Enumeration Complexity of logical query problems with second order variables

Yann Strozecki and Arnaud Durand

Université Paris Diderot - Paris 7

September 2011, CSL conference

Enumeration

A logical perspective on enumeration

A quantifier alternation hierarchy

Enumeration problems

Polynomially balanced predicate $A(x, y)$, decidable in polynomial time.

- $\exists ? y A(x, y)$: decision problem (class NP)
- $\sharp\{y \mid A(x, y)\}$: counting problem (class $\sharp\mathrm{P}$)
- $\{y \mid A(x, y)\}$: enumeration problem (class EnumP)

Example

Perfect matching:

- The decision problem is to decide if there is a perfect matching.
- The counting problem is to count the number of perfect matchings.
- The enumeration problem is to list every perfect matching.

Enumeration problems

Polynomially balanced predicate $A(x, y)$, decidable in polynomial time.

- $\exists ? y A(x, y)$: decision problem (class NP)
- $\sharp\{y \mid A(x, y)\}$: counting problem (class $\sharp\mathrm{P}$)
- $\{y \mid A(x, y)\}$: enumeration problem (class EnumP)

Example

Perfect matching:

- The decision problem is to decide if there is a perfect matching.
- The counting problem is to count the number of perfect matchings.
- The enumeration problem is to list every perfect matching.

Time complexity measures for enumeration

1. the total time related to the number of solutions
   - polynomial total time: $\mathbf{TotalP}$

2. the delay
   - incremental polynomial time: $\mathbf{IncP}$ (Circuits of a matroid)
   - polynomial delay: $\mathbf{DelayP}$ (Perfect Matching)
   - Constant or linear delay
     - A two steps algorithm: preprocessing + generation
     - An ad-hoc RAM model.

Time complexity measures for enumeration

1. the total time related to the number of solutions
   - polynomial total time: **TotalP**
2. the delay
   - incremental polynomial time: **IncP** (Circuits of a matroid)
   - polynomial delay: **DelayP** (Perfect Matching)
   - Constant or linear delay
     - A two steps algorithm: preprocessing + generation
     - An ad-hoc RAM model.

Enumeration problems

$\textsc{Enum}{\cdot}R$

Input: $x \in \mathcal{I}$

Output: an enumeration of elements in $R(x) = \{y \mid R(x, y)\}$

Definition

The problem $\textsc{Enum}{\cdot}R$ belongs to the class $\textsc{Delay}(g, f)$ if there exists an enumeration algorithm that computes $\textsc{Enum}{\cdot}R$ such that, for all input x:

- Preprocessing in time $O(g(|x|))$,
- Solutions $y \in R(x)$ are computed successively without repetition with a delay $O(f(|x|))$

$\textsc{Constant-Delay} = \bigcup_k \textsc{Delay}(n^k, O(1))$.

Enumeration

A logical perspective on enumeration

A quantifier alternation hierarchy

Enumeration problem defined by a formula

Let $\Phi(\mathbf{z}, \mathbf{T})$ be a first order formula.

To simplify, the tuple $\mathbf{T}$ contains only one relation T.

$\textsc{Enum}{\cdot}\Phi$

Input: A σ-structure $\mathcal{S}$

Output: $\Phi(\mathcal{S}) = \{(\mathbf{z}^*, \mathbf{T}^*) : (\mathcal{S}, \mathbf{z}^*, \mathbf{T}^*) \models \Phi(\mathbf{z}, \mathbf{T})\}$

Similar to parametrized complexity classes.

Let $\mathscr{F}$ be a subclass of first order formulas. We denote by $\textsc{Enum}{\cdot}\mathscr{F}$ the collection of problems $\textsc{Enum}{\cdot}\Phi$ for $\Phi \in \mathscr{F}$.

First-order queries with free second order variables

This work

- **FO** queries with free second-order variables
- Data complexity: the query is fixed
- The complexity in term of the size of the input structure's domain
- On arbitrary structures
- Quantifier depth as a parameter: $\textsc{Enum}{\cdot}\Sigma_1$

Example

Example

Independent sets:

$$IS(T) \equiv \forall x \forall y\ \ T(x) \wedge T(y) \Rightarrow \neg E(x,y).$$

The formula is in Π_1, thus $\textsc{Enum}{\cdot}\mathrm{IS} \in \textsc{Enum}{\cdot}\Pi_1$.

