

On the enumeration of signatures of XOR formulas

Nadia Creignou, Oscar Defrain,
Frédéric Olive, and Simon Vilmin
LIS, Aix-Marseille Université, France

WADS 2025
Toronto, Canada
July 12th




Enumeration problems

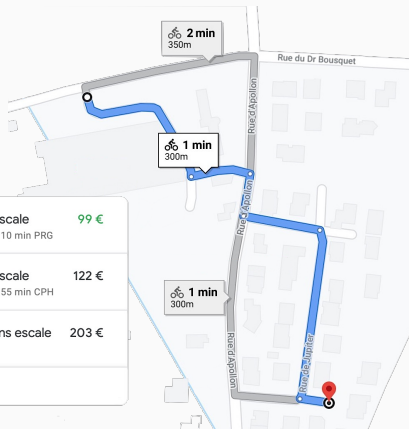
Typical question:

Given *input I*, list all *solutions in I*

Examples:

- *paths* to a *destination*
- *flights* to a *city*
- *answers* to a *query*

	20:10 – 08:35 ⁺¹ CSA · Smartwings	12 h 25 min CDG–WAW	1 escale 9 h 10 min PRG	99 €
	14:30 – 22:30 SAS	8 h 0 min CDG–WAW	1 escale 4 h 55 min CPH	122 €
	13:00 – 15:20 Air France	2 h 20 min CDG–WAW	Sans escale	203 €
▼	114 autres vols			



Two perspectives about complexity

Input-sensitive: in terms of **input** size

Theorem (Moon & Moser, IJM 65)

*There is an $O(3^{n/3})$ -time algorithm enumerating all the **maximal cliques** of a **n -vertex graph**.*

→ *basically upper-bounds the number of objects*

Output-sensitive: in terms of **input** + **output** size

Theorem (Tsukiyama et al., SICOMP 77)

*There is a $O(n + m + d)$ -time algorithm enumerating all the **d maximal cliques** of a **n -vertex m -edge graph**.*

→ *many techniques (reverse search, backtrack search, saturations algorithms, ordered generation, etc.)*

Efficiency for the output-sensitive approach

Let n be input size, e.g., number of vertices of a graph

Let d be the output size, e.g., number¹ of max. cliques

execution time



output-polynomial
stops in $\text{poly}(n + d)$ time



incremental-polynomial
outputs i^{th} solution in $\text{poly}(n + i)$ time



solution output

polynomial delay
 $\text{poly}(n)$ time between consec. outputs

¹For simplicity as solutions are of poly size

Definitions (1)

- variable set $V = \{x_1, \dots, x_n\}$
- **literal**: variable x_i or its negation $\overline{x_i}$
- **clause**: disjunction $C = \ell_1 \vee \dots \vee \ell_k$ of literals
- CNF: conjunction of clauses $\phi = C_1 \wedge \dots \wedge C_m$

Example:

$$\phi := (x_1 \vee x_2)(x_2 \vee x_3)(\overline{x_1} \vee \overline{x_3})$$

Definitions (2)

- **assignment**: function $\mathbf{a}: V \rightarrow \{0, 1\}$
- we note $C_j(\mathbf{a}) = 1$ if \mathbf{a} evaluates C_j to 1, 0 otherwise
- **signature** produced by \mathbf{a} : binary sequence
 $\sigma(\mathbf{a}) = (C_1(\mathbf{a}), \dots, C_m(\mathbf{a}))$
- $\sigma \leq \sigma'$: if $\sigma[j] \leq \sigma'[j]$, $\forall 1 \leq j \leq m$

Example:

$$\phi := (x_1 \vee x_2)(x_2 \vee x_3)(\overline{x_1} \vee \overline{x_3})$$

$$\mathbf{a} := \{x_1 \mapsto 1, x_2 \mapsto 0, x_3 \mapsto 0\}$$

$$\sigma(\mathbf{a}) = 101$$

$$\text{SIG}(\phi) = \{001, 011, 101, 110, 111\}$$

all signatures

min max

Observation

A formula admits **one maximal signature** iff it is **satisfiable**

Problems

Signatures Enumeration (Sig•Enum)

input: a formula ϕ

output: the set $\text{SIG}(\phi)$ of all signatures

Minimal Signatures Enumeration (MaxSig•Enum)

output: the set $\min_{\leq} \text{SIG}(\phi)$

Maximal Signatures Enumeration (MinSig•Enum)

output: the set $\max_{\leq} \text{SIG}(\phi)$

These problems were first stated and motivated during the Dagstuhl seminar 19211 on enumeration in data management which took place in 2019

