The Hardest Explicit Construction

September 23, 2021

Explicit Construction Problems

In an explicit construction problem, we have some "property" (language) $\Pi \subseteq \{0,1\}^*$, and the goal is to produce an n-bit string $x \in \Pi$ in time $poly(n)$. This can be phrased more formally as a search problem, where the input is 1^n, and the solutions are $\Pi \cap \{0,1\}^n$.

A familiar example is the construction of expander graphs: we have some fixed expansion parameters in mind, and given 1^n we want to print the adjacency matrix of an n-vertex expander holding these expansion parameters in time $poly(n)$.

Explicit Construction Problems (cont.)

For the problem to be of any interest, we should should have a reason to believe that objects with property Π exist for all n, i.e. $\Pi \cap \{0,1\}^n \neq \emptyset$. This makes the explicit construction problem a "total search problem."

One way to phrase totality is as a promise. However, as is known in the study of TFNP, a promise makes it difficult to define interesting classes with complete problems. Instead, it is often fruitful to identity the basic combinatorial lemma which guarantees the existence of solutions, and define a complexity class in terms of reducibility to some canonical problem corresponding to this lemma.

The Probabilistic Method

For many important explicit construction problems (e.g. the construction of expanders, good codes, rigid matrices, hard boolean functions, randomness extractors, ramsey graphs), the known existence proofs utilize the *probabilistic method*: proving that a random string posesses the desired property with high probability.

In all of the above cases, the probabilistic method can be rephrased as an *encoding argument*: any n-bit string which fails to possess the desired property can be encoded using $< n$ bits, implying that a random n bit string must possess the property with probability $\geq \frac{1}{2}$.

The Empty Pigeonhole Principle (APEPP)

In KKMP'21, a class is defined which captures the complexity of making such encoding arguments "explicit:"

EMPTY is the following search problem: given a circuit $C : \{0,1\}^m \to \{0,1\}^n$ with $m < n$, find an n-bit string outside the range of C. APEPP is then the class of problems poly-time reducible to EMPTY.

APEPP$\subseteq \Sigma_2^{\mathrm{P}}$ follows from the definition
APEPP $\subseteq \mathrm{ZPP}^{\mathrm{NP}}$ since $|\{0,1\}^n| \geq 2|\{0,1\}^m|$ for $m < n$

APEPP Captures Explicit Constructions

We show that the explicit construction problems for a wide range of objects fall into APEPP, in particular:

1. n-bit truth tables of hardness $\frac{2^n}{2n}$
2. Pseudorandom generators (in the Nisan-Wigderson sense)
3. Strongly-explicit 2-source randomness extractors and Ramsey graphs with optimal parameters
4. Matrices of high rigidity over any finite field
5. Strings of high time-bounded Kolmogorov complexity
6. Communication matrices outside PSPACE^{CC}
7. Hard data stru cture problems in the bit-probe model

