Enumeration and matroids

Yann Strozecki

Équipe de Logique Mathématique, Paris 7

September 5, 2008

Let P be a predicate such as $P(x, y)$ is decidable in polynomial time in $|x|$ and $|y| \leq Q(|x|)$ with Q a fixed polynomial.

Our problem is to find the set $A(x) = \{y \mid P(x, y)\}$ The problems of this form are the class **Enum**$\cdot\mathrm{P}$.

We want to find $A(x)$ as fast as possible.
As the process is dynamic, we want the time between each member of the set we output to be as short as possible.

Definition

A problem $\textsc{Enum}\cdot\mathcal{A}$ is decidable in polynomial total time **TotalP**, if there is an algorithm which on every instance x, return the set $A(x)$ in time polynomial in $|x|$ and $|A(x)|$.

Definition

A problem $\textsc{Enum}{\cdot}\mathcal{A}$ is decidable in polynomial total time **TotalP**, if there is an algorithm which on every instance x, return the set $A(x)$ in time polynomial in $|x|$ and $|A(x)|$.

To an enumeration problem $\textsc{Enum}{\cdot}\mathcal{A}$, we associate the following characteristic decision problem :

$\textsc{AllSolution}_A$
Input: An instance x of A and a set M included in $A(x)$
Output: Accept if $A(x) - M \neq \varnothing$, reject otherwise

Definition

A problem $\text{Enum}\cdot\mathcal{A}$ is decidable in polynomial total time **TotalP**, if there is an algorithm which on every instance x, return the set $A(x)$ in time polynomial in $|x|$ and $|A(x)|$.

To an enumeration problem $\text{Enum}\cdot\mathcal{A}$, we associate the following characteristic decision problem :

AllSolution_A
Input: An instance x of A and a set M included in $A(x)$
Output: Accept if $A(x) - M \neq \varnothing$, reject otherwise

If $\text{Enum}\cdot\mathcal{A} \in$ **TotalP** then $\text{AllSolution}_A \in \mathrm{P}$.

A formula is $\textsc{Horn}$ if every clause is of the form $x_1 \Rightarrow x_2 \cdots \Rightarrow x_n$

A formula is $\textsc{Horn}$ if every clause is of the form $x_1 \Rightarrow x_2 \cdots \Rightarrow x_n$

- $\textsc{Enum}\cdot$*Horn* is not in **TotalP** unless $\mathrm{P} = \mathrm{NP}$ thanks to the problem $\textsc{AllSolution}_{\textsc{Horn}}$

A formula is $\textsc{Horn}$ if every clause is of the form $x_1 \Rightarrow x_2 \cdots \Rightarrow x_n$

A formula is $2\mathrm{CNF}$ if every clause has only two literals.

- $\textsc{Enum}$·*Horn* is not in **TotalP** unless $\mathrm{P} = \mathrm{NP}$ thanks to the problem $\textsc{AllSolution}_{\textsc{Horn}}$

A formula is $\textsc{Horn}$ if every clause is of the form $x_1 \Rightarrow x_2 \cdots \Rightarrow x_n$

A formula is $\mathrm{2CNF}$ if every clause has only two literals.

- $\textsc{Enum}{\cdot}\mathit{Horn}$ is not in **TotalP** unless $\mathrm{P} = \mathrm{NP}$ thanks to the problem $\textsc{AllSolution}_{\textsc{Horn}}$
- $\textsc{Enum}{\cdot}2\mathit{CNF}$ is proven to be in **TotalP**.

Definition

A problem $\textsc{Enum}{\cdot}\mathcal{A}$ is decidable in incremental polynomial time **IncP**, if there is an algorithm which on every instance x and every integer $k \leq |A(x)|$ return the $(k+1)^{th}$ solution in time polynomial in $|x|$ and k.

Definition

A problem $\textsc{Enum}{\cdot}\mathcal{A}$ is decidable in incremental polynomial time **IncP**, if there is an algorithm which on every instance x and every integer $k \leq |A(x)|$ return the $(k+1)^{th}$ solution in time polynomial in $|x|$ and k.

$\textsc{Enum}{\cdot}\textsc{MatroidCircuit}$ is in **IncP** as we will explain it later.

Definition

A problem $\textsc{Enum}\cdot\mathcal{A}$ is decidable with polynomial delay **DelayP**, if there is an algorithm which on every instance x return $A(x)$ using a time polynomial in $|x|$ between two generated solutions.

