Descriptive Complexity of Approximate Counting CSPs

Andrei Bulatov[1] Victor Dalmau[2] Marc Thurley[3]

[1]Simon Fraser University

[2]Universitat Pompeu Fabra

[3]Oracle

November 19, 2014

Counting Problems

Let σ be a finite relational signature and let $\text{Struct}(\sigma)$ be the set of all structures with signature σ.

Definition

A *counting problem* is a mapping $\mathcal{C} : \text{Struct}(\sigma) \to \mathbf{Z}^+$

Counting Problems

Let σ be a finite relational signature and let Struct(σ) be the set of all structures with signature σ.

Definition

A *counting problem* is a mapping $\mathcal{C}$: Struct(σ) $\rightarrow \mathbf{Z}^+$

Examples:

- #SAT: given a CNF compute the number of its satisfying assignments.
- #INDEPENDENT SET: given a graph, compute the number of its independent sets.

(Non uniform) Counting Constraint Satisfaction Problem

Let $\mathbf{H}$ be a relational structure in Struct(σ)

Problem

$\#$CSP($\mathbf{H}$) *is the following counting problem:*
Given a finite structure $\mathbf{A} \in$ Struct(σ) *compute the number of homomorphisms from* $\mathbf{A}$ *to* $\mathbf{H}$.

An homomorphism from $\mathbf{A} = (A; E_1, \ldots, E_n)$ to $\mathbf{H} = (H; R_1, \ldots, R_n)$ is a mapping $f : A \to H$ such that for every $1 \leq i \leq n$ and every tuple $(a_1, \ldots, a_r) \in A^*$

$$(a_1, \ldots, a_r) \in E_i \Rightarrow (f(a_1), \ldots, f(a_r)) \in R_i$$

(Non uniform) Counting Constraint Satisfaction Problem

Let **H** be a relational structure in Struct(σ)

Problem

#CSP(**H**) *is the following counting problem:*
Given a finite structure **A** $\in$ Struct(σ) *compute the number of homomorphisms from* **A** *to* **H**.

An homomorphism from $\mathbf{A} = (A; E_1, \ldots, E_n)$ to $\mathbf{H} = (H; R_1, \ldots, R_n)$ is a mapping $f : A \to H$ such that for every $1 \leq i \leq n$ and every tuple $(a_1, \ldots, a_r) \in A^*$

$$(a_1, \ldots, a_r) \in E_i \Rightarrow (f(a_1), \ldots, f(a_r)) \in R_i$$

Choosing appropriately **H** one can encode many counting problems arising in logic, artificial intelligence, computational complexity, ...

Descriptive complexity of counting problems

Let $\mathcal{C} : \text{Struct}(\sigma) \to \mathbf{Z}^+$ be a counting problem and let φ be a (first order) formula with signature σ.

Definition

We say that φ *defines* $\mathcal{C}$ if for every $\mathbf{A} \in \text{Struct}(\sigma)$:

$\mathcal{C}(\mathbf{A}) = \#$ of assignments (to the free variables) that make φ true on $\mathbf{A}$

Note: the free variables of φ might be of first and second order.

Descriptive complexity of counting problems

Let $\mathcal{C} : \text{Struct}(\sigma) \to \mathbf{Z}^+$ be a counting problem and let φ be a (first order) formula with signature σ.

Definition

We say that φ *defines* $\mathcal{C}$ if for every $\mathbf{A} \in \text{Struct}(\sigma)$:

$\mathcal{C}(\mathbf{A}) = \#$ of assignments (to the free variables) that make φ true on $\mathbf{A}$

Note: the free variables of φ might be of first and second order.

Example: #INDEPENDENT SET is defined by the formula ($\sigma = \{E\}$)

$$\forall x, y \ \ \neg E(x,y) \vee \neg I(x) \vee \neg I(y)$$

Descriptive complexity of counting problems

Let $\mathcal{C} : \text{Struct}(\sigma) \to \mathbf{Z}^+$ be a counting problem and let φ be a (first order) formula with signature σ.

Definition

We say that φ *defines* $\mathcal{C}$ if for every $\mathbf{A} \in \text{Struct}(\sigma)$:
$\mathcal{C}(\mathbf{A}) = \#$ of assignments (to the free variables) that make φ true on $\mathbf{A}$

Note: the free variables of φ might be of first and second order.

Example: #INDEPENDENT SET is defined by the formula ($\sigma = \{E\}$)

$$\forall x, y \;\; \neg E(x, y) \vee \neg I(x) \vee \neg I(y)$$

Theorem (Saluja et al. 95)

$$\#\Sigma_0 = \#\Pi_0 \subset \#\Sigma_1 \subset \#\Pi_1 \subset \#\Sigma_2 \subset \#\Pi_2 = \#FO (= \#P)$$

$\#L$ is the class counting problems definable by a formula in logic L

The logic $\mathsf{RH\Pi}_1$

Observe that $\#\mathrm{CSP}(\mathbf{H}) \in \#\Pi_1$ for every $\mathbf{H}$.

