Complexity Classes of Counting Problems

Andreas-Nikolas Göbel

National Technical University of Athens, Greece

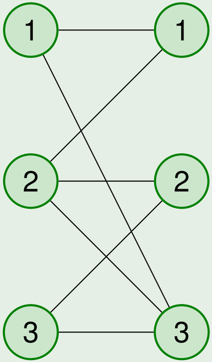
April 2012

Counting problems

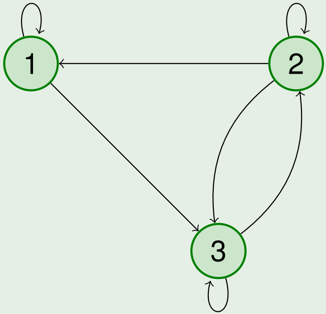
- #SAT: the function that counts the number of satisfying assignments of a given formula.
- #HAMILTONPATHS: the function that counts the number of Hamilton paths of a given graph.
- #DIV: the function that counts the number of divisors of a given number
- #PRDIV: the function that counts the number of prime divisors of a given number
- #PERFECTMATCHINGS: the function that counts the perfect matchings of a given bipartite graph.
- #CYCLECOVER: the function that counts the cycle covers of a given directed graph with self loops.
- (0,1)-PERMANENT: The permanent of a given matrix with elements from $\{0, 1\}$

Remark: The last 3 are equivalent

Perfect Matchings



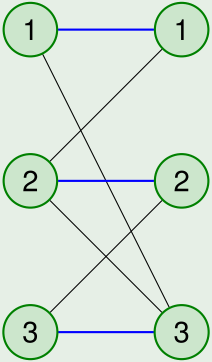
Cycle Covers



Adjacency Matrix

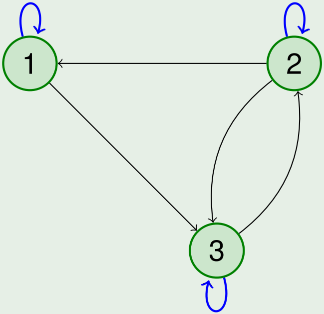
1	0	1
1	1	1
0	1	1

Perfect Matchings



(1,1),(2,2),(3,3)

Cycle Covers

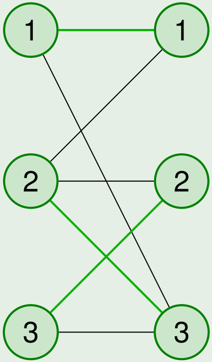


(1,1),(2,2),(3,3)

Adjacency Matrix

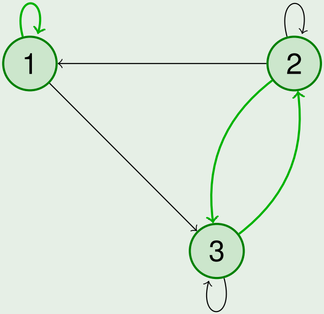
1	0	1
1	1	1
0	1	1

Perfect Matchings



(1,1),(2,3),(3,2)

Cycle Covers

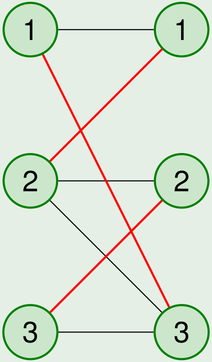


(1,1),(2,3),(3,2)

Adjacency Matrix

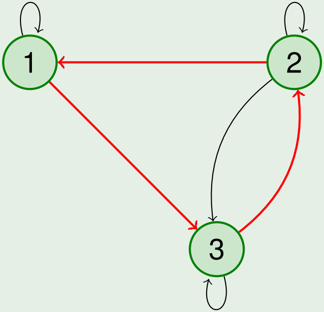
1	0	1
1	1	1
0	1	1

Perfect Matchings



(1,3),(2,1),(3,2)

Cycle Covers



(1,3),(2,1),(3,2)

Adjacency Matrix

1	0	1
1	1	1
0	1	1

Definition (Valiant '79)

$\#\mathrm{P}$ is the class of functions that can be computed by NDTM's of polynomial time complexity.

$$\#P = \{f : f(x) = \mathrm{acc}_M(x)\}$$

Or

The class of functions that count the number of *witnesses* (or certificates) of NP problems.

