Lecture 5-1. The Incompressibility Method, continued

- We give a few more examples using the incompressibility method. We avoid ones with difficult and long proofs, only give short and clean ones to demonstrate the ideas.
- These include:
  - Prime number Theorem,
  - Stack sort – T. Jiang notes
  - Boolean matrix rank
  - 1-tape Turing machine lower bound
  - Coin-weighing problem
- Then we will survey the ideas of the solutions of some major open questions. They have difficult proofs, but it is sufficient show you just the ideas.

Prime number Theorem

(Chebychev, Hadamard, de la Vallee Poussin)

- Let π(n) be #{primes ≤ n}. Then $\pi(n) \sim n/ \ln n$

Lemma 1. With incompressibility: $\pi(n) \geq \log n / \log \log n$ for some n of each length.

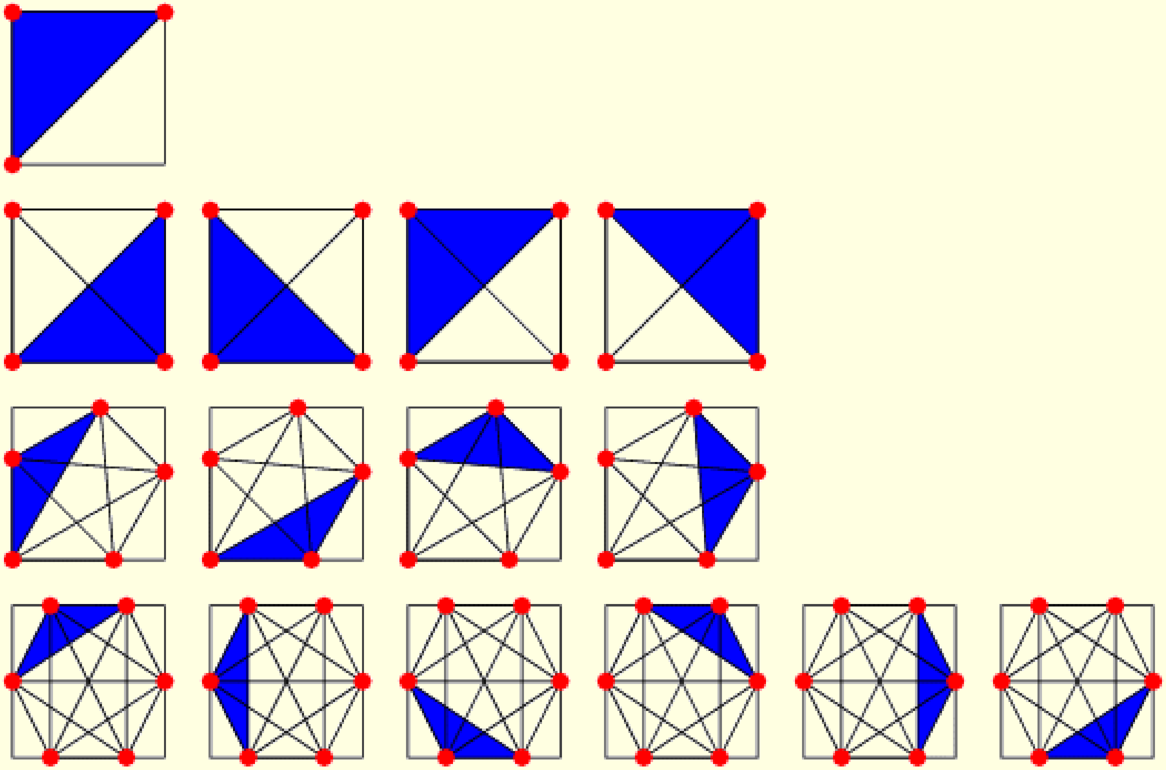
Proof. Write n= ∏ p^e , the product over all primes p ≤ n. Then e≤log n. Describe n by the exponents e in |e| bits (ignoring small terms for self-delimiting that will add a negligible –o(1) term in the lemma). If max prime is p_m, then m |log n |≥ C(n) ≥ |n| (for incompressible n). Hence m ≥ log n / log log n.