Definition

Let $\mathsf{RH\Pi}_1 \subseteq \Pi_1$ be the class of all formulas of the form $\forall \mathbf{y}\, \psi$ where ψ is a quantifier-free CNF in wich every clause contains at most one occurrence of a unnegated second-order variable and at most one occurrence of a negated second-order variable.

The logic $\mathrm{RH}\Pi_1$

Observe that $\#\mathrm{CSP}(\mathbf{H}) \in \#\Pi_1$ for every $\mathbf{H}$.

Definition

Let $\mathrm{RH}\Pi_1 \subseteq \Pi_1$ be the class of all formulas of the form $\forall \mathbf{y}\, \psi$ where ψ is a quantifier-free CNF in wich every clause contains at most one occurrence of a unnegated second-order variable and at most one occurrence of a negated second-order variable.

Example: Assume $\sigma = \{R, S\}$. The formula

$$\begin{array}{rc} \forall x, y, z & (R(x,u) \vee R(y,v) \vee \neg T(x,u) \vee U(v)) \\ & \wedge \\ & (U(x) \vee \neg U(v)) \\ & \wedge \\ & (R(x,y) \vee \neg R(y,z) \vee \neg S(x,z)) \end{array}$$

belongs to $\mathrm{RH}\Pi_1$.

Why do we care about $\#\mathrm{RH}\Pi_1$?

Theorem (Bulatov 08, Dyer and Richerby 10)

For any $\mathbf{H}$, $\#\mathrm{CSP}(\mathbf{H})$ *is either in FP or it is #P-complete. Furthermore there is a simple and decidable criterion.*

It turns out that $\#\mathrm{CSP}(\mathbf{H})$ is #P-complete for most $\mathbf{H}$.

Approximate counting

Theorem (Dyer et al. 10)

Let **H** *be a structure with 2 elements. Then:*

- *If* **H** *is invariant under* $x \oplus y \oplus z$ *then* $\#\mathrm{CSP}(\mathbf{H})$ *has a fully polynomial randomized approximation scheme (FPRAS),*
- *else,* **H** *is invariant under* $\vee$ *and* $\wedge$ *then* $\#\mathrm{CSP}(\mathbf{H})$ *is in* $\#\,\mathrm{RH}\Pi_1$*,*
- *else,* $\#\mathrm{CSP}(\mathbf{H})$ *is #P-complete under approximation preserving (AP) reductions.*

More generally, it appears a new class that seems to lie strictly between the problems that have a FPRAS and the problems for which approximation is hard. This class is precisely the closure under AP-reduction of $\#\,\mathrm{RH}\Pi_1$.

What we would like to do?

Our (ultimate) research goal:

Characterize those $\mathbf{H}$ such that $\#\mathrm{CSP}(\mathbf{H})$ is in $\#\,\mathrm{RH}\Pi_1$.

What we would like to do?

Our (ultimate) research goal:

Characterize those $\mathbf{H}$ such that $\#\mathrm{CSP}(\mathbf{H})$ is in $\#\,\mathrm{RH}\Pi_1$.

What we actually do?

In this talk:

We investigate for which $\mathbf{H}$, $\#\mathrm{CSP}(\mathbf{H})$ is in the monotone fragment of $\#\,\mathrm{RH}\Pi_1$.

Definition

Monotone $\mathrm{RH}\Pi_1$ is the fragment of $\mathrm{RH}\Pi_1$ in which every relation symbol from σ can only appear negatively.

Additionally, we deal only with unordered structures.

Main result

Theorem

Let **H** *be a structure.*

1. *If* **H** *has polymorphisms* $x \sqcap y$ *and* $x \sqcup y$ *for some distributive lattice* $(H; \sqcap, \sqcup)$ *then* $\#\mathrm{CSP}(\mathbf{H})$ *is in* $\#\mathrm{RH}\Pi_1$*.*
2. *Furthermore, if* **H** *contains equality then the converse also holds.*

Note that equality has all polymorphisms. This implies that if membership to $\#\mathrm{RH}\Pi_1$ is controlled by the polymorphisms of **H** then the converse of (1) holds.

Sketch of the proof (Sufficient condition)

Let **H** be a structure that has polymorphisms $x \sqcap y$ and $x \sqcup y$ for some distributive lattice $(H; \sqcap, \sqcup)$.

Sketch of the proof (Sufficient condition)

Let **H** be a structure that has polymorphisms $x \sqcap y$ and $x \sqcup y$ for some distributive lattice $(H; \sqcap, \sqcup)$.

