$\mu\Pi\lambda\forall$

Counting Complexity: #P, #P-completeness

Nikolidaki Aikaterini

Algorithms and Complexity II

Fall semester 2012-2013

Introduction

- So far we have discussed the below problems:
  - If a desired solution exists in a problem
  - If the solution exists then which is it
- So , the resources (time and space) that we need to find this solution, are varied according to the problem.
- But there is a third important and fundamentally different kind of problem:
  - How many solutions exist
- So, we need the number of different solutions in a problem.
- Then, the class that we will discuss is #P, which was defined at first by L.G. Valiant. The source of definition is the permanent problem.

The Class #P

- **#P** belongs to the category of counting classes (of classes that there are functions to compute the several solutions for some instance of a problem.
- **Definition 1** [Val79]
  - A counting Turing machine is a standard nondeterministic TM **with an auxiliary output device** that (magically) prints in binary notation on a special tape **the number of accepting computations** induced by the input. It has (worst- case) time-complexity f(n) if the longest accepting computation induced by the set of all inputs of size n takes f(n) steps (when the TM is regarded as a standard nondeterministic machine with no auxiliary device).
- **Definition 2** [Val79]
  - # P is the class of functions that can be computed by counting TMs of polynomial time complexity.

The Class #P

- **#P** contains functions whose output is a natural number, and not just 0/1.
- **Definition 3 (#P)** [Ar09]
  - A function f : {0,1}*→N is in **#P** if there exists a polynomial p: N→N and a polynomial-time TM *M* such that for every x Є {0,1}*:

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

- **#P** consists of all functions f such that f(x) is equal to **the number of paths** from the initial configuration to an accepting configuration (in brief, "**accepting paths**") in the configuration graph $G_{M,x}$ of a polynomial-time non deterministic TM *M* on input x.
- **FP** is the set of functions computable by a deterministic polynomial time TM, is the analog of efficiently computable functions (the analog of P for functions with more than one bit of output).

The Class #P

#P=FP ?

- If **#P=FP** then **NP=P**.
- If **PSPACE = P** then **#P=FP**.
- **PP=P** *iff* **#P=FP**.
  - Let f be a function in #P. Then there is some poly-time TM M such that for every x, $f(x) = \#_M(x)$ of strings u Є $\{0,1\}^m$. For every two TM's M0+ M1 taking m-bit certificates, the TM M' that takes n+1 bit certificate where $M'(x, bu)= M_b(x,u)$. Then $\#_{M0+M1}(x)=\#_{M0}(x)+\#_{M1}(x)$. For N Є $\{0,\ldots,2^m\}$, M_N the TM that on input x, u outputs 1 iff u is smaller than N. $\#_{M_N}(x)=N$. If PP=P then we can determine in poly-time if $\#_{M_N}(x)=N + \#_M(x) \geq 2^m$.

Counting Problems

- All the problems that belong to NP decisions problems, which are in effect of counting solutions, belongs to #P and then to #P-complete.
- But, there are decisions problems, which are easy to find a solution in P, then the counting corresponding problems belongs to #P-complete, such as the PERMANENT problem is **#P-complete**.

- Some examples in effect of "counting versions" of NP-complete decision problems:
  - **#SAT**: Given a Boolean expression φ, compute the number of different truth assignments that satisfies it.
  - **#CYCLE**: Given a graph G, compute the number of simple cycles.
  - **# HAMILTON PATH**: Given a graph G, compute the number of different paths.

Counting Problems

- **#SAT** has a polynomial-time algorithm, then SAT Є P and so P=NP.
- **#CYCLE** has a polynomial-time algorithm, then SAT Є P and so P=NP [Ar09].
  - Show: if #CYCLE can be computed in polynomial time, then Ham Є P, where Ham is the NP-complete problem of deciding whether or not a given digraph has a Hamiltonian cycle. Given a graph G with *n* vertices, we construct a graph G' such that G has a Hamiltonian cycle iff has at least n^{n^2} cycles.
  - To obtain G', replace each edge (u, v) in G by the gadget, which has $m=n\log n$ levels. It is an acyclic digraph, so cycles in G' correspond to cycles in G.
  - There are 2m directed paths from u to v in the gadget, so a simple cycle of length l in G yields $(2^m)^l$ simple cycles in G'.
  - If G has a Hamiltonian cycle, then G' has at least $(2^m)^n > n^{n^2}$ cycles.
  - If G has no a Hamiltonian cycle, then G' has at least $(2^m)^{n-1} * (n^{n-1}) < n^{n^2}$ cycles.

