STOC, June 2021

How do we succeed in tasks like proving Fermat's Theorem or predicting the Higgs boson?

Leonid A. Levin
Boston University

This talk aims at attracting attention to the following open problem:

Can every algorithms finding, say, 3-coloring be sped-up 10 times on an infinite set of graphs?

Or, there is a 'perfect' one that cannot be?

(Note: no speed-up above O(1) factor exists.)

But first some history.

Russian controversies of the 50s.

Sergey Yablonsky: Resolved ! I proved (to appear in 1959) the exponential complexity of some such (search) problem.

Kolmogorov: Not at all **!**
Such arguments, addressing only "customary" algorithms, fall short for any such claims.

We cannot even prove the universally believed quadratic complexity of multiplication !
Try answering that using an adequate (graph-based) model of Time Complexity [he defined].

Karatsuba, Toom (early 60s): In fact, multiplication has nearly linear complexity.

Kolmogorov (and independently Solomonoff): Universal Algorithm allows optimal definition of informational complexity, randomness, etc.

Levin: same arguments give optimal algorithm for Tiling, and thus for every search problem.

Kolmogorov: the optimality is a bit abstract, but do publish the completeness of Tiling !

Levin: I will if I can reduce it to some popular problems.
(Follow years of failure with isomorphism of graphs, small circuits for boolean tables, etc.)

Cook, Karp, David Johnson: 3-SAT reduces to great many important combinatorics problems.

[M.Dekhtiar 1969] (and independently [Baker, Gill, Solovay]): Under some oracles, inverting simple functions has exponential complexity.

And Kolmogorov had some curious questions. One (still open): Are there polynomial time algorithms that have no **linear**-sized circuits ?

Another one: would not a search for fast short (with $+O(1)$ slack for robustness) programs transforming x into y be a better focus than Tiling to see (in today's terms) if P$\neq$NP ?

[He felt Tiling is too generic (universal), some others – too narrow (e.g., factoring), and the best focus often is neither.]

This task is involved in another great set of issues: Inductive Inference via **Occam Razor**.

(Attributed to Einstein: Conjectures should be chosen as simple as possible, but no simpler.)

Solomonoff: Likelihoods of extrapolations (matching known data) drop exponentially with length of their shortest descriptions p.

Those short programs p run about as fast as the process that had generated the data.
But finding such short fast p may be hard.

There were many subtleties there. Most have been clarified, **except** for time to search for p.

Yet, this is an inversion task, thus
the optimal search algorithm applies!

Some discussion: L.Levin. Universal Heuristics: How Do Humans Solve Unsolvable Problems?

In: LNCS v. 7070; also posted on page 5 in
https://arxiv.org/abs/cs/0503039

Now. The optimal search algorithm ignores constant factors. What about them ?

Chorus: They must be huge, huge, huge !

Wait a minute ! But how our brains (evolved on the jumping in trees, not on writing math papers) could, say, prove Fermat's Theorem ?!

Actually:

Can every algorithms for complete search problems be sped-up **10 times** on an infinite set?

Or, there is one so good that it cannot be sped-up 10 times even on a subset !?

(Of course, the definition of time must care to exclude false speed-ups, e.g., those ignoring the alphabet size, or skipping the prescribed end verification of the input/output relation.)

But what are the constant factors issues?

Time-refine complexity to turn it computable: $\mathbf{Kt}(w|x) = \min\{\|p\| + \log T\colon U^T(p,x){=}w\}$ for universal U run in time T, prefixless on p.

Optimal Inverter **OI**: searches for solutions $w \in f^{-1}(x)$ in order of increasing complexity $\mathbf{Kt}(w|x)$. (**Not (!)** of length $\|w\|$, as e.g., shorter proofs may be much harder to find!)

In time 2^k, **OI** lists all w with $\mathbf{Kt}(w|x) < k$.

[And **OI** allows hardness, $\min_w \mathbf{Kt}(w|x)$, apply to specific instances x, not just to whole families. Say, how hard is Fermat's theorem, not theorems with short proofs in general. A tighter notion !]

CATCH: Each redundant bit that U requires of p **doubles** the time. Need **VERY** "pure" U.

Do our brains have one built-in **?** We do seem to have much agreement on what is "neat".