Def. $f \in \#\mathrm{P} \ : \exists \text{ pred. } Q \in \mathrm{P}, \ \forall x : \ f(x) = \#\{y \mid Q(x, y)\}$

PP

Definition (Wagner '86)

PP is the class of all languages L such that there exists a poly-NDTM M and an FP function f such that
$L = \{x : \mathrm{acc}_M(x) > f(x)\}$

- Originally defined by Gill '74 as a probabilistic class
- From the above definition we can show: $\mathrm{P}^{\#\mathrm{P}} = \mathrm{P}^{\mathrm{PP}}$
- $\mathrm{P}^{\mathrm{NP}[log]} \subseteq \mathrm{PP}$ (Beigel, Hemaspaandra, and Wechsung '89)

Definition (Wagner '89)

$\mathrm{C_{=}P}$ is the class of all languages L such that there exists a poly-NDTM M and an FP function f such that

$$L = \{x : \mathrm{acc}_M(x) = f(x)\}$$

Definition (Papadimitriou and Zachos '82)

$\oplus\mathrm{P}$ is the class of all languages L such that there exists a poly-NDTM M such that

$$L = \{x : \mathrm{acc}_M(x) \text{ is odd}\}$$

Definition (Allender '86)

For any language L, $L \in \mathrm{FewP}$ if and only if there exist a NDTM M and a polynomial p such that:

$$acc_M(x) \leq p(|x|)$$
$$x \in L \Leftrightarrow acc_M(x) > 0$$

Definition (Cai and Hemaspaandra '90)

For any language L, $L \in \mathrm{Few}$ if and only if there exist a NDTM M and a polynomial p and a polynomial time computable predicate $A(x, y)$ such that:

$$acc_M(x) \leq p(|x|)$$
$$x \in L \Leftrightarrow A(x, acc_M(x))$$

- $\oplus\mathrm{P}^{\oplus\mathrm{P}} = \oplus\mathrm{P}$ (PZ '82)
- $\mathrm{Mod}_k\mathrm{P}^{\mathrm{Mod}_k\mathrm{P}} = \mathrm{Mod}_k\mathrm{P}$ holds if k is a prime (BGH '90)
- $\mathrm{FewP} \subseteq \mathrm{NP}$
- This is not known for Few but $\mathrm{Few} \subseteq \mathrm{P}^{\mathrm{NP}[log]}$
- $\mathrm{Few} \subseteq \oplus\mathrm{P}$ (CH '90),
  $\mathrm{Few} \subseteq \mathrm{C}_=\mathrm{P}$ (KSTT '89),
  $\mathrm{Few} \subseteq \mathrm{Mod}_k\mathrm{P}$, for each prime k (BGH '90)
- Few is low for the classes PP, $\mathrm{C}_=\mathrm{P}$ and $\oplus\mathrm{P}$ (KSTT '89)
  - That is $\mathrm{PP}^{\mathrm{Few}} = \mathrm{PP}$

Only Positive?

- The counting classes defined so far may only contain positive functions.
- For a NDTM M we denote with $\overline{M}$ the machine identical to M but with the accepting and rejecting states interchanged
- $\mathrm{gap}_M(x) = \mathrm{acc}_M(x) - \mathrm{acc}_{\overline{M}}(x)$

Definition

$\mathrm{GapP} = \{\mathrm{gap}_M : M \text{ is a poly-NDTM}\}$

- $\#\mathrm{P} \subseteq \mathrm{GapP} = \#\mathrm{P} - \#\mathrm{P} = \#\mathrm{P} - \mathrm{FP} = \mathrm{FP} - \#\mathrm{P}$

SPP

Definition

SPP is the class of all languages L, such that there exists a poly-NDTM M, such that for all x

$$\begin{aligned} x \in L &\Rightarrow \mathrm{gap}_M(x) = 1, \\ x \notin L &\Rightarrow \mathrm{gap}_M(x) = 0. \end{aligned}$$

Köbler, Schöning and Torán ('92) showed that Graph Automorphism is in SPP
Arvind and Kurur ('02) showed that Graph Isomorphism is in SPP

Valiant's Framework

Definition

For every complexity class $\mathcal{C}$ of decision problems we define $\#\mathcal{C} = \bigcup_{A \in \mathcal{C}} (\#\mathrm{P})^A$, where $\#\mathrm{P}^A$ is the collection of all functions that count the accepting paths of polynomially bounded NDTM's having A as their oracle.

