Recent Advances in Meta-Complexity

Rahul Santhanam

(University of Oxford)

Plan of the Talk

- Metamathematics
- Learning
- Cryptography
- Complexity Lower Bounds

Plan of the Talk

- *Metamathematics*
- Learning
- Cryptography
- Complexity Lower Bounds

The Circuit Complexity Approach to P vs NP

- The P vs NP problem can be approached combinatorially through the study of *Boolean circuit complexity*
- Well-known: If L is a language in P, then $L_n = L \cap \{0,1\}^n$ has Boolean circuits of size poly(n)
- Therefore, to show NP ≠ P it suffices to show that there is a problem in NP that does not have polynomial-size circuits
- The circuit complexity approach aims to make progress by showing lower bounds in NP for restricted circuit classes

Success and Slowdown

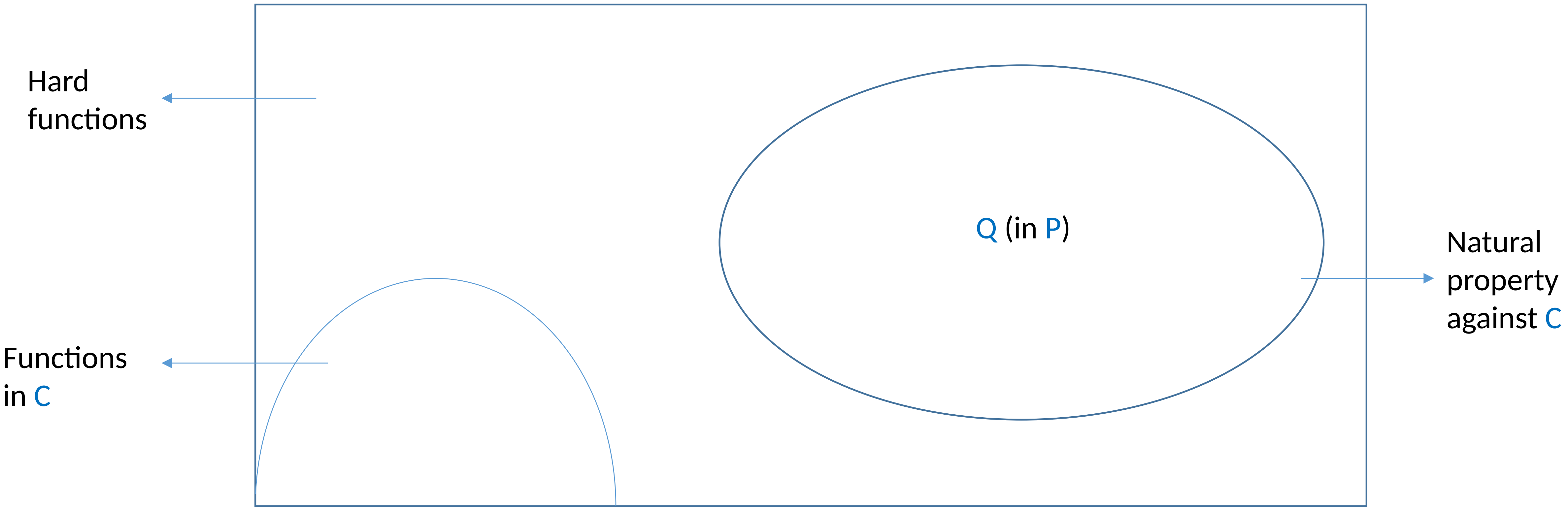
- Many circuit lower bounds shown in the 1980s for interesting circuit models
  - Constant-depth circuits [A83, FSS83, Y85, H86]
  - Monotone circuits [R85]
  - Constant-depth circuits with Mod p gates [R87, S87]
- However, progress ground to a halt in the 1990s and we still don't know if NP has polynomial-size constant-depth circuits with Mod 6 gates
- Is there a fundamental reason for this?

