Feasible Proofs and Computations: Partnership and Fusion

Alexander A. Razborov*
Institute for Advanced Study
School of Mathematics
Princeton, NJ 08540, USA
razborov@math.ias.edu

Abstract

A computation or a proof is called feasible if it obeys prescribed bounds on the resources consumed during its execution. It turns out that when restricted to this world of feasibility, proofs and computations become extremely tightly interrelated, sometimes even indistinguishable. Moreover, many of these rich relations, underlying concepts, techniques etc. look very different from their "classical" counterparts, or simply do not have any. This talk is intended as a very informal and popular (highly biased as well) attempt to illustrate these fascinating connections by several related developments in the modern complexity theory.

1. Introduction

Proofs and computations are among the most basic concepts pertinent to virtually any intellectual human activity. Both have been central to the development of mathematics for several millenniums. The effort to study these concepts themselves in a rigorous, metamathematical way initiated in the 20th century led to flourishing of the mathematical logic and derived disciplines like those which are the focus of attention of both conferences gathering here.

The relation between proofs and computation in mathematics never was easy. In particular, the debate as to what makes a worthy mathematical result – a deductive inference from accepted axioms, possibly shockingly nonconstructive or a practical procedure leading to the desired results but possibly lacking a rigorous justification – was sometimes hot (at the time of crises), sometimes lukewarm, but ever-present it was. And as we all know well, precise formalizations of both these fundamental concepts given by great logicians of the last century at least put this debate onto a solid ground, even if this did not seem to extinguish it completely. Since that time we at least more or less universally agree on *what* is a proof and *what* is a computation. The connections between them are extremely diverse, rich and mutually beneficial: we study computations with formal proofs, try to write programs for computer-aided theorem proving, use formal proofs for the program verification etc. Still, it appears (at least to the speaker) that no matter how we are playing with computations and proofs, they keep their unique identities, and at every particular moment it is utterly clear to which of the two realms the object of interest belongs. In our communities this difference is often articulated as the difference between the syntax and the semantics.

Most of the above are, of course, just common places for this audience (and we will go over them very quickly in the actual talk). But this makes a necessary background for the main point we will try to make. Namely, when we restrict our attention to *feasible* proofs and computations (which are most often defined as polynomially bounded in terms of length or time), then this clear classical (that is, in the absence of such restrictions) picture all of a sudden becomes rather different, and in general more intricate and saturated. Some of the relations between classical proofs and computations gain in importance, whereas some become almost obsolete. New unexpected and beautiful connections emerge on the conceptual level, as well as on the level of proof techniques. Finally, even the separate identities of proofs and computations are compromised by the important discovery of "interactive proofs" that can be (and are) thought of as both proofs and computations, subjectively and interchangeably[1]. And, to make the story even

* On leave from Steklov Mathematical Institute, Moscow, Russia. Supported by the State of New Jersey, The Bell Companies Fellowship, The James D. Wolfensohn Fund, and The Ellentuck Fund.

1 Needless to say, these latter creatures completely decline to behave like "straight-line programs" when viewed as a computation or as a Hilbert-style inference when viewed as a proof! Instead they pay quite a fair share in the analysis of "normal" proofs and "normal" computations.

more interesting, all these trends are interweaving and reenforcing each other.

Inherent reasons for these *qualitative* changes in behaviour resulting from a *quantitative* change in the framework (that is, when we place some *numerical* bounds on the resources) are far from being understood, and this talk will not pretend to offer any such explanation beyond one simple observation. Our main purpose will simply be to illustrate the wonderful interplay between feasible proofs and feasible computations by a few important examples. These will be borrowed from several rather different sub-areas of the modern complexity theory. As a result, this talk should not be considered as a survey in none of them, and it will, out of necessity, be lacking precise definitions and statements (Section 3 below contains some suggested literature for further study of the subjects we will only superficially touch in the talk). Moreover, the selection of topics from each area will be heavily biased by our main goal: illustrate various intricate connections existing between proofs and computations when both are restricted to the world of feasible objects.

