Questions Pool

Amnon Ta-Shma and Dean Doron

January 2, 2017

General guidelines

The questions fall into several categories:

(Know).	Make sure you know how to solve. Do not submit.
(Mandatory).	Mandatory questions.
(Bonus).	Bonus questions.

HW 1 – Error-correcting codes.

Out: 6.11.2016
Due: 20.11.2016

1. (Know). Let C be a q-ary linear error-correcting code. Prove that the minimal weight of a nonzero codeword is d if and only if the minimum Hamming distance between every two distinct codewords is at least d.

2. (Mandatory). Let C be an $(n,k)_q$ code. Prove that there exists a word $w \in \mathbb{F}_q^n$ such that $|B(w,(1-1/q)n) \cap C| \geq q^{k-o(n)}$.

3. (Mandatory). Let C be a (linear) $[n,k,d]_q$ code. Prove that $d \leq n-k+1$.

4. (Know). Let C be a (linear) $[n,k,d]_q$ code with a generating matrix G. Show to decode codewords where no error occurred.

5. (Mandatory). Let n_1 be a power of 2 and A a $[n_1,k_1,d_1]_{n_1}$ code. Let B be a $[n_2,\log n_1,d_2]_2$ code. Suppose $A(\bar{x})$ for $\bar{x}=x_1,\ldots,x_{k_1}$ is $A(\bar{x}) = A_1(\bar{x}) \circ \ldots \circ A_{n_1}(\bar{x})$, with $A_i(\bar{x}) \in \mathbb{F}_{n_1}$. Define $B \circ A$ to be $(B \circ A)(\bar{x}) = B(A_1(\bar{x})) \circ \ldots \circ B(A_{n_1}(\bar{x}))$. Prove that $B \circ A$ is a linear binary code, and find its dimension and distance.

6. (Mandatory). Suppose you can *efficiently* decode A and B up to half the distance. Show an efficient algorithm decoding the concatenated code. How many errors can you *efficiently* correct?

7. (Mandatory). Prove the Johnson bound (Theorem 2 from Lecture 2) for the case of $q=2$.

   Guidance: Fix a word y and let $c_1,\ldots,c_L \in B(y,e) \cap C$. Define $c_i' = c_i - y$ and let $S = \sum_{i<j} d(c_i',c_j')$. Find an upper bound and a lower bound on S.

For the upper bound, consider the matrix M whose columns are the c_i'-s and define m_i as the number of 1-s in each row. Express S using M and the m_i-s and obtain an upper bound.

8. (Know). In the Reed-Muller code, we encoded a message $x \in \mathbb{F}_q^k$ into a multivariate polynomial $p : \mathbb{F}_q^m \to \mathbb{F}_q$. Prove the existence and uniqueness of p and explain how to find it efficiently.

9. (Know). Fix $a \in \mathbb{F}_q^m$ and consider a random curve $\Gamma : \mathbb{F}_q \to \mathbb{F}_q^m$ of degree-k that passes through a. That is, $\Gamma(t) = a + \sum_{i=1}^{t} z_i t^i$ where the z_i-s are chosen uniformly and independently from $\mathbb{F}_q^m$. Prove that the random variables $\Gamma(1), \ldots, \Gamma(q-1)$ are uniform and k-wise independent.

10. (Know). The Hadamard code Had is a $[n = 2^k, k]_2$ code. For a string $z \in \{0,1\}^k$, the w-th coordinate of $\mathsf{Had}(z) \in \{0,1\}^{2^k}$ is $\langle z, w \rangle$ modulo 2, which we abbreviate as $\langle z, w \rangle_2$. Prove that the Hadamard code has relative distance $1/2$.

11. (Mandatory). Prove that the Hadamard code is δ-locally decodable for $\delta < 1/4$. How many queries do you have?

12. (Mandatory). Prove that $\mathsf{Had} : \{0,1\}^k \to \{0,1\}^{2^k}$ is $\left(\frac{1}{2} + \varepsilon, \frac{1}{4\varepsilon^2}\right)$-list-decodable in time $\mathrm{poly}(\frac{k}{\varepsilon})$.

