# Descriptive complexity for counting classes

Descriptive Complexity
ALMA Spring 2020

# The class #P

A function $f : \{0,1\}^* \to \mathbb{N}$ is in $\#P$ if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing Machine $M$ such that for every $x \in \{0,1\}^*$:

$$f(x) = |\{y \in \{0,1\}^{p(|x|)} \;:\; M(x,y) = 1\}|$$

For a nondeterministic polynomial-time Turing Machine $M$, we define the function $acc_M(x) : \{0,1\}^* \to \mathbb{N}$ as follows:

$$acc_M(x) = \#\text{ accepting paths of M on input } x$$

Then $\#P$ is the class:

$$\#P = \{acc_M \,|\, M \text{ is a PNTM } \}$$

# Counting vs Decision

- Every decision problem in $NP$ has a counting version in $\#P$ For example, $HamiltonCycle \in NP$ and $\#HamiltonCycle \in \#P$
- $FP \subseteq \#P \subseteq FPSPACE$
- $NP \subseteq P^{\#P[1]}$
- If $FP = \#P$, then $P = NP$

**Toda's Theorem**

$$PH \subseteq P^{\#P[1]}$$

# Reductions between functions

- Cook (poly-time Turing)

$$f \leqslant_T^p g : \; f \in FP^g$$

- Karp / parsimonious (poly-time many one)

$$f \leqslant_m^p g : \; \exists h \in FP, \; \forall x \; f(x) = g(h(x))$$

- $\#SAT$ is $\#P$-complete under parsimonious reductions.
- $\#PerfectMatching$ is $\#P$-complete under Turing reductions.

- A $\#P$-complete problem under parsimonious reductions
  1. has an $NP$-complete decision version, e.g. $SAT$ is $NP$-complete,
  2. cannot be aprroximated efficiently unless $RP = NP$.
- There are $\#P$-complete problem under Turing reductions that
  1. have a decision in $P$, e.g. $PerfectMatcing$ is in $P$,
  2. admit an FPRAS, e.g. $\#DNF$.

# Definition of an FPRAS

**Definition**

A **fully polynomial randomised approximation scheme (FPRAS)** for a function $f : \Sigma^* \to \mathbb{N}$ is a probabilistic TM that takes as input an instance $x$ of $f$, $\varepsilon > 0$ and $0 < \delta < 1$, and produces as output an integer random variable $Y$ satisfying the condition

$$Pr((1-\varepsilon)f(x) \leq Y \leq (1+\varepsilon)f(x)) \geq 1-\delta.$$

It also runs in time poly$(|x|, 1/\varepsilon)$.

- For a self-reducible counting problem,

randomized approximation poly-time algorithm within a polynomial factor $\Rightarrow$ *FPRAS*

# #PE and TotP

For a counting function $f : \{0,1\}^* \to \mathbb{N}$ we define the related language $L_f = \{x \mid f(x) > 0\}$. Then,

$$\#PE = \{f \mid f \in \#P \text{ and } L_f \in P\}$$

For a nondeterministic polynomial-time Turing Machine $M$, we define the function $tot_M(x) : \{0,1\}^* \to \mathbb{N}$ as follows:

$$tot_M(x) = \# \text{ paths of M on input } x - 1$$

Then $TotP$ is the class:

$$TotP = \{tot_M \mid M \text{ is a PNTM }\}$$

- $TotP$ is the Karp-closure of all self-reducible $\#PE$ functions.

For any $\#A \in \#P$, there exists:

- a randomized polynomial-time (in $|x|$ and $1/\varepsilon$) algorithm, which using an $NP$-oracle, approximates $\#A$ within ratio $(1+\varepsilon)$.
- a deterministic polynomial-time (in $|x|$ and $1/\varepsilon$) algorithm, which using an $\Sigma_2^p$-oracle, approximates $\#A$ within ratio $(1+\varepsilon)$.

# Our interests today

- Descriptive Complexity for counting
- How can descriptive complexity contribute to the classification of counting problems with respect to their approximability?

# Fagin's Theorem (reminder)

## Theorem (Fagin)

$\exists$**SO captures NP**: A language $L$ is *NP* computable iff it is definable by an existential second-order sentence, i.e. iff there is a sentence $\phi(\mathbf{T})$ with predicate symbols from $\mathbf{T} \cup \sigma$ such that

$$\mathcal{A} \in L \Leftrightarrow \mathcal{A} \models \exists \mathbf{T} \phi(\mathbf{T})$$

where $\mathcal{A}$ is an ordered finite structure over the vocabulary $\sigma$.