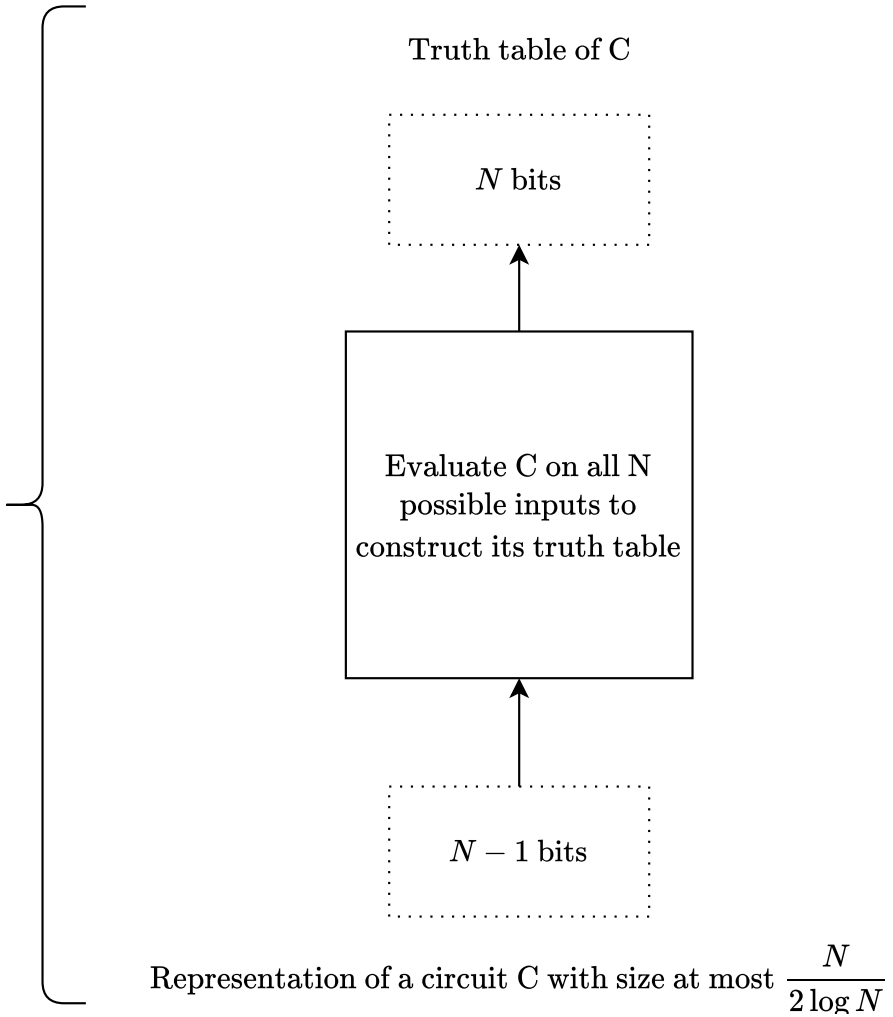
Example 1: Hard Truth Tables in APEPP

A truth table is an *explicit* representation of a function $f : \{0,1\}^n \to \{0,1\}$, written as a 2^n-bit binary string giving the values of $f(x)$ for each $x \in \{0,1\}^n$ in lexicographical order. When dealing with truth tables, we think of $N = 2^n$ as the input size, so we are satisfied with constructions running in time $poly(N) = poly(2^n)$.

Any circuit with s gates can be specified using $2s \log s + O(s)$ bits. This can be easily used to show that any circuit on $n = \log N$ inputs of size at most $\frac{N}{2 \log N} = \frac{2^n}{2n}$ can be represented using $N - 1$ bits (indeed significantly fewer bits suffice).

Hard Truth Tables in APEPP (cont.)

An instance of EMPTY such that any N-bit string outside its range has circuit complexity $\frac{N}{2 \log N}$