Definition

A problem $\textsc{Enum}{\cdot}\mathcal{A}$ is decidable with polynomial delay **DelayP**, if there is an algorithm which on every instance x return $A(x)$ using a time polynomial in $|x|$ between two generated solutions.

Several refinements of this class, depending on the level of information the algorithm keep during the enumeration.

Definition

A problem $\textsc{Enum}\cdot\mathcal{A}$ is decidable with polynomial delay **DelayP**, if there is an algorithm which on every instance x return $A(x)$ using a time polynomial in $|x|$ between two generated solutions.

Several refinements of this class, depending on the level of information the algorithm keep during the enumeration.

$\textsc{Enum}\cdot\textsc{MaxIndependentSet}$ is in **DelayP** and the algorithm enumerate the solutions in lexicographic order.
In reverse lexical order, it is not **DelayP** anymore [David S. Johnson, Christos H. Papadimitriou and Mihalis Yannakakis].

Definition

A problem $\textsc{Enum}{\cdot}\mathcal{A}$ is in the class **QueryP** if there are an order on the set $A(x)$ and an algorithm which computes the i^{th} solution of $A(x)$ in time polynomial in $|x|$. If there is no i^{th} solution the algorithm must tell so in polynomial time.

Definition

A problem $\textsc{Enum}\cdot\mathcal{A}$ is in the class **QueryP** if there are an order on the set $A(x)$ and an algorithm which computes the i^{th} solution of $A(x)$ in time polynomial in $|x|$. If there is no i^{th} solution the algorithm must tell so in polynomial time.

MSO queries on a term can be enumerated in linear time and the problem is also in **QueryP**.

Under widely accepted complexity hypothesis we have the following inclusions :

$$\textbf{QueryP} \subsetneq \textbf{StrongDelayP} \subsetneq \textbf{DelayP} \subseteq \textbf{IncP} \subsetneq \textbf{TotalP} \subsetneq \textbf{Enum}{\cdot}\mathrm{P}$$

Under widely accepted complexity hypothesis we have the following inclusions :

QueryP $\subsetneq$ **StrongDelayP** $\subsetneq$ **DelayP** $\subseteq$ **IncP** $\subsetneq$ **TotalP** $\subsetneq$ **Enum**$\cdot\mathrm{P}$

The enumeration of bipartite matching has be proven to be in **StrongDelayP** by Takeaki Uno but if it is in **QueryP** we would have $\mathrm{P} = \#\mathrm{P}$.

Under widely accepted complexity hypothesis we have the following inclusions :

QueryP $\subsetneq$ **StrongDelayP** $\subsetneq$ **DelayP** $\subseteq$ **IncP** $\subsetneq$ **TotalP** $\subsetneq$ **Enum**$\cdot\mathrm{P}$

The enumeration of maximal independent sets of a graph is in **DelayP** but not in **StrongDelayP** unless $\mathrm{P} = \mathrm{NP}$.

Under widely accepted complexity hypothesis we have the following inclusions :

$$\textbf{QueryP} \subsetneq \textbf{StrongDelayP} \subsetneq \textbf{DelayP} \subseteq \textbf{IncP} \subsetneq \textbf{TotalP} \subsetneq \textbf{Enum}{\cdot}\mathrm{P}$$

The inclusion between **DelayP** and **IncP** is not known to be strict and it would be very satisfying to have this separation too !

Under widely accepted complexity hypothesis we have the following inclusions :

$\textbf{QueryP} \subsetneq \textbf{StrongDelayP} \subsetneq \textbf{DelayP} \subseteq \textbf{IncP} \subsetneq \textbf{TotalP} \subsetneq \textbf{Enum}{\cdot}\mathrm{P}$

From a TFNP problem we create a problem wich has an exponential number of solution for each solution of the first one.
It is a **TotalP** problem, because the trivial algorithm is polynomial in the number of solution.
It is not **IncP** except if $\mathrm{NP} \cap \mathrm{coNP} = \mathrm{P}$.

Under widely accepted complexity hypothesis we have the following inclusions :

QueryP $\subsetneq$ **StrongDelayP** $\subsetneq$ **DelayP** $\subseteq$ **IncP** $\subsetneq$ **TotalP** $\subsetneq$ **Enum**$\cdot\mathrm{P}$

We want to enumerate the solutions of a FewP problem, if it is in **TotalP** then we have $\mathrm{FewP} = \mathrm{P}$.