$\#\mathcal{C} = \#\mathrm{co}\mathcal{C}$ holds for every complexity class

Span and Operators

Definition (Hemaspaandra Vollmer '95)

For any class $\mathcal{C}$, define $\# \cdot \mathcal{C}$ to be the class of functions f, such that for some $\mathcal{C}$-computable 2-ary predicate (relation) R and some polynomial p, for every string x it holds that:

$$f(x) = \|\{y : p(|x|) = |y| \text{ and } R(x,y)\}\|,$$

where "$\|A\|$" denotes the cardinality of the set A.

Definition (Köbler, Shöning, Torán '89)

For a non-deterministic transducer M define the function $\text{span}_M : \Sigma^* \to \mathbb{N}$ such that $\text{span}_M(x)$is the number of different valid outputs that occur in the nondeterministic computation tree induced by M on input x.
Define $\text{SpanP} = \{f : f = \text{span}_M \text{ and } M \text{ is a poly-NDTM}\}$

Definition

$\mathrm{span}_{M-N}(x)$ is the number of different outputs that $M(x)$ can produce and $N(x)$ can't.

Lemma

$\#NP = \{f : f = span_{M-N}\}$, *M,N PNTM's*

$\#\mathrm{NP} \subseteq \# \cdot \mathrm{NP} - \# \cdot \mathrm{NP}$

Some Results

- $\#NP = \# \cdot P^{NP}$ (from definitions)
- $\#NP = (\#P)^{NP[1]}$ (KST '89)
- $\#P \subseteq \# \cdot NP \subseteq \#NP$ (KST '89)
- $\#P = \# \cdot NP$ if and only if $UP = NP$ (KST '89)
- $\# \cdot NP = \#NP$ if and only if $NP = coNP$ (KST '89)

Toda's Result (Ph.D.'92)

Theorem

$\# \cdot coNP = \#NP$

- Generalizing to the $\#\mathrm{PH}$ we have:

$$\# \cdot \Sigma_k^p \subseteq \#\Sigma_k^p = \# \cdot \Pi_k^p$$

- Thus, $\#\mathrm{PH} = \bigcup \#\Sigma_k^p = \bigcup \# \cdot \Pi_k^p = \# \cdot \mathrm{PH}$

Valian-Vazirani Theorem

- For every known NP-complete problem the number of solutions of its instances varies from zero to exponentially many.
- Does this cause the inherent intractability of these problems?

Theorem (Valiant Vazirani '86)

There exists a probabilistic polynomial time algorithm f s.t. for every n-variable Boolean formula φ

$$\varphi \in \text{SAT} \Rightarrow \Pr[f(\varphi) \in \text{USAT}] \geq \frac{1}{8n}$$

$$\varphi \notin \text{SAT} \Rightarrow \Pr[f(\varphi) \in \text{SAT}] = 0$$

Therefore the above answer is no unless $NP = RP$

Pairwise Independent Hash Functions

Definition (Pairwise Independent Hash Functions)

Let $\mathcal{H}_{n,k}$ be a collection of functions from $\{0,1\}^n$ to $\{0,1\}^k$. We say that $\mathcal{H}_{n,k}$ is pairwise independent if for every $x, x' \in \{0,1\}^n$ with $x \neq x'$ and for every $y \neq y' \in \{0,1\}^k$, $\Pr_{h \in_R \mathcal{H}_{n,k}}[h(x) = y \wedge h(x') = y'] = 2^{-2k}$

Lemma: Let $\mathcal{H}_{n,k}$ be a pairwise independent hash function collection from $\{0,1\}^n$ to $\{0,1\}^k$ and $S \subseteq \{0,1\}^n$ s.t. $2^{k-2} \leq |S| \leq 2^{k-1}$. Then

$$\Pr_{h \in_R \mathcal{H}_{n,k}}[\text{there is a unique } x \in S \text{ satisfying } h(x) = 0^k] \geq \frac{1}{8}$$

proof on board

Proof of Valiant Vazirani Theorem

- Given φ on n variables, choose k at random from $\{2, \ldots, n+1\}$ and a random hash function $h \in_R \mathcal{H}_{n,k}$.
- The statement $\exists_{x \in \{0,1\}^n} \varphi(x) \wedge (h(x) = 0^k)$ is false if φ is unsatisfiable, and with probability $1/8n$ has a unique satisfying assignment if φ is satisfiable.
    - If S is the set of satisfying assignments of φ, with probability $1/n$, k satisfies $2^{k-2} \leq |S| \leq 2^{k-1}$.
    - With probability $1/8$ there is a unique x such that $\varphi(x) \wedge h(x) = 0^k$
- The implementation is based on Cook's reduction, and expresses tha deterministic computation inside the $\exists$ sign as a formula τ on variables $x, y \in \{0,1\}^{\mathrm{poly}(n)}$, s.t. $h(x) = 0$ iff there exists a unique y such that $\tau(x, y) = 1$. Output

$$\psi = \varphi(x) \wedge \tau(x, y)$$

What is the complexity of #P compared to Decision Classes?