    Guidance: Let $n = 2^k$ and view $f \in \{0,1\}^n$ as $f : \{0,1\}^k \to \{0,1\}$. Consider choosing $z_1, \ldots, z_m \in \{0,1\}^n$ uniformly at random and given $i \in \{0,1\}^k$, outputting the majority value among $\{f(z_j) \oplus f(z_j \oplus e_i)\}_{j \in [m]}$. This algorithm works (for a suitable choice of m) when ε is high (say 0.4). Why? How can we adopt it to handle an arbitrarily small ε?

13. (Mandatory). A function $f : \{0,1\}^\star \to \{0,1\}^\star$ is a one-way function if f can be computed by a polynomial-time algorithm and for any probabilistic polynomial-time algorithm A and any constant c, for every large enough n, it holds that $\Pr_{x \in \{0,1\}^n, r}[A(f(x), r) \in f^{-1}(f(x))] < n^{-c}$.

    Let f be a one-way function such that f is one-to-one. Prove that for every probabilistic polynomial-time algorithm A there is a negligible function $\varepsilon = \varepsilon(n)$ such that $\Pr_{x,r}[A(f(x), r) = \langle x, r \rangle_2] \leq 1/2 + \varepsilon$.

14. (due to Kopparty) (Mandatory). Let d be an odd integer and let C be an $[n, k, d]_2$ code. Show that there exists a linear code C' that is a $[n, k-1, d+1]_2$ code.

15. (due to Guruswami) (Bonus). Let $1 \leq k \leq n$ be integers and let $p_1 < \ldots < p_n$ be n distinct primes. Denote $K = \prod_{i=1}^{k} p_i$ and $N = \prod_{i=1}^{n} p_i$. Consider the mapping $E : \mathbb{Z}_K \to \mathbb{Z}_{p_1} \times \ldots \times \mathbb{Z}_{p_n}$ defined by:

$$E(m) \;=\; (m \bmod p_1, \ldots, m \bmod p_n).$$

    (a) Suppose that $m_1 \neq m_2$. For $i \in [n]$, define the indicator b_i such that $b_i = 1$ iff $E(m_1)_i \neq E(m_2)_i$. Prove that $\prod_{i=1}^{n} p_i^{b_i} > N/K$.

    Deduce that when $m_1 \neq m_2$, $\Delta(E(m_1), E(m_2)) \geq n - k + 1$.

    (b) We will now adopt the Welch-Berlekamp algorithm to handle E. Suppose $r = (r_1, \ldots, r_n)$ is the received word, where $r_i \in \mathbb{Z}_{p_i}$.

        i. Prove there can be at most one $m \in \mathbb{Z}_K$ such that

$$\prod_{i : E(m)_i \neq r_i} p_i^{b_i} \leq \sqrt{N/K}. \tag{1}$$

In what follows, let r be the unique integer in $\mathbb{Z}_N$ such that $r \bmod p_i = r_i$ for every $i \in [n]$ (note that the Chinese Remainder theorem guarantees that there is a unique such r).

ii. Assuming such an m exists, prove that there exist integers y, z with $0 \leq y < \sqrt{NK}$ and $1 \leq z \leq \sqrt{N/K}$ such that $y \equiv rz \pmod{N}$.

iii. Prove that if y, z are any integers satisfying the above conditions, then in fact $m = y/z$. Note that a pair of integers (y, z) satisfying the above can be found by integer linear programming in a fixed number of dimensions in polynomial time.

(c) Instead of condition (1), what if we want to decode under the more natural condition: $|\{i \mid E(m)_i \neq r_i\}| \leq \frac{n-k}{2}$? Show how this can be done by calling the above decoder many times and erasing the last i symbols for each choice of $i \in [n]$.