Lemma 2. With incompressibility: $\pi(n) \geq n / (\log n)^2$ for all n. With better prefix coding closer to Chebychev-Hadamard-Poussin.

Proof. Describe n by E(m)(n/p_m) with E(m) is self-delimiting description of m and p_m is largest prime. Length of description, |E(m)|+log n – log p_m ≥ C(n) ≥ |n| for incompressible n. Hence, log p_m ≤ |E(m)|. Taking |E(m)| ≤ log m + 2 log log m, we find p_m ≤ m $(\log m)^2$. Hence p_m ≤ $n_m := m (\log m)^2$. By Lemma 1 there are infinitely many primes, and $\pi(n_m) \geq n_m / (\log n_m)^2$. Since $n_{m+1}/n_m \rightarrow 1$ we are done. QED

Heilbronn’s Problem

- Throw n points in the unit square and look at the area of the smallest triangle. What is the largest possible area for any arrangement of the n points? Heilbronn thought (1950) that this is $O(1/n^2)$.



Heilbronn's Problem

- Work by Erdos, Roth, Schmidt, Komlos, Pintz, Szemeredi, and others show that it is between $O(1/n^{8/7})$ and $\Omega(\log n / n^2)$ disproving Heilbronn's conjecture.

Theorem (Jiang-Li-Vitanyi, RSA 2000). If we arrange the n points uniformly at random, then the **expected** area of the smallest triangle is $\Theta(1/n^3)$.

Proof (Sketch). Divide the unit square by a k x k grid and put the n points on grid crossings. Now we can talk about the Kolmogorov complexity of each arrangement. Assume the arrangement is (almost) incompressible. We can show that the description can be compressed too much, both if the area of the smallest triangle is too large, and if it is too small. The contradictions turn out to be such that the parameter k disappears, so hold for all k, that is for $k \to \infty$, and hence for using all points in the unit square. Since the result holds for arrangements that can be compressed a little, it holds with probability → 1 for $n \to \infty$ and on average (in expectation) QED

Boolean matrix rank (J. Seiferas and Y. Yesha)

- Consider matrix over GF(2): 0,1 elements, with usually Boolean x,+ operations. Such properties are often needed for example in proving tradeoff optimal bound TS=$\Omega(n^3)$ for multiplying 2 matrices.

Theorem. For each n, there is an n x n matrix over GF(2) s.t. every submatrix of s rows and n-r columns has at least rank s/2, for every 2log n<r,s<n/4.

Proof. Take $|x|=n^2$, $C(x)\geq n^2$. Form an n x n matrix with x, one bit per entry. For any submatrix R with s rows and n-r columns, if R does not have rank s/2, then s/2+1 rows can be linearly described by other rows. Then you can compress the original matrix, hence x. QED

Coin weighing problem

- A family ***D***={D_1,D_2, … , D_j} of subsets of N={1,...,n} is called a distinguishing family for N, if for every two distinct subsets M and M’ of N there exists an i (1≤i≤j) s.t. |D_i ∩ M| is different from |D_i ∩ M’|.
- Let f(n) denote the minimum of |***D***| over all distinguishing families for N.
- To determine f(n) is known as coin-weighting problem.
- Erdos, Renyi, Moser, Pippenger:

$$f(n)\geq(2n/\log n)[1+O(\log\log n/\log n)]$$

Theorem: $f(n) \geq (2n/\log n)[1+O(\log\log n/\log n)]$

Proof. Choose M such that $C(M|\boldsymbol{D}) \geq n$.

