Idea's behind theoretical computer science!

Sachin Kumar
University of Waterloo, Faculty of Mathematics

Every description of this sort is necessarily biased and reflects personal tastes of me. With this disclaimer, here are some concepts and ideas that we consider as milestones in the development of Theoretical Computer Science.

- The focus of the field changed from the study of computability in finite (but unbounded) time, to the more practical (but mathematically subtle) study of efficient computation. The fundamental notion of NP-completeness was formulated, and its near-universal impact was gradually understood. Long term goals, such as the P vs. NP question, were set.

- The theory of algorithms was developed, with the fundamental focus on asymptotic and worst-case analysis. Numerous techniques, of increasing mathematical sophistication, were invented to efficiently solve major computational problems.

- A variety of computational models, designed to explain and sometimes anticipate existing computer systems, were developed and studied. Among them are parallel and distributed models, asynchronous and fault-tolerant computation, on-line algorithms and competitive analysis.

- Randomness was introduced as a key tool and resource. This revolutionized the theory of algorithms. In many cases, probabilistic algorithms and protocols can achieve goals which are impossible deterministically. In other cases they enable much more efficient solutions than deterministic ones. Following this, a series of derandomization techniques developed to convert in general cases probabilistic algorithms to deterministic ones.

- The emergence of the notion of one-way functions (which are "easy to compute" but "hard to invert"), together with the use of randomness, has lead to the development of modern cryptography. This theory has provided essential ideas for secure communication and computation over networks, and has played a crucial role in the evolution of electronic commerce.

- The notion of interactive proof systems, in which knowledge is gained through interaction with untrusted parties, has been pivotal to both computational complexity theory and cryptography. Probabilistic proof systems, with their many variants — zero knowledge, Arthur-Merlin, multi-prover, and probabilistically checkable proofs (PCPs), have enriched to a tremendous extent our understanding of many basic computational problems, including ones which superficially have nothing to do with randomness or interaction. As a prime example, our understanding of PCPs has led to enormous strides in our knowledge of the difficulty of finding approximate solutions to many natural optimization problems.

- The theory of pseudo-randomness has revealed intimate connections between computational difficulty and the task of derandomizing algorithms, and has brought us closer to understanding the power of randomness in various computational settings.

- Complexity theory, attempting to classify problems according to their computational difficulty, has integrated many of the ideas above, and has become much more mature and intricate. There were many nontrivial successes in proving lower bounds on restricted models of computation. The failure to do so for general models (towards the P vs. NP question) has motivated the introspective field of proof complexity, which tries to quantify the resource requirements of proofs and the intrinsic difficulty of proving various mathematical statements.

Derandomization and Pseudo-Randomness

The P = BPP question and related questions about the power of randomness in computation have given rise to the notion of pseudo-random generator, a deterministic process that in some sense looks random to the computational model at hand. The fundamental insight here is that a hard function for that computational model can (sometimes) be efficiently converted into a pseudo-random number generator for the same model.

This insight, that hardness can be turned into randomness, has led to some surprising and deep connections between the complexity of randomness, cryptography, circuit complexity and combinatorics. And once we have such a generator at hand, it results in a derandomization procedure for all probabilistic algorithms in that model - simply try out all possible seeds of the generator (which are much fewer than all possible random strings the algorithm used).

The techniques for implementing this paradigm have improved tremendously in the two decades of its existence. For BPP (which is the class of algorithmic problems solvable by probabilistic polynomial time algorithms), such non-trivial derandomization is possible already if BPP is different from the class of problems solvable by deterministic exponential algorithms, EXP. Moreover, if EXP contains functions requiring exponential circuit size, then BPP = P (namely, randomness can be always eliminated from polynomial time computations). This brings up the problem of unconditionally separating BPP from EXP as a natural "next step", which may be feasible even with the current technology. The extensive technical progress of the last couple of years on different ways of constructing pseudo-random generators still has to be fully understood, simplified and generalized to realize its potential impact on this and related problems.

The analogous techniques developed for fooling probabilistic logarithmic space computations (again, some only very recently) seem to put us quite close to deciding unconditionally that randomness is useless in that context as well, e.g that RL = L (L stands for "logarithmic space", and R stands for "randomized"). But "en-route", there are many seemingly simpler problems that still remain challenges like derandomizing constant-width probabilistic computations or even combinatorial rectangles. Interestingly, the connection between pseudo-randomness and lower bounds is not as explicit in space bounded models as in time bounded models, and has yet to be clarified.

Finally, it seems that we have precious few ways of generating randomness from hardness. It will be extremely interesting to find drastically different pseudo-random generators, (even if these will not improve current results), or find that current constructions (like Nisan-Wigderson generators) are somehow universal.

Another line of research in the area is more combinatorial in nature and is aimed at improving sources of imperfect randomness so that after this improvement the outcome can be used for derandomizing various classes of algorithms. Combinatorial constructions like expanders , extractors and condensers were identified for the purpose, and the ultimate goal here is to be able to build explicit constructions of such objects that match the performance of randomly chosen objects. There has been a constant progress in this direction over several last years, but much still remains to be done.

Contributions

As we already mentioned in Introduction, this section can be viewed as annotations to some of the papers linked or referenced at our paper page, and for a more complete picture the latter should be consulted . We

confine ourselves here only to the two topics somewhat elucidated in the previous section. But many good results were proved in other branches of Computational Complexity, as well as in Combinatorics, Algebra and Topology, and more information about them can be gained from the abstracts of the respective papers.