A problem presented at our last meeting :

B is a matrix with coefficients in $\mathbb{F}_2$ of dimension $n * m$ and y a vector of length n .

A problem presented at our last meeting :

B is a matrix with coefficients in $\mathbb{F}_2$ of dimension $n * m$ and y a vector of length n .

We set the pointwise order on the vectors that is to say $x \leq z$ iff $x_i \leq z_i$ for all i.

A problem presented at our last meeting :

B is a matrix with coefficients in $\mathbb{F}_2$ of dimension $n * m$ and y a vector of length n .

We set the pointwise order on the vectors that is to say $x \leq z$ iff $x_i \leq z_i$ for all i.

We look for the minimal x for this order satisfying the equation $Bx = y$.

- Equivalent to finding minimal satisfying assignments of an affine formula, a problem known as ENUM·AFFINE CIRCUMSCRIPTION.

- Equivalent to finding minimal satisfying assignments of an affine formula, a problem known as ENUM·AFFINE CIRCUMSCRIPTION.
- Equivalent to enumerate all vertices of a given polytope in linear programming.
  Generalizable to any finite field.

- Equivalent to finding minimal satisfying assignments of an affine formula, a problem known as
  ENUM·AFFINE CIRCUMSCRIPTION.
- Equivalent to enumerate all vertices of a given polytope in linear programming.
  Generalizable to any finite field.
- Relationship with matroids circuits.

- Equivalent to finding minimal satisfying assignments of an affine formula, a problem known as ENUM·AFFINE CIRCUMSCRIPTION.
- Equivalent to enumerate all vertices of a given polytope in linear programming.
  Generalizable to any finite field.
- Relationship with matroids circuits.

Question by Dimitris J. Kavvadias, Martha Sideri, Elias C. Stavropoulos in “Generating all maximal models of a Boolean expression”.

Matroids have been design to abstract the notion of dependence.

Definition

A matroid is a pair $(E, \mathcal{I})$, E is a finite set and $\mathcal{I}$ is included in the power set of E. Elements of $\mathcal{I}$ are said to be independent sets, the others are dependent sets.

A matroid must satisfy the following axioms:

Matroids have been design to abstract the notion of dependence.

Definition

A matroid is a pair $(E, \mathcal{I})$, E is a finite set and $\mathcal{I}$ is included in the power set of E. Elements of $\mathcal{I}$ are said to be independent sets, the others are dependent sets.
A matroid must satisfy the following axioms:

1. $\varnothing \in \mathcal{I}$

Matroids have been design to abstract the notion of dependence.

Definition

A matroid is a pair $(E, \mathcal{I})$, E is a finite set and $\mathcal{I}$ is included in the power set of E. Elements of $\mathcal{I}$ are said to be independent sets, the others are dependent sets.
A matroid must satisfy the following axioms:

1. $\varnothing \in \mathcal{I}$
2. If $I \in \mathcal{I}$ and $I' \subseteq I$, then $I' \in \mathcal{I}$

Matroids have been design to abstract the notion of dependence.

Definition

A matroid is a pair $(E, \mathcal{I})$, E is a finite set and $\mathcal{I}$ is included in the power set of E. Elements of $\mathcal{I}$ are said to be independent sets, the others are dependent sets.
A matroid must satisfy the following axioms:

1. $\varnothing \in \mathcal{I}$
2. If $I \in \mathcal{I}$ and $I' \subseteq I$, then $I' \in \mathcal{I}$
3. If I_1 and I_2 are in $\mathcal{I}$ and $|I_1| < |I_2|$, then there is an element e of $I_2 - I_1$ such that $I_1 \cup e \in \mathcal{I}$.

The first concrete example of matroid is the vector matroid.

Let A be a matrix, the ground set E is the set of the columns and a set of columns is independent if the vectors are linearly independent.
In fact matroid have been designed from this example.

The first concrete example of matroid is the vector matroid.

Let A be a matrix, the ground set E is the set of the columns and a set of columns is independent if the vectors are linearly independent.
In fact matroid have been designed from this example.

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

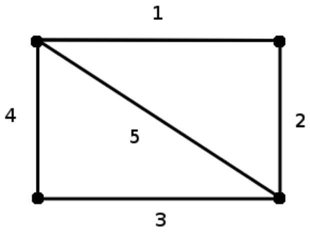
Here the set $\{1, 2, 4\}$ is independent and $\{1, 2, 3\}$ is dependent.