Example

Hitting sets (vertex covers) of a hypergraph represented by the incidence structure $\langle D, \{V, E, R\}\rangle$.

$$HS(T) \equiv \forall x\,(T(x) \Rightarrow\ V(x)) \wedge \forall y \exists x\, E(y) \Rightarrow (T(x) \wedge R(x,y))$$

The problem $\textsc{Enum}{\cdot}\mathrm{HS} \in \textsc{Enum}{\cdot}\Pi_2$.

Previous results

1. Only first-order free variables and bounded degree structures. Durand-Grandjean'07, Lindell'08, Kazana-Segoufin'10: linear preprocessing + constant delay.
2. Only first-order free variables and acyclic conjunctive formula. Bagan-Durand-Grandjean'07: linear preprocessing + linear delay

Example

Enumeration of the k-cliques of a graph of bounded degree.

Previous results

1. Only first-order free variables and bounded degree structures. Durand-Grandjean'07, Lindell'08, Kazana-Segoufin'10: linear preprocessing + constant delay.
2. Only first-order free variables and acyclic conjunctive formula. Bagan-Durand-Grandjean'07: linear preprocessing + linear delay
3. Monadic second order formula and bounded tree-width structure Bagan, Courcelle 2009: almost linear preprocessing + linear delay

Example

Typical database query. Simple paths of length k.

Previous results

1. Only first-order free variables and bounded degree structures. Durand-Grandjean'07, Lindell'08, Kazana-Segoufin'10: linear preprocessing + constant delay.
2. Only first-order free variables and acyclic conjunctive formula. Bagan-Durand-Grandjean'07: linear preprocessing + linear delay
3. Monadic second order formula and bounded tree-width structure Bagan, Courcelle 2009: almost linear preprocessing + linear delay

Example

Enumeration of the cliques of a bounded tree-width graph.

A hierarchy result for counting functions

From a formula $\Phi(\mathbf{z}, \mathbf{T})$, one defines the counting function:

$$\#\Phi : \mathcal{S} \mapsto |\Phi(\mathcal{S})|.$$

Theorem (Saluja, Subrahmanyam, Thakur 1995)

On linearly ordered structures:
$\#\Sigma_0 \subsetneq \#\Sigma_1 \subsetneq \#\Pi_1 \subsetneq \#\Sigma_2 \subsetneq \#\Pi_2 = \sharp\mathrm{P}$.

Some $\sharp\mathrm{P}$-hard problems in $\#\Sigma_1$ (but existence of FPRAS at this level).

Corollary

On linearly ordered structures:
$\textsc{Enum}{\cdot}\Sigma_0 \subsetneq \textsc{Enum}{\cdot}\Sigma_1 \subsetneq \textsc{Enum}{\cdot}\Pi_1 \subsetneq \textsc{Enum}{\cdot}\Sigma_2 \subsetneq \textsc{Enum}{\cdot}\Pi_2$.

A hierarchy result for counting functions

From a formula $\Phi(\mathbf{z}, \mathbf{T})$, one defines the counting function:

$$\#\Phi : \mathcal{S} \mapsto |\Phi(\mathcal{S})|.$$

Theorem (Saluja, Subrahmanyam, Thakur 1995)

On linearly ordered structures:
$\#\Sigma_0 \subsetneq \#\Sigma_1 \subsetneq \#\Pi_1 \subsetneq \#\Sigma_2 \subsetneq \#\Pi_2 = \sharp\mathrm{P}$.

Some $\sharp\mathrm{P}$-hard problems in $\#\Sigma_1$ (but existence of FPRAS at this level).

Corollary

On linearly ordered structures:
$\textsc{Enum}{\cdot}\Sigma_0 \subsetneq \textsc{Enum}{\cdot}\Sigma_1 \subsetneq \textsc{Enum}{\cdot}\Pi_1 \subsetneq \textsc{Enum}{\cdot}\Sigma_2 \subsetneq \textsc{Enum}{\cdot}\Pi_2$.

Enumeration

A logical perspective on enumeration

A quantifier alternation hierarchy

The first level: Enum$\cdot\Sigma_0$

Theorem

For $\varphi \in \Sigma_0$, $\textsc{Enum}\cdot\varphi$ can be enumerated with preprocessing $O(|D|^k)$ and delay $O(1)$ where k is the number of free first order variables of φ and D is the domain of the input structure.