Permanent

- Suppose that G=(U,V,E) is a bipartite graph with $U=\{u_1,\ldots, u_n\}$ and $V=\{v_1,\ldots, v_n\}$ and $E \subseteq U\mathrm{x}V$. The determinant of A^G is:

$$\det A^G = \sum_{\pi} \sigma(\pi) \prod_{i=1}^{n} A^G_{i,\pi(i)}$$

- The permanent of A^G is precisely the number of perfect matchings in G [Val79]:

$$perm A^G = \sum_{\pi} \prod_{i=1}^{n} A^G_{i,\pi(i)}$$

#P-complete

- **#SAT** is **#P-complete** [Pap 84].
  - It's a parsimonious variant of Cook's Theorem. Suppose that we have an arbitrary counting problem in #P, defined in terms of the relation Q. Show that this problem reduces to #SAT. Q can be decided by a poly-time TM M, also is poly balanced, that is, for each x the only possible solutions y have length at most $|x|^k$, the alphabet of the solutions y is {0,1}. From Cook's Theorem, on M and x, construct in O(log|x|) space a circuit C(x), with $|x|^k$ inputs, such that an input y makes the output of C(x) equal to **true** iff (x,y) Є Q. Thus the construction of C(x) is a parsimonious reduction from the counting problem of Q to the counting problem of CIRCUIT SAT. And from CIRCUIT SAT to #SAT.

- **#HAMILTON PATH** is **#P-complete** [Pap 84].
  - Not using the known reductions from 3SAT to HAMILTON PATH because there is not 1-1 reduction (for one assignment exist many Hamilton paths). But, using the TSP problem which is FP^{NP}-complete.

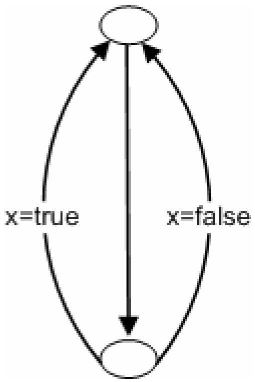
Valiant's Theorem

- **PERMANENT** is **#P-complete** [Val79].
- Steps of proof:
  - Reduction from the counting problem of paths a TM to the assignment values, that satisfy a proper construction f (#3SAT).
  - Reduction from the #3SAT to Integer – permanent.
  - Reduction from the Integer – permanent to (0,1)- permanent (mod N).
  - Reduction from the (0,1) permanent (mod N) to (0,1)– permanent.

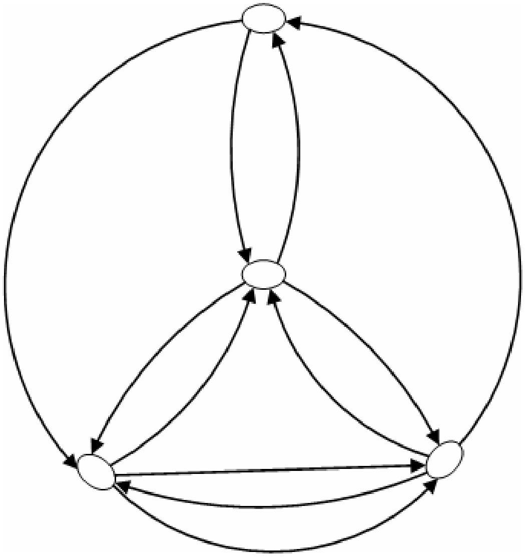
- #3SAT to Integer – permanent : There is a f Є FP where corresponds formulas in 3CNF with m clauses to matrices with entries -1,0,1,2,3: Perm (f(F))=4^{m}*s(F), where s is the number of satisfying truth assignments.