## Corollary (Cook)

*SAT* is *NP*-complete

- $3COL$: A graph can be encoded by a finite structure $\mathcal{A} = \{(x_1, ..., x_n), E^2\}$ and $\psi_{3COL} = (\exists R^1)(\exists B^1)(\exists G^1)(\forall x)\big[(R(x) \vee B(x) \vee G(x)) \wedge (\forall y)\Big(E(x,y) \to \neg(R(x) \wedge R(y)) \wedge \neg(B(x) \wedge B(y)) \wedge \neg(G(x) \wedge G(y))\Big)\big]$
- $SAT$: A boolean formula in conjunctive normal form can be encoded by a finite structure $\mathcal{A} = \{(v_1, ..., v_n, c_1, ..., c_m), C^1, P^2, N^2\}$ and $\psi_{SAT} = (\exists S^1)(\forall c)(\exists v)\big[C(c) \to (P(c,v) \wedge S(v)) \vee (N(c,v) \wedge \neg S(v))\big]$

- Let $\sigma$ be a vocabulary containing a relation symbol $\leq$.
- Let $f$ be a counting function with finite structures $\mathcal{A}$ over $\sigma$, as instances.
- Let $\mathbf{T} = \{T_1, ..., T_r\}$ and $\mathbf{z} = \{z_1, ..., z_m\}$ be sequences of predicate symbols and first-order variables respectively.

A counting function belongs to $\#FO$ iff there is a first-order formula with predicate symbols from $\mathbf{T} \cup \sigma$ and free first-order variables from $\mathbf{z}$ such that

$$f(\mathcal{A}) = |\{< \mathbf{T}, \mathbf{z} > : \mathcal{A} \models \phi(\mathbf{T}, \mathbf{z})\}|$$

- Let $\sigma$ be a vocabulary containing a relation symbol $\leq$.
- Let $f$ be a counting function with finite structures $\mathcal{A}$ over $\sigma$, as instances.
- Let $\mathbf{T} = \{T_1, ..., T_r\}$ and $\mathbf{z} = \{z_1, ..., z_m\}$ be sequences of predicate symbols and first-order variables respectively.

A counting function belongs to $\#FO$ iff there is a first-order formula with predicate symbols from $\mathbf{T} \cup \sigma$ and free first-order variables from $\mathbf{z}$ such that

$$f(\mathcal{A}) = |\{< \mathbf{T}, \mathbf{z} > : \mathcal{A} \models \phi(\mathbf{T}, \mathbf{z})\}|$$

- If the formula $\phi$ in the above definition is a $\Sigma_i$ ($\Pi_i$ resp.), $i \in \mathbb{N}$, then we obtain the subclasses $\#\Sigma_i$ ($\#\Pi_i$ resp.), $i \in \mathbb{N}$, of $\#FO$.

# Saluja, Sabrahmanyama and Thakur (1995)

**Theorem**

The class $\#P$ coincides with the class $\#FO$.
In fact, $\#\Pi_2$ captures $\#FO$.

*Proof.* $\#FO \subseteq \#P$: The NP machine guesses a tuple $< \mathbf{T}, \mathbf{z} >$ and verifies in polynomial time that $\mathcal{A} \models \phi(\mathbf{T}, \mathbf{z})$.

$\#P \subseteq \#FO$: For an $f \in \#P$, the decision version $L_f \in NP$. By Fagin's Theorem, $\mathcal{A} \in L_f$ iff $\mathcal{A} \models \exists \mathbf{T} \phi(\mathbf{T})$. The formula $\phi$ is such that every accepting computation of the NP machine on input $\mathcal{A}$ corresponds to a unique value of $\mathbf{T}$ that satisfies $\phi(\mathbf{T})$. So, the number of accepting paths is equal to $|\{< \mathbf{T} >: \mathcal{A} \models \phi(\mathbf{T})\}|$.
Furthermore, from the proof of Fagin's Theorem, $\phi$ is a $\Pi_2$ first-order formula. □

- $\#DNF$: A DNF formula can be encoded by a finite structure $\mathcal{A} = \{(v_1, ..., v_n, d_1, ..., d_m), D^1, P^2, N^2\}$ and $f_{\#DNF}(\mathcal{A}) = |\{T : \mathcal{A} \models \exists d \, \forall v \Big( D(d) \wedge \big( P(d, v) \rightarrow T(v) \big) \wedge \big( N(d, v) \rightarrow \neg T(v) \big) \Big) \}|$.
  Hence $\#DNF \in \#\Sigma_2$.

- $\#DNF$: A DNF formula can be encoded by a finite structure $\mathcal{A} = \{(v_1, ..., v_n, d_1, ..., d_m), D^1, P^2, N^2\}$ and $f_{\#DNF}(\mathcal{A}) = |\{T : \mathcal{A} \models \exists d \, \forall v \Big(D(d) \wedge \big(P(d,v) \to T(v)\big) \wedge \big(N(d,v) \to \neg T(v)\big)\Big)\}|$. Hence $\#DNF \in \#\Sigma_2$.

