Counting Complexity: #P and subclasses

Chalki Angeliki

Advanced Topics in Algorithms and Complexity

Computation and Reasoning Laboratory
National Technical University of Athens

April 2017

Basic Definitions

- There are many problems where we want to *count* the **number** of solutions.
- Of course, this is more "difficult" than finding if a solution exists!
- We want to define the class of counting the number of solutions to **NP** problems:

Definition

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 $x \in \{0,1\}^*$:

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

Basic Definitions

Definition (Reductions between functions)

- Cook (poly-time Turing) $f \leq_T^p g$: $f \in FP^g$.
  In specific, $f \leq_{1-T}^p g \Leftrightarrow \exists h_1, h_2 \in FP, \forall x \ f(x) = h_1(x, g(h_2(x)))$.
- Karp (poly-time many-one) $f \leq_m^p g$: $\exists h \in FP, \forall x \ f(x) = g(h(x))$.

- There are two notions of **#P**-completeness.
- The counting versions of all known **NP**-complete problems are **#P**-complete! No counterexamples to this phenomenon are known, so it remains a possibility that this empirically observed relationship is actually a theorem.
- Valiant presented a Cook reduction with one oracle call from any problem in **#P** to #Perfect Matchings [Va79].
- There are **#P**-complete problems, the decision version of which is in **P**: #Perfect Matchings, #DNF, #Independent Sets.

Toda's Theorem

#P is of high complexity.

Theorem

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

- **FP** $\subseteq$ **#P** $\subseteq$ **PSPACE**.
- If **#P**=**FP**, then **P**=**NP**.
- If **P**=**PSPACE**, then **#P**=**FP**.

Relativization [Fo97]

- There exists an oracle A, such that $\mathbf{P}^A = \mathbf{PSPACE}^A$.
- If **P**=**PSPACE**, then **#P**=**P**=**PSPACE** and **#P** is closed under Cook reductions.
- Any proof that **#P** differs from **P**, or **PSPACE**, or that **#P** is not closed under Cook reductions requires nonrelativizing techniques.

Basic Definitions

Definition

A *Randomized Approximation Scheme* (*RAS*) for a function $f : \Sigma^* \to \mathbb{N}$ is a Probabilistic Turing Machine that takes as input a pair $(x, \varepsilon) \in \Sigma^* \times (0, 1)$ and produces as output an integer random variable Y satisfying the condition:

$$\mathbf{Pr}\left[e^{-\varepsilon} f(x) \leq Y \leq e^{\varepsilon} f(x)\right] \geq \frac{3}{4}$$

A RAS is said to be *fully polynomial* (*FPRAS*) if it runs in time $poly(|x|, \varepsilon^{-1})$.

Approximability of counting functions in **#P**

- There are problems in **#P** that can be solved exactly using a polynomial-time deterministic algorithm, such as #Spanning Trees, and #Perfect Matchings in planar graphs.
- There are **#P**-complete problems under Cook reductions which admit FPRAS, such as #Matchings, and #DNF.
- There is no polynomial-time deterministic algorithm for a **#P**-complete problem, unless **P**=**NP** (and **#P**=**FP**).
- There is no FPRAS for #SAT unless **NP**=**RP** [Zuckerman96].
- There is no FPRAS for a **#P**-complete problem under Karp reductions, unless **NP**=**RP**.

Basic Definitions

Definition

An *approximation-preserving* reduction from f to g is a probabilistic oracle Turing Machine M that takes as input a pair $(x, \varepsilon) \in \Sigma^* \times (0, 1)$, and satisfies the following conditions:

1. Every oracle call made by M is of the form (w, δ), where w is an instance of g, and $\delta \in (0, 1)$ is an error bound satisfying $\delta^{-1} \leq poly(|x|, \varepsilon^{-1})$.
2. M is a RAS for f whenever its oracle is a RAS for g.
3. M runs in $poly(|x|, \varepsilon^{-1})$.

If such a reduction from f to g exists, we write $f \leq_{AP} g$ (*AP-reducible*).
If $(f \leq_{AP} g)$ and $(g \leq_{AP} f)$, we write $f \equiv_{AP} g$ (*AP-interreducible*).

#P-complete problems under AP reductions

Theorem

Let A be an **NP**-complete decision problem. Then the corresponding counting problem, #A, is complete for **#P** with respect to AP-reducibility.