Problems

Signatures Enumeration (Sig•Enum)

input: a formula ϕ

output: the set $\text{SIG}(\phi)$ of all signatures

Minimal Signatures Enumeration (MaxSig•Enum)

output: the set $\min_{\leq} \text{SIG}(\phi)$

Maximal Signatures Enumeration (MinSig•Enum)

output: the set $\max_{\leq} \text{SIG}(\phi)$

Theorem (Berczi et al., TCS 2021)

- Sig•Enum can be solved in *inc-poly* time for $O(1)$ -CNF's
- MinSig•Enum can be solved with *poly delay* for any CNF
- MaxSig•Enum cannot be solved in *output-poly* time if $P \neq NP$

Open question

Question (Berczi et al., TCS 2021)

*What is the status of **MaxSig·Enum** for tractable² formulas?*

Theorem (Schaefer's dichotomy theorem, STOC 78)³

Non-trivial classes of tractable formulas are precisely

- *2-CNF's*
- *Horn-CNF's (and their dual)*
- *XOR formulas*

Also posed as open problems in the WEPA 2022 workshop

This talk addresses the latter case

²Admitting a poly-time satisfiability check

³Rough reformulation

XOR formulas

XOR formulas differ from CNF's:

- **eXclusive or** operator: \oplus which is associative
- **XOR clause**: $(\ell_1 \oplus \cdots \oplus \ell_k)$ where $\ell_j = x_j$ or $\ell_j = \overline{x_j}$, $\forall j$
- XOR formula: conjunction of XOR clauses

Thus the results of Berczi et al. do not directly apply

Observation

A clause may be seen as an eq. $x_1 + \cdots + x_k = \varepsilon$, $\varepsilon \in \{0, 1\}$

A XOR formula may be seen as a system of equations in \mathbb{F}_2

We consider this formulation form now on:

- variable are no longer negated
- clauses are of two types: even ($\varepsilon = 0$) or odd ($\varepsilon = 1$)

Contributions

Lemma⁴

*On XOR formulas **MinSig·Enum** is equivalent to **MaxSig·Enum***

Theorem (Creignou, D., Olive, and Vilmin)

On XOR formulas:

- *Sig·Enum can be solved with **poly delay***
- *Min/MaxSig·Enum can be solved in **inc-poly** time*
- *Min/MaxSig·Enum can be solved with **poly delay** when restricted to clauses of size at most two*

⁴Crucial distinction with CNF's

Properties (1)

If $C_i = (x_1 + \dots + x_k = \varepsilon_i)$ let $\overline{C}_i := (x_1 + \dots + x_k = 1 - \varepsilon_i)$

For ϕ a m -clause formula, $A, B \subseteq \{1, \dots, m\}$:

$$\phi(A, B) := \left(\bigwedge_{i \in A} C_i \right) \wedge \left(\bigwedge_{j \in B} \overline{C}_j \right)$$

For σ is a signature:

- $1(\sigma) := \{j : \sigma[j] = 1\}$
- $0(\sigma) := \{j : \sigma[j] = 0\}$

Lemma⁵

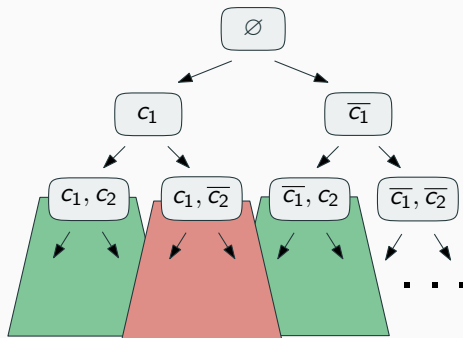
σ is a *signature* of ϕ iff $\phi(1(\sigma), 0(\sigma))$ is *satisfiable*

⁵True for CNF's as well

Flashlight search

General idea:

- determine the value of clauses one by one
- recursively call for each value
- at **the bottom of the recursion**: a signature is determined
- **do not explore subtree if no solution lies in descendants**



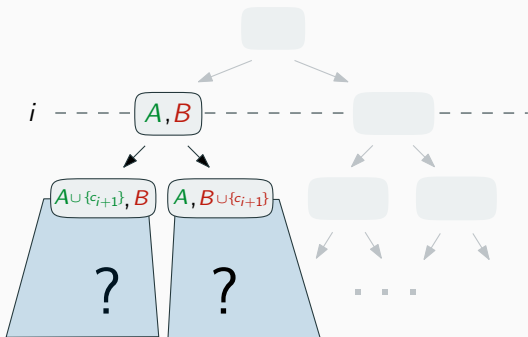
Extension problem

To guarantee the last condition, we want to solve:

Signature Extension (Sig-Ext)

input: a formula ϕ , $A, B \subseteq \{C_1, \dots, C_m\}$

output: is there a signature σ such that $\sigma[i] = 1$ if $C_i \in A$, and
 $\sigma[i] = 0$ if $C_i \in B$?