- $\#3CNF$: A boolean formula in conjunctive normal form with three literals per clause can be encoded by a finite structure $\mathcal{A} = \{(v_1, ..., v_n), C_0^3, C_1^3, C_2^3, C_3^3\}$ and $f_{\#3CNF}(\mathcal{A}) = |\{T : \mathcal{A} \models (\forall x_1)(\forall x_2)(\forall x_3)\Big[\big(C_0(x_1, x_2, x_3) \to (T(x_1) \wedge T(x_2) \wedge T(x_3))\big) \wedge \big(C_1(x_1, x_2, x_3) \to (\neg T(x_1) \wedge T(x_2) \wedge T(x_3))\big) \wedge \big(C_2(x_1, x_2, x_3) \to (\neg T(x_1) \wedge \neg T(x_2) \wedge T(x_3))\big) \wedge \big(C_3(x_1, x_2, x_3) \to (\neg T(x_1) \wedge \neg T(x_2) \wedge \neg T(x_3))\big)\Big]\}|$. Hence $\#3CNF \in \#\Pi_1$.

- $\#DNF$: A DNF formula can be encoded by a finite structure $\mathcal{A} = \{(v_1, \ldots, v_n, d_1, \ldots, d_m), D^1, P^2, N^2\}$ and $f_{\#DNF}(\mathcal{A}) = |\{T : \mathcal{A} \models \exists d\, \forall v \Big( D(d) \wedge \big(P(d, v) \rightarrow T(v)\big) \wedge \big(N(d, v) \rightarrow \neg T(v)\big)\Big)\}|$. Hence $\#DNF \in \#\Sigma_2$.

- $\#3CNF$: A boolean formula in conjunctive normal form with three literals per clause can be encoded by a finite structure $\mathcal{A} = \{(v_1, \ldots, v_n), C_0^3, C_1^3, C_2^3, C_3^3\}$ and $f_{\#3CNF}(\mathcal{A}) = |\{T : \mathcal{A} \models (\forall x_1)(\forall x_2)(\forall x_3)\Big[\big(C_0(x_1, x_2, x_3) \rightarrow (T(x_1) \wedge T(x_2) \wedge T(x_3))\big) \wedge \big(C_1(x_1, x_2, x_3) \rightarrow (\neg T(x_1) \wedge T(x_2) \wedge T(x_3))\big) \wedge \big(C_2(x_1, x_2, x_3) \rightarrow (\neg T(x_1) \wedge \neg T(x_2) \wedge T(x_3))\big) \wedge \big(C_3(x_1, x_2, x_3) \rightarrow (\neg T(x_1) \wedge \neg T(x_2) \wedge \neg T(x_3))\big)\Big]\}|$. Hence $\#3CNF \in \#\Pi_1$.

- $\#SAT$: A boolean formula in conjunctive normal form can be encoded by a finite structure $\mathcal{A} = \{(v_1, \ldots, v_n, c_1, \ldots, c_m), C^1, P^2, N^2\}$ and $f_{\#SAT}(\mathcal{A}) = |\{T : \mathcal{A} \models (\forall c)(\exists v)\Big[C(c) \rightarrow \big(P(c, v) \wedge T(v)\big) \vee \big(N(c, v) \wedge \neg T(v)\big)\Big]\}|$. Hence $\#SAT \in \#\Pi_2$.

# Hierarchy in $\#FO$

Proposition 1:

$$\begin{array}{ccccc} & & \#\Pi_1 & & \\ & \subseteq & & \subseteq & \\ \#\Sigma_0 = \#\Pi_0 & & & & \#\Sigma_2 \subseteq \#\Pi_2 = \#\mathrm{P}. \\ & \subseteq & & \subseteq & \\ & & \#\Sigma_1 & & \end{array}$$

Proposition 2:

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

*Proof.* $\#\Sigma_1 \subseteq \#\Pi_1$:
Let $f \in \#\Sigma_1$ with $f(\mathcal{A}) = |\{< \mathbf{T}, \mathbf{z} >: \mathcal{A} \models \exists \mathbf{x}\, \psi(\mathbf{x}, \mathbf{z}, \mathbf{T})\}|$.
Instead of counting the tuples $< \mathbf{T}, \mathbf{z} >$, we count the tuples $< \mathbf{T}, (\mathbf{z}, \mathbf{x}^*) >$ where $\mathbf{x}^*$ is the lexicographically smallest $\mathbf{x}$ such that $\mathcal{A} \models \psi(\mathbf{x}, \mathbf{z}, \mathbf{T})$. Let $\theta(\mathbf{x}, \mathbf{x}^*)$ be the quantifier-free formula which expresses that $\mathbf{x}^*$ is lexicographically smaller than $\mathbf{x}$ under $\leq$. Then,

$$f(\mathcal{A}) = |\{< \mathbf{T}, (\mathbf{z}, \mathbf{x}^*) >: \mathcal{A} \models \psi(\mathbf{x}^*, \mathbf{z}, \mathbf{T}) \wedge (\forall \mathbf{x})\big(\psi(\mathbf{x}, \mathbf{z}, \mathbf{T}) \rightarrow \theta(\mathbf{x}, \mathbf{x}^*)\big)\}|$$

The second part of the proof includes the following:

- $\#3DNF \in \#\Sigma_1 \setminus \#\Sigma_0$
- $\#3CNF \in \#\Pi_1 \setminus \#\Sigma_1$
- $\#DNF \in \#\Sigma_2 \setminus \#\Pi_1$
- $\#HamiltonCycle \in \#\Pi_2 \setminus \#\Sigma_2$

$\square$

The above classes are not closed under parsimonious reductions. For example, $\#3CNF \in \#\Pi_1$, but $\#HamiltonCycle \notin \#\Pi_1$.