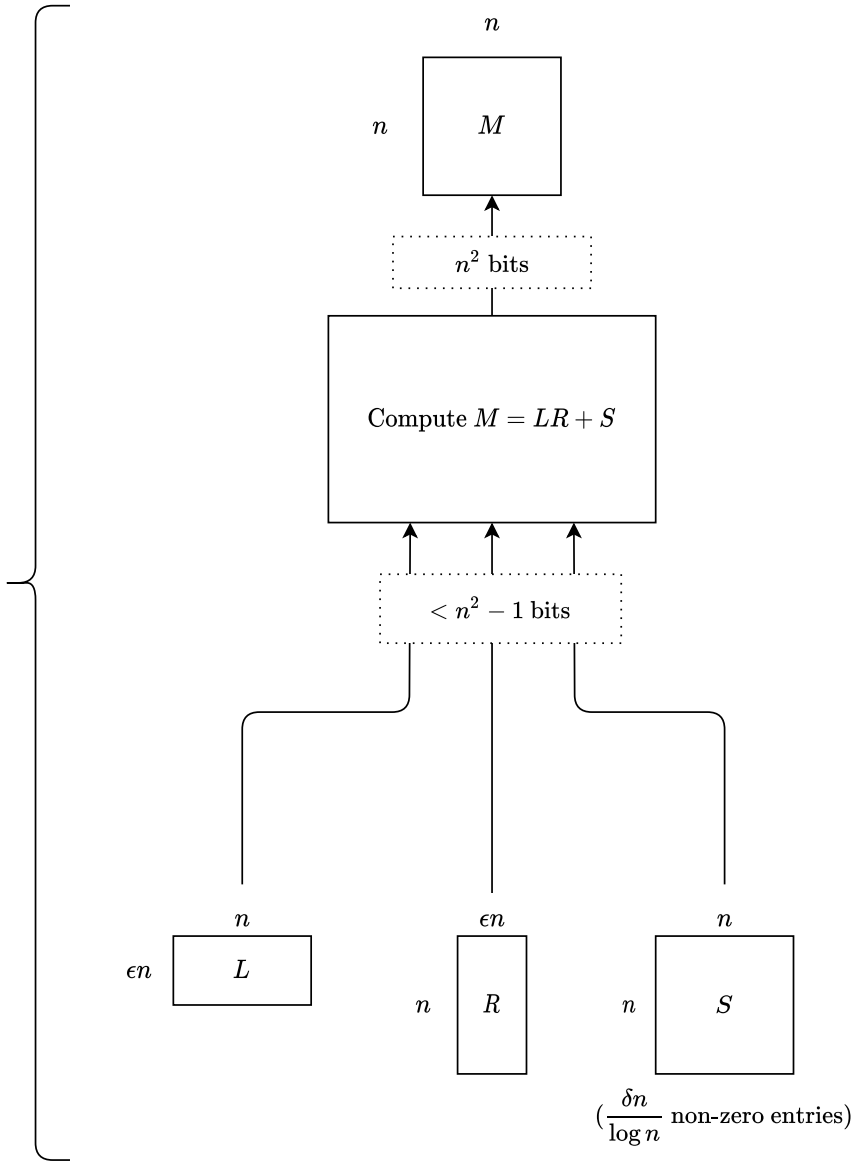
Example 2: Rigid Matrices in APEPP

An $n \times n$ matrix M is (r, s)-rigid if there *do not exist* $n \times n$ matrices A, S such that A has rank at most r, S has at most s non-zero entries, and $A + S = M$.

Explicit construction of a $(\Omega(n), \Omega(n^{1+\varepsilon}))$-rigid $n \times n$ matrix has been a notoriously open problem since the 70's, even if the construction algorithm is allowed an NP-oracle.

Rigid Matrices in APEPP (cont.)

An instance of EMPTY such that any n^2-bit string outside its range encodes an $n \times n$ matrix with rigidity $(\epsilon n, \frac{\delta n^2}{\log n})$



Example 3: PRG's

Theorem

Construction of PRGs (in the complexity theoretic sense) reduces in polynomial time to EMPTY.

Though we won't go into detail here, important note is that the proof of this is *quite simple*, we'll see why that's interesting later...

A Complete Problem for APEPP

Since APEPP appears to be the natural class for explicit construction problems from the probabilistic method, one would hope that some natural explicit construction problem is *complete* for it as well.

As it turns out, construction of hard truth tables is APEPP-complete under P^{NP} reductions!

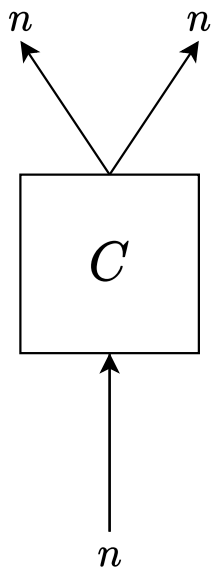
The core of this theorem was in fact proven almost 20 years ago by Emil Jeřábek, where it was phrased as a logical expressibility result in Bounded Arithmetic, and the main construction dates back further to the PRF generator of GGM'86. We show how this argument can be applied to give a *reduction* from EMPTY to the construction of hard truth tables.

The Problem ε-HARD

For any fixed $0 < \varepsilon < 1$, ε-HARD is the following search problem: given 1^N, find an N-bit truth table which requires circuits of size N^ε.