The second example is the cycle matroid of a graph.

Let G be a graph, the ground set of his cycle matroid is E the set of his edges.
A set is said to be dependent if it contains a cycle.

The second example is the cycle matroid of a graph.

Let G be a graph, the ground set of his cycle matroid is E the set of his edges.
A set is said to be dependent if it contains a cycle.



Here the set $\{1, 2, 4\}$ is independent and $\{1, 2, 3, 4\}$ or $\{1, 2, 5\}$ is dependent.

Lemma

Any cycle matroid is a representable matroid, i.e. it is isomorph to a vector matroid.

Label the vertices of a graph by $1, \ldots, n$ and the edges by $1, \ldots, m$.

Lemma

Any cycle matroid is a representable matroid, i.e. it is isomorph to a vector matroid.

Label the vertices of a graph by $1, \ldots, n$ and the edges by $1, \ldots, m$. We build the matrix A such as $A_{i,j} = 1$ iff the edge j is incident to the vertex i.

Lemma

Any cycle matroid is a representable matroid, i.e. it is isomorph to a vector matroid.

Label the vertices of a graph by $1, \ldots, n$ and the edges by $1, \ldots, m$.
We build the matrix A such as $A_{i,j} = 1$ iff the edge j is incident to the vertex i.
The dependence relation is the same over the edges and over the vectors representing the edges.

Lemma

Any cycle matroid is a representable matroid, i.e. it is isomorph to a vector matroid.

Label the vertices of a graph by $1, \ldots, n$ and the edges by $1, \ldots, m$. We build the matrix A such as $A_{i,j} = 1$ iff the edge j is incident to the vertex i.
The dependence relation is the same over the edges and over the vectors representing the edges.

This matrix represents the former graph:

$$\mathbf{X} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Definition

A base of a matroid is a independent set maximal for the inclusion.

Definition

A base of a matroid is a independent set maximal for the inclusion.

Lemma

Every base is of the same length.

Definition

A base of a matroid is a independent set maximal for the inclusion.

Lemma

Every base is of the same length.

In a vectorial matroid a base is a maximal free family.

Definition

A base of a matroid is a independent set maximal for the inclusion.

Lemma

Every base is of the same length.

In a vectorial matroid a base is a maximal free family.

In a cycle matroid it is a spanning tree of the graph.

Definition

A circuit is a dependent set minimal for the inclusion.

Definition

A circuit is a dependent set minimal for the inclusion.

Lemma (Circuit elimination property)

Let C_1 and C_2 two circuits of a matroid, $e \in C_1 \cap C_2$, then there is a circuit C_3 such that $C_3 \subseteq (C_1 \cup C_2) - e$.

Definition

A circuit is a dependent set minimal for the inclusion.

Lemma (Circuit elimination property)

Let C_1 and C_2 two circuits of a matroid, $e \in C_1 \cap C_2$, then there is a circuit C_3 such that $C_3 \subseteq (C_1 \cup C_2) - e$.

Remark

We can axiomatically define the matroids with a predicate base or circuit rather than the independence predicate.

We can also define concept as the rank or the span of a set, in the same way we do in linear algebra.

There is a strong link between our problem and the circuit of a vector matroid.

B a matrix and y a vector defining a linear system.
Consider the vector matroid M represented by the matrix $(B|y)$.

There is a strong link between our problem and the circuit of a vector matroid.

B a matrix and y a vector defining a linear system.
Consider the vector matroid M represented by the matrix $(B|y)$.

Lemma

There is a bijection between the minimal solutions of the system and the circuits of M containing the vector y.

For complexity, we use the following presentation of matroids : Let S be a finite ground set, the matroid is defined by an *independence oracle* wich given $X \subseteq S$ determines in unit time if X is independent.

For complexity, we use the following presentation of matroids : Let S be a finite ground set, the matroid is defined by an *independence oracle* wich given $X \subseteq S$ determines in unit time if X is independent.

To interpret the complexity results founds thanks to this representation, we have to decide the independence of a set in polynomial time in the cardinal of the base set of the matroid.

For complexity, we use the following presentation of matroids : Let S be a finite ground set, the matroid is defined by an *independence oracle* wich given $X \subseteq S$ determines in unit time if X is independent.

