Simple Probabilistic Algorithm

TECHNISCHE UNIVERSITÄT DARMSTADT

Complexity Theory

Example: MaMu := $\{ (A,B,C)\in\mathbb{Z}_2^{3(m\times m)} : m\in\mathbb{N},\ C=A\cdot B \}$

- deterministic algorithm: running time $O(m^3)=O(n^{1.5})$
- world record [Strassen],[Coppersmith&Winograd]: $O(m^{2.38})$
- randomized algorithm, running time $O(m^2)=O(n)$:
  - Guess $\underline{x}\in\mathbb{Z}_2^m$ identically independently at random
  - Calculate $\underline{y}:=B\cdot\underline{x}$, $\underline{z}:=A\cdot\underline{y}$, $\underline{w}:=C\cdot\underline{x}$.
  - If $\underline{w}=\underline{z}$, accept; otherwise reject.

Amplifiable to near certainty

Lemma: a) Every $(A,B,C)\in$ MaMu gets accepted.

b) A d-dimensional $\mathbb{Z}_2$-vector space has 2^d elements.

c) For $A,B,C\in\mathbb{Z}_2^{m\times m}$ with $C\neq A\cdot B$, dim kern$(C-A\cdot B)\leq m-1$

d) Each $(A,B,C)\notin$ MaMu is rejected with probability $\geq 1/2$.

Randomized Algorithm for 3SAT

Uwe Schöning, Ulm

Sei $\underline{z}$ eine erfüllende Belegung von f

Mit Wahrscheinlichkeit $\binom{n}{\ell}\cdot 2^{-n}$ unterscheidet sich $\underline{y}$ von $\underline{z}$ an ℓ Stellen;

nach einem Durchlauf mit W'keit ≥⅓ nur noch an ℓ-1 Stellen, sonst an ℓ+1; erreiche $\underline{y}=\underline{z}$ mit Wahr'keit $\geq(⅓)^{\ell}$.

Wähle z.B. $\ell=n/2$

und $a:=20\cdot 3^{n/2}$ (Übung)

besser $a:=20\cdot 2^{n}/\binom{n}{\ell}\cdot 3^{\ell}$

für $\ell=n/4$

Exponentialzeit algorithmen

Laufzeit $(1.5)^{n}\cdot \text{poly}(n)$

Gegeb. 3KNF Formel $f(x_1,\ldots,x_n)$

Wiederhole $a(n)$-mal:

- rate Start-Belegung $\underline{y}\in\{0,1\}^{n}$
- Wiederhole $\ell(n)$-mal:
  - Falls $f(\underline{y})=1$, akzeptiere.
  - Sei C Klausel in f mit $C(\underline{y})=0$
  - Rate Literal x_k in C
  - und setze $y_k := 1-y_k$

Verwerfe. $1/\binom{n}{cn} \approx c^{cn}\cdot(1-c)^{(1-c)n}$

Randomized Complexity Classes

TECHNISCHE UNIVERSITÄT DARMSTADT

Las Vegas algorithms:	**Monte Carlo** algorithms:
always correct result,	always polynomial time,
expected time polynomial	results expected correct

Def: $L \in$ **RP** if there is a polytime NTM which

- on inputs $\underline{x} \notin L$ has only rejecting computations
- on inputs $\underline{x} \in L$ has ≥50% accepting computations.

P ⊆ RP ⊆ NP | **RP ⊆ BPP**

∃strong pseudo-random number generators?

Example: MaMu ∈ **coRP**.

Open Question: P versus **RP** versus **NP** versus **BPP**

Def: $L \in$ **BPP** if there is a polytime NTM which

- on inputs $\underline{x} \notin L$ has ≥75% rejecting computations
- on inputs $\underline{x} \in L$ has ≥75% accepting computations.

Randomized Complexity Classes

Def: $L\in$ **PP** if there is a polytime NTM which

- on inputs $\underline{x}\notin L$ has ≥50% rejecting computations
- on inputs $\underline{x}\in L$ has >50% accepting computations.

Lemma: $L\in$ **BPP** if there is polyn. p and NTM which

- on all inputs $\underline{x}\in\Sigma^n$ makes exactly $p(n)$ steps
- on $\underline{x}\notin L$ has $\geq(1-2^{-n})\cdot 2^{p(n)}$ rejecting computations
- on $\underline{x}\in L$ has $\geq(1-2^{-n})\cdot 2^{p(n)}$ accepting computations.

Exercise

Proof: Repeat O(n) times and report the majority vote

Def: $L\in$ **BPP** if there is a polytime NTM which

- on inputs $\underline{x}\notin L$ has ≥75% rejecting computations
- on inputs $\underline{x}\in L$ has ≥75% accepting computations.