

The Tale of One-way Functions: Part 1.

Leonid A. Levin*
Boston University†

Fall 2000

*All the king's horses, and all the king's men,
Couldn't put Humpty together again.*

Abstract

This is the first part of an article based on my talks [Levin 99]. I discuss the task of inverting functions, mention frivolous models, such as quantum factoring machines, and explore subtleties of probabilistic aspects of the problem. I release this part since it may be fun to read by itself: subsequent parts, concentrating on the ideas of completeness, may be more technical. I also hope to benefit from criticism I may receive while working on the completion.

1 Intro I: Inverting Functions

From times immemorial, humanity gets frequent, often cruel, reminders that many things are easier to do than to reverse. When the foundations of mathematics started to be seriously analyzed, this experience found immediately a formal expression.

1.1 An Odd Axiom

Over a century ago George Cantor reduced all the great variety of math concepts to just one – the concept of sets – and derived all math theorems from just one axiom scheme – the Cantor's Postulate. For each set-theoretical formula $A(x)$ it postulates the existence of a set containing those and only those x satisfying A . This axiom looked a triviality, almost a definition, but was soon found to yield more than Cantor wanted, including contradictions. To salvage its great promise, Zermelo, Fraenkel, and others pragmatically replaced the Cantor's Postulate with a collection of its restricted cases, limiting the types of allowed properties A . The restrictions turned out to bring little inconvenience, precluded (so far) any contradictions, and the axioms took their firm place in the foundation of math.

In 1904 Zermelo noticed that one more axiom was needed to derive all known math, the (in)famous Axiom of Choice: every function f has an inverse g s.t. $f(g(x)) = x$ for x in the range of f . It was accepted reluctantly; to this day proofs dependent on it are being singled out. Its strangeness was not limited to going beyond Cantor's Postulate; it brought paradoxes! Allow me a simple illustration.

Consider the additive group $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ of reals mod 1 as points $x \in [0, 1)$ on a circle; take its subgroup $\mathbf{Q} \subset \mathbb{T}$ of decimal fractions $a/10^b$. Let $f(x)$ be the (countable) coset $x + \mathbf{Q}$, i.e., f projects \mathbb{T} onto its factor-group \mathbb{T}/\mathbf{Q} . Any inverse g of f then selects one representative from each coset. Denote $G = g(f(\mathbb{T}))$ the image of such a g ; then each $x \in \mathbb{T}$ is brought into G by exactly one rational shift $x + q, q \in \mathbf{Q}$. Now I will deviate from the standard path to emphasize the elementary nature of the paradox. One last notation: $q' = (10q \bmod 1)$ is $q \in \mathbf{Q}$, shortened by the removal of its most significant digit.

For a pair $p, q \in \mathbf{Q}$, I bet 2:1 that $x + p$ rather than $x + q$ falls in G for a random $x \in \mathbb{T}$. The deal should be attractive to you since my bet is higher while conditions to win are completely symmetric under rotation of x . To compound my charitable nature to its extreme, I offer such bets for all $q \in \mathbf{Q}, p = q'$, not just one pair. If you rush to accept, we choose a random x by rolling dice for all its digits and find the unique $\mathbf{q} \in \mathbf{Q}$ for which $x + \mathbf{q} \in G$. Then I lose one bet for this \mathbf{q} and win ten (for each q such that $q' = \mathbf{q}$). Generosity pays!

This paradox is not as easy to dismiss as is often thought. Only 11 bets are paid in each game: no infinite pyramids. Moreover, if x is drawn from a sphere S_2 , a finite number even of unpaid bets suffices: [Banach, Tarski, 24] construct 6 pairs, each including a set $A_i \in S_2$ and a rotation T_i ; betting $x \in A_i$ versus $T_i(x) \in A_i$, they lose one bet and win two for each x . Our above $x + p$ and $x + q$ are tested for the same condition, differ in finitely many digits whose finitely many combinations are equally distributed, given any string of less significant digits of x . One can refuse a thought experiment of rolling the infinite number

*Supported by NSF grant CCR-9820934.

†Computer Science department, 111 Cummington St., Boston, MA 02215; (e-mail to Lnd@bu.edu)

of digits of x or a question of whether $x + q \in G$. But this amounts to rejecting basic concepts of set theory. It is simpler to interpret the refusal to bet as a hidden disbelief in the Axiom of Choice.