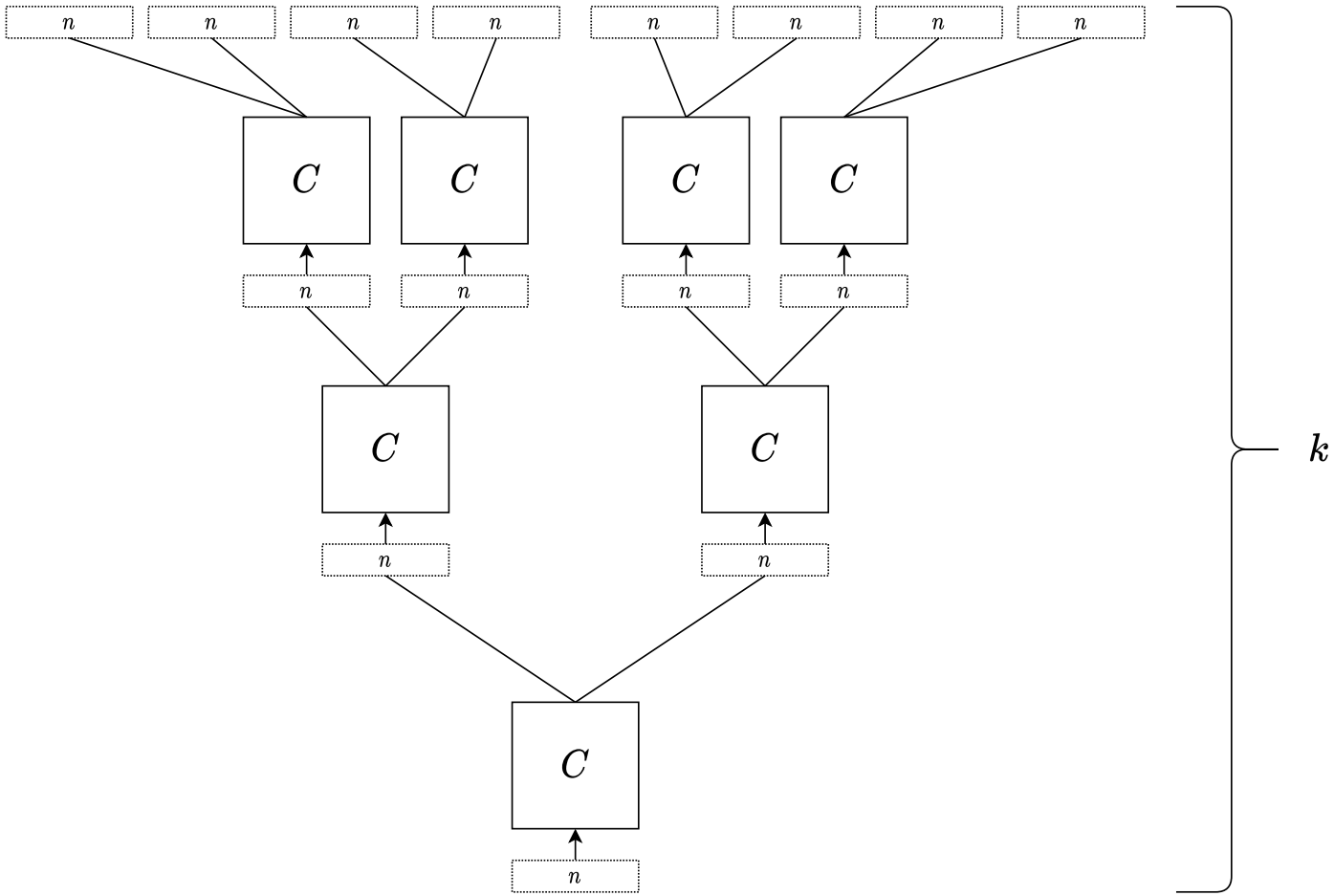
In the typical case where $N = 2^n$, this is equivalent to finding the truth table of an n-variable function requiring circuits of size $2^{\varepsilon n}$, the same object used to build the Impagliazzo-Wigderson generator.