HW 2 – Hardness implies derandomization

Out: 23.11.2016
Due: 11.12.2016

1. (Mandatory). Revisit the list-decoding algorithm for Reed-Solomon codes we gave in class, and re-prove it taking care also of the output list size. That is, prove Theorem 6 from Lecture 2 (taken from [2]):

   Theorem 1. *There exists an algorithm that given as input:*

   - *Code parameters: q, $n \leq q$, deg,*
   - *A sequence of n distinct pairs $\{(\alpha_i, y_i)\}_{i=1}^{n}$, $\alpha_i, y_i \in \mathbb{F}_q$ and*
   - *An agreement parameter $\tau > \sqrt{\frac{2deg}{n}}$,*

   outputs a list of all polynomials $p_1, \ldots, p_\ell$ of degree at most deg satisfying $|\{i \in [n] : p_j(\alpha_i) = y_i\}| \geq \tau n$. Furthermore, the list size ℓ is at most $\frac{2}{\tau}$. The algorithm runs in time $\mathrm{poly}(n, \log q)$.

   Notice that the list $\{(\alpha_i, y_i)\}_{i=1}^{n}$ may have several values for the same α_i.

2. (Mandatory). Prove that there exists an explicit $[n, k]_2$ code that is $(\frac{1}{2} + \varepsilon, L)$ locally list-decodable where $n = \mathrm{poly}(k, 1/\varepsilon)$ and $L = \mathrm{poly}(n/\varepsilon)$. Notice that the code is binary. The (local) list-decoding procedure runs in time $\mathrm{poly}(\log k, 1/\varepsilon)$.

   For the proof you may take the Reed-Muller code we have analyzed in class, and concatenate it with the Hadamard code. Also recall that Had is $\left(\frac{1}{2} + \varepsilon, \frac{1}{4\varepsilon^2}\right)$-list-decodable.

3. (Know). Last item is Mandatory.

   Prove that if there exists $f \in \mathsf{PSPACE}$ with $\mathsf{Size}(f) \geq s(n)$ then for every $\varepsilon(n) > 0$ there exists another $f' \in \mathsf{SPACE}(\mathrm{poly}(n, \log \frac{1}{\varepsilon}))$ such that $\mathsf{Size}_{\frac{1}{2}+\varepsilon}(f) \geq \frac{s(n/10)}{\mathrm{poly}(\frac{n}{\varepsilon})}$.

   The proofs puts together what we have done in class:

   - Suppose $f \in \mathsf{PSPACE}$ and $\mathsf{Size}(f) \geq s(n)$. Given $f_n : \{0,1\}^n \to \{0,1\}$ construct $f'_{n'} : \{0,1\}^{n'} \to \{0,1\}$ that extends f_n and is supposed to be hard on average (and you need the binary version as in the previous question). What is n' as a function of n? Show that $\left\{f'_{n'}\right\} \in \mathsf{PSPACE}$ by using Lagrange's multi-variate interpolation.
   - Assume C' is of size s' and computes f' correctly with $\frac{1}{2} + \varepsilon$ average-case success. Show a randomized circuit computing f on inputs of length n, such that for every input it succeeds with success probability $2/3$. Which splitting point do you use?
   - Get a deterministic circuit and conclude the theorem.
   - For which of the classes $\mathsf{PSPACE}, \mathsf{E}, \mathsf{EXP}, \mathsf{NEXP}, \mathsf{PSPACE}^{\mathrm{SAT}}, \mathsf{E}^{\mathrm{SAT}}, \mathsf{EXP}^{\mathrm{SAT}}, \mathsf{NEXP}^{\mathrm{SAT}}$ this worst-case to average-case reduction holds?

4. (Know). Let $\varepsilon > 0$ and set $\delta = \varepsilon/2$. Prove that there exists an integer c such that given access to a Boolean function on n^δ variables with circuit complexity at least $n^{c\delta}$, there is a pseudorandom generator $G : \{0,1\}^{n^\varepsilon} \to \{0,1\}^n$ computable in $2^{O(n^\varepsilon)}$ time which fools circuits of size n.

