Counting Classes, the Parity Class and Toda's Theorem

Zacharias Pitouras

MPLA

zacharias.pitouras@gmail.com

Outline

- The Parity Class and Toda's Theorem
  - References:
    - C. Papadimitriou, *Computational Complexity*, ch. 18.2
    - S. Arora and B. Barak, *Computational Complexity: A Modern Approach*, ch. 17.4
- Counting Classes
  - Reference:
    - Fenner SA, Fortnow LJ and Kurtz S, *Gap-Definable Counting Classes*, J. of Computer and System Sciences, **48**, 116-148 (1994)

Parity Class

- Any counting problem in $\#\mathbf{P}$ can be solved in polynomial space
  - reusing space enumerate all solutions in lexicographic order, keeping a counter of the ones that we have seen
- Thus $\#\mathbf{P}$ like the polynomial hierarchy is no more powerful than polynomial space
- A question arises:
  - how do the polynomial hierarchy and $\#\mathbf{P}$ compare in power?
  - or, does counting takes you further than quantifiers?

Parity Class continued

- **Definition.** A language L is in the class $\oplus$**P** ("odd P" or "parity P") if there is a nondeterministic Turing machine M such that for all strings x we have $x \in L$ if and only if the number of accepting computations of M on input x is odd.
  - equivalently if there is a polynomially balanced and polynomially desidable relation R such that $x \in L$ if and only if the number of y's such that $(x, y) \in R$ is odd
- The following problems are defined:
  - $\oplus$SAT: given a set of clauses, is the number of satisfying truth assignments odd?
  - $\oplus$HAMILTON PATH: given a graph, does it have an odd number of Hamilton Paths?

Parity Class continued

- **Theorem**: $\oplus$SAT and $\oplus$HAMILTON PATH are $\oplus$**P**-complete.
  - **Proof**: They are in $\oplus$**P** based on the previous second definition of $\oplus$**P** and the definition of the problems. Completeness follows from the parsimonious reductions of any problem in $\#$**P** to $\#$SAT and from that to $\#$HAMILTON PATH.

Parity Class continued

- **Theorem**: $\oplus$**P** is closed under complement.
  - **Proof**: The complement of $\oplus$SAT (whether there is an even number of satisfying truth assignments) is obviously **co** $\oplus$ **P**-complete. Next we reduce this language to $\oplus$SAT. Given any set of clauses on n variables $x_1, \cdots, x_n$ we add the new variable z, we add to all clauses the literal z, and add the n clauses $(z \Rightarrow x_i)$ for $i = 1, ..., n$. Thus any satisfying truth assignment of the old expression is still satisfying (with $z =$ false), and we have the extra all-true satisfying truth assignment (the only one with $z =$ true). Hence we increased the number of satisfying truth assignments by one and this is a reduction from the complement of $\oplus$SAT to $\oplus$SAT. Since $\oplus$SAT is both $\oplus$**P**-complete and **co** $\oplus$ **P**-complete and these classes are closed under reductions it follows that $\oplus$**P** $=$ **co** $\oplus$ **P**.

Toda's Theorem

- **Theorem** (Toda's Theorem): $\mathbf{PH} \subseteq \mathbf{P}^{\#\mathrm{SAT}}$
  - Which means that we can solve any problem in the polynomial hierarchy given an oracle to a $\#\mathbf{P}$-complete problem.
- Because **PP** (more than half of the computations of a nondeterministic machine are accepting) is closely related to $\#$SAT
  - $\mathbf{PH} \subseteq \mathbf{P}^{\mathrm{PP}}$

An alternative to Toda's Theorem

- $\oplus$**P** captures a fairly weak aspect of counting: the parity of the number of solutions.
- But it can be shown that if an **RP** machine is equipped with an $\oplus$**P** oracle it can simulate all of **NP**.
- This result uses oracle machines that are more powerful and a much weaker oracle. The class captured is the lowest level of **PH**.
- **Theorem**: $\mathbf{NP} \subseteq \mathbf{RP}^{\oplus \mathbf{P}}$
  - the proof uses similar arguments of the proof of Toda's theorem

Definitions

- **Definition.** A *counting machine* (CM) is a nondeterministic Turing machine running in polynomial time with two halting states: accepting and rejecting.
  - Every computational path must end in one of these states.
- **Definition.** Let M be a CM. We define the function $\#M : \Sigma^* \to \mathbf{Z}^+$ to be such that for all $x \in \Sigma^*$, $\#M(x)$ is the number of accepting computation paths of M on input x.
  - Similarly, $\text{Total}_M : \Sigma^* \to \mathbf{Z}^+$ is the total number of computation paths of M on input x.
  - The CM $\bar{M}$ is the machine identical to M but with the accepting and rejecting states interchanged.