1.2 Finite objects: exhaustive search

These problems with inverting functions have limited relevance to computations. The latter deal with finite objects which are naturally well-ordered, rendering Axiom of Choice unnecessary. There are other mitigating considerations. Shannon Information theory says a random variable x has as much information about a function $f(x)$ of it as $f(x)$ has about x . Kolmogorov Information theory extended the concept to arbitrary x , not necessarily random: $I(x : y)$ is the difference between the smallest lengths of programs generating y and of those transforming x to y respectively. Kolmogorov and myself proved in 1967 that this quantity is, like Shannon's, symmetric too, albeit approximately [Zvonkin Levin, 1970].

The proof involved a caveat: computationally prohibitive exhaustive search of all strings of a given length. For instance, the product pq of two primes contains as much, i.e. all, information about them as vice versa, but [RSA] and great many other things in modern cryptography depend on the assumption that recovering the factors of pq is infeasible. Kolmogorov suggested at the time that this information symmetry theorem may be a good test case to prove that for some tasks exhaustive search cannot be avoided (in today's term $P \neq NP$).

The RSA application marked a dramatic twist in the role of the inversion problem: from a notorious troublemaker to a priceless tool. RSA was the first instance of the flood of bewildering applications which soon followed. At heart of many of them was the discovery that the hardness of every one-way function $f(x)$ can be focused into a *hard-core* bit, i.e. a predicate $b(x)$ which is as hard to determine from $f(x)$, or even to guess with any noticeable correlation, as it is to recover x completely. [Blum, Micali, 82] found the first instance for $f(x) = a^x \bmod p$, which was soon followed with the cases of RSA and Rabin function $x^2 \bmod n, n = pq$, and of [Yao 82] "grinding" functions $f^*(x_1, \dots, x_n) = f(x_1), \dots, f(x_n)$. [Goldreich Levin, 89] proved the general case.

The importance of such hard-cores comes from their use, proposed in [Blum Micali], [Yao] for deterministic generation of unlimited flow of perfectly random bits from a small random seed s . In the case of permutation f such generators are straightforward $g_s(i) = b(f^i(s))$; the general case was worked-out in [Hastad Impagliazzo Levin Luby]. With the

barrier between random and deterministic processes thus broken, many previously unthinkable feats were demonstrated in the 80s. Zero-knowledge proofs, implementation of arbitrary protocols between distrusting parties (e.g., card games) as full information games, are just a couple of many well known examples. This period was truly a golden age of Computer Theory brought about by the discovery of the use of one-way functions.

2 Intro II: Extravagant Models

2.1 The Downfall of the RSA

This development was all the more remarkable as the very existence of one-way (i.e., easy to compute, infeasible to invert) functions remains unproven and subject to repeated assaults. The first came from Adi Shamir himself, the "S" in RSA. He proved that factoring (on infeasibility of which RSA depends) can be done in polynomial number of arithmetic operations. This result uses a so called "unit-cost model" which charges one unit for each arithmetic operation, however long the operands. Squaring a number repeatedly doubles its length and quickly brings it to cosmological sizes. Embedding a huge array of ordinary numbers into such a long one allows one arithmetic operation to replace a great amount of work, e.g., checking exponentially many factor candidates. The closed minded cryptographers, however, were not convinced and this result brought a dismissal of the unit-cost model, not the RSA :-).

Another, not a dissimilar attack is raging this very moment. It started with the brilliant result of Peter Shor. He factors integers in polynomial time using an imaginary device Quantum Computer (QC) inspired by the laws of Quantum physics taken to their extreme.

2.2 Quantum Computers.

QC has n interacting elements called q-bits. A pure state of each is a unit vector on a complex plane \mathbb{C}^2 . Its projections on the two coordinates are quantum amplitudes of its two boolean values. A state of the entire machine is a vector in the tensor product of n planes. Its 2^n coordinate vectors are tensor-products of q-bit basis states, one for each n -bit combination. The machine is cooled, isolated from the environment nearly perfectly, and initialized in one of its basis states representing the input and empty memory bits. The computation is arranged as a sequence of perfectly reversible interactions of the q-bits, putting

their combination in the superposition of rapidly increasing number of basis states, each having an exponentially small amplitude. The environment may intervene with errors; the computation is done in an error-correcting way, immune to such errors as long as they are few and of special restricted forms. Otherwise, the equations of Quantum Mechanics are obeyed with unlimited precision. This is crucial since the amplitudes are exponentially small and deviations in remote (hundredth or even millionth) decimal places would overwhelm the content completely. In [Shor] such computers are shown capable of factoring in polynomial time. The exponentially many coordinates of their states can, roughly speaking, explore one potential factor each and concentrate the amplitudes in the one which works.