Let $d_i=|D_i|$ and $m_i=|D_i \cap M|$. Since M is random, the value m_i is within the range $d_i/2 \pm O(\sqrt{(d_i \log n)})$. Therefore, given d_i, each m_i can be described by its discrepancy with $d_i/2$, with gives

$$C(m_i|D_i) \leq \frac{1}{2} \log d_i + O(\log\log n)$$
$$\leq \frac{1}{2} \log n + O(\log\log n)$$

Since $\boldsymbol{D}$ is a distinguishing family for N, given $\boldsymbol{D}$, the values of $m_1, \ldots, m_j$ determine M. Hence

$$C(M|\boldsymbol{D}) \leq C(m_1, \ldots ,m_j|\boldsymbol{D}) \leq \Sigma_{i=1..j} [\frac{1}{2} \log n+O(\log\log n)]$$

This implies $f(n) \geq (2n/\log n)[1+O(\log\log n / \log n)]$.

QED

A simple Turing machine lower bound

- Consider one-tape TM. Input tape is also work tape, allow read/write, two-way head.

Theorem. It takes $\Omega(n^2)$ time for such TM M to accept $L=\{ww \mid w \in \Sigma^*\}$.

Proof (W. Paul). Take w s.t. $C(w|n) \geq |w| = n$. Consider M's computation on input: $0^n w 0^n w$. Consider the shortest crossing sequence on the second block 00…0.

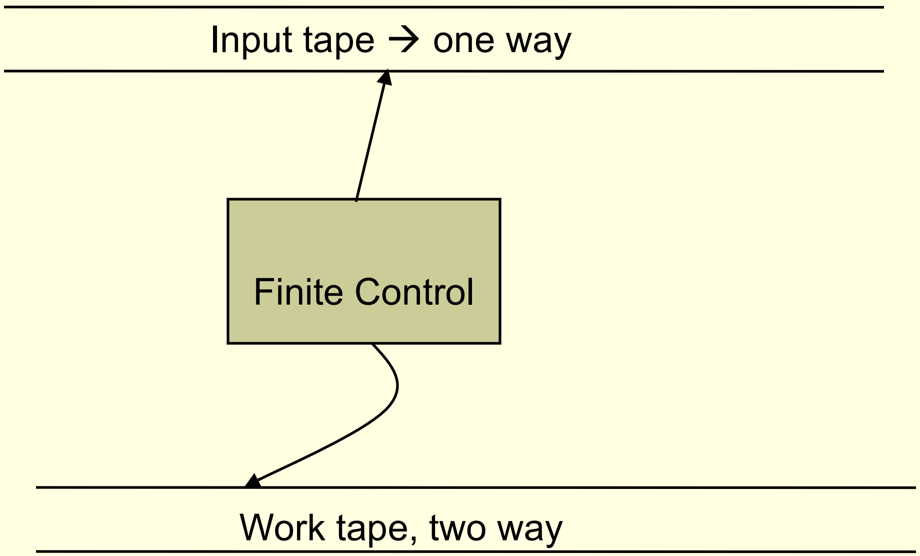
  - If it is $\Omega(n)$, then the computation time is $\Omega(n^2)$.
  - If it is $o(n)$, we can use this crossing sequence to find w by simulating M's computation on the "right side" of the crossing sequence, in order to match the crosing sequence, trying all the strings of length n. Only w has the correct c.s (or we accept another language). This way we find w searching through all candidate strings of length n using |c.s.| +O(1) bits. Then $C(w|n)=o(n)$: contradiction. QED

Solutions to open questions

- We will tell the histories of some open questions and the ideas of how they were solved by the incompressibility method.
- We will not be able to give detailed proofs to these problems … but hopefully by telling you the ideas, you will be convinced enough and able to reconstruct the details on your own.
- Through these stories, we hope to convince you of the power of Kolmogorov complexity and hope you will extend it. As Fortnow puts it: May your proofs be short and elegant.

1 tape vs 2 tape Turing machines

- Standard (on-line) TM Model:



- Question since the 1960's: Are two work tapes better than 1 work tape? How many works tapes are needed?