The Classes

- $\#\mathbf{P} \stackrel{\text{df}}{=} \{\#M \mid M \text{ is a CM}\}$
- **PP** is the class of all languages L such that there exists M and **FP** function f such that, for all x,

$$x \in L \Leftrightarrow \#M(x) > f(x)$$

the function f is the threshold of M

- $C_{=}\mathbf{P}$ is the class of all languages L such that there exists M and an **FP** function f such that, for all x,

$$x \in L \Leftrightarrow \#M(x) = f(x)$$

The Classes continued

- For $k \geq 2$, define Mod$_k$**P** to be the class of all languages L such that there exists M such that, for all x,

  $$x \in L \Leftrightarrow \#M(x) \neq 0 \text{ mod} k$$

  the class Mod$_k$**P** is also called $\oplus\mathbf{P}$ ("Parity P") (Papadimitriou and Zachos, Goldschlager and Parberry)

- For any language L, $L \in$ **FewP** if and only if there exist a CM M and a polynomial p such that for all $x \in \Sigma^*$, $\#M(x) \leq p(\mid x \mid)$ and

  $$x \in L \Leftrightarrow \#M(x) > 0$$

The Classes continued

- For any language L, $L \in$ **Few** if and only if there exist a CM M, a polynomial p, and a polynomial-time computable predicate $A(x, y)$ such that for all $x \in \Sigma^*$, $\#M(x) \leq p(| x |)$ and

$$x \in L \Leftrightarrow A(x, \#M(x))$$

  we know **FewP** $\subseteq$ **NP** but this is not known for **Few**

Need for Closure

- The function class $\#$**P** lacks an important closure property
  - $\#$**P** functions cannot take on negative values
  - it is not closed under subtraction
- Remedy: the function class Gap**P** is a natural alternative
  - Gap**P** is the closure of $\#$**P** under subtraction
  - has all the other useful properties of $\#$**P** as well
- Gap**P** is a function class consisting of differences, or "gaps"
  - between the number of accepting and rejecting paths of **NP** Turing machines

Gaps

- **Definition.** If M is a CM, define the function $\text{gap}_M : \Sigma^* \rightarrow \mathbf{Z}$

$$\text{gap}_M \stackrel{\text{df}}{=} \#M - \#\bar{M}$$

- gap_M represents the gap between the number of accepting and the number of rejecting paths of M
- The natural gap analog of the function class $\#\mathbf{P}$
  - **Definition.**

$$\text{Gap}\mathbf{P} \stackrel{\text{df}}{=} \{\text{gap}_M \mid M \text{ is a CM}\}$$

Gaps continued

- **Lemma**. For every CM M, there is a CM N such that $\text{gap}_N = \#M$. That is $\#\mathbf{P} \subseteq \text{Gap}\mathbf{P}$.
- **Proposition**. $\text{Gap}\mathbf{P} = \#\mathbf{P} - \#\mathbf{P} = \#\mathbf{P} - \mathbf{FP} = \mathbf{FP} - \#\mathbf{P}$

Closure Properties 1-2

- Closure Property 1.

$$\text{Gap}\mathbf{P} \circ \mathbf{FP} = \text{Gap}\mathbf{P} \text{ and } \mathbf{FP} \subseteq \text{Gap}\mathbf{P}$$

- Closure Property 2.

$$\text{If } f \in \text{Gap}\mathbf{P} \text{ then } -f \in \text{Gap}\mathbf{P}$$

Closure Properties 3-4

- Closure Property 3.

  If $f \in$ Gap**P** and q is a polynomial, then the function

  $$g(x) \stackrel{\text{df}}{=} \sum_{|y| \leq q(|x|)} f(\langle x, y \rangle)$$

  is in Gap**P**

- Closure Property 4.

  If $f \in$ Gap**P** and q is a polynomial, then the function

  $$g(x) \stackrel{\text{df}}{=} \prod_{0 \leq y \leq q(|x|)} f(\langle x, y \rangle)$$

  is in Gap**P**

Closure Properties 5-6

- Closure Property 5.

  If $f \in$ Gap**P**, $k \in$ **FP**, and $k(x)$ is bounded by a polynomial in $|x|$, then the function $g(x) \stackrel{\text{df}}{=} \binom{f(x)}{k(x)}$ is in Gap**P**

- Closure Property 6.

  If $f, g \in$ Gap**P** and $0 \leq g(x) \leq q(|x|)$ for some polynomial q, then the function $h(x) \stackrel{\text{df}}{=} f(\langle x, g(x) \rangle)$ is in Gap**P**

Corollary

- An outcome of the closure properties is the following
  - **Corollary**. Gap**P** is closed under adition, subtraction and multiplication

Thank You!