5. (Mandatory). Prove that if there exists a function $f \in \mathsf{E}$ such that $\mathsf{Size}(f) = 2^{\Omega(n)}$ then $\mathsf{BPP} = \mathsf{P}$.

6. (Mandatory). Prove that if there exists $f \in \mathsf{E}$ such that $\mathsf{Size}(f) = 2^{\Omega(n)}$ then $\mathsf{MA} = \mathsf{NP}$.

7. (Mandatory). [1] Let $\mathsf{Size}^{\mathrm{SAT}}(f_n)$ be the minimal size of a circuit C with oracle gates to SAT that solves f_n on inputs of length n.

   Prove that if there exists $f \in \mathsf{E}$ such that $\mathsf{Size}^{\mathrm{SAT}}(f) = 2^{\Omega(n)}$ then $\mathsf{AM} = \mathsf{NP}$.

8. (Know). Let $f : \{0,1\}^n \to \{0,1\}$ and suppose $C : \{0,1\}^n \times \{0,1\} \to \{0,1\}$ is a circuit such that

$$\Pr_{x \sim U_n}[C(x, f(x)) = 1] - \Pr_{x \sim U_n, b \sim U_1}[C(x, b) = 1] > \delta.$$

   Prove that there exists another circuit $C' : \{0,1\}^n \to \{0,1\}$ such that

$$\Pr_{x \sim U_n}[C'(x) = f(x)] > \frac{1}{2} + \delta.$$

9. (Mandatory). Prove that for every large enough n there exists a function $f : \{0,1\}^n \to \{0,1\}$ such that $\mathsf{Size}_{\frac{1}{2}+\varepsilon}(f) \geq 2^{n/10}$ for $\varepsilon = 2^{-\Omega(n)}$.

10. (Mandatory). Prove that (non-explicitly) there exists a (ℓ, a)-design $S_1, \ldots, S_m \subseteq [t]$ where $a = O(\ell^2/t)$ and $m = 2^{\Omega(\ell)}$.

11. Two norm-one vectors $v_1, v_2 \in \mathbb{R}^n$ are almost orthogonal if $|\langle v_1, v_2 \rangle| \leq \varepsilon$.

    (a) (Mandatory). Show how to convert an (ℓ, a)-design $S_1, \ldots, S_m \subseteq [t]$ into:
    - A set of m nearly orthogonal norm-one vectors.
    - A binary error-correcting code of length t with m codewords and large distance.

    (b) (Mandatory). How many norm-one orthogonal vectors can one put into $\mathbb{R}^d$?

    (c) (Mandatory). How many norm-one ε-almost orthogonal vectors can one put into $\mathbb{R}^d$? Give a lower bound.

    (d) (Bonus). How many norm-one ε-almost orthogonal vectors can one put into $\mathbb{R}^d$? Give an upper bound. Can you reach tight estimations?

12. (Mandatory). Consider the parity function $\mathrm{Parity} : \{0,1\}^\ell \to \{0,1\}$. It is known that for every d, Parity cannot be computed on more than a $\frac{1}{2} + 2^{-\Omega(\ell^{1/d})}$ fraction of the inputs by circuits of depth d and size $2^{O(\ell^{1/d})}$ (you do not need to prove this).

    With that, prove that the class RAC^0 (of constant-depth, polynomial-size circuits that has access to random input bits) is contained in $\bigcup_c \mathsf{DSPACE}(\log^c n)$.

13. (Mandatory). Prove: If there exists an $(\varepsilon = \frac{1}{4})$–PRG $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ against circuits of size s running in time exponential in ℓ then there exists a function f in EXP that is worst-case hard for circuits of size s.