2.3 Small Difficulties.

There are many problems with such QCs, though. For instance, thermal isolation cannot be perfect. Tiny backgrounds of neutrinos, gravitational waves, and other exotics, cannot be shielded. Their effects on quantum amplitudes need not satisfy the restrictions error-correcting tools depend on. Moreover, non-dissipating computing gates, even classical, remain a speculation. Decades past, their existence was cheerfully proclaimed and even proven for worlds where the laws of physical interaction can be custom-designed. In our world, where the electromagnetic interaction between electrons, nuclei, and photons is about the only one readily available, circuits producing less entropy than computing remain hypothetical. So, low temperatures have limits and even a tiny amount of heat can cause severe decoherence problems. Furthermore, the uncontrollable degrees of freedom need not behave simply like heat. Interaction with the intricately correlated q-bits may put them in devilish states capable of conspiracies which beat the imagination.

2.4 Remote Decimals.

All such problems, however, are peanuts. The major problem is the requirement that basic quantum equations hold to multi-hundredth if not millionth decimal positions where the significant digits of the relevant quantum amplitudes reside. We have never seen a physical law valid to over a dozen decimals. Typically, every few new decimal places require major rethinking of most basic concepts. Are quantum amplitudes still complex numbers to *such* accuracies or they become quaternions, colored graphs, or sick-humored gremlins? I suspect physicists would doubt

even the laws of arithmetic pushed that far :-). In fact, we know that the most basic laws cannot all be correct to hundreds of decimals: this is where they stop being consistent with each other!

And what is the physical meaning of 500 digit long numbers? What could one possibly mean by saying “This box has a remarkable property: its many q-bits contain the Ten Commandments with the amplitude whose first 500 decimal places end with 666”? What physical interpretation could this statement have even for just this one amplitude? Close to the tensor product basis one might have opportunities to restate the assertions using several short measurable numbers instead of one long. Such opportunities may also exist for large systems, such as lasers or condensates where individual states matter little. But QC factoring uses amplitudes of exponential number of highly individualized basis states. I doubt anything short of the most generic and direct use of these huge precisions would be easy to substitute.

2.5 Too Small Universe.

QC proponents often say they win either way, by making a working QC or by finding a correction to Quantum Mechanics. E.g., in [21] Peter Shor says: “If there are non-linearities in quantum mechanics which are detectable by watching quantum computers fail, physicists will be VERY interested (I would expect a Nobel prize for conclusive evidence of this).”

Consider, however, this scenario. With few q-bits, QC is eventually made to work. The progress stops, though, long before QC factoring starts competing with ordinary PCs. The QC people then demand some noble prize for the correction to the Quantum Mechanics. But the committee wants more specifics than simply a non-working machine, so something like observing the state of the QC is needed. Then they find the Universe too small for observing individual states of the needed dimensions and accuracy. (Raising sufficient funds to compete with paper and pencil factoring may justify a Nobel Prize in Economics :-).

Let us make some calculations. In cryptography the length n of the integers to factor may be a thousand bits (and could easily be millions.) By $\sim n$ I will mean a reasonable power of n . A $2^{\sim n}$ dimensional space H has $2^{2^{\sim n}}$ of roughly different vectors. Take a generic $v \in H$. The minimal size of a machine which can recognize or generate v (approximately) is $K = 2^{\sim n}$ – far larger than our Universe. This comes from a cardinality argument: $2^{\sim K}$ machines of K atoms. Let us call such v “mega-states”.

