.

**Time complexity :**

- Output polynomial : $O\left(poly(|\mathfrak{A}|, |P(\mathfrak{A})|)\right)$.
- Polynomial Delay : The delay between two consecutive outputs is $O\left(poly(|\mathfrak{A}|)\right)$.

**Space complexity :**

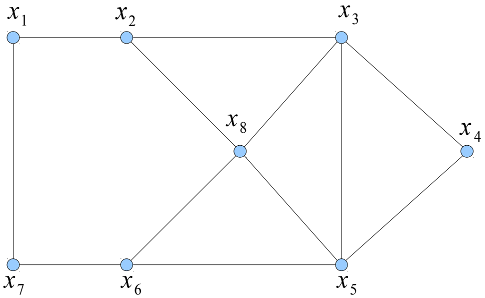
- Polynomial : $O(poly(|\mathfrak{A}|))$

# Dominating Set

- $D \subseteq V$ is a dominating set of $G$ if for all $y \in V$, $N[y] \cap D \neq \emptyset$.

# Dominating Set

- $D \subseteq V$ is a dominating set of $G$ if for all $y \in V$, $N[y] \cap D \neq \emptyset$.



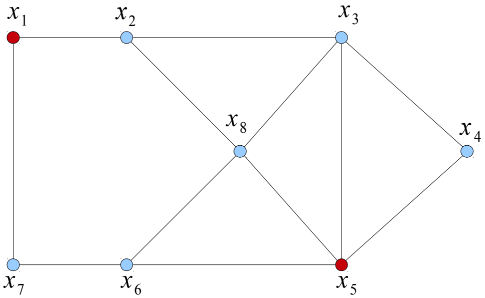

$\{x_1, x_5, x_3\}$

# Dominating Set

- $D \subseteq V$ is a dominating set of $G$ if for all $y \in V$, $N[y] \cap D \neq \emptyset$.
- $D$ dominating set is minimal if no $D' \subseteq D$ is dominating.

# Dominating Set

- $D \subseteq V$ is a dominating set of $G$ if for all $y \in V$, $N[y] \cap D \neq \emptyset$.
- $D$ dominating set is minimal if no $D' \subseteq D$ is dominating.



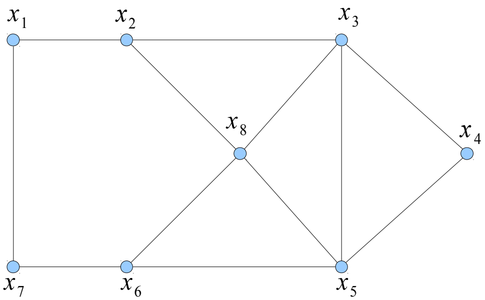

$\{x_1, x_5\}$

# Dominating Set

- $D \subseteq V$ is a dominating set of $G$ if for all $y \in V$, $N[y] \cap D \neq \emptyset$.
- $D$ dominating set is minimal if no $D' \subseteq D$ is dominating.
- DOM problem : Enumerate the set $\mathcal{D}(G)$ of minimal dominating sets of $G$.

# Dominating Set

- $D \subseteq V$ is a dominating set of $G$ if for all $y \in V$, $N[y] \cap D \neq \emptyset$.
- $D$ dominating set is minimal if no $D' \subseteq D$ is dominating.
- DOM problem : Enumerate the set $\mathcal{D}(G)$ of minimal dominating sets of $G$.

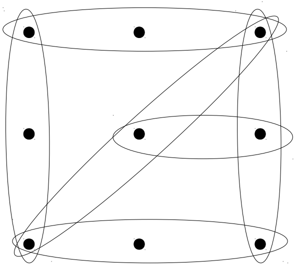



$\{x_1, x_5\}, \{x_7, x_2, x_5\}, \{x_8, x_7, x_5\}, \{x_5, x_2, x_6\}, \{x_4, x_8, x_7\}, \{x_4, x_8, x_1\}, \{x_4, x_2, x_6\}, \{x_4, x_6, x_1\}$

...

# Dominating Set and Transversal

- A hypergraph $\mathcal{H}$ is a couple $(V, \mathcal{E})$ where $V$ is called vertex set and $\mathcal{E} \subseteq 2^V$ is the hyperhedge set.
- Hypergraphs generalise graphs.