Natural Proofs

Given a circuit class C, a natural proof against C is a property Q of Boolean functions (represented by their truth tables of size N) such that:

- Constructivity: Q in P
- Usefulness: Q(F) = 1 => F not in C
- Density: At least a $1/N^{O(1)}$ fraction of Boolean functions F satisfy Q

Natural Proofs

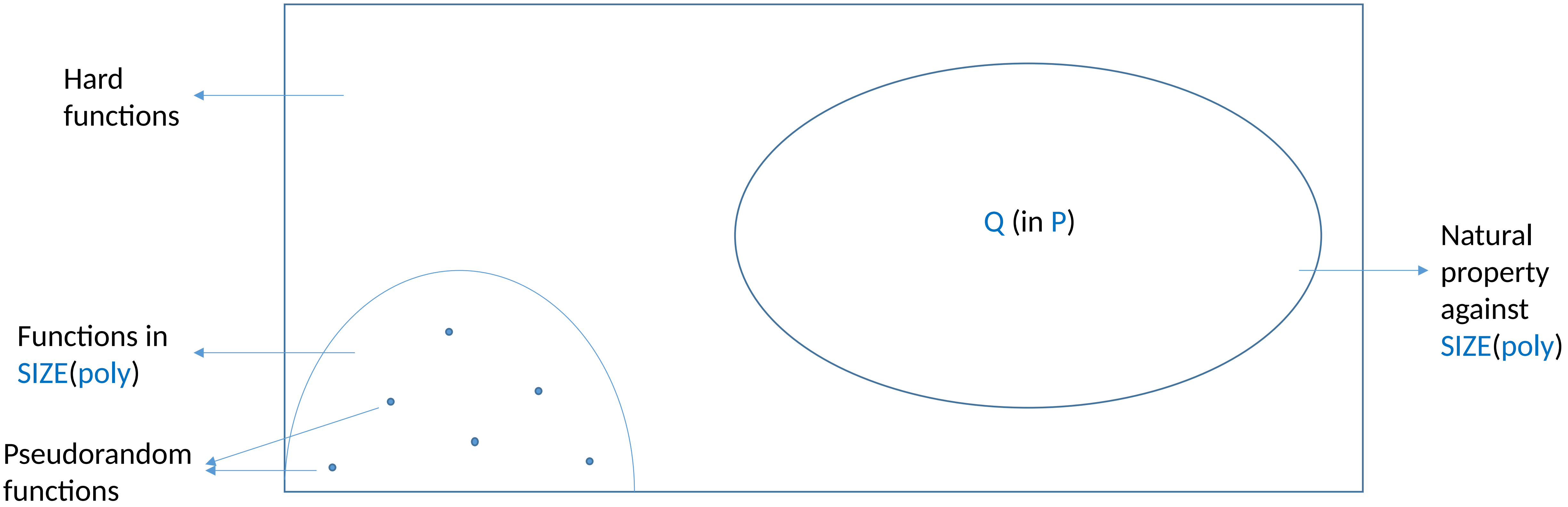


Natural Proofs

- Given a circuit class C, a natural proof against C is a property Q of Boolean functions (represented by their truth tables of size N) such that:
  - Constructivity: Q in P
  - Usefulness: Q(F) = 1 => F not in C
  - Density: At least a $1/N^{O(1)}$ fraction of Boolean functions F satisfy Q
- Razborov and Rudich observed that standard circuit lower bound proofs against restricted circuit classes yield natural proofs against C
- Main theorem [RR97]: If exponentially hard one-way functions exist, there are no natural proofs against SIZE(poly)

Natural Proofs: Proof of Main Theorem

Lemma [GGM86]: If exponentially hard one-way functions exist, then there is pseudorandom function family in SIZE(poly) against SIZE($2^{O(n)}$)



Q distinguishes random from pseudorandom, and is poly-time computable. Contradiction!