Proof complexity

One of the long-standing open problems in propositional proof complexity was to decide whether the weak Pigeonhole principle is hard for Resolution or not (here "weak" refers to the fact that the number of pigeons is much larger than the number of holes, potentially infinite). This problem was completely solved in the papers Resolution Lower Bounds for the Weak Pigeonhole Principle by R. Raz and Improved Resolution Lower Bounds for the Weak Pigeonhole Principle by A. A. Razborov (very recently this result was extended to the functional version of the Pigeonhole principle, see Resolution Lower Bounds for the Weak Functional Pigeonhole Principle).

In the paper Resolution is Not Automatizable Unless W[P] is Tractable, M. V. Alekhnovich and A. A. Razborov studied the question whether Resolution is automatizable or not. Under a rather strong hardness assumption from the theory of parameterized complexity they were able to completely answer this question in negative.

Another important contribution to understanding the power of Resolution was made by Elli, Ben-Sasson and Nicola Galesi in Space Complexity of Random Formulae in Resolution . The behaviour of a proof system on random tautologies is traditionally considered as a good indicator of its strength. Ben-Sasson and Galesi were able to show that Resolution performs rather badly on such formulas in terms of space consumed by the proof (a similar result for the ordinary bit size measure was known for a long time).

Among the papers devoted to stronger proof systems, we can mention Monotone simulations of non-monotone propositional proof by Albert Atserias, Nicola Galesi and Pavel Pudlak that contains the following (rather surprising) result. An interesting fragment of the Frege proof system called monotone sequent calculus was introduced several years ago and was long believed to be substantially weaker than (non-monotone) Frege. This paper showed that, contrary to this belief, the monotone sequent calculus can in fact efficiently simulate every Frege proof.

Derandomization and pseudo-randomness

The paper In Search of an Easy Witness: Exponential Time vs. Probabilistic Polynomial Time by R. Impagliazzo , V. Kabanets and A. Wigderson (awarded the Best Paper Award at the 16th IEEE Conference on Computational Complexity) makes very important contributions to the goal of extracting randomness from hardness described above. It proves (in a sense, and in an intermediate form) the universality of this approach: no derandomization of the complexity class MA (that stands for "Arthur-Merlin games") is possible unless there are hard functions in NEXPTIME. As another application of this technique, they showed a number of the so-called downward closure results (that are very rare in Complexity Theory). The paper Entropy Waves, The Zig-Zag Graph Product, and New Constant-Degree Expanders and Extractors by O. Reingold, S. Vadhan and A. Wigderson introduces one elegant combinatorial construction, zig-zag product of graphs. Iterating this construction, one in particular gets simple explicit expander graphs of every size, starting from one constant-size expander. The subsequent paper Semi-direct Products in Groups and Zig-Zag products in graphs: Connections and Applications by N. Alon , A. Lubotsky and A. Wigderson reveals deep connections between this zig-zag product and basic group-theoretical primitives. As an important application, they give an example showing that the expansion property of the Cayley graph of a group may be not invariant under the choice of generators.

Extracting Randomness via Repeated Condensing by O. Reingold, R. Saltiel and A. Wigderson constructs efficient explicit condensers and extractors that give significant qualitative improvements over previously known constructions for sources of arbitrary min-entropy.

References

[GW96] O. Goldreich, A. Wigderson. Theory of Computation: A Scientific Perspective, an essay.

[Raz00] A. Razborov. Theoretical Computer Science: a mathematician's look (Russian), full version of an essay written for Computerra.

[GJ79] M. R. Garey, D. S. Johnson. Computers and Intractability. A guide to the theory of NP-completeness, W. H. Freeman, 1979.

[Weg87] I. Wegener. The Complexity of Boolean Functions. John Wiley & Sons, 1987.

[Han90] Handbook of Theoretical Computer Science, vol. A (Algorithms and Complexity). Elsevier Science Publishers B.V. and The MIT Press, 1990.

[Pap94] C. Papadimitriou. Computational Complexity, Addison-Wesley, 1994.

[Bus86] S. R. Buss. Bounded arithmetic, Bibliopolis, Napoli, 1986.

[Kra95] J. Krajicek. Bounded arithmetic, propositional logic and complexity theory, Cambridge University Press, 1995.

[Urq95] A. Urquhart. The complexity of propositional proofs, Bulletin of Symbolic Logic, 1:425-467, 1995.

[Raz96] A. Razborov. Lower bounds for propositional proofs and independence results in Bounded Arithmetic, in Proceedings of the 23rd ICALP, Lecture Notes in Computer Science vol. 1099, 1996, 48-62.

[BP98] P. Beame, T. Pitassi. Propositional Proof Complexity: Past, Present, and Future, Bulletin of the European Association for Theoretical Computer Science, 65:66-89, June 1998. The Computational Complexity Column (ed. E. Allender).

[Pud98] P. Pudlak. The lengths of proofs, Handbook of Proof Theory, pages 547-637. Elsevier, 1998.

[Raz99] A. Razborov. Review of the book Proof Complexity and Feasible Arithmetics, Journal of Symbolic Logic.

[ABRW00] M. Alekhnovich, E. Ben-Sasson, A. Razborov, A. Wigderson. Pseudorandom generators in propositional complexity, Proceedings of the 41st IEEE FOCS, 43-53.

[MTCS] Some milestones in the evolution of Theoretical Computer Science.

[DPR] Derandomization and Pseudo-Randomness.