Solving the extension problem + limitations

Lemma

σ is a *signature* of ϕ iff $\phi(1(\sigma), 0(\sigma))$ is *satisfiable*

The problem **Sig•Ext** can be solved using this lemma, as XOR formulas are tractable by Schaefer's theorem

We derive the following

Theorem (Creignou, D., Olive, and Vilmin)

The problem Sig•Enum can be solved with poly delay

Theorem (Creignou, D., Olive, and Vilmin)

The problems Min/MaxSig•Enum are NP-complete even when restricted to XOR formulas

Properties (2)

For ϕ a XOR m -clause formula and σ a signature:

- $\bar{\sigma} := (1 - \sigma[1], \dots, 1 - \sigma[m])$
- $\bar{\phi} := \bigwedge_{i=1}^m \bar{C}_i$

Lemma⁶

σ is a signature of ϕ iff $\bar{\sigma}$ is a signature of $\bar{\phi}$

Corollary

*On XOR formulas **MinSig•Enum** is equivalent to **MaxSig•Enum***

⁶Crucial distinct behavior compared to CNF's

Detour to matroid theory

Lemma

The *maximal signatures* of a XOR formula ϕ are in bijection with the *maximal feasible subsystems* of the system that ϕ describes

Theorem (Boros et al., ISAAC 03)

The *maximal feasible subsystems* can be listed in *inc-poly* time

This algorithm is based on one enumerating the *circuits* of a *matroid* within the same time bounds

Corollary

Min/MaxSig•Enum can be solved in *inc-poly* time

Reducing to poly delay is a long-standing open question

What about 2-XOR formulas ?

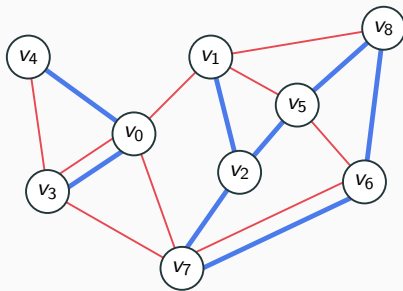
Bicolored graph

We assume all clauses have size precisely 2.⁷

We define a (multi)graph $G(\phi)$ on the variables with

- a **blue edge** xy if there exists a clause $x + y = 1$ in ϕ
- a **red edge** xy if there exists a clause $x + y = 0$ in ϕ

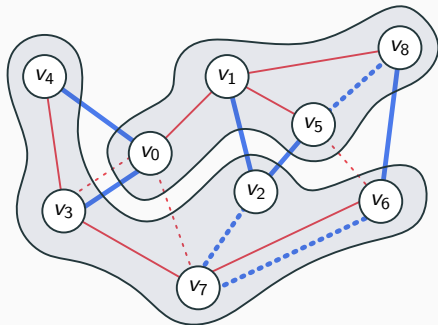
Let $B(G)$ be the blue edges, and $R(G)$ be the red edges



⁷This can be assumed by adding a dummy vertex adjacent to clauses of size 1

Bicolored partitions

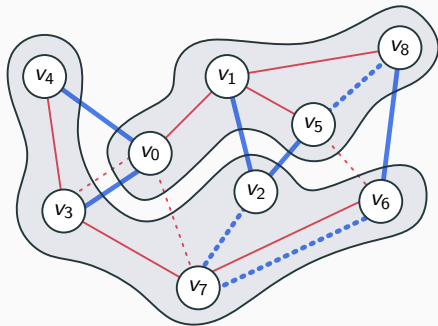
- $\delta(X, Y)$: set of edges having an endpoint in X , the other in Y
- G **red-blue bipartite**: if there exists a partition (X, Y) of its vertex set s.t. $B(G) = \delta(X, Y)$



Bicolored partitions

Lemma

The *maximal signatures* of a XOR formula ϕ are in bijection with the maximal *red-blue bipartite* (edge) *subgraphs* of $G(\phi)$



Particular case

In particular, **Min/MaxSig•Enum** is harder than **maximal bipartite subgraphs enumeration**, a non-trivial problem