Natural Proofs and Meta-Complexity

- Natural proofs are closely related to meta-complexity
- Natural proofs distinguish *easy* Boolean functions from *random* Boolean functions
  - Relaxation of MCSP to the average-case setting
- Thus the average-case hardness of MCSP might explain the difficulty of proving lower bounds (including for MCSP itself!)
- This is reminiscent of Chaitin's incompleteness result
  - Chaitin's result says that because strings are incompressible, it is hard to prove that strings are incompressible
  - The natural proofs barrier suggests that because MCSP is hard, it is hard to prove that MCSP (and other Boolean functions) are hard

Plan of the Talk

- Metamathematics
- *Learning*
- Cryptography
- Complexity Lower Bounds

Search to Decision Reductions

- Let L be a problem in NP
- The decision problem for L is to decide, given x, whether x in L
- The search problem for L is to find, given x in L, a *proof* or *witness* that x in L
- Classical result: SAT is decidable in polynomial time iff the search problem for SAT is solvable in polynomial time
  - Proof idea: Iteratively determine the witness bit by bit, using one oracle call to the decision problem for each bit of the witness

Search to Decision for MCSP?

- The idea of the search-to-decision reduction for SAT doesn't seem to work for MCSP
  - Unclear how to find a circuit for a given truth table bit by bit just by asking questions about MCSP
- Until recently, nothing was known about whether search reduces to decision for MCSP
- The search version of MCSP is closely related to *learning*

Learning and MCSP

- Learning model: The learner is given oracle access to a target Boolean function F and outputs a "good" hypothesis (i.e., small circuit) C approximating the target function if there is a good hypothesis consistent with F
- Search version of MCSP: Given a truth table of a Boolean function F, output a small circuit C for the truth table if one exists
- Intuitively, if there is an efficient learner, one can solve (approximately) the search version of MCSP, simply using the input truth table to answer oracle queries

Learning from Solving MCSP Efficiently

- Theorem [CIKK16]: Let C be a "reasonable" circuit class. If C-MCSP[2^{n^ε}] can be solved in time poly(N) (on average over the uniform distribution), then C-circuits of poly(n) size can be learned in time $2^{\mathrm{polylog}(n)}$
- Corollary [CIKK16]: The class AC^0[Parity] of constant-depth unbounded fan-in circuits with Parity gates can be learned in quasi-polynomial time
  - Average-case algorithms for AC^0[Parity]-MCSP had been known since [RR97], based on lower bound techniques against AC^0[Parity]

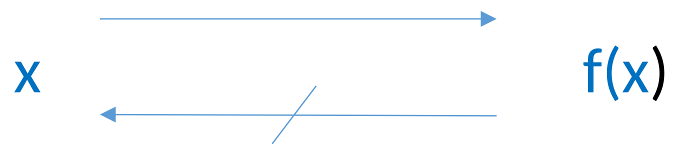
Speedup for Learning

- Theorem [OS17]: Let C be a "reasonable" circuit class. There is $\varepsilon > 0$ such that C-circuits of 2^{n^ε} size can be learned in time $2^{O(n)}$ if and only if C-circuits of poly(n) size can be learned in time $2^{\mathrm{polylog}(n)}$
- The statement of this result doesn't directly involve MCSP or meta-complexity, but the proof crucially uses the main result of [CIKK16]

Plan of the Talk

- Metamathematics
- Learning
- *Cryptography*
- Complexity Lower Bounds

One-Way Functions (OWFs)



- Efficient computability: f can be computed in polynomial time
- No efficient invertibility: There is no probabilistic poly-time procedure A that for most x, produces an inverse to f(x)

OWFs and Cryptography

- OWFs are the most fundamental primitive in theoretical cryptography
  - Cryptographic tasks such as private-key encryption, pseudorandom generation, bit commitment, message authentication and digital signatures are all *equivalent* to the existence of OWFs
- OWFs are based on various well-studied complexity assumptions such as the hardness of the Discrete Logarithm problem, Factoring problem and the Shortest Vector problem in certain lattices

Should We Believe in the Existence of OWFs?