Simple ingredients:

1. Transformation of a f.o. formula $\Phi(\mathbf{z}, T)$ into a propositional formula:
   - Disjunction on all values for first order variables, $\bigvee_{i=0}^{|D|^k-1} \Phi(\mathbf{z}_i, T)$.
   - Replace the atomic formulas by their truth value.
   - Obtain a propositional formula with variables $T(\mathbf{w})$.
2. Gray Code Enumeration.

Bounded degree structure

Remark: The k-clique query is definable.
No hope to improve the $O(n^k)$ preprocessing.

Theorem

Let $d \in \mathbb{N}$, on d-degree bounded input structures, $\textsc{Enum}\cdot\Sigma_0 \in \textsc{Delay}(|D|, 1)$ where D is the domain of the input structure.

Bounded degree structure

Remark: The k-clique query is definable.
No hope to improve the $O(n^k)$ preprocessing.

Theorem

Let $d \in \mathbb{N}$, on d-degree bounded input structures, $\textsc{Enum}\cdot\Sigma_0 \in \textsc{Delay}(|D|, 1)$ where D is the domain of the input structure.

Idea of proof:

- Another transformation: $\Phi(\mathbf{z}, T)$ seen as a propositional formula whose variables are the atoms of Φ.
- From each solution, create a quantifier free formula without free second order variables.
- Enumerate the solutions of this formula thanks to [DG 2007].

Bounded degree structure

Remark: The k-clique query is definable.
No hope to improve the $O(n^k)$ preprocessing.

Theorem

Let $d \in \mathbb{N}$, on d-degree bounded input structures, $\textsc{Enum}{\cdot}\Sigma_0 \in \textsc{Delay}(|D|, 1)$ where D is the domain of the input structure.

Idea of proof:

- Another transformation: $\Phi(\mathbf{z}, T)$ seen as a propositional formula whose variables are the atoms of Φ.
- From each solution, create a quantifier free formula without free second order variables.
- Enumerate the solutions of this formula thanks to [DG 2007].

Second level: Enum$\cdot\Sigma_1$

Theorem

$\textsc{Enum}\cdot\Sigma_1 \subseteq \textsc{DelayP}$. *More precisely,* $\textsc{Enum}\cdot\Sigma_1$ *can be computed with precomputation* $O(|D|^{h+k})$ *and delay* $O(|D|^k)$ *where* h *is the number of free first order variables of the formula,* k *the number of existentially quantified variables and* D *is the domain of the input structure.*

Idea of Proof: $\Phi(\mathbf{y}, T) = \exists \mathbf{x} \varphi(\mathbf{x}, \mathbf{y}, T)$

- Substitute values for $\mathbf{x}$. Collection of formulas of the form:

$$\varphi(\mathbf{x}^*, \mathbf{y}, T)$$

- Need to enumerate the (non necessarily disjoint) union.

Enumeration of an union

How to enumerate an union?

Order on the solutions.

Enumeration of an union

How to enumerate an union?

Order on the solutions.

Lemma

Let R and S be two polynomially balanced predicates such that S can be decided in time $O(h(n))$. Assume that one can solve $\text{Enum}{\cdot}R$ and $\text{Enum}{\cdot}S$ with preprocessing $f(n)$ and delay $g(n)$, then one can solve $\text{Enum}{\cdot}R \cup S$ with preprocessing $2f(n) + c$ and delay $2g(n) + h(n) + c$, where c is a constant.

Enumeration of an union

How to enumerate an union?

Order on the solutions.

Lemma

Let R and S be two polynomially balanced predicates such that S can be decided in time $O(h(n))$. Assume that one can solve $\textsc{Enum}{\cdot}R$ *and* $\textsc{Enum}{\cdot}S$ *with preprocessing* $f(n)$ *and delay* $g(n)$*, then one can solve* $\textsc{Enum}{\cdot}R \cup S$ *with preprocessing* $2f(n) + c$ *and delay* $2g(n) + h(n) + c$*, where* c *is a constant.*

Remark

The problem $\textsc{Enum}{\cdot}l - DNF$ is equivalent to problems in $\textsc{Enum}{\cdot}\Sigma_1$.

Enumeration of an union

How to enumerate an union?