Theorem (Conte & Uno, STOC 19)

Maximal bipartite subgraphs can be enumerated with poly delay

Can it be extended to red-blue bipartitions? Yes

Theorem (Creignou, D., Olive, and Vilmin)

Max. red-blue bip. subgraphs can be enumerated with poly delay

The latter algorithm is based on the framework introduced by Conte & Uno known as *proximity search*: go from solutions to solutions and ensure that you get closer to any target solution

Algorithm outline

Let \mathcal{S} denote the solution set

Key steps:

- show that a first solution can be **computed in poly time**
- define a **reconfiguration function** $\mathcal{N} : \mathcal{S} \rightarrow 2^{\mathcal{S}}$
- show that \mathcal{N} can be **computed in poly time**
- show that \mathcal{N} defines a **strongly connected digraph**

Theorem (Folklore)

*The family \mathcal{S} can be **enumerated with poly delay** if these conditions are fulfilled by launching a traversal of the solutions graph*

The approach of Conte & Uno is to define an asymmetric proximity measure to argue of the **strong connectivity**

Reconfiguration function

Given $H \in \mathcal{S}$, let $\text{GC}(H)$ be a maximal red-blue bipartite subgraph containing H obtained greedily by adding edges as long as possible

Given $H \in \mathcal{S}$, for every edge $ab \in G(\phi) - H$:

- compute $H_a = H + ab - \{av : av \in E(H)\}$
- compute $H_b = H + ab - \{bv : vb \in E(H)\}$
- add $\text{GC}(H_a)$ and $\text{GC}(H_b)$ to $\mathcal{N}(H)$

Observation

The family $\mathcal{N}(H)$ can be computed in poly time

It remains to argue that $\mathcal{N}(H)$ defines a strongly connected digraph

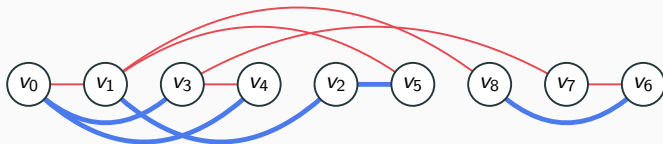
Proximity search (1)

General ideal: introduce a measure of proximity between solutions

Given $H \in \mathcal{S}$:

- ρ : BFS ordering of its vertices
- τ : increasing ordering of the edges of H
with respect to their endpoint occurring later ρ

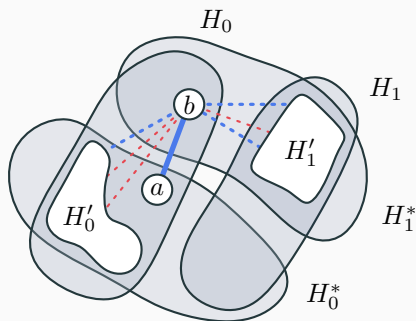
Proximity between two solutions H and H^* : size of the largest prefix of $\tau(H)$ which is a subset of H^*



Proximity search (2)

Key arguments:

- solutions are **connected**, and **cover all the vertices**
- solutions **agree on their bipartition** up to the prefix witnessing the proximity
- **delete edges** of H_a or H_b may not lie in such a prefix



Open questions

MaxSig·Enum stays open in other tractable cases




Question

Can **MaxSig·Enum** be solved in *output-poly time* in Horn-CNF's and 2-CNF's?

For k -XOR formulas, it remains open whether **Min/MaxSig·Enum** can be solved:

- with *poly delay* for fixed values of k
- with *poly delay and poly space*⁸ for $k = 2$

⁸Indeed, solutions are stored in the current algorithm

-  Kristóf Bérczi, Endre Boros, Ondřej Čepek, Khaled Elbassioni, Petr Kučera, and Kazuhisa Makino. **Generating clause sequences of a cnf formula.** *Theoretical computer science*, 856:68–74, 2021.
-  Eugene L. Lawler, Jan K. Lenstra, and Alexander H. G. Rinnooy Kan. **Generating all maximal independent sets: NP-hardness and polynomial-time algorithms.** *SIAM Journal on Computing*, 9(3):558–565, 1980.
-  John W. Moon and Leo Moser. **On cliques in graphs.** *Israel journal of Mathematics*, 3(1):23–28, 1965.



Shuji Tsukiyama, Mikio Ide, Hiromu Ariyoshi, and Isao Shirakawa. **A new algorithm for generating all the maximal independent sets.** *SIAM Journal on Computing*, 6(3):505–517, 1977.