- The existence of OWFs implies that NP $\neq$ P (and even the hardness of NP problems on average) but the reverse implication is unknown
- Problems such as Discrete Logarithm and Factoring are known to be efficiently solvable by quantum algorithms
- Other standard assumptions such as hardness of lattice problems could be much stronger than what we require

Characterizing OWFs using Meta-Complexity

- Liu and Pass [LP20] showed how to characterize OWFs using a natural average-case meta-complexity assumption
- Given a polynomial time bound t, we say that K^t is mildly hard on average over the uniform distribution if there is a polynomial p such that any probabilistic poly-time algorithm must fail to compute K^t on at least a $1/p(n)$ fraction of strings for large enough n
- Theorem [LP20]: Fix any polynomially bounded $t > 1.1\, n$. OWFs exist iff K^t is mildly hard on average over the uniform dist
- This is the first characterization of OWFs using average-case hardness of a natural problem

A Further Characterization of OWFs

- Theorem [IRS22]: The following are equivalent:
  - One-way functions exist
  - Kolmogorov complexity is hard to approximate on average over some "samplable" distribution, i.e., distribution sampled by some poly-time procedure
- Characterization based on hardness over *any* samplable distribution, while previous characterizations relied on the uniform distribution
- Works even for the uncomputable problem K!

Plan of the Talk

- Metamathematics
- Learning
- Cryptography
- *Complexity Lower Bounds*

Uniform vs Non-Uniform Lower Bounds

- Major open questions in complexity theory, such as the NP vs P question and the PSPACE vs P question, are about *uniform* lower bounds
- Since the 1980s, approaches to these questions have focused on showing stronger *non-uniform* lower bounds, i.e., that there is a problem in NP or in PSPACE that does not have polynomial-size Boolean circuits
  - These approaches have been largely unsuccessful and barriers such as the natural proof barrier [RR97] are known
- We are interested in new ways of exploiting the uniformity condition when proving lower bounds

Algorithmic Approaches to Lower Bounds

- While the area of complexity lower bounds has seen infrequent progress, research in algorithms is thriving [CKLPPS22, BNW22]
- Lower bounds are *impossibility* results while algorithms results are *possibility* results
- Counter-intuitive idea: Could we approach a lower bound by designing and analysing an algorithm for some computational task that we believe to be feasible?

Algorithmic Approaches to Lower Bounds

- Williams [W10] proposed an algorithmic approach to proving circuit lower bounds for NEXP (non-deterministic exponential time), and applied the approach [W11] to show that a new circuit lower bound for NEXP against ACC^0 circuits
- He showed in general that if SAT can be solved on C-circuits of size m on n variables in time $poly(m)2^{n-\omega(\log(n))}$, then NEXP does not have polynomial-size C-circuits

Algorithmic Approaches to Lower Bounds

- Williams' approach only has the potential to yield lower bounds against size s circuits for problems that require time more than s to solve, eg., lower bounds for exponential time against polynomial size
- However, in order to attack the NP vs P problem, we need to find an approach that applies to a problem solvable non-deterministically in some fixed polynomial amount of time (such as SAT) and yields arbitrary polynomial size lower bounds
- We give such an algorithmic approach, but for *uniform* rather than *non-uniform* lower bounds for PSPACE and NP

A Circuit-Based Sampling Task

- Input: A circuit C on n variables and of size $s = poly(n)$, such that C accepts at least a 2/3 fraction of all inputs
- Task: Output some element of SAT(C) with probability $>> 2^{-n}$
  - Here SAT(C) is the set of satisfying assignments of C

- The trivial algorithm that outputs a random bitstring of length n runs in time n and outputs each element of SAT(C) with probability 2^{-n}
  - Can we find an algorithm that is almost as efficient but beats random guessing for some element of SAT(C)?

A Simulation-Based Algorithm

- Input: A circuit C on n variables and of size $s = poly(n)$, such that C accepts at least a $2/3$ fraction of all inputs
- Task: Output some element of $SAT(C)$ with probability $>> 2^{-n}$
  - Here $SAT(C)$ is the set of satisfying assignments of C

- The following simple algorithm runs in time (and space) $O(sn^5)$ and outputs some element of $SAT(C)$ with probability $>= n^4/2^n$: pick n^5 strings of length n independently and uniformly at random, and output the lexicographically first one that satisfies C

An Algorithmic Approach

Input: A circuit C on n variables of size poly(n), accepting $\geq 2/3$ fraction of inputs