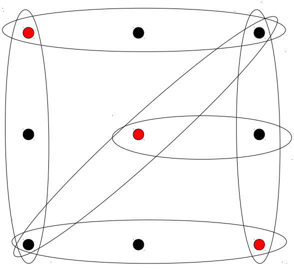


# Dominating Set and Transversal

- A hypergraph $\mathcal{H}$ is a couple $(V, \mathcal{E})$ where $V$ is called vertex set and $\mathcal{E} \subseteq 2^V$ is the hyperhedge set.
- Hypergraphs generalise graphs.
- $T \subseteq 2^V$ is a Transversal of $\mathcal{H}$ if $\forall E \in \mathcal{E},\ T \cap E \neq \emptyset$.

# Dominating Set and Transversal

- A hypergraph $\mathcal{H}$ is a couple $(V, \mathcal{E})$ where $V$ is called vertex set and $\mathcal{E} \subseteq 2^V$ is the hyperhedge set.
- Hypergraphs generalise graphs.
- $T \subseteq 2^V$ is a Transversal of $\mathcal{H}$ if $\forall E \in \mathcal{E},\ T \cap E \neq \emptyset$.



# Dominating Set and Transversal

- A hypergraph $\mathcal{H}$ is a couple $(V, \mathcal{E})$ where $V$ is called vertex set and $\mathcal{E} \subseteq 2^V$ is the hyperhedge set.
- Hypergraphs generalise graphs.
- $T \subseteq 2^V$ is a Transversal of $\mathcal{H}$ if $\forall E \in \mathcal{E},\ T \cap E \neq \emptyset$.
- The TransHyp problem is the enumeration of $tr(\mathcal{H})$.

## Dominating Set and Transversal

- A hypergraph $\mathcal{H}$ is a couple $(V, \mathcal{E})$ where $V$ is called vertex set and $\mathcal{E} \subseteq 2^V$ is the hyperhedge set.
- Hypergraphs generalise graphs.
- $T \subseteq 2^V$ is a Transversal of $\mathcal{H}$ if $\forall E \in \mathcal{E}$, $T \cap E \neq \emptyset$.
- The TransHyp problem is the enumeration of $tr(\mathcal{H})$.
- For $G$, $\mathcal{N}(G) = (V(G), \{N[x] \mid x \in V(G)\})$.

# Dominating Set and Transversal

- A hypergraph $\mathcal{H}$ is a couple $(V, \mathcal{E})$ where $V$ is called vertex set and $\mathcal{E} \subseteq 2^V$ is the hyperhedge set.
- Hypergraphs generalise graphs.
- $T \subseteq 2^V$ is a Transversal of $\mathcal{H}$ if $\forall E \in \mathcal{E}$, $T \cap E \neq \emptyset$.
- The TransHyp problem is the enumeration of $tr(\mathcal{H})$.
- For $G$, $\mathcal{N}(G) = (V(G), \{N[x] \mid x \in V(G)\})$.

**Theorem (Folklore)**

$$\mathcal{D}(G) = tr(\mathcal{N}(G)).$$

# TransHyp Problem

- Well-studied problem because of its connection to several problems :
  - datamining where frequently occurring patterns are of interest
  - Dualisation of monotone Boolean formulas.
  - ...
- However, no known polynomial-time algorithm. Best is the following.

**Theorem (Fredman, Khachiyan)**

*The TransHyp problem is quasi polynomial, i.e., there exists an $O(n^{O(\log(n))})$-time algorithm where $n = |tr(\mathcal{H})| + |\mathcal{H}|$.*

- But ...

# Back to Dominating Sets

- $\mathcal{D}(G)$ in constant-delay if $G \in CWD(\leq k)$.
- $\mathcal{D}(G)$ in polynomial-delay if $G$ is strongly-chordal.
  - directed path-graphs, chordal comparability graphs, ...
- $\mathcal{D}(G)$ in polynomial-delay if $G$ is degenerate.
  - planar graphs, bounded degree graphs, bounded genus graphs, ...
- $\mathcal{D}(G)$ in polynomial-delay if $G$ is a split graph.
- ...

