Counting complexity

Arnaud Durand

Université de Paris

July 6, 2021 - Counting and Sampling Workshop 2021

Introduction

- Subject: Characterizing hardness of counting problems
- Main instruments: reductions (as for any algorithmic task)
- Well-known that defining adequate reductions in this context is not easy
- Classical reductions are either too weak or too powerful
- Talk: No new results, short series of remarks based on old contributions

A hierarchy of counting problems

$\sharp\cdot C$: the classes of all witness functions R that satisfy the following conditions:

1. There is a polynomial p, such that for each input x and each $y \in R(x)$, the relation $|y| \leq p(|x|)$ holds.
2. The decision problem

   Given x and y, does the relation $y \in R(x)$ hold?

   is in the class C.

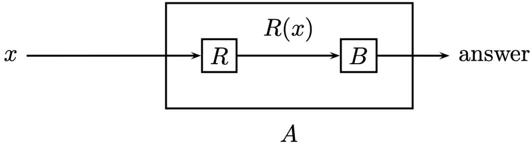
It holds : $\mathbf{FP} \subseteq \sharp\cdot\mathbf{P} \subseteq \sharp\cdot\mathbf{NP} \subseteq \sharp\cdot\mathbf{coNP}$

Remark: $\sharp\mathbf{P} = \sharp\cdot\mathbf{P}$

The difficulty to define reductions

- Karp reduction (parsimonious: special case):

$$A \leq_m^p B$$

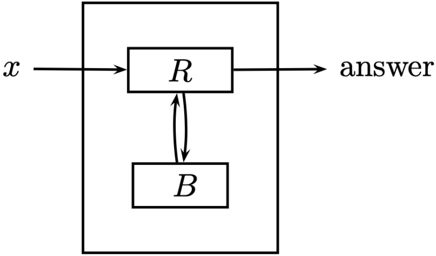


- #SAT complete for $\sharp\cdot\mathbf{P}$, generalization of #SAT for higher classes.

The difficulty to define reductions

- Turing reduction

$$A \leq_T^p B$$



A

- #PerfectMatching complete for $\sharp \cdot \mathbf{P}$ under Turing reduction.

Motivations

- There are counting problems that seem to reside above the class $\sharp\cdot\mathbf{P}$:
  #Hilbert, #Circumscription, #CQ (conjunctive queries)
- Toda and Watanabe (1992) : $\sharp\cdot\mathrm{PH} \subseteq \mathbf{P}^{\sharp\mathbf{P}}$.
  Bad news: The counting classes are not closed under the Turing reductions.

Motivations

- Karp reductions or parsimonious reductions are insufficient to prove complete problems for counting classes
  Common folklore: there exists seemingly hard counting problems (e.g. #PerfectMatching) whose underlying decision problem is easy. So no reduction based on a direct mapping of the solution set is enough to capture all hard problems.
- Interesting to design reduction technics under which counting classes are closed
- Need to map solution sets in a indirect way (not too indirect though)

Multisets

Let D be a nonempty domain. *Multiset* M is a function $M : D \to \mathbb{N}$ that assigns to each input x its number of occurrences $M(x)$ in M.

Operations on multisets

Union: $(A \oplus B)(x) = A(x) + B(x)$ for each $x \in D$

Difference: $(A \ominus B)(x) = \max(A(x) - B(x), 0)$

Observation

$(A \ominus B)(x) = A(x) - B(x)$ holds for each $x \in D$ if $B \subseteq A$.

$A_1 \oplus \cdots \oplus A_n$ is denoted by $\bigoplus_{i=1}^{n} A_i$

Subtractive reductions

Let R be a binary predicate and #R the associated counting problem that computes the cardinality of the set $R(x)$.

Definition (D., Hermann, Kolaitis'00-05)

The counting problem #A reduces to the problem #B by a **subtractive reduction**, if there exist polynomial-time computable functions f_i and g_i, $i = 1, \ldots, n$, such that the following conditions hold for the predicates A and B:

- $\bigoplus_{i=1}^{n} B(f_i(x)) \subseteq \bigoplus_{i=1}^{n} B(g_i(x))$,
- $|A(x)| = \sum_{i=1}^{n} |B(g_i(x))| - \sum_{i=1}^{n} |B(f_i(x))|$.

Simpler useful form

Definition

The counting problem #A reduces to the problem #B by a **subtractive reduction**, if there exist polynomial-time computable functions f and g, such that the following conditions hold for the predicates A and B:

- $B(f(x)) \subseteq B(g(x))$,
- $|A(x)| = |B(g(x))| - |B(f(x))|$.

Other variants (such as complementive reductions) defined by Bauland and al.

Properties

Property

The subtractive reductions are transitive (only the general case).

Property

For each $k \in \mathbb{N}$, the class $\sharp\cdot\Pi_k\mathbf{P}$ (in particular $\sharp\cdot\mathbf{P}$) is closed under the subtractive reductions for each k. It is not the case of $\sharp\cdot\Sigma_k\mathbf{P}$ classes.

Subtractive reduction are weaker than Turing reductions (but stronger than parsimonious ones).
Can we prove some interesting problem is hard under this reduction?

Complete problems - Some examples

- #DNF: count the models of a DNF propositional formula (in $\sharp\mathbf{P}$)
- #Circumscription: count the *minimal* (or the pointwise ordering) models of a propositional formulas (in $\sharp\cdot\mathbf{coNP}$)
- #CQ: count the number of tuples solutions of a conjunctive queries (with projections) (in $\sharp\cdot\mathbf{NP}$)

Under some reasonable complexity assumption: none of them can be proved complete for the corresponding class.
However :

- #DNF is $\sharp\mathbf{P}$-complete for subtractive reduction (obvious)
- #Circumscription is $\sharp\cdot\mathbf{coNP}$-complete for sub. red. (DHK'00)
- #CQ is $\sharp\cdot\mathbf{NP}$-complete for complementive reduction (Bauland and al')

Remarks

- The approach is powerful for proving hardness of some natural counting problems
- However, not clear if it can substitute to Turing reductions for most classical problems (e.g. #PerfectMatching)
- If yes, it would probably provide some interesting insight on the nature of counting problems.

Concluding remarks

- The reductions above are based on a more general principle of polynomial time witness reductions (D, Hermann, Wagner) where $A \leq_w B$ if (roughly speaking) there exists a polytime function f, polynomial time predicate $D_1, ..., D_m$ and a $(\cap, \cup, \neg)$-formula F such that, for all x

$$A(x) = F(B(f(x)), D_1(x), ..., D_m(x)))$$

- Depending on the choice of F (monotone, affine, conjunctive, disjunctive, etc): different closure properties for counting (but also enumeration, approximation etc) can be obtained
- Makes sense for smaller counting classes also