As $(H, \sqcup, \sqcap)$ is distributive we can rename the elements of H using boolean vectors so that $\sqcup$ and $\sqcap$ are $\vee$ and $\wedge$ resp. applied coordenate-wise. This implies that every instance of $\#\,\mathrm{CSP}(\mathbf{H})$ can be rewritten as a boolean #CSP in which all relations are invariant under $\vee$ and $\wedge$.

Sketch of the proof (Necessary condition)

Assume that $\#\mathrm{CSP}(\mathbf{H})$ is definable by a formula φ in monotone $\mathsf{RH}\Pi_1$.
Then:

Sketch of the proof (Necessary condition)

Assume that $\#\mathrm{CSP}(\mathbf{H})$ is definable by a formula φ in monotone $\mathsf{RH\Pi}_1$. Then:

- One can assume that φ, additionally, does not have free first-order variables and that all second-order free variables are monadic.

Sketch of the proof (Necessary condition)

Assume that $\#\mathrm{CSP}(\mathbf{H})$ is definable by a formula φ in monotone $\mathrm{RH\Pi_1}$. Then:

- One can assume that φ, additionally, does not have free first-order variables and that all second-order free variables are monadic.
- If, furthermore, $\mathbf{H}$ contains equality, then one can assume that every clause of φ contains at most one occurrence of a predicate in σ.

Sketch of the proof (Necessary condition)

Assume that #CSP($\mathbf{H}$) is definable by a formula φ in monotone $\mathsf{RH}\Pi_1$. Then:

- One can assume that φ, additionally, does not have free first-order variables and that all second-order free variables are monadic.
- If, furthermore, $\mathbf{H}$ contains equality, then one can assume that every clause of φ contains at most one occurrence of a predicate in σ.
- There exists a structure $\mathbf{J}$ such that for every structure $\mathbf{A} \in \mathsf{Struct}(\sigma)$, the number of assignments that make φ true on $\mathbf{A}$ is equal to the number of homomorphisms from $\mathbf{A}$ to $\mathbf{J}$. By construction $\mathbf{J}$ has polymorphisms $x \sqcap y$ and $x \sqcup y$ for some distributive lattice $(\sqcap, \sqcup)$

Sketch of the proof (Necessary condition)

Assume that $\#\mathrm{CSP}(\mathbf{H})$ is definable by a formula φ in monotone $\mathrm{RH}\Pi_1$. Then:

- One can assume that φ, additionally, does not have free first-order variables and that all second-order free variables are monadic.
- If, furthermore, $\mathbf{H}$ contains equality, then one can assume that every clause of φ contains at most one occurrence of a predicate in σ.
- There exists a structure $\mathbf{J}$ such that for every structure $\mathbf{A} \in \mathrm{Struct}(\sigma)$, the number of assignments that make φ true on $\mathbf{A}$ is equal to the number of homomorphisms from $\mathbf{A}$ to $\mathbf{J}$. By construction $\mathbf{J}$ has polymorphisms $x \sqcap y$ and $x \sqcup y$ for some distributive lattice $(\sqcap, \sqcup)$
- We have that $\#\,\hom(\mathbf{A}, \mathbf{H}) = \#\,\hom(\mathbf{A}, \mathbf{J})$ for every $\mathbf{A}$. It follows by the Möbius inversion formula that $\mathbf{H}$ and $\mathbf{J}$ are isomorphic.

Definability by Linear Datalog

As a consequence of our result we obtain some results on the definability by linear Datalog.

Observation

Every linear Datalog program is equivalent to a monotone $\#\mathrm{RH}\Pi_1$ *sentence in which every clause contains an unnegated second-order variable. The converse also holds.*

Theorem

Let $\mathbf{H}$ *be a structure.*

1. *If* $\mathbf{H}$ *has polymorphisms* $x \sqcap y$ *and* $x \sqcup y$ *and* $\top$ *for some distributive lattice* $(H; \sqcap, \sqcup)$ *then* $\#\mathrm{CSP}(\mathbf{H})$ *is definable by a linear datalog program.*
2. *Furthermore, if* $\mathbf{H}$ *contains equality then the converse also holds.*

Open problems

- Prove or disprove: For every structure **H**, if $\#\text{CSP}(\mathbf{H})$ is in $\#\,\text{RH}\Pi_1$ then also is in monotone $\#\,\text{RH}\Pi_1$.
- Prove or disprove: For every structure **H**, if $\#\text{CSP}(\mathbf{H})$ is in $\#\,\text{RH}\Pi_1$ then so is $\#\text{CSP}(\mathbf{H} \cup \{\text{eq}\})$

THANKS FOR YOUR ATTENTION!!!!