- Every counting function in $\#\Sigma_0$ is computable in deterministic polynomial time.
- Every counting function in $\#\Sigma_1$ has an FPRAS.
    1. Every $\#\Sigma_1$ function is reducible to a restricted version of $\#DNF$ under a reducibility which preserves approximability.
    2. $\#DNF$ has an FPRAS.

- Poly-time product reduction

$$f \leqslant_{pr} g: \ \exists h_1, h_2 \in FP, \forall x \, f(x) = g(h_1(x)) \cdot h_2(|x|)$$

**Definition**

For any $k \in \mathbb{N}$, $\#k \cdot logDNF$ is the problem of counting the satisfying assignments for a DNF formula with at most $k \cdot logn$ literals in each disjunct, where $n$ is the number of variables in the formula.

**Proposition**

For every counting function $f \in \#\Sigma_1$ there is a positive constant $k$ such that $f \leqslant_{pr} \#k \cdot logDNF$.

*Proof.* $f(\mathcal{A}) = |\{< \mathbf{T}, \mathbf{z} >: \mathcal{A} \models \exists \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z}, \mathbf{T}\}|$, where $\psi$ is in DNF, $\mathbf{y}$ has arity $p$, $\mathbf{z}$ has arity $m$ and $T_i$ has arity $a_i$, $1 \leq i \leq r$.

- For every $\mathbf{z_i} \in A^m$, we write $\exists \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z_i}, \mathbf{T}\}$ as a disjunct $\bigvee_{j=1}^{|A|^p} \psi(\mathbf{y_j}, \mathbf{z_i}, \mathbf{T}\}$.

*Proof.* $f(\mathcal{A}) = |\{< \mathbf{T}, \mathbf{z} >: \mathcal{A} \models \exists \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z}, \mathbf{T}\}|$, where $\psi$ is in DNF, $\mathbf{y}$ has arity $p$, $\mathbf{z}$ has arity $m$ and $T_i$ has arity $a_i$, $1 \leq i \leq r$.

• For every $\mathbf{z_i} \in A^m$, we write $\exists \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z_i}, \mathbf{T}\}$ as a disjunct $\bigvee_{j=1}^{|A|^p} \psi(\mathbf{y_j}, \mathbf{z_i}, \mathbf{T}\}$.

• We replace every subformula that is satisfied by $\mathcal{A}$ by TRUE and every subformula that is not satisfied by $\mathcal{A}$ by FALSE and we obtain $\psi'(\mathbf{z_i}, \mathbf{T})$.

*Proof.* $f(\mathcal{A}) = |\{< \mathbf{T}, \mathbf{z} >: \mathcal{A} \models \exists \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z}, \mathbf{T}\}|$, where $\psi$ is in DNF, $\mathbf{y}$ has arity $p$, $\mathbf{z}$ has arity $m$ and $T_i$ has arity $a_i$, $1 \leq i \leq r$.

• For every $\mathbf{z_i} \in A^m$, we write $\exists \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z_i}, \mathbf{T}\}$ as a disjunct $\bigvee_{j=1}^{|A|^p} \psi(\mathbf{y_j}, \mathbf{z_i}, \mathbf{T}\}$.

• We replace every subformula that is satisfied by $\mathcal{A}$ by TRUE and every subformula that is not satisfied by $\mathcal{A}$ by FALSE and we obtain $\psi'(\mathbf{z_i}, \mathbf{T})$.

• The formula $\psi'(\mathbf{z_i}, \mathbf{T})$ is a propositional formula in DNF with variables of the form $T_i(w_i)$, $w_i \in A^{a_i}$, $1 \leq i \leq r$.

*Proof.* $f(\mathcal{A}) = |\{< \mathbf{T}, \mathbf{z} >: \mathcal{A} \models \exists \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z}, \mathbf{T}\}|$, where $\psi$ is in DNF, $\mathbf{y}$ has arity $p$, $\mathbf{z}$ has arity $m$ and $T_i$ has arity $a_i$, $1 \leq i \leq r$.

• For every $\mathbf{z_i} \in A^m$, we write $\exists \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z_i}, \mathbf{T}\}$ as a disjunct $\bigvee_{j=1}^{|A|^p} \psi(\mathbf{y_j}, \mathbf{z_i}, \mathbf{T}\}$.