2. Plan of the talk

We will try to do as much of the following as time permits (although, it does not seem realistic to cover all these issues). The arrangement of topics is somewhat arbitrary, although we will try to stick to this general principle: begin with concepts that still look similar to their "classical" counterparts, gradually moving to the regions where these similarities fade away.

It is (or at least should be) already well-known these days what is a "feasible" computation: this is a computation that obeys a prescribed bound on computational resources like time, space etc. It is intuitively less clear what is a feasible proof. It is natural to assume that a feasible proof should not involve objects that are unfeasibly (= exponentially) large, but is this sufficient? We will begin with one canonical example (Fermat's Little Theorem) illustrating that the right answer should be "no", and that all objects involved in a feasible *proof* must be also feasibly *computable*.

We then move on to *Bounded Arithmetic* which is a generic name for a series of first-order or second-order theories capturing this notion of a feasible proof, and cover a few (relatively simple) *witnessing theorems*. The question whether the hierarchy of these theories is proper is the central open problem in this area, which is reminiscent of just the same situation in the computational world.

From this point on, almost everything in this talk will be about propositional (as opposed to first-order) logic, and we will be mostly concerned with two fundamental (and dual to each other) questions:

- How to prove efficiently that a propositional formula ϕ is a tautology?
- How to prove efficiently that a propositional formula ϕ is satisfiable?

We compare these questions with their first-order counterparts, and note a drastic difference in their behaviour.

Then we will address the "textbook" approach to proving that a propositional formula ϕ is a tautology, which consists in deriving ϕ in a Hilbert-style (or Gentzen-style) propositional calculus. We will be interested in the complexity (most often measured by the bit size) of such *propositional proofs*, and we will indicate numerous connections existing between the complexity of propositional proofs and circuit complexity. We will give a couple of lower bound results illustrating the fruitfulness of these connections for both areas. In particular, we will spend some time on the so-called *Feasible Interpolation Theorem*, as well as on the results limiting its use for stronger proof systems that are based upon complexity hardness assumptions.

Next we will move on to the question of feasible provability of a non-uniform version of $\mathbf{P} \neq \mathbf{NP}$. In particular, we will address one approach to this question based upon an adaptation of the notion of a pseudorandom function generator to proof complexity. Specifically, we will talk about the conjecture stating that for the so-called *Nisan-Wigderson generators* their computational hardness always implies hardness for the corresponding proof system, and survey known results for weak proof systems supporting this conjecture.

In everything we have encountered so far, proofs and computations often came very close to each other, but still they did not blend together. The Pandora's box was open in one of the most influential mathematical articles of the last century, [16] which is even entitled suggestively "The complexity of theorem *proving procedures*". Namely, the definition of the fundamental complexity class **NP** given in that paper is inherently dual, and can be viewed both in terms of non-deterministic *computations* and *proofs* of membership. We will reflect a little bit on this duality.

After the news of the marriage between proofs and computations spread around, and this idea soon became one of the main paradigms of the modern complexity theory, it was only a matter of time before more offsprings would be conceived, and the most fruitful notion born in this way was that of *interactive proofs*. The prover no longer prepares a proof on a sheet of paper in the silence of her office and then submits it to a journal for verification, but rather interacts with the verifier trying to convince him in the validity of her results in the "good common sense". Remarkably, both are making a non-trivial *computational* effort during this interaction. One of the most unexpected results of the complexity theory states, in a weaker form, that one can efficiently

prove in this way that a propositional formula ϕ is a tautology, something we strongly believe no "ordinary" (say, strictly propositional) proof system will ever achieve! Unfortunately, even a sketch of this remarkable result is way too technical to fit into this lecture, but we will at least try to illustrate the power of interactive proofs using (more or less standard) example of GRAPH NON-ISOMORPHISM.