Proof Sketch.

1. $\#A \in \#P$
2. Also, #SAT is AP-reducible to #A:
   - #SAT can be approximated, in the FPRAS sense, by a PTM M equipped with an oracle for the decision problem of SAT [VV86].
   - This oracle can be replaced by an approximate counting oracle (RAS) for #A.
   - Thus, M consists an approximation-preserving reduction from #SAT to #A.

Relative complexity of approximate counting

In [DGGJ03] three classes of AP-interreducible problems are studied:

1. The first is the class of counting problems that admit an FPRAS.
2. The second is the class of counting problems AP-interreducible with #SAT.
3. The third is the class of counting problems AP-interreducible with #BIS.

Counting problems AP-interriducible with #SAT

Theorem

$$\#IS \equiv_{AP} \#SAT$$

Proof.

1. $\#LARGEIS \equiv \#SAT$.
2. $\#LARGEIS \leq_{AP} \#IS$:
   Let m and $G = (V, E)$, $|V| = n$, be an instance of $\#LARGEIS$.
   Construct $G' = (V', E')$ such that:
   $V' = V \times [r]$, and
   $E' = \{\{(u, i), (v, j)\} : u, v \in E \text{ and } i, j \in [r]\}$.

Counting problems AP-interridducible with #SAT

Proof cont. An independent set I' in G' projects to an independent set $I = \pi(I')$ in G in the following way:

$$I = \pi(I') = \{v \in V : \text{ there exists } i \in [r] \text{ such that } (v, i) \in I'\}$$

- For every k-sized independent set in G there are exactly $(2^r - 1)^k$ independent sets in G' that project to it.
- Let $I_m(G)$ the set of all m-sized independent sets in G, and $I(G')$ the set of all independent sets in G. Then:

$$|I(G')| \geq (2^r - 1)^m \cdot |I_m(G)|$$

- On the other hand, at most $(2^r - 1)^{m-1}$ independent sets I' in G' project to each independent set $I = \pi(I')$ in G of size $< m$. Thus:

$$|I(G')| \leq (2^r - 1)^m \cdot |I_m(G)| + (2^r - 1)^{m-1} \cdot 2^n$$

Counting problems AP-interridducible with #SAT

Proof cont. We have:

$$|I(G')| \geq (2^r - 1)^m \cdot |I_m(G)| \textbf{ (1)}$$

$$|I(G')| \leq (2^r - 1)^m \cdot |I_m(G)| + (2^r - 1)^{m-1} \cdot 2^n \textbf{ (2)}$$

If we choose $r \geq n + 3$, then $|I_m(G)| \leq \frac{|I(G')|}{(2^r-1)^m} \leq |I_m(G)| + \frac{1}{4}$.
Thus we can take,

$$|I_m(G)| = \left\lfloor \frac{|I(G')|}{(2^r - 1)^m} \right\rfloor$$

Counting problems AP-interriducible with #SAT

Theorem

- #IS is complete for **#P** with respect to AP-reducibility.
- #IS remains complete for **#P** with respect to AP-reducibility even when restricted to graphs of maximum degree 25.

#P problems with FPRAS

An unbiased estimator for #DNF using sampling:
Let U be the universe of possible assignments for a DNF formula f, and $S \subseteq U$ the set of satisfying assignments, i.e. $\#f = |S|$.

1. Repeat the following t times. At the i-th iteration:
   - Pick u uniformly at random from U.
   - If u belongs to S, $X_i = 1$.
   - If not, count $X_i = 0$.
2. Take an average of the above counts, $\tilde{X} = \sum_{i=1}^{t} \frac{X_i}{t}$, and return that as an estimate of $\frac{|S|}{|U|}$.

Obviously, $E[X_i] = Pr\Big(\{u \in S\}\Big) \cdot 1 + Pr\Big(\{x \notin S\}\Big) \cdot 0 = \frac{|S|}{|U|} \cdot 1 = \frac{|S|}{|U|}$ and $E[\tilde{X}] = \frac{|S|}{|U|}$. One can use the value $\tilde{X} \cdot |U|$ as an estimator for the size of $|S|$.

FPRAS for #DNF

Theorem

Let $\mu = \frac{|S|}{|U|}$ and $\varepsilon \leqslant 2$. If $t \geqslant (1/\mu) \cdot (4 \cdot ln(2/\delta)/\varepsilon^2)$ then the algorithm described above is an ε, δ approximation algorithm.