Order on the solutions.

Lemma

Let R and S be two polynomially balanced predicates such that S can be decided in time $O(h(n))$. Assume that one can solve $\textsc{Enum}{\cdot}R$ and $\textsc{Enum}{\cdot}S$ with preprocessing $f(n)$ and delay $g(n)$, then one can solve $\textsc{Enum}{\cdot}R \cup S$ with preprocessing $2f(n) + c$ and delay $2g(n) + h(n) + c$, where c is a constant.

Remark

The problem $\textsc{Enum}{\cdot}l - DNF$ is equivalent to problems in $\textsc{Enum}{\cdot}\Sigma_1$.

The case Enum$\cdot\Pi_1$

Proposition

Unless $\mathrm{P} = \mathrm{NP}$, *there is no polynomial delay algorithm for* $\textsc{Enum}\cdot\Pi_1$.

Proof Direct encoding of SAT.

Hardness even:

- on the class of bounded degree structure
- if all clauses but one have at most two occurences of a second-order free variable

Tractable cases

Transformation of a Σ_i formula into a propositional formula.

Proposition (Creignou, Hebrard'97)

The problem $\textsc{Enum}{\cdot}\mathrm{SAT}(\mathcal{C})$ *is in* $\textsc{DelayP}$ *when* $\mathcal{C}$ *is one of the following classes: Horn formulas, anti-Horn formulas, affine formulas, bijunctive (2CNF) formulas.*

Tractable cases

Transformation of a Σ_i formula into a propositional formula.

Proposition (Creignou, Hebrard'97)

The problem $\textsc{Enum}{\cdot}\mathrm{SAT}(\mathcal{C})$ *is in* $\textsc{DelayP}$ *when* $\mathcal{C}$ *is one of the following classes: Horn formulas, anti-Horn formulas, affine formulas, bijunctive (2CNF) formulas.*

Corollary

Let $\Phi(\mathbf{z}, T)$ *be a formula, such that, for all* σ *structures, all propositional formulas* $\tilde{\Phi}_i$ *are either Horn, anti-Horn, affine or bijunctive. Then* $\textsc{Enum}{\cdot}\Phi \subseteq \textsc{DelayP}$.

Tractable cases

Transformation of a Σ_i formula into a propositional formula.

Proposition (Creignou, Hebrard'97)

The problem $\textsc{Enum}{\cdot}\mathrm{SAT}(\mathcal{C})$ *is in* $\textsc{DelayP}$ *when* $\mathcal{C}$ *is one of the following classes: Horn formulas, anti-Horn formulas, affine formulas, bijunctive (2CNF) formulas.*

Corollary

Let $\Phi(\mathbf{z}, T)$ *be a formula, such that, for all* σ *structures, all propositional formulas* $\tilde{\Phi}_i$ *are either Horn, anti-Horn, affine or bijunctive. Then* $\textsc{Enum}{\cdot}\Phi \subseteq \textsc{DelayP}$.

Example: independent sets and hitting sets.

Tractable cases

Transformation of a Σ_i formula into a propositional formula.

Proposition (Creignou, Hebrard'97)

The problem $\textsc{Enum}{\cdot}\mathrm{SAT}(\mathcal{C})$ *is in* $\textsc{DelayP}$ *when* $\mathcal{C}$ *is one of the following classes: Horn formulas, anti-Horn formulas, affine formulas, bijunctive (2CNF) formulas.*

Corollary

Let $\Phi(\mathbf{z}, T)$ *be a formula, such that, for all* σ *structures, all propositional formulas* $\tilde{\Phi}_i$ *are either Horn, anti-Horn, affine or bijunctive. Then* $\textsc{Enum}{\cdot}\Phi \subseteq \textsc{DelayP}$.

Example: independent sets and hitting sets.

Conlusions and open problems

$\textsc{Enum}\cdot\Sigma_0 \subsetneq \textsc{Enum}\cdot\Sigma_1 \subsetneq \textsc{Enum}\cdot\Pi_1 \subsetneq \textsc{Enum}\cdot\Sigma_2 \subsetneq \textsc{Enum}\cdot\Pi_2 = \mathrm{EnumP}$.

- Nice but small hierarchy.
- Other tractable classes above Σ_1 (optimization operator)?
- Efficient probabilistic enumeration procedure?