So far we concentrated on our first task itemized above, and there seems to be a very good reason for this: it is very easy to prove that ϕ is satisfiable simply be exhibiting a satisfying assignment (it is an entirely different story, of course, how to *find* such an assignment). This trivial proof is easily checkable, and it is feasible in any sense we have seen so far. It, however, turns out, that in the realm of interactive proofs we can employ much more severe notion of feasibility than just "poly-time verifiable" and require a proof to be presented in such a format that its validity can be verified by checking a constant number of (randomly chosen) places in the proof. This is the celebrated *PCP*[2] *theorem* which is in fact extremely tightly connected with interactive proof systems.

In another unexpected turn, the PCP theorem and its many variants became the major tool in analyzing the complexity of finding approximate solutions to combinatorial optimization problems. This is one of the most practical areas of Theoretical Computer Science that, prior to the PCP breakthrough, did not have anything to do with proofs, and in general was not too successful at solving its major problems. We will sample several typical applications in this area.

Next, we we will link this latter topic with propositional proof complexity by considering the optimization problem of finding the best proof for a given tautology in a given propositional proof system. This naturally leads to the important concept of *automatizability* of such systems. We will mention one tight connection between Feasible Interpolation and automatizability, and give one example of a proof system for which they (apparently) differ.

Finally, we will return to the question of feasible provability of the $\mathbf{P} \neq \mathbf{NP}$ question previously considered by us in the context of the propositional proof complexity. It was also studied in the framework of so-called *Natural Proofs*, where "proofs" are defined this time by a set of properties (of computational nature) shared by all known arguments. We will show the main theorem of this minitheory, which is exactly the result of a kind we are still missing in the propositional proof complexity.

2 for Probabilistically Checkable Proofs

3. Historical remarks and literature for further reading

Some of the topics above (especially those from the first, "classical" part) appeared in my ICALP lecture 8 years ago in much more elaborated and systematic way, and the extended abstract of that talk [38] contains some additional literature.

Bounded Arithmetic was apparently considered for the first time by Parikh in [31], and was studied by Paris and Wilkie in the 1980s (see e.g. [32]). Systematically this subject was treated in Buss's book [15] which still remains a very good source for a quick introduction to it. Other choices include the monograph [25] and the last section of [23].

The first non-trivial lower bound in propositional proof complexity is due to Tseitin [45] which well preceded the general theory developed by Cook and Reckhow in [17]. Feasible Interpolation Theorem evolved from a sequence of papers [24, 37, 13, 26], and its elegant proof sketched in this talk is due to Pudlák [33]. The limitations of Feasible Interpolation for stronger proof systems were established in [29, 14, 12].

The proper formalization of the non-uniform version of $\mathbf{P} \neq \mathbf{NP}$ was proposed by Razborov in [36], and it was also stressed there that the proofs of (apparently) all known partial results toward this goal become feasible in this framework. The approach based upon pseudo-random generators was proposed by Alekhnovich, Ben-Sasson, Razborov, Wigderson [1] and Krajíček [27]; the first paper also contained specific suggestions as to the use of Nisan-Wigderson generators. Partial results in that direction were proved in [1, 3, 35, 40, 28, 39], and the introduction in [39] also contains an extended summary of the whole approach, and of the speaker's view of the subject.

There are several good surveys on propositional proof complexity as a whole, see e.g. [46, 25, 38, 11, 34].

Interactive proofs were introduced by Goldwasser, Micali, Rackoff [21] and Babai [8], and the protocol for the GRAPH NON-ISOMORPHISM was given by Goldwasser, Micali, Wigderson [22]. The result that all of PSPACE has interactive proofs (which is much stronger than its partial case mentioned in the talk) is due to Lund, Fortnow, Karloff, Nisan [30] and Shamir [44].

The original form of the PCP theorem evolved from [10, 18], and was proved in the papers by Arora, Safra [6] and Arora, Lund, Motwani, Sudan, Szegedy [5]. The connection to the complexity of approximation was understood already in those early papers (in fact, it was one of their primary motivations).