- $\frac{1}{\mu} = \frac{|U|}{|S|}$ can be exponential in the size of the input.
- We decrease the size of the universe!

1. Let S_i be the set of truth assignments which satisfy clause C_i.
2. Let $S = \bigcup_{i=1}^{m} S_i$. Observe that $\#f = |S|$.
3. Let $U' = S_1 \uplus \ldots \uplus S_m$, i.e. U' is the disjoint union of the S_i's.
4. An element $a_j \in U'$ is represented by (a_j, i), where $a_j \in S_i$ and $1 \leqslant i \leqslant m$.
5. U' contains only the satisfying assignments. However, $|U'| = \sum_{i=1}^{m} |S_i| \geq |S|$, since if an assignment a_j satisfies k clauses, U' contains k copies of a_j.
6. For each row containing at least one star, make the first one a 'special' star $\overline{*}$. If C_i is the first clause that is satisfied by a_j, then (a_j, i) is $\overline{*}$.

$$\begin{array}{cccccc}
 & S_1 & S_2 & S_3 & \ldots & S_m \\
a_1 & \overline{*} & & * & & \\
a_2 & \overline{*} & * & & & \\
a_3 & & \overline{*} & & & \\
\vdots & & & & &
\end{array}$$

Figure: The number of rows that contain at least one * is equal to $|S|$, and the number of special stars is equal to $|S|$.

To count the number of special stars we use the same idea as before:

1. Repeat the following t times. At the i-th iteration:
   - Sample an (a_j, i) (a star $*$) uniformly at random from U'.
   - If (a_j, i) is a special star $(\overline{*})$, $X_i = 1$.
   - If not, count $X_i = 0$.
2. Take an average of the above counts, $\tilde{X} = \sum_{i=1}^{t} \frac{X_i}{t}$, and return that as an estimate of $\frac{|S|}{|U'|}$.

The value $\tilde{X} \cdot |U'|$ can be used as an estimator for the size of $|S|$.

Now $\frac{1}{\mu} = \frac{|U'|}{|S|} \leq m$, where m is the number of clauses in f.

For picking up an (a_j, i) (a star $*$) uniformly at random we use the following algorithm:

1. Compute $|S_i|$, for every $1 \leq i \leq m$.
2. Pick i with probability $\frac{|Si|}{\sum_i |S_i|}$.
3. Pick a random assignment satisfying the corresponding clause C_i.

This concludes to picking a satisfying assignment with probability $\frac{1}{|U'|}$.

The above procedure is an FPRAS for #DNF.

Counting problems AP-interreducible with #BIS

$\#P_4$-COL Definition

Instance: A graph G.
Output: The number of P_4 colourings of G, where P_4 is the path of length 3.

#DOWNSETS Definition

Instance: A partially ordered set $(X, \preceq)$.
Output: The number of downsets in $(X, \preceq)$.

#1P1NSAT Definition

Instance: A CNF Boolean formula ϕ, with at most one unnegated literal per clause, and at most one negated literal.
Output: The number of satisfying assignments to ϕ.

Definition of H-colourings

Definition

An H-colouring of a graph G is simply a homomorphism $f : G \to H$:

$$(u, v) \in E_G \Rightarrow (f(u), f(v)) \in E_H$$

Regard the vertices of H as representing colours, then $f : G \to H$ induces a q-colouring of G that *respects* the structure of H: *two colours may be adjacent in G only if the corresponding vertices are adjacent in H.*

- K_q -colourings, where K_q is the complete q-vertex graph, are simply the usual q-colourings.
- K_2^1-colourings, where K_2^1 is K_2 with one loop added, correspond to independent sets.

Counting problems AP-interreducible with #BIS

Definition

The problems #BIS, $\#P_4$-COL, #DOWNSETS, #1P1NSAT are all AP-interreducible.