There is a big difference between untested and

untestable regimes. Claims about individual megastates are untestable. I can imagine a feasible way to separate any *two* QC states *from each other*. However, as this calculation shows, no machine can separate a generic QC state from the set of all states more distant from it than QC tolerates. So, what thought experiments can probe the QC to be in the state described with the accuracy needed? I would allow to use the resources of the entire Universe, but *not more!*

2.6 Metric versus Topology.

A gap in quantum formalism may be contributing to the confusion. Approximation has two subtly different aspects: metric and topology. Metric tells how close is our ideal point to a specific wrong one. Topology tells how close it is to the combination of all unacceptable (non-neighboring) points. This may differ from the distance to the closest unacceptable point, especially for quantum systems.

In infinite dimensions the distinction between 0 and positive separation varies with topologies. In finite dimensions 0-vs.-positive distinction is too coarse: all topologies agree. Since 2^{500} is finite only in a very philosophical sense, one needs some quantitative refinement, some sort of a weak-topological (not metric) *depth* of a neighborhood such that resources required for precision to a given depth are polynomial in it. Then, precision to reasonable depths would be physical, e.g., one could generate points inside the neighborhood, distinguish its center from the outside, etc.

Metric defines ε -neighborhoods and is richer in that than topology where the specific value of ε is lost (only $\varepsilon > 0$ is assured). However, metric is restricted by the axiom that the intersection of any set of ε -neighborhoods is always another ε -neighborhood. Quantum proximity may require both benefits: defined depth ε and freedom to express it by formulas violating the “intersection axiom”. Here is an example of such violation, without pretense of relevance to our needs. Suppose a neighborhood of 0 is given by a set of linear inequalities $f_i(x) < 1$; then its depth may be taken as $1/\sum_i \|f_i\|$. Restricting x to the unit sphere would render this depth quadratically close to metric depth. A more relevant formula may need preferred treatment of tensor product basis.

2.7 The Cheaper Boon.

QC of the sort that factors long numbers seems firmly rooted in science fiction. It is pity that popular accounts do not distinguish it from much more believ-

able ideas, like Quantum Cryptography, Quantum Communications, and the sort of Quantum Computing that deals primarily with locality restrictions, such as fast searching long arrays. It is worth noting that the reasons why QC must fail are by no means clear; they merit thorough investigation. The answer may bring much greater benefits to understanding of basic physical concepts than any factoring device could ever promise. The present attitude is analogous to, say, Maxwell selling the Daemon of his famous thought experiment as a path to cheaper electricity from heat. If he did, much of insights of today's thermodynamics might be lost or delayed.

For the rest of the article we will ignore any extravagant models and will stand firmly rooted in the Polynomial Overhead Church-Turing Thesis: Any computation that take t steps on s -bit device can be simulated by a Turing Machine in $s^{O(1)}t$ steps within $s^{O(1)}$ cells.

3 The Treacherous Averaging

Worst-case hardness of inverting functions may come with no significant implications. Imagine that all instances come in two types: “easy” and “hard”. The easy instances x take $\|x\|^2$ time. An exponential expected time is required *both* to solve any hard instance, *and to find* any such instance. So, the Universe would be too small to ever produce instances that it is too small to solve. Then, inversion problem could still never pose a practical difficulty. It is “generic”, not worst-case, instances that both frustrate algorithm designers and empower cryptographers to do their incredible feats. The definition of “generic,” however, requires a great care.

3.1 Las Vegas Algorithms

First we must agree on how to measure the performance of inverters. Beside instances $x = f(w)$, algorithms $A(x, \alpha)$ inverting one-way functions f can use random coin flips sequences $\alpha \in \{0, 1\}^N$. They never need a chance for (always filterable) wrong answers. So, we restrict ourselves to *Las Vegas* algorithms which can only produce a correct output, abort, or diverge.

For any given instance, the algorithm’s performance has two aspects: the expected volume¹ of computation, and a chance of producing the output. It is important that both volume average and success probability are taken only over A ’s own coin-flips α ;

¹I say *volume* rather than *time*, for greater robustness in case of massively parallel models.

the instance x is chosen by the adversary. This multiplicity of performance aspects, however, is illusive: they are freely interchangeable. One can always double the chance of success at the price of doubling the expected computation, or vice versa, by simply running the algorithm twice or with half-a-chance.

Thus, one can always, without loss of generality, normalize the algorithms to any standard expected volume of computation and consider the chance of success as the only relevant performance measure. This volume bound may be $O(1)$ if the model of computation is very specific. If a flexibility between several reasonable models is desired, polynomial bounds, specific to each algorithm, may be preferable. There is one obstacle: the set of algorithms with a restricted expected complexity is not recursively enumerable. We can circumvent this problem by using the following enforceable form of the bound.