Theorem (Papadimitriou Zachos '82)

$P^{NP[log]} \subseteq P^{\#P[1]}$

Theorem (Toda '91)

$PH \subseteq P^{\#P}$

Proof

Lemma (Randomized Reduction from PH to $\oplus$SAT)

There exists a prob. poly-time algo A s.t. given a parameter m and any QBF ψ of size n with c levels of alternations, runs in $\mathrm{poly}(n, m)$ *and satsisfies:*

$$\psi \text{ is true } \Rightarrow \Pr[A(\psi) \in \oplus\textsc{Sat}] \geq 1 - 2^{-m}$$

$$\psi \text{ is false } \Rightarrow \Pr[A(\psi) \in \oplus\textsc{Sat}] \leq 2^{-m}$$

Proof on board

- By derandomizing the above lemma the proof of Toda's theorem is concluded.

Hard To Count - Easy to Decide

- Surprisingly enough, some problems in P have counting versions that are complete for $\#\mathrm{P}$ under less restrictive reductions.
  - Cook reductions ($f \leq_T^p g : f \in \mathrm{FP}^g$ — *aka* poly-time Turing).
- Examples: #PERFECT MATCHINGS, #DNFSAT, #MONSAT
- They cannot be complete for $\#\mathrm{P}$ under parsimonious reductions unless $\mathrm{P} = \mathrm{NP}$.
- Note that $\#\mathrm{P}$ is not closed (under likely assumptions) under Cook reductions.
  Therefore these problems *are Cook-complete also for superclasses (and subclasses!) of* $\#\mathrm{P}$.
- Hard to count - easy to decide problems neither are well represented by $\#\mathrm{P}$, nor are well classified by means of Cook reductions.

Further Motivation for Studying HCED Problems

- Three degrees of approximability within problems of $\#\mathrm{P}$ [DGGJ'00]:
  - Solvable by an *FPRAS*:
    #PERFECT MATCHINGS, #DNFSAT, ...
  - AP-interreducible with SAT:
    SAT, #IS, ...
  - An Intermediate Class (AP-Interreducible with #BIS)
    #BIS, ...

Hard to Count Easy to Decide

- #P contains counting versions of known NP problems
- there exist other #P problems with decision version in P

Definition (Pagourtzis '01)

Let $\#\mathrm{PE}$ be the class that contains functions of $\#\mathrm{P}$ whose related language is in P.

Definition (Kiayias, Pagourtzis, Sharma and Zachos)

$\mathrm{TotP} = \{\mathrm{tot}_M : M \text{ is a poly-NDTM}\}$

$$\mathrm{tot}_M(x) = (\text{The number of paths of M on input x}) - 1$$

Inclusions among $\#\mathrm{P}, \#\mathrm{PE}$, and TotP

- $\#\mathrm{P} \subseteq \mathrm{TotP} - \mathrm{FP}$
- $\mathrm{FP} \subseteq \mathrm{TotP} \subseteq \#\mathrm{PE} \subseteq \#\mathrm{P}$.
  These inclusions are proper unless $\mathrm{P} = \mathrm{NP}$
- From these propositions we get the following Corollaries:
  - $\mathrm{TotP}, \#\mathrm{PE}$ and $\#\mathrm{P}$ are not Karp equivalent unless $\mathrm{P{=}NP}$.
  - The above classes are Cook[1] interreducible:

$$\mathrm{FP}^{\mathrm{TotP}[1]} = \mathrm{FP}^{\#\mathrm{PE}[1]} = \mathrm{FP}^{\#\mathrm{P}[1]}$$