A Logical Characterisation #BIS and its relatives

- A counting problem is identified with a sentence ϕ in FO Logic, an instance with a model **A**, and solutions can be counted by counting relations that make ϕ true in **A**.
- Standard Definitions:
  - *Vocabulary*: $\sigma = \{\widetilde{R}_0, \ldots \widetilde{R}_{k-1}\}$
  - $\widetilde{R}_i$'s are relation symbol of arities $r_0, \ldots, r_{k-1}$
  - *Structure* $\mathbf{A} = (A, R_0, \ldots, R_{k-1})$ over σ consists a universe A
  - Each relation $R_i \subseteq A^{r_i}$ is an interpretation of $\widetilde{R}_i$.
- We present counting problems as *structures* over suitable vocabularies:

Example

An instance of #IS is a graph which can be 0regarded as a structure $\mathbf{A} = (A, \sim)$, where A is the vertex set, and "$\sim$" is the symmetric binary relation of adjacency.

A Logical Characterisation #BIS and its relatives

- The solutions to be counted are represented as sequences of relations $\mathbf{T} = (T_1, \ldots, T_{r-1})$ and first-order variables $\mathbf{z} = (z_0, \ldots, z_{m-1})$.

Definition

A counting problem f (from structures over σ to $\mathbb{N}$) is in the class $\#\mathcal{FO}$ if it can be expressed as:

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

where ϕ is a FO formula with relation symbols from $\sigma \cup \mathbf{T}$ and (free) variables from $\mathbf{z}$.

A Logical Characterisation #BIS and its relatives

Example

If we encode an IS as a unary relation I, then #IS:

$$f_{IS}(\mathbf{A}) = |\{(I) : \mathbf{A} \models \forall x, y : x \sim y \Rightarrow \neg I(x) \vee \neg I(y)\}|$$

- $\#$IS is in the subclass $\#\Pi_1 \subseteq \#\mathcal{FO}$ (since the formula contains only universal quantification).
- In general, we have a (strict) hierarchy of subclasses:

$$\#\Sigma_0 = \#\Pi_0 \subset \#\Sigma_1 \subset \#\Pi_1 \subset \#\Sigma_2 \subset \#\Pi_2 = \#\mathcal{FO} = \#\mathbf{P}$$

- All functions in $\#\Sigma_1$ admit an *FPRAS*!
- All AP-interreducible problems we saw are in the (*syntactically* restricted) subclass $\#RH\Pi_1 \subseteq \#\Pi_1$:

A Logical Characterisation #BIS and its relatives

Definition

A counting problem f is in the class $\#RH\Pi_1$ if it can be expressed in the form:

$$f(\mathbf{A}) = |\{(\mathbf{T}, \mathbf{z}) : \mathbf{A} \models \forall \mathbf{y} : \psi(\mathbf{y}, \mathbf{z}, \mathbf{T})\}|$$

where ψ is an *unquantified* CNF formula in which each clause has at most one occurrence of an unnegated relation symbol from **T**, and at most one occurrence of a negated relation symbol from **T**.

- "RH" stands for "Restricted Horn"
- The restriction on clauses of ψ applies only to terms involving symbols from **T**.

A Logical Characterisation #BIS and its relatives

- An instance of #DOWNSETS can be expressed as a structure $\mathbf{A} = (A, \preceq)$.
  Then, $\#DOWNSETS \in \#RH\Pi_1$, since the number of downsets may be expressed as:

$$f_{DS}(\mathbf{A}) = |\{(D) : \mathbf{A} \models \forall x, y \in A : D(x) \wedge (y \preceq x) \rightarrow D(y)\}|$$

- An instance of #BIS can be expressed as a structure $\mathbf{A} = (A, \preceq)$.
  Then, $\#BIS \in \#RH\Pi_1$, since:

$$f_{BIS}(\mathbf{A}) = |\{(X) : \mathbf{A} \models \forall x, y \in A : L(x) \wedge (y \preceq x) \wedge X(x) \rightarrow X(y)\}|$$

A Logical Characterisation #BIS and its relatives

Theorem

- *#1P1NSAT is complete for $\#RH\Pi_1$ under Karp reductions.*
- *The problems #BIS, $\#P_4$-COL, #DOWNSETS, #1P1NSAT are all complete for $\#RH\Pi_1$, with respect to AP-reducibility.*

Definitions of #PE and TotP

Definitions

- The class **#PE** contains the functions of **#P**, the decision version of which are in **P**.
- The definition of **TotP** involves a function associated with every PNTM M:

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

Then, **TotP** $= \{tot_M | M \text{ is a } PNTM\}$.