To interpret the complexity results founds thanks to this representation, we have to decide the independence of a set in polynomial time in the cardinal of the base set of the matroid.

Clearly the vector matroids verify this condition, because gaussian elimination allows to decide the independence of a set of vector in cubic time.

The simpler enumeration problem is to find all the bases of a matroid.

The simpler enumeration problem is to find all the bases of a matroid.

Set an order on the ground set and do a depth first search on the tree of the independent sets.

Theorem

$\textsc{Enum}\cdot\textsc{MatroidBase} \in$ **StrongDelayP**.

We want to enumerate the circuits of a matroid, we add the constraint that they contain a fixed element.

The followings are simpler cases of this problem :

We want to enumerate the circuits of a matroid, we add the constraint that they contain a fixed element.

The followings are simpler cases of this problem :

1. If the matroid is is defined by a whole subspace of a vector space, then it is in **DelayP**.

We want to enumerate the circuits of a matroid, we add the constraint that they contain a fixed element.

The followings are simpler cases of this problem :

1. If the matroid is is defined by a whole subspace of a vector space, then it is in **DelayP**.
2. If the matroid is a binary matroid with only two non zero coefficients in each row, this is the 2-AFFINE problem. Equivalent to find the 2 coloration of a graph, in **DelayP** or better as there is either 0 or 2 colorations for each connected component.

We want to enumerate the circuits of a matroid, we add the constraint that they contain a fixed element.

The followings are simpler cases of this problem :

1. If the matroid is is defined by a whole subspace of a vector space, then it is in **DelayP**.
2. If the matroid is a binary matroid with only two non zero coefficients in each row, this is the 2-AFFINE problem. Equivalent to find the 2 coloration of a graph, in **DelayP** or better as there is either 0 or 2 colorations for each connected component.
3. If the matroid is the cycle matroid of a graph, it is in **DelayP**.

Definition

Let B be a basis of a matroid M and x not in B then there exists a unique circuit $C = C(B, x)$ such that $x \in C \subseteq B \cap x$. This circuit is called the fundamental circuit of x in the base B.

Exactly as the unicity of the decomposition of a vector in a basis.

Definition

Let B be a basis of a matroid M and x not in B then there exists a unique circuit $C = C(B, x)$ such that $x \in C \subseteq B \cap x$. This circuit is called the fundamental circuit of x in the base B.

Exactly as the unicity of the decomposition of a vector in a basis.

We compute a fundamental circuit in a basis B by at most $|B|$ call to the independence oracle.
Hence we compute all the fundamental circuits in a certain base in poly time.

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

After gaussian elimination the matrix look like this.

$$\mathbf{A} = \left(\begin{array}{ccc|cc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{array}\right)$$

The three first columns are the base and were the columns $1, 2$ and 4.

The two other columns are defined by their fundamental circuit.

Let E be a ground set and C included in the power set of E.

Suppose that C verifies the two following conditions :

Let E be a ground set and C included in the power set of E.

Suppose that C verifies the two following conditions :

C.e.p. $\forall C_1, C_2 \in C$, $e \in C_1 \cap C_2$, then $\exists C_3 \in C$ such that $C_3 \subseteq (C_1 \cup C_2) - e$.

Sperner $\forall C_1, C_2 \in C \Rightarrow C_1 \subsetneq C_2$

Let E be a ground set and C included in the power set of E.

Suppose that C verifies the two following conditions :

- **C.e.p.** $\forall C_1, C_2 \in C$, $e \in C_1 \cap C_2$, then $\exists C_3 \in C$ such that $C_3 \subseteq (C_1 \cup C_2) - e$.
- **Sperner** $\forall C_1, C_2 \in C \Rightarrow C_1 \subsetneq C_2$

Then C is the set of the circuits of a matroid.

Let E be a ground set and C included in the power set of E.

Suppose that C verifies the two following conditions :

C.e.p. $\forall C_1, C_2 \in C$, $e \in C_1 \cap C_2$, then $\exists C_3 \in C$ such that $C_3 \subseteq (C_1 \cup C_2) - e$.

Sperner $\forall C_1, C_2 \in C \Rightarrow C_1 \subsetneq C_2$

Then C is the set of the circuits of a matroid.

If C contains all the fundamental circuits of a matroid M, then C is the set of circuits of M.

Theorem

Computing the circuits of a matroid is in **IncP**.