  - Combining the latter with Toda's result we have

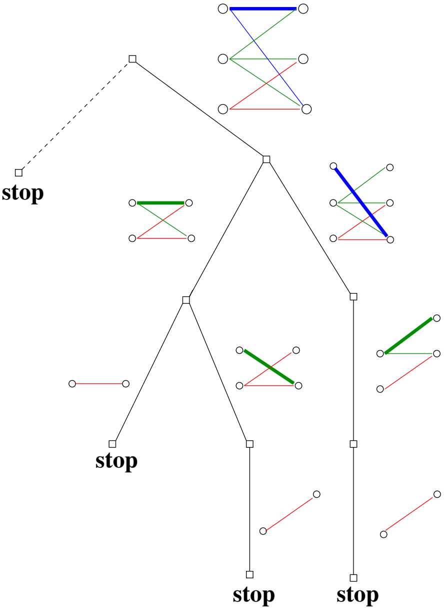
$$\mathrm{PH} \subseteq \mathrm{P}^{\mathrm{TotP}[1]} = \mathrm{P}^{\#\mathrm{PE}[1]}$$

#PERFECT MATCHINGS is in TotP

Sketch of the proof:

- It can be decided in polynomial time whether a graph G has any perfect matchings.
- Let $e = (v, u)$ be an edge of G. Then, the set of perfect matchings of G can be partitioned into two subsets, S_0 and S_1, where S_0 consists of those perfect matchings that match u and v through e, and S_1 consists of the remaining perfect matchings.
- Each S_i has the same cardinality with the set of of perfect matchings of an appropriate subgraph G_i of G.

Path Counting



Self-reducibility of #DNF-SAT

Select a variable x and construct formulae
$\phi_0 := \phi|_{x=0}$,
$\phi_1 := \phi|_{x=1}$. Clearly:
$[\#\text{DNFSAT}(\phi) = \#\text{DNFSAT}(\phi_0) + \#\text{DNFSAT}(\phi_1)]$

Corollary

#DNFSAT *is in* TotP.

Self-reducibility for well-known problems that are therefore in TotP

- #PerfectMatchings (equiv. Permanent and CycleCover)
- #DNF-Sat
- #2-Sat
- #NonCliques
- #NonIndSets
- #IndSetsAll
- Ranking

They are all Cook[1]-complete for TotP.

TotP $= \textit{Karp-closure}(\#\mathrm{PE}_{\mathrm{SR}})$

Self Reducibility

A function $f : \Sigma^* \to \mathbb{N}$ is called poly-time self-reducible, if there exist polynomials r and q and polynomial time computable functions $h : \Sigma^* \to \mathbb{N}$, $g : \Sigma^* \to \mathbb{N}$ and $t : \Sigma^* \to \mathbb{N}$, such that for all $x \in \Sigma^*$

1. $f(x) = t(x) + \sum_{i=0}^{r(|x|)} g(x,i) f(h(x,i))$, that is, f can be processed recursively by reducing x to $h(x,i)$, $(0 \leq i \leq r(|x|))$, and
2. the recursion terminates after at most polynomial depth (that is, the value of f on instance $h(\ldots h(h(x,i_1),i_2)\ldots, i_{q(|x|)})$ can be computed deterministically in polynomial time).

Theorem (Pagourtzis, Zachos)

TotP *is exactly the closure under Karp reductions of* $\#\textbf{\textit{PE}}_{SR}$

Some Definitions

- A is a Partial Order ($x \leq_A y$)
  - reflexive
  - antisymmetric
  - transitive
- A is a Total Order
  - partial order
  - all strings comparable
- A is a p-order
  $\exists q, \forall x, y$ with $x <_A y : \quad |x| \leq q(|y|)$
  (i.e. lengths of all strings are polynomially related)
- A has Efficient Adjacency Checks (a feasibility constraint)
  We can check efficiently (in P) whether:
  $(x <_A y) \wedge (\nexists z : x <_A z <_A y)$