History

- 1965. Hartmanis & Stearns: 1 work tape TM can simulate k>1 tape TM in $O(n^2)$ time.
- 1963. Rabin: 2 work tapes are better than 1.
- 1966. Hennie-Stearns: 2 work tapes can simulate k tapes in O(nlogn) time.
- 1982. Paul: $\Omega(n(\log n)^{1/2})$ lower bound for 1 vs 2 work tapes.
- 1983. Duris-Galil: Improved to $\Omega(n\log n)$.
- 1985. Maass, Li, Vitanyi: $\Omega(n^2)$ tight bound, by incompressibility method, settling the 30 year effort.

How did we do it

- Here is the language we have used to prove a (simpler) $\Omega(n^{1.5})$ lower bound:

  $L=\{x_1@x_2@ \dots @x_k\#y_1@ \dots @y_l\#0^i1^j : x_i=y_j\}$
- Choose random x, $C(x)\geq|x|=n$, evenly break x into $x_1 \dots x_k$, $k=\sqrt{n}$.
- Then the two work tape machine can easily put x_i blocks on one tape and y_j blocks on the other. Then it accepts this language in linear time.
- However, the one work tape machine has trouble where to put these blocks. Whichever way it does it, there are bound to be some x_i and y_j blocks that are far away, then our previous proof works. The proof needs to worry that not many blocks can be stored in a small region (they are non-compressible strings, hence intuitively we know they can't be). The nice thing about Kolmogorov complexity is that it can directly formulate your intuition into formal arguments.
- To improve to $\Omega(n^2)$ lower bound, we just need to extend the reasoning from 1 pair to n pairs. Then argue there are O(n) pairs of (x_i,y_j) need to be matched and they are O(n) away.

Continued ....

- Li, Vitanyi,1988 Inf.Contr., Nondeterministic TMs, stacks, tapes, queues, Simulating 1 queue by one tape takes n^2 deterministically and $n^{4/3}/\log n$ nondeterministically (about the upper bounds)
- Li, Longpre, Vitanyi, Structure Compl. Conf. 1986, SIAM J Compl 1992, Simulating 1 stack (thus also tape) by 1 queue takes $n^{4/3}/\log n$
- 1 queue by 1 tape takes n^2 determintic case, $n^{4/3}/\log n$ in nondeterministic case.
- 2 queues (2 tapes) by 1 queue takes $n^2/(\log^2 n \log\log n)$ nondeterministically, and n^2 deterministically.

2 heads are better than 2 tapes

- Question: Are 2 heads on one storage tape better than 2 storage tapes with 1 head each (apart from input- and output tapes).
- J. Becvar Kybernetica 1965, A.R. Meyer, A.L. Rosenberg, P.C. Fisher 1967 IEEE-SSAT = FOCS raised question 2 heads versus 2 tapes.
- H.J. Stoss k tapes can linear-time simulate k heads , Computing 1970
- Fisher, Meyer, Rosenberg 11k-9 tapes real-time simulation of k heads 1972 JACM
- Leong, Seiferas 4k-4 tapes real-time simulate k heads 1981 JACM
- 2 heads better than 2 tapes in real-time, for 2-dimensional tapes, W. Paul TCS 1984 (using K-complexity) This is much easier.
- Papers by R. Reischuk, R. Tarjan, Fan Chung, W. Paul, ...
- P. Vitanyi used L = {xy#x: x,y over {0,1}} to prove "if 2 tapes accept L in real-time, then both heads get linearly far from origin'' using K-complexity, JCSS 1984
- T. Jiang, J. Seiferas, P. Vitanyi, 2 heads better than two tapes in real-time! Using K-complexity etc. STOC 1994, JACM 1997

How we did it

- Use FIFO language {xy#x} to separate. Clearly, 2 heads on 1 tape can accept in real time. So we need to show that 2 one-headed tapes cannot.
- Vitanyi JCSS1984 showed that for 2 tapes in real time both heads go linear far away from origin: ``Far-out lemma''.
- Using a single incompressible input xy the two heads describe a trajectory in 2-dimensions, moving out of the square with linear sides (in input length) with left bottom corner at origin.
- Using a complicated `overlap' lemma show that the trajectory contains linearly many points with no coordinate in common.
- Use `anti-holography' lemma
- Use `symmetry of information theorem' many times...