• We replace every subformula that is satisfied by $\mathcal{A}$ by TRUE and every subformula that is not satisfied by $\mathcal{A}$ by FALSE and we obtain $\psi'(\mathbf{z_i}, \mathbf{T})$.

• The formula $\psi'(\mathbf{z_i}, \mathbf{T})$ is a propositional formula in DNF with variables of the form $T_i(w_i)$, $w_i \in A^{a_i}$, $1 \leq i \leq r$.

• We introduce $l$ new variables $x_1, ..., x_l$, where $l = log(|A|^m)$. The binary representation $s$ of an integer between 0 and $2^l - 1$ can be encoded by the conjunction $x(s)$ of these variables in which $x_i$ appears negated iff the $i^{th}$ bit of $s$ is 0.

*Proof.* $f(\mathcal{A}) = |\{< \mathbf{T}, \mathbf{z} >: \mathcal{A} \models \exists \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z}, \mathbf{T}\}|$, where $\psi$ is in DNF, $\mathbf{y}$ has arity $p$, $\mathbf{z}$ has arity $m$ and $T_i$ has arity $a_i$, $1 \leq i \leq r$.

• For every $\mathbf{z_i} \in A^m$, we write $\exists \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z_i}, \mathbf{T}\}$ as a disjunct $\bigvee_{j=1}^{|A|^p} \psi(\mathbf{y_j}, \mathbf{z_i}, \mathbf{T}\}$.

• We replace every subformula that is satisfied by $\mathcal{A}$ by TRUE and every subformula that is not satisfied by $\mathcal{A}$ by FALSE and we obtain $\psi'(\mathbf{z_i}, \mathbf{T})$.

• The formula $\psi'(\mathbf{z_i}, \mathbf{T})$ is a propositional formula in DNF with variables of the form $T_i(w_i)$, $w_i \in A^{a_i}$, $1 \leq i \leq r$.

• We introduce $l$ new variables $x_1, ..., x_l$, where $l = log(|A|^m)$. The binary representation $s$ of an integer between 0 and $2^l - 1$ can be encoded by the conjunction $x(s)$ of these variables in which $x_i$ appears negated iff the $i^{th}$ bit of $s$ is 0.

• We define
$\theta_{\mathcal{A}} = [\psi'(\mathbf{z_0}, \mathbf{T}) \wedge x(0)] \vee [\psi'(\mathbf{z_1}, \mathbf{T}) \wedge x(1)] \vee ... \vee [\psi'(\mathbf{z_{|A|^m-1}}, \mathbf{T}) \wedge x(|A|^m - 1)]$.

*Proof.* $f(\mathcal{A}) = |\{< \mathbf{T}, \mathbf{z} >: \mathcal{A} \models \exists \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z}, \mathbf{T}\}|$, where $\psi$ is in DNF, $\mathbf{y}$ has arity $p$, $\mathbf{z}$ has arity $m$ and $T_i$ has arity $a_i$, $1 \leq i \leq r$.

• For every $\mathbf{z_i} \in A^m$, we write $\exists \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z_i}, \mathbf{T}\}$ as a disjunct $\bigvee_{j=1}^{|A|^p} \psi(\mathbf{y_j}, \mathbf{z_i}, \mathbf{T}\}$.

• We replace every subformula that is satisfied by $\mathcal{A}$ by TRUE and every subformula that is not satisfied by $\mathcal{A}$ by FALSE and we obtain $\psi'(\mathbf{z_i}, \mathbf{T})$.

• The formula $\psi'(\mathbf{z_i}, \mathbf{T})$ is a propositional formula in DNF with variables of the form $T_i(w_i)$, $w_i \in A^{a_i}$, $1 \leq i \leq r$.

• We introduce $l$ new variables $x_1, ..., x_l$, where $l = log(|A|^m)$. The binary representation $s$ of an integer between 0 and $2^l - 1$ can be encoded by the conjunction $x(s)$ of these variables in which $x_i$ appears negated iff the $i^{th}$ bit of $s$ is 0.

• We define
$\theta_{\mathcal{A}} = [\psi'(\mathbf{z_0}, \mathbf{T}) \wedge x(0)] \vee [\psi'(\mathbf{z_1}, \mathbf{T}) \wedge x(1)] \vee ... \vee [\psi'(\mathbf{z_{|A|^m-1}}, \mathbf{T}) \wedge x(|A|^m - 1)]$.

• Finally, $\theta_{\mathcal{A}}$ can be easily rewritten as a DNF formula with variables of the form $T_i(w_i)$ and $x_1, ..., x_l$. Also, each disjunct will contain $\mathcal{O}(logn)$ literals.

*Proof.* $f(\mathcal{A}) = |\{<\mathbf{T},\mathbf{z}>: \mathcal{A} \models \exists \mathbf{y}\, \psi(\mathbf{y},\mathbf{z},\mathbf{T}\}|$, where $\psi$ is in DNF, $\mathbf{y}$ has arity $p$, $\mathbf{z}$ has arity $m$ and $T_i$ has arity $a_i$, $1 \leq i \leq r$.