There are many good surveys on interactive proofs, PCP and hardness of approximation, see e.g. [9, 19, 4]. We would also like to mention the books

[20, 7], as well as the unique on-line compendium `http://www.nada.kth.se/~viggo/problemlist/compendium.html` compiling known results on the complexity of approximation.

The concept of automatizability was introduced by Bonet, Pitassi, Raz [14], and the remark that automatizability implies feasible interpolation is due to Impagliazzo (unpublished). Alekhnovich, Razborov [2] proved (modulo strong complexity assumptions) that Resolution (which does allow Feasible Interpolation) is not automatizable.

Natural proofs were introduced by Razborov, Rudich [41], and [42] gave some unexpected applications in quite a different area. This theory was further developed by Rudich [43].

References

[1] M. Alekhnovich, E. Ben-Sasson, A. Razborov, and A. Wigderson. Pseudorandom generators in propositional complexity. In *Proceedings of the* 41*st IEEE FOCS*, pages 43–53, 2000. Journal version to appear in *SIAM Journal on Computing*.

[2] M. Alekhnovich and A. Razborov. Resolution is not automatizable unless $W[P]$ is tractable. In *Proceedings of the* 42*nd IEEE Symposium on Foundations of Computer Science*, pages 210–219, 2001.

[3] M. Alekhnovich and A. Razborov. Lower bounds for the polynomial calculus: non-binomial case. *Proceedings of the Steklov Institute of Mathematics*, 242:18–35, 2003.

[4] S. Arora. The approximability of NP-hard problems. In *Proceedings of the* 30*th ACM Symposium on the Theory of Computing*, pages 337–348, 1998.

[5] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proceedings of the* 33*rd IEEE Symposium on Foundations of Computer Science*, pages 13–22, 1992.

[6] S. Arora and M. Safra. Probabilistic checking of proofs: a new characterization of np. *Journal of the ACM*, 45(1):70–122, 1998.

[7] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation. Combinatorial optimization problems and their approximability properties*. Springer-Verlag, 1999.

[8] L. Babai. Trading group theory for randomness. In *Proceedings of the* 17*th ACM Symposium on the Theory of Computing*, pages 421–429, 1985.

[9] L. Babai. Transparent proofs and limits to approximations. In *Proceedings of the First European Congress of Mathematics, Vol. I*, pages 31–91. Birkhauser, 1994.

[10] L. Babai, L. Fortnow, C. Lund, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the* 23*rd ACM Symposium on the Theory of Computing*, pages 21–31, 1991.

[11] P. Beame and T. Pitassi. Propositional proof complexity: Past, present and future. Technical Report TR98-067, Electronic Colloquium on Computational Complexity, 1998. Available at ftp://ftp.eccc.uni-trier.de/pub/eccc/reports/1998/TR98-067/index.html.

[12] M. Bonet, C. Domingo, R. Gavaldá, A. Maciel, and T. Pitassi. Non-automatizability of bounded-depth Frege proofs. In *Proceedings of the* 14*th Annual IEEE Conference on Computational Complexity*, pages 15–23, 1999.

[13] M. Bonet, T. Pitassi, and R. Raz. Lower bounds for cutting planes proofs with small coefficients. *Journal of Symbolic Logic*, 62(3):708–728, 1997.

[14] M. Bonet, T. Pitassi, and R. Raz. On interpolation and automatization for Frege systems. *SIAM Journal on Computing*, 29(6):1939–1967, 2000.

[15] S. R. Buss. *Bounded Arithmetic*. Bibliopolis, Napoli, 1986.

[16] S. A. Cook. The complexity of theorem proving procedures. In *Proceedings of the* 3*rd Annual ACM Symposium on the Theory of Computing*, pages 151–158, 1971.

[17] S. A. Cook and A. R. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.

[18] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996.

[19] O. Goldreich. Probabilistic proof systems. In *Proceedings of the International Congress of Mathematicians (Zurich, 1994)*, pages 1395–1406. Birkhauser, 1995.

[20] O. Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness, Algorithms and Combinatorics, Vol. 17*. Springer-Verlag, 1998.