The Las Vegas algorithm $A(x, \alpha)$ would then start with a given bound $b(x)$ on expected computation volume. We can denote this $A \in LV(b)$, doubling the meaning of V as “Vegas” and “volume”. At any time the algorithm can bet a part of the remaining volume, so that it is doubled or subtracted depending on the next coin flip. $LV(O(1))$ suffices for most purposes and we will abbreviate it as L .²

Despite its tight $O(1)$ expected complexity bound, L is robust since any Las Vegas algorithm can be put in this form, preserving the ratio between the complexity bound and success probability. The role of running time is played by the inverse success probability, i.e., the number of runs required for a constant chance of success. An extra benefit is that a reader adverse to bothering with the inner workings of computers can just accept their restriction to L and view all further analysis in purely probabilistic terms!

3.2 Multi-median Time

Averaging over the instance x is, however, much trickier. It is not robust to define generic complexity of an algorithm $A(x)$ running in $t(x)$ steps as its expected time $\mathbf{E}_x t(x)$. A different device may have a quadratic time overhead. For instance, reversing an input string requires quadratic time on a Turing machine with one tape, but only linear time with two tapes. It may be that a similar overhead exists for

²Pronounced “Las” algorithms, hinting at Las Vegas, its inventor Laszlo Babai, and the Spanish definite article :-). I would like to stress that no YACC (Yet Another Complexity Class) is being introduced here. L is a *form* of algorithms; this is much less abstract than a class of algorithms or, especially, than a class of problems to be solved by a class of algorithms. Besides, it is not really new, just a slight tightening of the Las Vegas restriction.

much slower algorithms too. Then $t(x)$ may be, say, $\|x\|^2$ for $x \notin 0^*$, while $t(0\dots 0)$ may be $2^{\|x\|}$ for one device and $4^{\|x\|}$ for another. Take x uniformly distributed on $\{0, 1\}^n$. Then $\mathbf{E}_x t(x)$ for these devices would be quadratic and exponential respectively: averaging does not commute with squaring. Besides, this exponential average hardness is misleading, since the hard instances would never appear in practice!

More device-independent would be the *median* time, the minimal number of steps spent for the *harder half* of instances. This measure, however, is not robust in another respect: it can change dramatically if its *half* threshold is replaced with, say, a *quarter*.

Fortunately these problems disappear as one of the many benefits of our Las Vegas conventions. One can simply take algorithms in L and measure their chance of success for inputs chosen randomly with a given distribution. The inverse of this chance, as a function of, say, input length is a robust measure of security of one-way function. This measure is important in cryptography, where any noticeable chance of breaking the code must be excluded.

A different measure is required for positive tasks aimed at success for almost all instances. We start with considering a combined distribution over all instance lengths. Typically, we take an L -distribution, i.e., a distribution of an L -algorithm’s outputs on empty input. If the generator is not algorithmic, the output length can serve instead of complexity in the definition of L . The instances of length n should have a combined polynomial probability, e.g., $\theta(1)/(n \log n)^2$.

Now, we run the generator k times (spending, an average time of $O(k)$) and apply inverter until all k generated instances are solved. (If the generator can produce unsolvable instances too, they are ignored for the definition of this measure.) The number of trials is a random variable depending on the inverter’s coin flips. Its median value $MT(k)$ we call *multi-median* time of inverting f by A .

This measure is robust in many respects. It commutes with squaring of the inverter’s complexity and, thus, is robust against variation of models. It does not depend much on the $1/2$ probability cut-off used for median. Indeed, increasing k by a factor of c raises $MT(k)$ as much as does tightening the inverter’s failure probability to 2^{-c} .

MT is relevant for both upper and lower bounds. Let $T(x)$ be high for ε fraction of $x \in \{0, 1\}^n$. Then $MT(k)$ is as high for $k = n^3/\varepsilon$. Conversely, let $MT(k)$ be high. Then, with overwhelming probability, $T(x_i)$ is at least as high for some of $n = k^2$ random x_1, \dots, x_n , and $\sum_i \|x_i\| = O(n)$.