This result has been proven in the article "On the Complexity of Some Enumeration Problems for Matroids" by Leonid G. Khachiyan, Endre Boros, Khaled M. Elbassioni, Vladimir Gurvich and Kazuhisa Makino.

1. We first compute the set S of fundamental circuit of the matroid.

1. We first compute the set S of fundamental circuit of the matroid.
2. For each couple of circuit in S having an element in common test if the Circuit elimination property is true

1. We first compute the set S of fundamental circuit of the matroid.
2. For each couple of circuit in S having an element in common test if the Circuit elimination property is true
3. If not add a new circuit to S to make the property true and go back in 2

1. We first compute the set S of fundamental circuit of the matroid.
2. For each couple of circuit in S having an element in common test if the Circuit elimination property is true
3. If not add a new circuit to S to make the property true and go back in 2
4. Otherwise S is closed by the property then stop

1. We first compute the set S of fundamental circuit of the matroid.
2. For each couple of circuit in S having an element in common test if the Circuit elimination property is true
3. If not add a new circuit to S to make the property true and go back in 2
4. Otherwise S is closed by the property then stop

This algorithm work in incremental polynomial time and output all the circuits of the matroid.

Definition

We define the relation $\sim$ on a matroid M by $e \sim f$ iff there is a circuit of M containing both e and f.

Definition

We define the relation $\sim$ on a matroid M by $e \sim f$ iff there is a circuit of M containing both e and f.

This relation is an equivalence relation, it is not trivial to prove the transitivity.

Definition

We define the relation $\sim$ on a matroid M by $e \sim f$ iff there is a circuit of M containing both e and f.

This relation is an equivalence relation, it is not trivial to prove the transitivity.

Definition (connectivity)

The equivalence classes of $\sim$ are called the connected components of the matroid and the matroid is said to be connected if there is only one component.

If we look for the circuit containing a fixed element, we have to search in the connected component of this element.

Theorem

Let e be an element of a connected matroid M, and C_e be the set of circuits of M containing e. Then the circuits of M not containing e are the minimal sets of the form

$$(C_1 \cup C_2) - \bigcap \{C \in C_e | C \subseteq C_1 \cup C_2\}$$

where C_1 and C_2 are distinct members of C_e.

Theorem

Let e be an element of a connected matroid M, and $\mathcal{C}_e$ be the set of circuits of M containing e. Then the circuits of M not containing e are the minimal sets of the form

$$(C_1 \cup C_2) - \bigcap \{C \in \mathcal{C}_e | C \subseteq C_1 \cup C_2\}$$

where C_1 and C_2 are distinct members of $\mathcal{C}_e$.

Then we have a surjection from $\mathcal{C}_e \times \mathcal{C}_e$ to $\mathcal{C}$ the set of circuit, therefore $|\mathcal{C}_e|^2 > |\mathcal{C}|$.

Theorem

Computing all the circuits through one fixed element is in **IncP***.*

Again it is a result in the article "On the Complexity of Some Enumeration Problems for Matroids".

We look for all the circuits through the element e.

- First compute the set S of the fundamental circuits.

We look for all the circuits through the element e.

- First compute the set S of the fundamental circuits.
- For each couple $s_1, s_2 \in S$ such as $s_1 \cap s_2 \neq \varnothing$ replace them by $s_1 \cup s_2$.

We look for all the circuits through the element e.

- First compute the set S of the fundamental circuits.
- For each couple $s_1, s_2 \in S$ such as $s_1 \cap s_2 \neq \varnothing$ replace them by $s_1 \cup s_2$.
- Select the connected component containing e, its a new matroid M.

We look for all the circuits through the element e.

- First compute the set S of the fundamental circuits.
- For each couple $s_1, s_2 \in S$ such as $s_1 \cap s_2 \neq \varnothing$ replace them by $s_1 \cup s_2$.
- Select the connected component containing e, its a new matroid M.
- Compute all the circuits of M thanks to the previous algorithm, even those not containing e but discard them.

We look for all the circuits through the element e.

- First compute the set S of the fundamental circuits.
- For each couple $s_1, s_2 \in S$ such as $s_1 \cap s_2 \neq \varnothing$ replace them by $s_1 \cup s_2$.
- Select the connected component containing e, its a new matroid M.
- Compute all the circuits of M thanks to the previous algorithm, even those not containing e but discard them.