[21] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1985.

[22] S. Goldwasser, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity, and a methodology of cryptographic protocol design. In *Proceedings of the* 27*th IEEE Symposium on Foundations of Computer Science*, pages 39–48, 1986.

[23] P. Hájek and P. Pudlák. *Metamathematics of First-Order Arithmetic*. Springer-Verlag, 1993.

[24] J. Krajíček. Lower bounds to the size of constant-depth propositional proofs. *Journal of Symbolic Logic*, 59(1):73–86, 1994.

[25] J. Krajíček. *Bounded arithmetic, propositional logic and complexity theory*. Cambridge University Press, 1995.

[26] J. Krajíček. Interpolation theorems, lower bounds for proof systems and independence results for bounded arithmetic. *Journal of Symbolic Logic*, 62(2):457–486, 1997.

[27] J. Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170(1-3):123–140, 2001.

[28] J. Krajíček. Dual weak pigeonhole principle, pseudo-surjective functions, and provability of circuit lower bounds. *Journal of Symbolic Logic*, 69(1):265–286, 2004.

[29] J. Krajíček and P. Pudlák. Some consequences of cryptographical conjectures for S^1_2 and EF. *Information and Computation*, 142:82–94, 1998.

[30] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

[31] R. J. Parikh. Existence and feasibility in arithmetic. *Journal of Symbolic Logic*, 36:494–508, 1971.

[32] J. Paris and A. Wilkie. Counting problems in bounded arithmetic. In *Methods in Mathematical Logic, Lecture Notes in Mathematics 1130*, pages 317–340. Springer-Verlag, 1985.

[33] P. Pudlák. Lower bounds for resolution and cutting planes proofs and monotone computations. *Journal of Symbolic Logic*, 62(3):981–998, 1997.

[34] P. Pudlák. The lengths of proofs. In S. Buss, editor, *Handbook of Proof Theory*, pages 547–637. Elsevier, 1998.

[35] R. Raz. Resolution lower bounds for the weak pigeonhole principle. *Journal of the ACM*, 51(2):115–138, 2004.

[36] A. Razborov. Bounded Arithmetic and lower bounds in Boolean complexity. In P. Clote and J. Remmel, editors, *Feasible Mathematics II. Progress in Computer Science and Applied Logic, vol.* 13, pages 344–386. Birkhaüser, 1995.

[37] A. Razborov. Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic. *Izvestiya of the RAN*, 59(1):201–224, 1995.

[38] A. Razborov. Lower bounds for propositional proofs and independence results in Bounded Arithmetic. In F. M. auf der Heide and B. Monien, editors, *Proceedings of the* 23*rd ICALP, Lecture Notes in Computer Science*, 1099, pages 48–62, New York/Berlin, 1996. Springer-Verlag.

[39] A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. Manuscript available at http://www.genesis.mi.ras.ru/˜razborov, 2002.

[40] A. Razborov. Resolution lower bounds for perfect matching principles. In *Proceedings of the* 17*th IEEE Conference on Computational Complexity*, pages 29–38, 2002.

[41] A. Razborov and S. Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

[42] K. Regan, D. Sivakumar, and J. Cai. Pseudorandom generators, measure theory, and natural proofs. In *Proceedings of the* 36*th IEEE Symposium on Foundations of Computer Science*, pages 26–35, 1995.

[43] S. Rudich. Super-bits, demi-bits, and NP/qpoly-natural proofs. In *Proceedings of the International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM 97), Lecture Notes in Computer Science*, 1269, pages 85–93, New York/Berlin, 1997. Springer-Verlag.

[44] A. Shamir. $IP = PSPACE$. *Journal of the ACM*, 39(4):869–877, 1992.

[45] G. C. Tseitin. On the complexity of derivations in propositional calculus. In *Studies in constructive mathematics and mathematical logic, Part II*. Consultants Bureau, New-York-London, 1968.

[46] A. Urquhart. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 1:425–467, 1995.