3.3 Nice Distributions

So far, we addressed the variance of performance of a randomized algorithm over its variable coin flips for a fixed input, as well as the issue of averaging it over variable input with a given distribution. Now we must address the variance of distributions. In practice, choosing the right distribution is not trivial which is often dismissed by declaring it uniform. Such declarations are confusing, though, since many different distributions deserve the name.

For instance, consider graphs $G = (V, E \in V^2)$, $\|V\| = n$, where n is chosen with probability $\theta(1)/n^2$. For a given n , graphs G are chosen with two distributions, both with a claim to uniformity: μ_1 chooses G with equal probability among all 2^{n^2} graphs; μ_2 first chooses $k = \|E\|$ with uniform probability $1/n^2$ and then G with equal probability among all the $C_{n^2}^k$ candidates. The set $S_n = \{G : k = n^{1.5}\}$ has then μ_2 probability $1/n^2$, while its μ_1 is exponentially small. In fact, all nice distributions can be described as uniform in a reasonable representation. Let me reproduce the argument sketched briefly in [Levin 86].

Let us use set-theoretic representation of integers: $n = \{0, 1, \dots, n-1\}$. A measure μ is an additive real function of *sets* of integers; $\mu(n) = \mu(\{0\}) + \mu(\{1\}) + \dots + \mu(\{n-1\})$ is its monotone *distribution* function. Its *density* $\mu(\{n\})$ is the probability of $\{n\}$ as a singleton, rather than of a set $n = \{0, 1, \dots, n-1\}$. We round the real-valued μ keeping only as many binary digits as needed for constant factor accuracy of probabilities. We call μ *perfectly rounded* if $\mu(x)$ is the shortest binary fraction within $(\mu(x-1), \mu(x+1))$ interval, $\mu(1) = 1/2$, $-\log \mu(\{x\}) = O(\|x\|)$. The last two conditions are just for convenience and can be met simply by mixing μ with some simple distribution.

The first condition can be achieved by rounding. It preserves μ 's polynomial time computability, if present, and keeps $\mu(\{x\})$ greater than a quarter of its previous value. First, round $\mu(x)$ to the shortest binary p that is closer to it than to any other $\mu(y)$ and call these rounded values *points*. Find all *slots*, i.e. closest to p shorter binary fractions of each binary length. Then, for each slot in order of increased lengths, find the point that fills it in the successive roundings until the slot for x is found.

All perfectly rounded μ have a curious property: both $m(x) = \mu(x)/\mu(\{x\})$ and $-\log \mu(\{x\}) = \|m(x)\|$ are always integers, making $\mu(x) = .m(x) \in [1/2, 1]$. So m is quite uniformly distributed: $2k\mu(m^{-1}(k)) \in [1, 2]$ for $k \in m(\mathbb{N})$. It is also computable in polynomial time, as is m^{-1} (by binary search). So, we can use $m(x)$ as an alternative representation for x in which the distribution μ is remarkably uniform.

Simple distributions are not normally general enough. They may be the ultimate source of the information in the instances x of our problems, but the original information r is transformed into x by some process A that may itself be something like a one-way function. We can assume that A is an algorithm with a reasonable time bound, but not that its output distribution is simple. Such distributions are called *samplable*. We will take $LV(P)$ bound as "reasonable." When a tighter precision is needed, it will be just L .

[Impagliazzo, Levin, 90] deal with samplable distributions in a similar manner as with those in the section 3.3, though through a different trick.

Here ends the finished part. After more discussion of samplable distributions, the next part Completeness starts with the following complete owf:

Tiles: unit squares with a letter at each corner; may be joined if the letters match. Expansion: maximal tile-by-tile unique (using given tiles) extension of a partial tiling of a square with marked border. One Way Function: Expand a given top line to the square; output the bottom line and permitted tiles.

a	x	x	c
e	r	r	z
e	r	r	z
n	s	s	z

I then discuss concepts of completeness for random NP problems, for one-way functions, for inverting algorithms, hopeful and hopeless open problems, etc.