• For every $\mathbf{z_i} \in A^m$, we write $\exists \mathbf{y}\, \psi(\mathbf{y},\mathbf{z_i},\mathbf{T}\}$ as a disjunct $\bigvee_{j=1}^{|A|^p} \psi(\mathbf{y_j},\mathbf{z_i},\mathbf{T}\}$.

• We replace every subformula that is satisfied by $\mathcal{A}$ by TRUE and every subformula that is not satisfied by $\mathcal{A}$ by FALSE and we obtain $\psi'(\mathbf{z_i},\mathbf{T})$.

• The formula $\psi'(\mathbf{z_i},\mathbf{T})$ is a propositional formula in DNF with variables of the form $T_i(w_i)$, $w_i \in A^{a_i}$, $1 \leq i \leq r$.

• We introduce $l$ new variables $x_1, ..., x_l$, where $l = log(|A|^m)$. The binary representation $s$ of an integer between 0 and $2^l - 1$ can be encoded by the conjunction $x(s)$ of these variables in which $x_i$ appears negated iff the $i^{th}$ bit of $s$ is 0.

• We define
$\theta_{\mathcal{A}} = [\psi'(\mathbf{z_0},\mathbf{T}) \wedge x(0)] \vee [\psi'(\mathbf{z_1},\mathbf{T}) \wedge x(1)] \vee ... \vee [\psi'(\mathbf{z_{|A|^m-1}},\mathbf{T}) \wedge x(|A|^m - 1)]$.

• Finally, $\theta_{\mathcal{A}}$ can be easily rewritten as a DNF formula with variables of the form $T_i(w_i)$ and $x_1, ..., x_l$. Also, each disjunct will contain $\mathcal{O}(logn)$ literals.

• Let $c(\mathcal{A})$ be the variables of the form $T_i(w_i)$ that do not appear in $\theta_{\mathcal{A}}$. It holds that:

$$f(\mathcal{A}) = 2^{c(\mathcal{A})} \cdot \text{ (the number of satisfying assignments of } \theta_{\mathcal{A}})$$

□

- Assuming $NP \neq RP$, the following is undecidable: Given a first-order formula $\phi(\mathbf{z}, \mathbf{T})$ over $\sigma \cup \mathbf{T}$, does the counting function defined by $\phi(\mathbf{z}, \mathbf{T})$ have a polynomial-time $(\epsilon, \delta)$ randomized approximation algorithm for some constants $\epsilon, \delta > 0$?
- Assuming $P \neq P^{\#P}$, the following is undecidable: Given a first-order formula $\phi(\mathbf{z}, \mathbf{T})$ over $\sigma \cup \mathbf{T}$, is the counting function defined by $\phi(\mathbf{z}, \mathbf{T})$ polynomial-time computable?

A counting function $f$ belongs to $\#R\Sigma_2$ iff there is a first-order formula $\psi$ with predicate symbols from $\mathbf{T} \cup \sigma$ and free first-order variables from $\mathbf{z}$ such that

$$f(\mathcal{A}) = |\{< \mathbf{T}, \mathbf{z} >: \mathcal{A} \models \exists \mathbf{x} \forall \mathbf{y}\, \phi(\mathbf{x}, \mathbf{y}, \mathbf{T}, \mathbf{z})\}|$$

where $\psi$ is quantifier-free and when it is expressed in CNF, each conjunct has at most one occurrence of a predicate symbol from $\mathbf{T}$.

## Proposition

Every function in $\#R\Sigma_2$ has an FPRAS.

*Proof.* $\#DNF$ is complete for $\#R\Sigma_2$ under product reductions. The proof is similar to the previous one. □

- The decision version of every function in $\#\Sigma_0$, $\#\Sigma_1$ and $\#R\Sigma_2$ is in $P$.
- $\#\mathit{Triangle} \in \#\Sigma_0$
- $\#\mathit{NonClique}$, $\#\mathit{NonVertexCover} \in \#\Sigma_1$,
- $\#\mathit{NonDominatingSet}$, $\#\mathit{NonEdgeDominatingSet} \in \#R\Sigma_2$.

Given a relational vocabulary $\sigma$, the set of **Quantitative Second-Order** logic formulae (or just *QSO*-formulae) over $\sigma$ is given by the following grammar:

$\alpha := \phi \mid s \mid (\alpha + \alpha) \mid (\alpha \cdot \alpha) \mid \Sigma x.\alpha \mid \Pi x.\alpha \mid \Sigma X.\alpha \mid \Pi X.\alpha$

where $\phi$ is an *SO*-formula, $s \in \mathbb{N}$, $x$ is a first-order variable and $X$ is a second-order variable (or a predicate that does not belong to $\sigma$).