Reduction from EMPTY to ε-HARD



We focus here on instances of EMPTY with twice as many output bits as input bits. Such an instance can equivalently be viewed as a candidate cryptographic PRG which we are attempting to break.

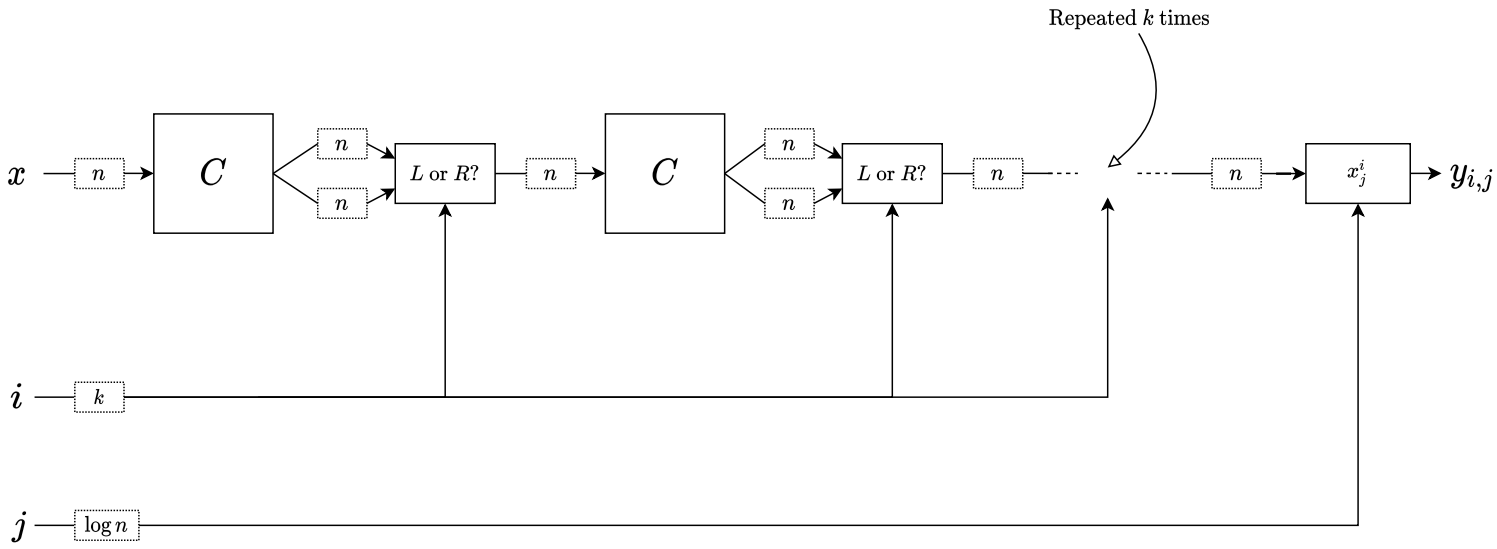
GGM Construction



Extending a map $C : \{0,1\}^n \to \{0,1\}^{2n}$ to a map $C^* : \{0,1\}^n \to \{0,1\}^{2^k n}$