K-head PDA's

- Model: Normal finite or pushdown automaton with k one-way input heads. Thus k-FA or k-PDA.
- These are natural extensions of our standard definition of FA and PDA.
- Two conjectures:
  - 1965, Rosenberg Conjecture: (k+1)-FA > k-FA
  - 1968, Harrison-Ibarra Conjecture: (k+1)-PDA > k-PDA

A tale of twin conjectures

- 1965 Rosenberg actually claimed a proof for (k+1)-FA > k-FA. But Floyd subsequently found error and the proof fell apart.
- 1971 (FOCS), Sudborough proved 3-FA > 2-FA.
- Ibarra-Kim: 3-FA > 2-FA
- 1976 (FOCS) Yao-Rivest: (k+1)-FA > k-FA.
- 1973 Ibarra: both conjectures true for 2-way input. This is by diagonalization, does not work for 1-way machines.
- 1982, Miyano: If change pushdown store to counter, then Harrison-Ibarra conjecture is true.
- 1983, Miyano: If input is not bounded, then H-I true.
- 1985,Chrobak: H-I conjecture true for deterministic case – using traditional argument, extremely complicated and tedious.
- 1987 (FOCS), Chrobak-Li: Complete solution to Harrison-Ibarra conjecture, using incompressibility method. (The same argument also gives a cute simplification to Yao-Rivest proof.)

How we did it

- The language we have used is:

$$L_b = \{w_1\# \ldots \#w_b \$ w_b \# \ldots \# w_1 \mid w_i \in \{0,1\}^*\}$$

Theorem. L_b can be accepted by a k-PDA iff $b \leq k(k-1)/2$.

When $b \leq k(k-1)/2$, then a k-FA can do it by pairing its k heads at right places at right time.

When $b > k(k-1)/2$, then we can again choose random w and break it into w_i blocks. Then we say there must be a pair of (w_i, w_i) that are indirectly matched (via the pushdown store). But when storing into pushdown store, w_i is reversed, so it cannot be properly matched with its counter part w_i. We will also need to argue information cannot be reversed, compressed etc. But these are all easy with Kolmogorov complexity.

String-matching by k-DFA

- String matching problem:

  L={x#y | x is a substring of y}
- This one of the most important problems in computer science (grep function for example)
- Hundreds of papers written.
- Many efficient algorithms – KMP(Knuth-Morris-Pratt), BM (Boyer-Moore), RK (Rabin-Karp). Main features of these algorithms:
  - Linear time
  - Constant space (not KMP, BM), i.e. multihead finite automaton. In fact, a two-way 6-head FA can do string matching in linear time (Galil-Seiferas, 1981, STOC)
  - No need to back up pointers in the text (e.g. KMP).
- Galil-Seiferas Conjecture: Can k-DFA for any k, do string matching?

History

- Li-Yesha: 2-DFA cannot.
- Gereb-Graus-Li: 3-DFA cannot
- Jiang-Li 1993 STOC: k-DFA cannot, for any k.

How we did it

- I will just tell you how we did it for 2-DFA.
- Remember the heads are one-way, and DFA does not remember much.
- We can play a game with the 2-DFA with input (of course with Kolmogorov random blocks):

  xy # y'x'

  such that x' can be x and y' can be y, so if the 2-DFA decides to match x, x' directly, then it won't be able to match y, y' directly (and vice versa), so then we simply make x' different from x, but y'=y. Then without the two heads simultaneously at y and y', we will argue, as before, that finite control cannot do it.

How far can we go?

- We have presented at least a dozen of problems that were solved by the incompressibility methods. There are many more … such problems (these include the important switching lemma in circuit complexity as well quantum complexity bounds).
- But can Kolmogorov complexity help to prove higher bounds? Or it is limited to linear, nlogn, n^2 bounds?
- Can we import some probabilistic method tools?
- If such a tool simply does not work for certain things, like NP $\neq$ P, can we be certain about it? (prove this?)