- **#PE** contains all hard-to-count-easy-to-decide problems.
- **FP** $\subseteq$ **TotP** $\subseteq$ **#PE** $\subseteq$ **#P**. Inclusions are proper, unless **P** $=$ **NP**.
- $\mathbf{FP}^{\mathbf{TotP}} = \mathbf{FP}^{\#\mathbf{PE}} = \mathbf{FP}^{\#\mathbf{P}}$.

Functions in TotP

Theorem

#Perfect Matchings is in **TotP**.

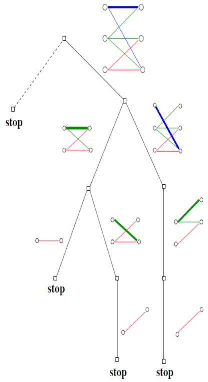


Figure: The computation tree of the nondeterministic algorithm for #Perfect Matchings for an input graph with 3 perfect matchings.

Functions in TotP

Theorem

#DNF, #NonCliques are in **TotP**.

Theorem

TotP is exactly the closure under Karp reductions of $\#\mathbf{PE}_{SR}$.

The class SpanL

Definition

Let the function

$$span_M(x) = \text{ the number of different valid outputs of } M \text{ on input } x$$

which is associated with a nondeterministic transducer M.
Then, **SpanL**$=\{span_M | M$ is some NL transducer $M\}$.

Definition

#NFA:
Input: An encoding of an NFA M and a string $x \in \{0,1\}^*$.
Output: The number of words $\leq x$ accepted by M.

Theorem

#NFA is complete for **SpanL** under logspace Karp reductions.

SpanL is a hard counting class

Theorem

#NFA is **#P**-complete with respect to Cook reducibility.

Proof. $\#DNF \leq_T^p \#NFA$.
Let f be a boolean formula in disjunctive normal form. Let $x_1, ..., x_m$ and $C_1, ..., C_l$ be the variables and the clauses of f.
We construct an NFA, the language of which is exactly the satisfying assignments of f.

- For each C_i we construct an NFA M_i.
- M_i consists of a chain of $m+1$ states, ${s_0}_i, ..., {s_m}_i$, ${s_0}_i$ is the start state and ${s_m}_i$ the accepting state.
- The edge $({s_j}_i, {s_{j+1}}_i)$ is labelled 1 if $x_{j+1} \in C_i$, 0 if $\overline{x}_{j+1} \in C_i$, and $0, 1$ otherwise.

1. M_i accepts exactly the strings corresponding to truth assignments for C_i.
Let M be the NFA with a start state s and an ϵ-transition to the start state ${s_0}_i$ for each $1 \leqslant i \leqslant l$. The final states are exactly the final states of each M_i.
2. M accepts exactly the strings corresponding to satisfying assignments of f and $\#NFA(M, m) = \#DFA(f)$.

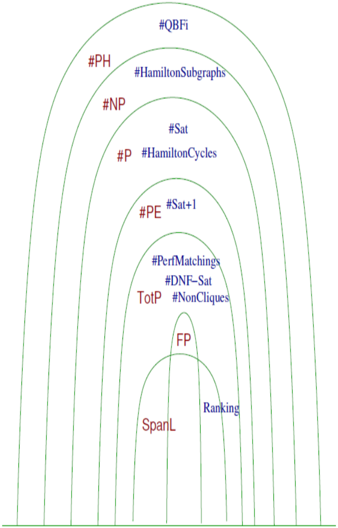
The complexity of #NFA

- There exists a $n^{\mathcal{O}(logn)}$ randomised approximation scheme for $\#NFA$.
- There exists a $n^{\mathcal{O}(logn)}$ almost uniform generator for $R_{\#NFA_M} = \{((M, 1^m), x) : x \in \{0,1\}^m \ and \ x \in L(M)\}$.

The class SpanL

Theorem

- $\#\mathbf{L} \subseteq \mathbf{SpanL} \subseteq \mathbf{TotP}$.
- $\mathbf{FP}^{\mathbf{SpanL}[1]} = \mathbf{FP}^{\mathbf{TotP}[1]} = \mathbf{FP}^{\#\mathbf{P}[1]}$.


#QBFi
#PH
#HamiltonSubgraphs
#NP
#Sat
#P
#HamiltonCycles
#PE
#Sat+1
#PerfMatchings
#DNF–Sat
TotP
#NonCliques
FP
Ranking
SpanL