Thanks to the previous remark we do only a polynomial number of step in the number of solution. We have proven that the problem is in **TotalP**, but with a few improvements we have a **IncP** algorithm.

The next natural question is to find circuits with k fixed elements.

The next natural question is to find circuits with k fixed elements.

- When k is part of the instance the decision problem is NP-complete.
  This problem is a generalization of the problem to find a minimal extension of a partial solution for an affine formula which is NP-complete too.

The next natural question is to find circuits with k fixed elements.

- When k is part of the instance the decision problem is NP-complete.
  This problem is a generalization of the problem to find a minimal extension of a partial solution for an affine formula which is NP-complete too.
- When $k = 2$ and the matroid is binary, the decision problem is in P.

The next natural question is to find circuits with k fixed elements.

- When k is part of the instance the decision problem is NP-complete.
  This problem is a generalization of the problem to find a minimal extension of a partial solution for an affine formula which is NP-complete too.
- When $k = 2$ and the matroid is binary, the decision problem is in P.
- The problem to find a cycle in a graph with k fixed edges is fixed parameter tractable in k.

Some open questions :

Some open questions :

1. Decide if there is a circuit with k fixed elements in a generic matroid in poly time

Some open questions :

1. Decide if there is a circuit with k fixed elements in a generic matroid in poly time
2. Find a simpler structure, where this problem is in P

Some open questions :

1. Decide if there is a circuit with k fixed elements in a generic matroid in poly time
2. Find a simpler structure, where this problem is in P
3. Enumerate the circuits of a representable matroid with k fixed elements in **IncP** or **TotalP** for at least $k = 2$

Some open questions :

1. Decide if there is a circuit with k fixed elements in a generic matroid in poly time
2. Find a simpler structure, where this problem is in P
3. Enumerate the circuits of a representable matroid with k fixed elements in **IncP** or **TotalP** for at least $k = 2$
4. Find a simpler structure where the problem is in **DelayP**

Some open questions :

1. Decide if there is a circuit with k fixed elements in a generic matroid in poly time
2. Find a simpler structure, where this problem is in P
3. Enumerate the circuits of a representable matroid with k fixed elements in **IncP** or **TotalP** for at least $k = 2$
4. Find a simpler structure where the problem is in **DelayP**
5. Prove that the enumeration of the circuits of a matroid is not **DelayP**

We can define a branch-width notion on the matroids with a corresponding parse tree.

We can define a branch-width notion on the matroids with a corresponding parse tree.

We define a monadic second order logic over the matroid, called MSO_M, just by adding the second order predicate *Indep* to the classical definition of this logic.

We can define a branch-width notion on the matroids with a corresponding parse tree.

We define a monadic second order logic over the matroid, called MSO_M, just by adding the second order predicate *Indep* to the classical definition of this logic.

Theorem

Let t be a fixed parameter and ϕ a MSO_M formula. We can decide ϕ over the matroids of branch-width t in polynomial time.

We can define a branch-width notion on the matroids with a corresponding parse tree.

We define a monadic second order logic over the matroid, called MSO_M, just by adding the second order predicate *Indep* to the classical definition of this logic.

Theorem

Let t be a fixed parameter and ϕ a MSO_M formula. We can decide ϕ over the matroids of branch-width t in polynomial time.

A theorem proven by Petr Hlinený. It's a FPT result, the dependence is not polynomial in t or in the length of the formula.

The previous theorem can be extended to enumeration as it has already been done for graph of bounded tree-width by Bruno Courcelle.

The previous theorem can be extended to enumeration as it has already been done for graph of bounded tree-width by Bruno Courcelle.

Theorem in progress

Let $\phi(\bar{x})$ a formula of MSO_M with free variables $\vec{x}$. Given a matroid, we want to enumerate all the $\bar{x}$ which makes the formula $\phi(\bar{x})$ true. This problem is in **QueryP** *over the matroids of bounded branch-width t.*

The previous theorem can be extended to enumeration as it has already been done for graph of bounded tree-width by Bruno Courcelle.

Theorem in progress

Let $\phi(\bar{x})$ a formula of MSO_M with free variables $\vec{x}$. Given a matroid, we want to enumerate all the $\bar{x}$ which makes the formula $\phi(\bar{x})$ true. This problem is in **QueryP** *over the matroids of bounded branch-width t.*

The problem to find the circuits of a matroid with k fixed elements is therefore easy when the matroids are of bounded branch-width.