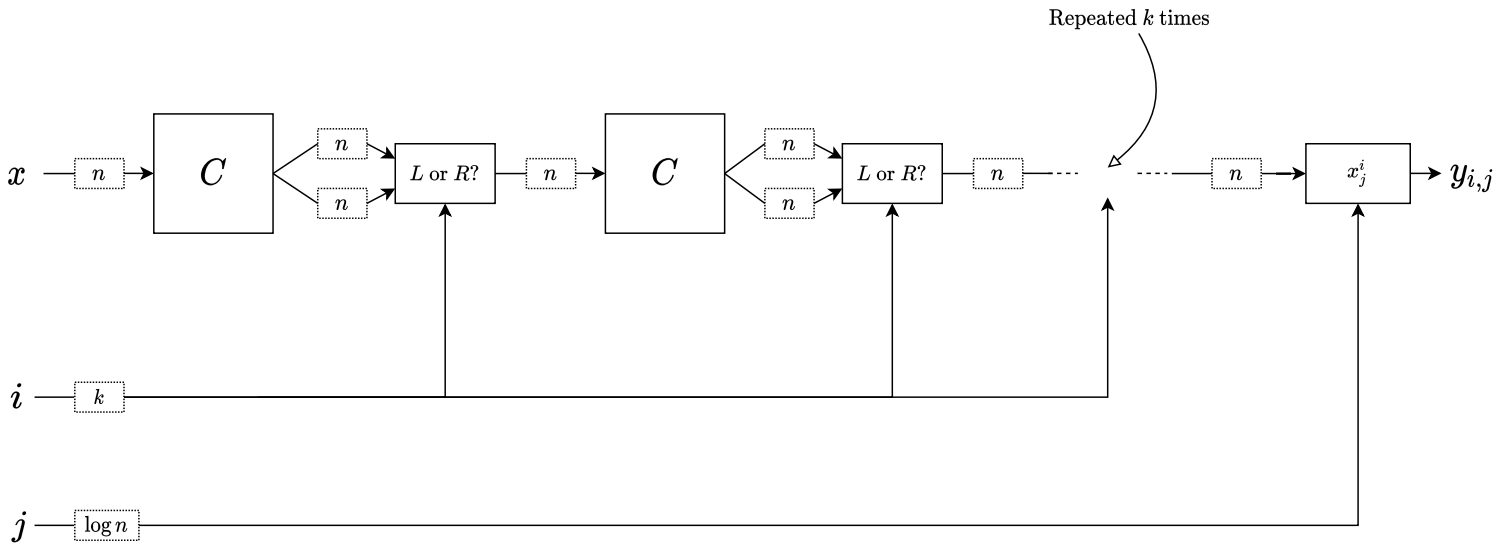
GGM Construction (cont.)

A circuit of size $O(|C|k)$ exists for any y in the range of C^*:



GGM Construction (cont.)

A circuit of size $O(|C|k)$ exists for any y in the range of C^*:



Setting $k = 2\lceil \log |C| \rceil \lceil \frac{1}{\varepsilon} \rceil$, any solution to ε-HARD on input $1^{2^k n}$ must therefore be outside the range of C^*, since:
$(2^k n)^\varepsilon > (|C|^{\frac{2}{\varepsilon}})^\varepsilon = |C|^2 >> O(|C|k) = O(|C| \log |C|)$

Reduction from EMPTY to ε-Hard

We have established that every solution to ε-Hard is outside the range of C^*. GGM proved the second critical property of C^*:

Lemma (GGM '86)

Given a statistical test T breaking C^*, we can construct a statistical test breaking C of poly(n) circuit complexity using a T-oracle.

Lemma (This work and J'04)

Given a string outside the range of C^*, we can construct a string outside the range of C in polynomial time using an oracle for inverting C.

This immediately implies the completeness result: a solution to ε-Hard will be outside the range of C^*, and with the aid of an NP-oracle (which allows us to invert C) we can use this to find a string outside the range of C.

Summary

Thus, the construction C^* highlights two ways in which the property of having circuit complexity n^ε is a *universal pseudorandom property of n-bit strings*:

1. An efficient test determining if strings have circuit complexity n^ε can be bootstrapped into an efficient test breaking *any* cryptographic pseudorandom generator.
2. An explicit example of a string with circuit complexity n^ε can be used to efficiently construct a string outside the range of an arbitrary circuit $C : \{0,1\}^n \to \{0,1\}^{2n}$ (or more generally a circuit $C : \{0,1\}^n \to \{0,1\}^{n+1}$ as shown in the paper).

Implications of Completeness

The existence of a P^{NP} algorithm for ε-HARD is equivalent to the existence of a language in E^{NP} of circuit complexity $2^{\Omega(n)}$. So such a circuit lower bound holds *if and only if* every problem in APEPP has a P^{NP} algorithm.

Proving circuit lower bounds for E^{NP} is thus a **universal explicit construction problem**, and is equivalent to showing a generic derandomization of the probabilistic method.