References

- [FOCS] *Proceedings of the Annual IEEE Symposium on Foundations of Computer Science.*
- [STOC] *Proceedings of the Annual ACM Symposium on Theory of Computing.*
- [1] S. Banach, A. Tarski. Sur la decomposition des ensembles de points en parties respectivement congruentes. *Fund. Math.* 6, 244-277, 1924.
- [2] Manuel Blum, Silvio Micali. *How to generate cryptographically strong sequences of pseudo-random bits.* SIComp. 13:850-864 (1984).
- [3] M. Blum, S. Goldwasser. An Efficient Probabilistic Encryption Scheme Hiding All Partial Information. *Crypto-1982.*
- [4] Yuri Gurevich. Average Case Complexity. *Internat. Symp. on Information Theory, Proc., IEEE,* 1985.
- [5] Oded Goldreich, Russell Impagliazzo, Leonid A. Levin, Ramarathnam Venkatesan, David Zuckerman. Security Preserving Amplification of Hardness. [FOCS] 1990, pp. 318-326.

- [6] Oded Goldreich, Leonid A. Levin. A Hard-Core Predicate for all One-Way Functions. [STOC] pp. 25-32, 1989.
- [7] Johan Hastad, Russell Impagliazzo, Leonid A. Levin, Michael Luby. A Pseudorandom Generator from any One-way Function. *SICOMP* 28(4):1364-1396, 1999.
- [8] Russell Impagliazzo, Leonid A. Levin. No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random. [FOCS] 1990, pp. 812-821.
- [9] David S. Johnson. The NP-Completeness Column. *J. of Algorithms* 5:284-299, 1984.
- [10] Richard Karp. The probabilistic analysis of some combinatorial search algorithms. *Algorithms and Complexity*. (J.F.Traub, ed.) Academic Press, NY 1976, pp. 1-19.
- [11] Donald E. Knuth. The Art of Computer Programming. Vol. 2. *Seminumerical Algorithms*. Addison-Wesley, 3d ed., 1997. Section 3.5.F. also available on pp. 10, 29-36 of <http://www-cs-faculty.stanford.edu/~knuth/err2-2e.ps.gz>.
- [12] Andrei N.Kolmogorov, Vladimir A. Uspenskii. Algorithms and Randomness. *Theoria Veroyatnostey i ee Primeneniya = Theory of Probability and its Applications*, 3(32):389-412, 1987.
- [13] Leonid A. Levin. One-way functions and pseudorandom generators. *Combinatorica* 7(4):357-363, 1987.
- [14] Leonid A. Levin. The Tale of One-Way Functions. 11/1/1999. Workshop on Probabilistic Analysis of Algorithms for Hard Problems. DIMACS, Rutgers University. Also a related talk at European Summer School "Pavages", Branville France, 5/15-19/00.
- [15] Leonid A. Levin. Randomness and Non-determinism. *J. Symb. Logic*, 58(3):1102-1103, 1993. A less technical version: *International Congress of Mathematicians*, Zurich, August 1994. *Proceedings* (Invited Addresses), pp. 1418-1419. Birkhauser Verlag, 1995.
- [16] Ming Li, Paul M.B. Vitányi. Introduction to Kolmogorov Complexity and its Applications. Springer Verlag, New York, 1993.
- [17] Michael Rabin. *Digitalized Signatures as Intractable as Factorization*. MIT/LCS/TR-212, 1979.
- [18] Ronald L. Rivest, Adi Shamir, Leonard M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *CACM* 21(2):120-126, 1978.
- [19] Adi Shamir. Factoring Numbers in $O(\log n)$ Arithmetic Steps. *Information Processing Letters* 8(1):28-31, 1979.
- [20] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIComp*. 26(5):1484-1509 (1997).
- [21] Peter W. Shor. "Re: When will quantum computers become practical?" Moderated usenet group *sci.physics.research* article FrsHL0.Ilr@research.att.com, 3/22/2000. The thread can be retrieved by search at the archive http://www.deja.com/usenet_home.epl or accessed directly as <http://www.deja.com/viewthread.xp?AN=599065748>.
- [22] Ramarathnam Venkatesan, Leonid A. Levin. Random Instances of a Graph Coloring Problem are Hard. *STOC*, 1988.
- [23] Andrew C. Yao. Theory and applications of trapdoor functions. [FOCS], 1982, pp. 80-91.
- [24] Alexander K. Zvonkin, Leonid A. Levin. The Complexity of finite objects and the Algorithmic Concepts of Information and Randomness. *UMN = Russian Math. Surveys* 25(6):83-124, 1970.