Tutorial II

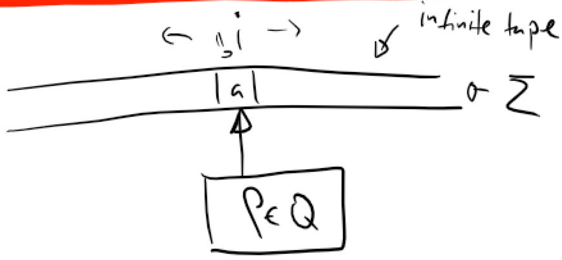
27 February 2020 14:01

Classification of randomized algorithms

- → Comlexity classes
- → Algorithm types
- → ZPP and Las Vegas Algorithms
- → 1-sided probability amplification

Complexity classes

DTM - deterministic TM



$$(Q\times\Sigma)\to(Q\times\{\leftarrow,\rightarrow,\downarrow\})$$

$$Acc \subseteq Q$$

Decision problems x - input does x belong to a language L

$L \subseteq \Sigma^*$ (set of all strings over alphabet Σ)

TM ends in an accepting state $p \in Acc$ whenever $x \in L$

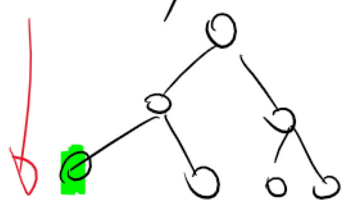
TM doesn't end in an accepting state whenever $x \notin L$

Poly inside of the input

NTM - non-deterministic

$$\to (Q\times\Sigma)\times(Q\times\{\leftarrow,\rightarrow,\downarrow\})$$

multiple options



$x \in L \Rightarrow \exists$ accepting terminating state

There exists a calculation which ends in an accepting state

Probabilistic TM this is like NTM but multiple rules for the same pair $(Q \times \Sigma)$ are assigned probabilities

$\rightarrow$

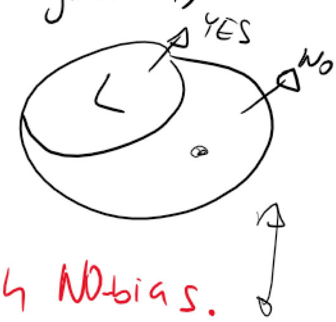
Pr to accept is the total probability to end in an accepting state

$\nearrow$ random polynomial

RP contains problems, for which there exists a TM (algorithm)

s.t. $x \in L : Pr(TM(x) \text{ accepts}) \geq 1/2 \geq \varepsilon > 0$

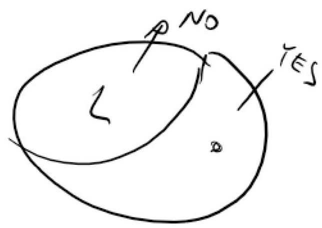
$x \notin L : Pr(TM(x) \text{ accepts}) = 0$



Problems in RP have 1-sided MC algorithms with NO-bias.

Monte Carlo polynomial

co-RP contains problems for which there exists a TM (algorithm)



s.t. $x \in L : Pr(TM(x) \text{ accepts}) = 1$ | $= 1$

$x \notin L : Pr(TM(x) \text{ accepts}) \leq 1/2$ | $\leq 1/4 \leq 1/8 \leq \varepsilon$

Problems in Co-RP have 1-sided MC algorithms with YES-bias

polynomial

1-sided probability amplification (in RP)

We want to show that a 1 sided MC with NO bias and polynomial probability of error δ can be used to construct 1-sided MC polynomial algorithm with $\varepsilon < \delta$ error.

$x \notin L$ if you run the algorithm k-times you will k answers 'NO'

$x \in L$ if you run the algorithm k-times one answer 'YES' is enough to convince you of this fact

$$\Pr\left[x \in L,\ A^k(x) = YES\right] \geq 1 - \delta^k$$

$\sqcup \sqcup \sqcup \sqcup \ldots \sqcup$

$\underset{\delta}{\sqcup}\underset{\delta}{\sqcup}\underset{\delta}{\sqcup}\underset{\delta}{\sqcup} \quad \underset{\delta}{\sqcup} \Leftarrow NO$

i.e. error is now $\varepsilon = \delta^k$

to find how many repetitions are needed to achieve chosen error ε we need to solve $\varepsilon \geq \delta^k$ for

$$\log \varepsilon \geq \log \delta^k$$

$$\log \varepsilon \geq k \log \delta$$

$$\frac{\log(\varepsilon)}{\log(\delta)} \leq k \quad \Big|$$

$$\boxed{\frac{\log(\varepsilon)}{\log(\delta(n))} \leq k(n)}$$

$\dfrac{1}{\log(\delta(n))}$ is a polynomial

$\log(\delta(n))$ in an inverse polynomial

$\delta(n) = \frac{1}{2}n$???

$\delta(n) = \frac{1}{2^n}$???

the idea is that to get an arbitrarily small error ε we need only polynomially many repetitions k.

$$\text{BPP} - \quad x \in L: \Pr[TM(x) \text{ accepts}] \geq 3/4 \quad (c > 1/2)$$
$$- \quad x \notin L: \Pr[TM(x) \text{ accepts}] < 1/4 \quad (c < 1/2)$$

arbitrary

This is called 2-sided MC algorithm

Again polynomially many repetitions are enough to get arbitrarily small error → Majority voting

k-repetitions

Answer is the majority of answers you see

We need Chernoff bounds for an easy proof ⇒ Next tutorial

2.

$\cap|$

$$\text{PP} - \quad x \in L: \Pr[TM(x) \text{ accepts}] > 1/2 \qquad (\text{consider } 1/2 + \tfrac{1}{2^n})$$
$$x \notin L: \Pr[TM(x) \text{ accepts}] \leq 1/2$$

$$\boxed{ZPP = \text{co-}RP \cap RP}$$

→ class associated with Las Vegas algorithms.

0) Problems in ZPP have both 1-MC with NO-bias (A_Y) ←
and 1-MC with YES-bias (A_N) ←

⇕

||

1.) It has LV algorithm of type 1: Always gives a correct answer with probability 1, and has $\boxed{\text{expected}}$ polynomial running time

$\Updownarrow$

2.) It has LV algorithm of type 2: Always runs in polynomial time it gives a correct answer w.p. $\geq 1/2$ or it says "I don't know".

expected n. of steps
$E(n)$

LV_1 LV_2

1.) $\Leftrightarrow$ 2.) ✓

1.) $\Rightarrow$ 2.) Run $LV_1(x)$. If the number of steps exceeds $2E(n)$ without finding a solution stop and say "I don't know"

Large majority of random choices lead to a short calculation, so we find the solution within $2E(n)$ steps w.p. more than $1/2$.

1.) this can be calculated exactly with Markov inequality

2.) $\Rightarrow$ 1.) Probability amplification

Run LV_2. if it says "I don't know" run it again (with different random choices)

$A_N A_Y$
0.) $\Rightarrow$ 1.)

1.) Run $A_Y(x)$ if it says YES it's the correct answer otherwise
2.) Run $A_N(x)$ if it says NO it is the correct answer otherwise go to 1.)

LV_2
2.) $\Rightarrow$ 0.) 1.) Run $LV_2(x)$ if correct answer is found give it as an output
2.) In case of "I don't know" output NO (NO bias

2.) => 0.) 1.) Run $C_2(x)$ if correct answer is found give it as an output

2.) In case of "I don't know" output NO (NO bias 1-sided MC)

or YES (YES-bias 1-sided MC)