- $QSO(\mathcal{L})$ is the fragment of *QSO* obtained by restricting $\phi$ to be in $\mathcal{L}$.
- *QFO* is the fragment of *QSO* where second-order sum and product are not allowed.
- $\Sigma QSO$ is the fragment of *QSO* where first- and second-order products ($\Pi x.$ and $\Pi X.$) are not allowed.

**Semantics**: Let $\mathfrak{A}$ be a structure, $v$ a first-order assignment for $\mathfrak{A}$ and $V$ a second-order assignment for $\mathfrak{A}$. Then the evaluation of a *QSO*-formula $\alpha$ over $(\mathfrak{A}, v, V)$ is defined as a function $\|\alpha\|$ that on input $(\mathfrak{A}, v, V)$ returns a number in $\mathbb{N}$.

$$[\![\varphi]\!](\mathfrak{A}, v, V) = \begin{cases} 1 & \text{if } (\mathfrak{A}, v, V) \models \varphi \\ 0 & \text{otherwise} \end{cases}$$

$$[\![s]\!](\mathfrak{A}, v, V) = s$$

$$[\![\alpha_1 + \alpha_2]\!](\mathfrak{A}, v, V) = [\![\alpha_1]\!](\mathfrak{A}, v, V) + [\![\alpha_2]\!](\mathfrak{A}, v, V)$$

$$[\![\alpha_1 \cdot \alpha_2]\!](\mathfrak{A}, v, V) = [\![\alpha_1]\!](\mathfrak{A}, v, V) \cdot [\![\alpha_2]\!](\mathfrak{A}, v, V)$$

$$[\![\Sigma x.\, \alpha]\!](\mathfrak{A}, v, V) = \sum_{a \in A} [\![\alpha]\!](\mathfrak{A}, v[a/x], V)$$

$$[\![\Pi x.\, \alpha]\!](\mathfrak{A}, v, V) = \prod_{a \in A} [\![\alpha]\!](\mathfrak{A}, v[a/x], V)$$

$$[\![\Sigma X.\, \alpha]\!](\mathfrak{A}, v, V) = \sum_{B \subseteq A^{\mathrm{arity}(X)}} [\![\alpha]\!](\mathfrak{A}, v, V[B/X])$$

$$[\![\Pi X.\, \alpha]\!](\mathfrak{A}, v, V) = \prod_{B \subseteq A^{\mathrm{arity}(X)}} [\![\alpha]\!](\mathfrak{A}, v, V[B/X])$$

Table I

THE SEMANTICS OF QSO FORMULAE.

- Counting triangles in a graph:
  $\alpha_1 = \Sigma x.\Sigma y.\Sigma z.(E(x,y) \wedge E(y,z) \wedge E(z,x) \wedge x < y \wedge y < z)$
- Counting cliques in a graph:
  $\alpha_2 = \Sigma X.\forall x \forall y (X(x) \wedge X(y) \wedge x \neq y) \rightarrow E(x,y)$
- Computing the permanent of a $(0,1)$ $n \times n$ matrix $A$,
  $perm(A) = \Sigma_{\sigma \in S_n} \Pi_{i=1}^{n} A[i, \sigma(i)]$

  $$\alpha_3 = \Sigma S.permut(S) \cdot \Pi x.(\exists y(S(x,y) \wedge M(x,y)))$$

  where $permut(S)$ is a first-order sentence that is true iff $S$ is a permutation (a total bijective function).

# Arenas, Muñoz and Riveros (2017)

Let $F$ be a fragment of $QSO$ and $C$ a counting complexity class. Then $F$ captures $C$ over ordered structures if the following conditions hold:

1. for every $\alpha \in F$, there exists $f \in C$ such that $\|\alpha\|(\mathfrak{A}) = f(\mathfrak{A})$ for every ordered structure $\mathfrak{A}$.
2. for every $f \in C$, there exists $\alpha \in F$ such that $f(\mathfrak{A}) = \|\alpha\|(\mathfrak{A})$ for every ordered structure $\mathfrak{A}$.

**Theorem**

$\Sigma QSO(FO)$ captures $\#P$ over ordered structures

# Hierarchy in $\Sigma QSO(FO)$

Proposition:

$$\#\Sigma_0 \subsetneq \#\Sigma_1 \subsetneq \Sigma\mathrm{QSO}(\Sigma_1), \qquad \#\Sigma_0 \subsetneq \Sigma\mathrm{QSO}(\Sigma_0) \subsetneq \Sigma\mathrm{QSO}(\Sigma_1)$$

$$\Sigma\mathrm{QSO}(\Sigma_1) \subsetneq \Sigma\mathrm{QSO}(\Pi_1) \subsetneq \Sigma\mathrm{QSO}(\Sigma_2) \subsetneq \Sigma\mathrm{QSO}(\Pi_2) = \#\mathrm{FO}$$

$$\Sigma\mathrm{QSO}(\Pi_1) = \#\Pi_1, \qquad \Sigma\mathrm{QSO}(\Sigma_2) = \#\Sigma_2, \qquad \Sigma\mathrm{QSO}(\Pi_2) = \#\Pi_2$$

# Robust counting classes with easy decision

- The goal is to give logical characterizations of robust subclasses of $\#PE$.
- A class is defined to be robust if it is closed under sum, multiplication and subtraction by one and it has natural complete problems.
- $\#PE$ is not robust, since it contains $\#SAT_{+1}$, but not $\#SAT$, unless $P = NP$.
- $TotP$ is robust.