Interval-size Function Classes

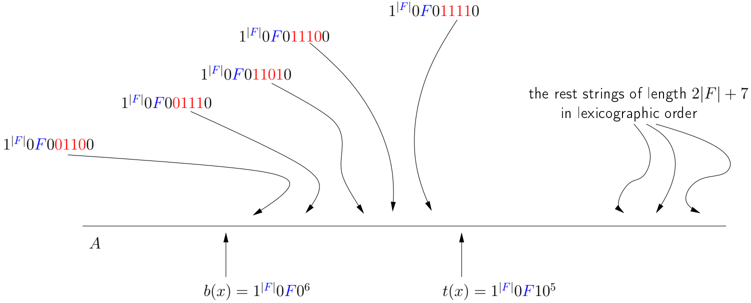
Definition

- $\mathrm{IF_p}$ ($\mathrm{IF_t}$) is the class of all functions $f : \Sigma^* \rightarrow \mathbb{N}$ for which there exists a partial (total) p-order $A \in \mathbf{P}$, with $f(x) = \|\{z : l(x) <_A z <_A u(x)\}\|$, for every $x \in \Sigma^*$, where $l, u \in \mathrm{FP}$.
- $\mathrm{IF_p^{\prec}}$ ($\mathrm{IF_t^{\prec}}$) is the class of all functions $f : \Sigma^* \rightarrow \mathbb{N}$ for which there exists a partial (total) p-order $A \in \mathbf{P}$, with $f(x) = \|\{z : l(x) <_A z <_A u(x)\}\|$, for every $x \in \Sigma^*$, where $l, u \in \mathrm{FP}$, and A has efficient adjacency checks.

- $\mathrm{IF_p^{\prec}}$ contains #DIV and #PRDIV
- $\mathrm{IF_t^{\prec}}$ contains #MONSAT (also Cook-complete for this class). (A monotone CNF formula contains no $\neg$)

Example: an $\mathrm{IF}_t^{\prec}$ computation for #MonSat

$F = \mathrm{enc}[(x_0 \vee x_2) \wedge (x_0 \vee x_1 \vee x_3) \wedge (x_1) \wedge (x_2 \vee x_3)]$



Interval-size Function Classes (cont.)

Theorem (Hemaspaandra, Homan, Kosub, Wagner '01)

For any function f the following are equivalent:

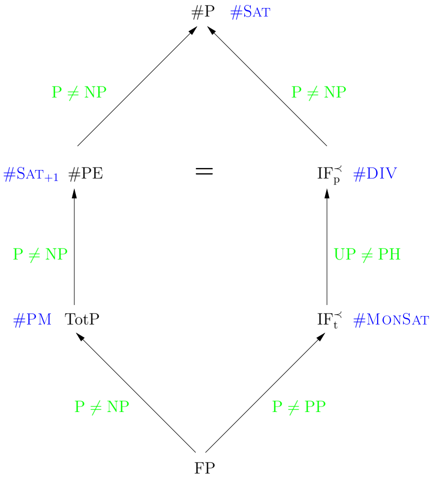
1. $f \in \#\mathbf{P}$.
2. There exist a partial p-order $A \in \mathbf{P}$ with $f(x) = \|\{z : l(x) <_A z <_A u(x)\}\|$, for all $x \in \Sigma^*$, for some $l, u \in \mathrm{FP}$.
3. There exist a total p-order $A \in \mathbf{P}$ with $f(x) = \|\{z : l(x) <_A z <_A u(x)\}\|$, for all $x \in \Sigma^*$, for some $l, u \in \mathrm{FP}$ with $l(x) <_A u(x)$.

Corollary: $\#\mathrm{P} = \mathrm{IF_t} = \mathrm{IF_p}$.

Some Inclusions

- $\mathrm{IF}_p^{\prec} = \#\mathrm{PE}$ *[Hemaspaandra et. al.'01]*
- $\#\mathrm{P} \subseteq \mathrm{IF}_t^{\prec} - \mathrm{FP}$
- $\mathrm{FP}^{\mathrm{IF}_t^{\prec}} = \mathrm{FP}^{\mathrm{IF}_p^{\prec}} = \mathrm{FP}^{\#\mathrm{P}}$
- $\mathrm{FP} \subseteq \mathrm{IF}_t^{\prec} \subseteq \mathrm{IF}_p^{\prec} \subseteq \#\mathrm{P} = \mathrm{IF}_t = \mathrm{IF}_p$
- The inclusion $\mathrm{FP} \subseteq \mathrm{IF}_t^{\prec}$ is proper unless $\mathrm{FP} = \#\mathrm{P}$.

Summarizing Earlier Results



Other Feasibility Constraints

Consider other polynomial time feasibility constraints, besides efficient adjacency checks.