## Back to Dominating Sets

- $\mathcal{D}(G)$ in constant-delay if $G \in CWD(\leq k)$.
- $\mathcal{D}(G)$ in polynomial-delay if $G$ is strongly-chordal.
  - directed path-graphs, chordal comparability graphs, ...
- $\mathcal{D}(G)$ in polynomial-delay if $G$ is degenerate.
  - planar graphs, bounded degree graphs, bounded genus graphs, ...
- $\mathcal{D}(G)$ in polynomial-delay if $G$ is a split graph.
- ...

**Question**

Can we expect a polynomial-time algorithm for the DOM problem ?

## Back to Dominating Sets

- $\mathcal{D}(G)$ in constant-delay if $G \in CWD(\leq k)$.
- $\mathcal{D}(G)$ in polynomial-delay if $G$ is strongly-chordal.
  - directed path-graphs, chordal comparability graphs, ...
- $\mathcal{D}(G)$ in polynomial-delay if $G$ is degenerate.
  - planar graphs, bounded degree graphs, bounded genus graphs, ...
- $\mathcal{D}(G)$ in polynomial-delay if $G$ is a split graph.
- ...

**Question**

Can we expect a polynomial-time algorithm for the DOM problem ?
NO unless one for TransHyp.

# TransHyp to DOM

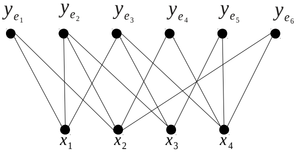
**Proposition**

No function $f$ such that for each hypergraph $\mathcal{H}$, $tr(\mathcal{H}) = \mathcal{D}(f(\mathcal{H}))$.

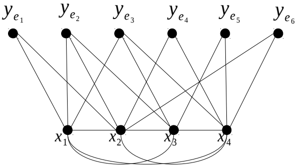
**Theorem**

*If there is an output polynomial-time algorithm for the DOM problem, then there exists one for the TransHyp problem.*

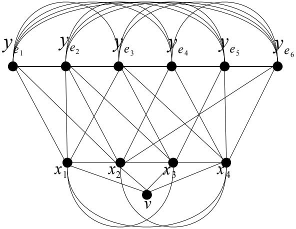
# TransHyp to DOM




$I(H)$



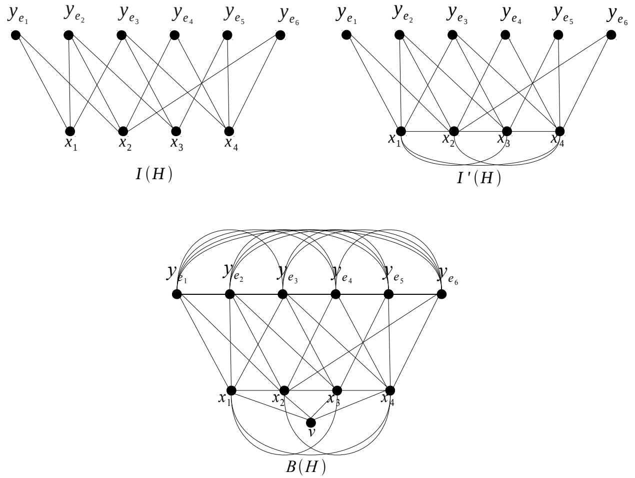

$I'(H)$




$B(H)$

# TransHyp to DOM

- $\mathcal{D}(\mathcal{B}(\mathcal{H})) = tr(\mathcal{H}) \cup \{\{x, y_e\}\}$.



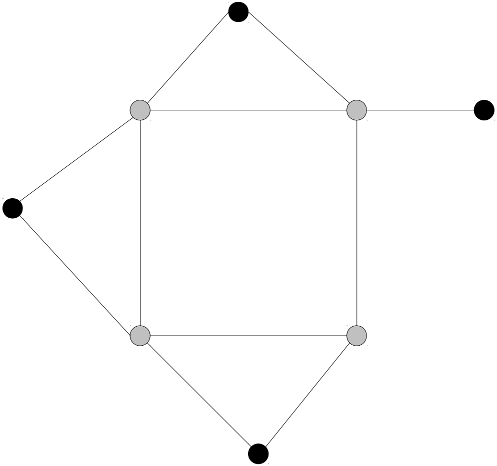
# Completion

- A vertex $x \in V(G)$ is said to be irredundant if for all $y \neq x$, $N[y] \not\subseteq N[x]$. Otherwise it is called redundant.
- The set of irredundant(resp. redundant) vertices is denoted by $M(G)$ (resp. $RN(G)$).