# Characterization of robust subclasses of $\#PE$

- $\Sigma QSO(\Sigma_1[FO])$: **A subclass of TotP closed under sum, multiplication and subtraction by one**
- $\Sigma QSO(\Sigma_2\text{-}HORN)$: **A subclass of TotP with a natural complete problem**

# Reductions that preserve approximability

- Parsimonious and product reductions preserve approximability: If $\#A \leq \#B$ and $\#B$ has an FPRAS, then $\#A$ has also an FPRAS.
- Approximation preserving reduction: $f \leq_{AP} g$ iff there is a probabilistic oracle TM $M$ that takes as input an instance $x$ of $f$ and $0 < \varepsilon < 1$ and satisfies the following:
  1. every oracle call made by M is of the form $(w, \delta)$, where $w$ is an instance of g, and $0 < \delta < 1$ is an error bound satisfying $\delta^{-1} \leq poly(|x|, \epsilon^{-1})$,
  2. the TM $M$ meets the specification for being a randomised approximation scheme for $f$ whenever the oracle meets the specification for being a randomised approximation scheme for $g$, and
  3. the run-time of $M$ is polynomial in $|x|$ and $\varepsilon^{-1}$.
- If $f \leq_{AP} g$ and $g \leq_{AP} f$ then we say that $f$ and $g$ are AP-interreducible, and write $f \equiv_{AP} g$.

# Dyer, Goldberg, Greenhill and Jerrum (2004)

We define three counting classes to categorize counting problems with respect to their approximability:

- The class of counting problems with an FPRAS.
  For example, $\#MatchingOfAllSizes$, $\#DNF$.
- The class of counting problems AP-interridducible with $\#SAT$.
  This class contains all counting problems with $NP$-complete decision version and others, such as $\#IndependentSetOfAllSizes$.
- The class of counting problems AP-interridducible with $\#BIS$. For example, $\#P_4\text{-}Coloring$, $\#1P1NSAT$, $\#Downset$.

*Name.* $\#BIS$
*Instance.* A bipartite graph $G$.
*Output.* The number of independent sets in $G$.

*Name.* $\#P_4$-*Coloring*
*Instance.* A graph $G$.
*Output.* The number of homomorphisms from $G$ to $P_4$, where $P_4$ is the path of length 3.

*Name.* $\#Downset$
*Instance.* A partially ordered set $(X, \leq)$.
*Output.* The number of downsets in $(X, \leq)$.

*Name.* $\#1P1NSAT$
*Instance.* A Boolean formula $\phi$ in conjunctive normal form, with at most one unnegated literal per clause, and at most one negated literal.
*Output.* The number of satisfying assignments to $\phi$.

$\#BIS$, $\#P_4$-$Coloring$, $\#1P1NSAT$ and $\#Downset$:

- are AP-interridubible
- belong to $\#RH\Pi_1$

We say that a counting problem $f$ is in the class $\#RH\Pi_1$ if it can be expressed in the form

$$f(\mathcal{A}) = |\{< \mathbf{T}, \mathbf{z} >: \mathcal{A} \models \forall \mathbf{x}\, \psi(\mathbf{x}, \mathbf{z}, \mathbf{T})\}|$$

where $\psi$ is an quantifier-free CNF formula in which each clause has at most one occurrence of an unnegated predicate symbol from $\mathbf{T}$, and at most one occurrence of a negated predicate symbol from $\mathbf{T}$.

- For example,

$$f_{DS} = |\{D : \mathcal{A} \models (\forall x)(\forall y)(D(x) \wedge y \leq x \rightarrow D(y))\}|$$

- $\#1P1NSAT$ is complete for $\#RH\Pi_1$ under parsimonious reductions.
- $\#BIS$, $\#P_4$-*Coloring*, $\#1P1NSAT$ and $\#Downset$ are complete for $\#RH\Pi_1$ under AP reductions.

# References


- Sanjeev Saluja, K. V. Subrahmanyam, Madhukar N. Thakur: Descriptive Complexity of $\#P$ Functions. Computational Complexity Conference 1992: 169-184.
- Marcelo Arenas, Martin Muñoz, Cristian Riveros: Descriptive Complexity for counting complexity classes. LICS 2017: 1-12.
- Martin E. Dyer, Leslie Ann Goldberg, Catherine S. Greenhill, Mark Jerrum: The Relative Complexity of Approximate Counting Problems. Algorithmica 38(3): 471-500 (2004).