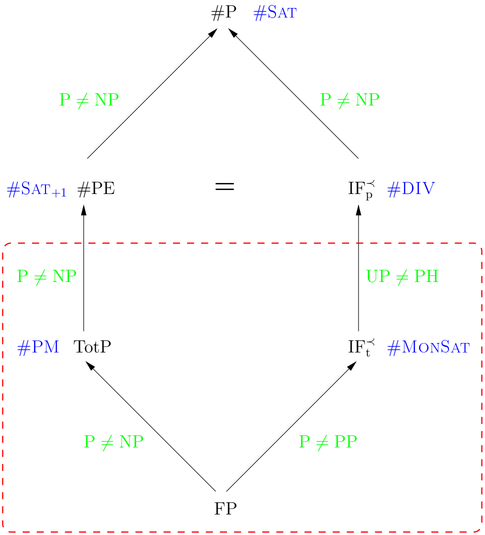
Def: $\mathrm{IF}_{\mathrm{t}}^{\mathrm{LN}}$: Lexicographical Nearest Function (Given a string compute the lex-nearest string within a defined interval)
Theorem: $\mathrm{IF}_{\mathrm{t}}^{\mathrm{LN}} = \mathrm{TotP}$

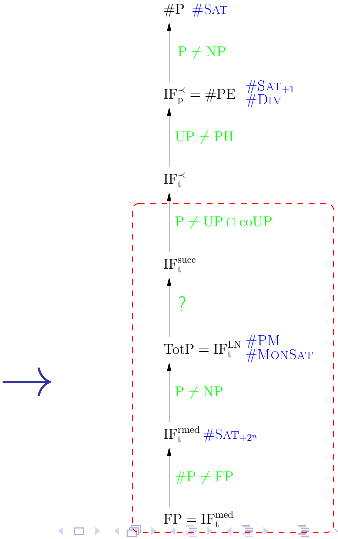
Def: $\mathrm{IF}_{\mathrm{t}}^{\mathrm{med}}$: median function.
(Fact: $\mathrm{IF}_{\mathrm{t}}^{\mathrm{med}} = \mathrm{FP}$)

Def: $\mathrm{IF}_{\mathrm{t}}^{\mathrm{rmed}}$: “relaxed” median function (there is a whole family of such).
(Fact: Contains the problem $\#\mathrm{SAT}_{+2^n}$)

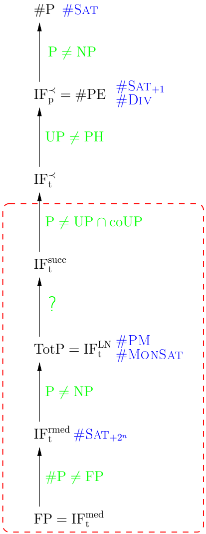
Corollary:
$\mathrm{FP} = \mathrm{IF}_{\mathrm{t}}^{\mathrm{med}} \subseteq \mathrm{IF}_{\mathrm{t}}^{\mathrm{rmed}} \subseteq \mathrm{IF}_{\mathrm{t}}^{\mathrm{LN}} = \mathrm{TotP}$

The Results [Bampas G. Pagourtzis Tentes '09]





The Results (cont.)



- Interval size characterizations: $\mathrm{TotP} = \mathrm{IF}_t^{LN}$ (also $\mathrm{FP} = \mathrm{IF}_t^{med}$).
- $\mathrm{TotP} \subseteq \mathrm{IF}_t^{\prec}$; the inclusion is proper unless $\mathrm{P} = \mathrm{UP} \cap \mathrm{coUP}$.
- A new interval size class, IF_t^{rmed}, s.t. $\mathrm{FP} \subseteq \mathrm{IF}_t^{rmed} \subseteq \mathrm{TotP}$, the inclusions being proper unless $\mathrm{FP} = \#\mathrm{P}$ and $\mathrm{P} = \mathrm{NP}$ resp.
- $\#\mathrm{SAT}_{+2^n} \in \mathrm{IF}_t^{rmed}$. Does IF_t^{rmed} contain natural “easy to decide - hard to count” problems?

Interval size characterizations

- The main idea is to consider new polynomial time feasibility constraints, other than efficient adjacency checks.
- For FP we consider the median function $\mathrm{med}_A(x, y)$
- For TotP:
  - Lexicographical Nearest Function on a p-order A: $\mathrm{LN}_A(x, y, w)$ returns the string $z \in (x, y)_A$, which is the lex nearest to w among strings in $(x, y)_A$.
  - Captures the property that given a NPTM M and a string of non-deterministic choices, we can find the lex nearest string that encodes a computation path of M.
  - All $\#\mathrm{PE}$ self reducible problems(i.e. TotP) can be expressed as interval size functions with lex-nearest function in P
  - And *vice versa*