Task: Output some fixed satisfying input y of C with probability $\geq n^4/2^n$, using space $O(n^2)$

Theorem [S23]: If the task is solvable, then PSPACE ≠ P

- This gives an *algorithmic* formulation of the PSPACE ≠ P problem, which is about *lower bounds*
- Proof of the implication uses meta-complexity

An Algorithmic Approach

Input: A circuit C on n variables of size $poly(n)$, accepting $\geq 2/3$ fraction of inputs, described by a *compressed* representation of size n

Task: Output some fixed satisfying input y of C with probability $\geq n^4/2^n$, using time $O(n^2)$

Theorem [S23]: If the task is solvable, then $NP \neq P$

- This gives an *algorithmic* formulation of the $NP \neq P$ problem, which is about *lower bounds*
- Proof of the implication uses meta-complexity

Features of the Approach

- It is an approach to NP vs P that exploits the power of NP
  - Several previous approaches to circuit lower bounds for circuit classes C yielded hard functions in P against C, and therefore are not useful in the most general setting
- It exploits uniformity of the lower bound
  - Previous approaches applied to non-uniform lower bounds and ran up against the natural proofs barrier [RR97]
  - It is possible that uniform lower bounds are much easier to prove than non-uniform ones
- It is very general, applying to any circuit class C, and therefore could be useful in making gradual progress

Proof Template

- Reminder of circuit-based sampling task for PSPACE lower bounds
  - Given: A circuit C on n variables of size poly(n), accepting $\geq 2/3$ fraction of inputs
  - Output: Some fixed satisfying input y of C with probability $\geq n^4/2^n$
  - The algorithm should use space $O(n^2)$
- Theorem: If the circuit-based sampling task is solvable, then PSPACE $\neq$ P
- The statement of the theorem does not involve meta-complexity, but the proof will use meta-complexity as a tool

Proof Template

- Theorem: If the circuit-based sampling task is solvable, then PSPACE ≠ P
- We assume, for the sake of contradiction, that PSPACE = P
- We consider a version of Kolmogorov complexity called *probabilistic time-bounded Kolmogorov complexity* pK^{poly} [GKLO22]
  - Informally, the pK^{poly} complexity of a string x is the size of the smallest program that can generate x in polynomial time given access to a random string
- Let R be the set of strings with pK^{poly} complexity at least $n-1$
- Easy to show that R includes at least half the strings of length n

Proof Template

- Theorem: If the circuit-based sampling task is solvable, then PSPACE $\neq$ P
- Let R be the set of strings with pK^{poly} complexity at least n-5
- Easy to show that R includes at least half the strings of length n and also that R is in PSPACE
- Since PSPACE = P, we have that R has uniform Boolean circuits $\{C_n\}$, where $pK^{poly}(C_n)$ is at most $\log(n) + O(1)$ by uniformity
- By the solvability of the circuit sampling task, we can show that there is a string y accepted by C_n such that $pK^{poly}(y|C_n)$ is at most $n-3\log(n)$
- Therefore $pK^{poly}(y)$ is at most $n-\log(n)$ for large n, which contradicts the assumption that $y \in R$

Necessity of the Approach

- Theorem: Under standard circuit lower bound assumptions for exponential time (i.e., that DTIME($2^{O(n)}$) requires circuits of size $2^{\Omega(n)}$), PSPACE ≠ P if and only if the sampling task is solvable
- Thus the approach is without loss of generality if we believe in strong circuit lower bounds

Applications of the Approach

- The approach can be used to give new proofs of old results such as the space hierarchy theorem and Allender's uniform lower bound for the Permanent [A99]
- It can also be used to show some new uniform lower bounds in NP (but still very far off from saying anything interesting about NP vs P)

Open Problems

- Find other applications of meta-complexity to learning and cryptography, eg., show that the task of learning in general is NP-complete
- Use the new algorithmic approach to lower bounds to make progress, eg., show that NP does not have uniform depth-2 neural networks of polynomial size
- Better understanding of the meta-mathematics of circuit lower bounds, eg., give evidence that circuit lower bounds for NP do not have efficient proofs in the Frege proof system