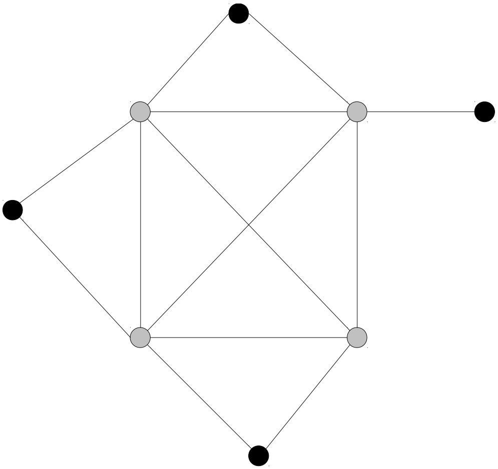
**Definition**

The completion graph of $G$ is the graph $G_{co}$ with vertex set $V(G)$ and edge set $E(G) \cup \{xy \mid x, y \in RN(G), x \neq y\}$

# Completion



# Completion



# Completion

**Proposition**

$D$ is a minimal dominating set of $G$ iff $D$ is a minimal dominating set of $G_{co}$, *i.e.*, $\mathcal{D}(G) = \mathcal{D}(G_{co})$

# Completion

**Proposition**

$D$ is a minimal dominating set of $G$ iff $D$ is a minimal dominating set of $G_{co}$, *i.e.*, $\mathcal{D}(G) = \mathcal{D}(G_{co})$

**Proposition**

$G_{co}$ is chordal $\Leftrightarrow$ $G_{co}$ is a split graph.

# Completion

**Proposition**

$D$ is a minimal dominating set of $G$ iff $D$ is a minimal dominating set of $G_{co}$, *i.e.*, $\mathcal{D}(G) = \mathcal{D}(G_{co})$

**Proposition**

$G_{co}$ is chordal $\quad\Leftrightarrow\quad$ $G_{co}$ is a split graph.

Corollary : $\mathcal{D}(G)$ is polynomial-delay on $P_6$-free chordal graphs.

# Other Reductions

- Sometimes, it is better to remove edges/vertices.
- $x$ and $y$ d-equivalent if $N[x] \cap M(G) = N[y] \cap M(G)$.

# Other Reductions

- Sometimes, it is better to remove edges/vertices.
- $x$ and $y$ d-equivalent if $N[x] \cap M(G) = N[y] \cap M(G)$.

## Properties

- $x$ and $y$ d-equivalent iff $\mathcal{D}_x(G) = \{D \cup \{x\} \setminus y \mid D \in \mathcal{D}_y(G)\}$.

# Other Reductions

- Sometimes, it is better to remove edges/vertices.
- $x$ and $y$ d-equivalent if $N[x] \cap M(G) = N[y] \cap M(G)$.

**Properties**

- $x$ and $y$ d-equivalent iff $\mathcal{D}_x(G) = \{D \cup \{x\} \setminus y \mid D \in \mathcal{D}_y(G)\}$.
- $\overline{\mathcal{D}_x(G)} = \mathcal{D}(G \setminus x)$ for $x \in RN(G)$.

# Other Reductions

- Sometimes, it is better to remove edges/vertices.
- $x$ and $y$ d-equivalent if $N[x] \cap M(G) = N[y] \cap M(G)$.

**Properties**

- $x$ and $y$ d-equivalent iff $\mathcal{D}_x(G) = \{D \cup \{x\} \setminus y \mid D \in \mathcal{D}_y(G)\}$.
- $\overline{\mathcal{D}_x(G)} = \mathcal{D}(G \setminus x)$ for $x \in RN(G)$.
- The partition $\{M(G), RN(G)\}$ is preserved if we delete all irredundant vertices (except one if a class contains only irredundant vertices).

# Related problems

- The enumeration of dominating sets containing a set is coNP-complete.
- The enumeration of total dominating sets is equivalent to the TransHyp problem.
- We have proved that the connected dominating sets is at least as hard as the TransHyp problem
  - In some graph classes the two problems are equivalent : chordal graphs, cobipartite graphs, etc...
  - In some graph classes, the TransHyp problem is harder : graphs with polynomial number of minimal separators.
- Directions :
  - Identify new graph classes where DOM is polynomial.
  - Deepen these notions of reductions.

Thank you ! !