$\mathrm{TotP} = \mathrm{IF}_{\mathrm{t}}^{\mathrm{LN}}$

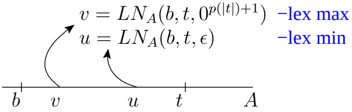
- $\mathrm{TotP} \subseteq \mathrm{IF}_{\mathrm{t}}^{\mathrm{LN}}$
  - Let M be a TotP machine.
  - The boundary functions b, t are set according to $p(|x|)$, where p is the polynomial bounding the length of M's computation.
  - An appropriate order A is defined, such that for each path of M, a string is contained in $(b(x), t(x))_A$ iff it encodes a valid computation path of $M(x)$.
  - The computation of LN_A is based on the property that given a string x, we can verify efficiently whether it encodes a computation path of M and if not, we can efficiently find the encoding of a path y, that is lex-nearest to x.

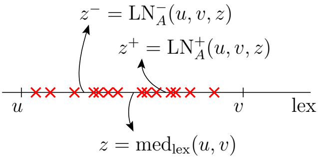
$\mathrm{TotP} = \mathrm{IF}_t^{\mathrm{LN}}$ (concl.)

- $\mathrm{IF}_t^{\mathrm{LN}} \subseteq \mathrm{TotP}$
  - We construct an NPTM M s.t. on input x, $\mathrm{tot}_M(x) = \|(b(x), t(x))_A\|$.
  - We use the LN_A function in order to re-arrange strings in $(b(x), t(x))_A$ lexicographically.
  - The lex-first and last strings of $(b(x), t(x))_A$, say f, l are computed first.
  - Then, the lex-nearest to the lex-median of $(f, l)_A$ is computed, say m.
  - The process is repeated recursively by splitting (into two computation paths) as long as (f, m) and (m, l) are nonempty.

$\mathrm{IF}_t^{\mathrm{LN}} \subseteq \mathrm{TotP}$

- If we can compute LN_A efficiently then we can also compute LN_A^+ and LN_A^- efficiently.
- We use LN_A function in order to re-arrange strings in $(b(x), t(x))_A$ lexicographically.





Functions-to-Languages Operators

- We define class operators that produce appropriate decision versions of counting problems.
- We observe that by using efficient adjacency checks ($\mathrm{IF}_t^{\prec}$) we can only tell (efficiently) whether there are any strings in the interval.
- On the other hand, in a TotP computation we can decide whether there are more than one paths.

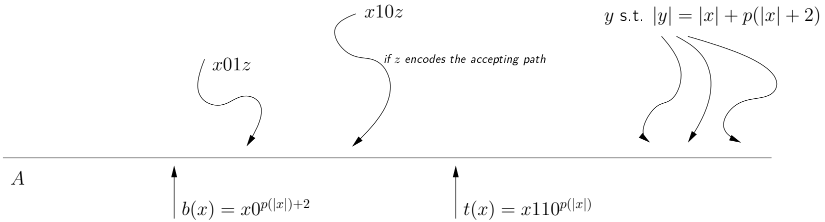
Definition

Let $\mathcal{F}$ be a function class; then $\mathcal{C}_{>1} \cdot \mathcal{F}$ is the following class of languages:

$$\mathcal{C}_{>1} \cdot \mathcal{F} = \{L \mid \exists f \in \mathcal{F}\ \forall x\, (x \in L \iff f(x) > 1)\} \ .$$

$\text{TotP} \subsetneq \text{IF}_t^{\prec}$ if $\text{P} \neq \text{UP} \cap \text{coUP}$

- $\mathcal{C}_{>1} \cdot \text{TotP} \subseteq \text{P}$
- $\text{UP} \cap \text{coUP} \subseteq \mathcal{C}_{>1} \cdot \text{IF}_t^{\prec}$



- If $\text{IF}_t^{\prec} \subseteq \text{TotP}$, then $\text{UP} \cap \text{coUP} \subseteq \mathcal{C}_{>1} \cdot \text{IF}_t^{\prec} \subseteq \mathcal{C}_{>1} \cdot \text{TotP} \subseteq \text{P}$

Open Problems

- Can the Karp closure of #PERFECT MATCHINGS, #DNFSAT, etc., be described in terms of interval size functions? In terms of path counting functions?
- Are there complete problems for the studied classes under Karp reductions? Under other suitable reductions?
- Can the approximation subclasses of $\#\mathrm{P}$ (DGGJ'00) be related to interval size functions or to path counting functions?

THANK YOU!!