Comparison to NW-type Generators

As mentioned at the start, $\mathrm{APEPP} \subseteq \mathrm{ZPP}^{\mathrm{NP}}$. Known extensions of the Impagliazzo-Wigderson Generator to oracle classes give us the following:

If E^{NP} requires *nondeterministic* circuits of size $2^{\Omega(n)}$, then $\mathrm{APEPP} \subseteq \mathrm{P}^{\mathrm{NP}}$

So our result weakens the assumption to standard circuits, and in doing so is able to prove *equivalence* between a certain lower bound and a certain derandomization, since $\mathrm{APEPP} \subseteq \mathrm{P}^{\mathrm{NP}}$ conversely implies that E^{NP} has circuit complexity $2^{\Omega(n)}$ for standard circuits (it does not seem to imply the same for nondeterministic circuits).

Simple Proof of (Weaker) Hardness-Pseudorandomness Connection

As mentioned earlier, there is a simple proof that construction of pseudorandom generators (in the NW-sense) can be reduced to EMPTY. In combination with the completeness result, this yields a self-contained proof that construction of pseudorandom generators reduces to construction of hard truth tables (via P^{NP} reductions).

This proof is notably simpler then that of Nisan, Wigderson, and Impagliazzo, though their result is significantly stronger as it does not require an NP oracle.

Worst-Case to Worst-Case Hardness Amplification

Theorem

E^{NP} *contains a language of circuit complexity* $2^{\Omega(n)}$ *if and only if it contains a language of circuit complexity* $\frac{2^n}{2n}$.

Earlier we saw that construction of truth tables of length 2^n with hardness $\frac{2^n}{2n}$ reduces to EMPTY, and in turn that EMPTY reduces to ε-HARD for any fixed $0 < \varepsilon < 1$. The above result follows.

Tweaking the reduction slightly we also obtain:

Theorem

$\mathrm{EXP}^{\mathrm{NP}}$ *contains a language of circuit complexity* $2^{n^{\Omega(1)}}$ *if and only if it contains a language of circuit complexity* $\frac{2^n}{2n}$.

Hardness Extraction

Unpacking the proof of the above corollaries reveals an efficient algorithm to "extract hardness" from truth tables using an oracle for circuit minimization, a prospect previously considered by Buresh-Oppenheim and Santhanam (BS06):

Theorem

There is a polynomial time algorithm using a circuit minimization oracle (or more generally an NP oracle) which, given a truth table x of length M and circuit complexity s, outputs a truth table y of length $N = \Omega(\sqrt{\frac{s}{\log M}})$ and circuit complexity $\Omega(\frac{N}{\log N})$.

Barriers to Improving the Extractor

The square root in the above theorem arises from the following: in the proof we consider the function $C : \{0,1\}^{\frac{N}{2}} \to \{0,1\}^N$ which maps descriptions of circuits of size $\approx \frac{N}{4 \log N}$ to their corresponding truth tables, and argue that there is a clear $O(N^2)$ upper bound on $\mathit{size}(C)$. Improving this upper bound to $N^{2-\varepsilon}$ would allow us to extract $\approx \left(\frac{s}{\log M}\right)^{\frac{1}{2}+\varepsilon}$ bits of hardness.

However, as observed by Williams (W'10), such an upper bound on $\mathit{size}(C)$ would in fact imply a (nonuniform) break-through from 3SUM.

Direct Reductions to Hard Truth Table Construction

Is ε-Hard complete for APEPP under polynomial time reductions? This would imply APEPP $\subseteq$ P/poly, not clear that this should hold.

However, we demonstrate that the following problems reduce in polynomial to hard truth table construction:

- Matrices of rigidity $(\Omega(n), \Omega(n^2))$ over $\mathbb{F}_2$
- Communication matrices outside $\text{PSPACE}^{\text{CC}}$
- Explicit hard data structure problems in the bit-probe model

Example: Rigid Matrices

Theorem

If E contains a language of circuit complexity $\Omega(\frac{2^n}{n})$, then there is a polynomial time construction of $(\Omega(n), \Omega(n^2))$-rigid matrices over $\mathbb{F}_2$.

Example: Rigid Matrices (cont.)