Valiant's Theorem

- For each variable, we will have a copy of the choice gadget. In any cycle cover the nodes of the graph must be covered by either the cycle to the left (x=true) or the cycle to the right (x=false).

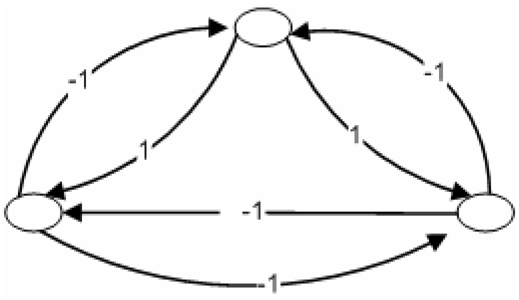


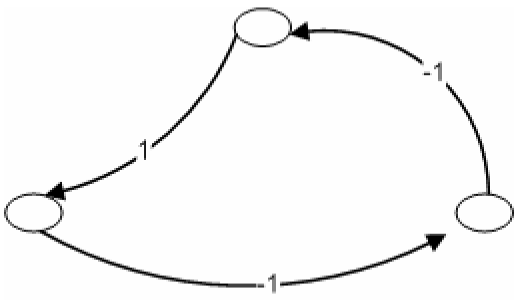
- For each clause, we will have a copy of the clause gadget. The three "external" edges are all-connected by exclusive – or's with the edges of the choice gadgets.

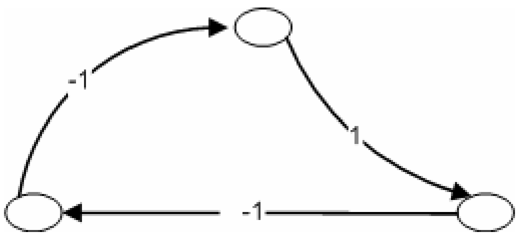


Valiant's Theorem

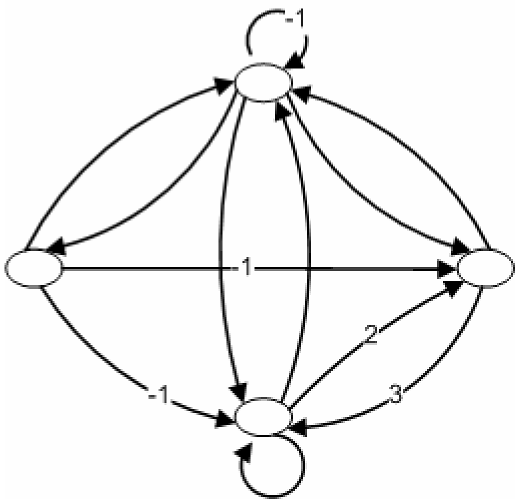
- This gadget has only two cycle covers with weight 1 and -1, corresponds







- The exclusive – or gadget



References

- [Val79] L.G. Valiant, “The complexity of computing the permanent”, Theoretical Computer Science, 189-201, 1979.
- [Ar09] S. Arora, B. Barak, “Computational Complexity: A Modern Approach”, Cambridge University Press, 2009.
- [Pap84] C. Papadimitriou, “Computational Complexity”, Addison Wesley, 1994.

Leslie Gabriel Valiant

- Leslie Gabriel Valiant (born 28 March 1949) is a British computer scientist and computational theorist. Valiant is world-renowned for his work in theoretical computer science.
- He introduced the notion of #P-completeness to explain why enumeration and reliability problems are intractable.
- Also, he introduced the "probably approximately correct" model of machine learning that has helped the field of computational learning theory grow, and the concept of holographic algorithms.
- He works in automata theory includes an algorithm for concept-free parsing, which is the asymptotically fastest known.
- He worked in computational neuroscience focusing on understanding memory and learning.
- Proved UNIQUE-SAT Є P then NP=RP (Valiant-Vazirani theorem)

Thank You!

Questions?