HW 3 – Non-uniform computation and the IKW theorem

Out: 11.12.2016
Due: 30.12.2016

1. (Mandatory). (Luca Trevisan) Let $S(n) \leq \frac{2^n}{n}$. Show a function f on n bits such that
$$f(n) - O(n) \leq \mathsf{Size}(f) \leq f(n).$$
2. (Mandatory). Prove: If $\mathsf{P} = \mathsf{NP}$ then $\mathsf{EXP} \not\subseteq \mathsf{P/poly}$.
3. (Know). Prove: If $\mathsf{NP} \subseteq \mathsf{P/poly}$ then $\mathsf{PH} \subseteq \mathsf{P/poly}$.
4. (Know). Prove that $\mathsf{NP} = \mathsf{P}$ implies $\Sigma_2 = \mathsf{P}$ and $\mathsf{PH} = \mathsf{P}$.
5. (Mandatory). Prove: If $\mathsf{NP} \subseteq \mathsf{BPP}$ then $\mathsf{BPP} = \mathsf{PH}$.
6. (Mandatory). Prove: $\mathsf{MA} \subseteq \mathsf{AM}$.
7. (Know). Prove: Succinct3SAT is NEXP-complete (under polynomial-time reductions).
   Hint: Recall the reduction from NP to SAT.
8. (Mandatory). Prove the following hierarchy theorems:
   (a) For any fixed c, $\mathsf{EXP} \not\subseteq \mathsf{io\text{-}DTIME}(2^{n^c})/n^c$.
   (b) If $\mathsf{NEXP} = \mathsf{EXP}$ then there is a fixed d such that $\mathsf{NTIME}(2^n)/n \subseteq \mathsf{DTIME}(2^{n^d})/n$.
9. (Mandatory). We will prove that if $\mathsf{NEXP} = \mathsf{MA}$ then $\mathsf{NEXP} \subseteq \mathsf{P/poly}$.
   (a) Prove: If $\mathsf{EXP} \not\subseteq \mathsf{P/poly}$ then $\mathsf{MA} \subseteq \mathsf{io\text{-}NTIME}(2^n)$.
   (b) Prove: If $\mathsf{NEXP} = \mathsf{EXP}$ then $\mathsf{NEXP} \not\subseteq \mathsf{io\text{-}NTIME}(2^{n^a})/n$.
   (c) Conclude that if $\mathsf{NEXP} = \mathsf{MA}$ then $\mathsf{NEXP} \subseteq \mathsf{P/poly}$.
10. (a) (Mandatory). Prove that $\mathsf{coNEXP} \subseteq \mathsf{NEXP/poly}$.
    (b) (Bonus). Prove that if $\mathsf{coNP} \subseteq \mathsf{NP/poly}$ then PH collapses to the third level.
11. (Mandatory). What is wrong with the following proof that $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$:
    Define $\Sigma_2\mathsf{EXP}$ the class of languages solvable by $\exists y \forall x \phi(x, y, z)$, where $|y|$,$|z|$,$|\phi(x, y, z)|$ are exponential in the size of $|x|$. Similarly define $\mathsf{PH} - \mathsf{EXP}$.
    - If $\mathsf{EXP} = \mathsf{NEXP}$ then $\mathsf{EXP} = \mathsf{PH} - \mathsf{EXP}$. However, in $\mathsf{PH} - \mathsf{EXP}$ there are languages not in $\mathsf{P/poly}$, hence: $\mathsf{EXP} = \mathsf{NEXP}$ implies $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$.
    - But, $\mathsf{NEXP} \subseteq \mathsf{P/poly}$ implies $\mathsf{NEXP} = \mathsf{EXP} = \mathsf{MA}$ which implies $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$. A contradiction.
    - Thus, we may conclude that $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$.
12. (Mandatory). Prove that for every k, Σ_4 contains a language that does not belong to $\mathsf{SIZE}(n^k)$.
13. (Mandatory). (Arbel Admoni) Prove: If $\mathsf{DTIME}(n^{\log n}) \subseteq \mathsf{NP}$ then $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$.
14. (Mandatory). (Arbel Admoni) Prove: If $\mathsf{NP} = \mathsf{PH}$ then $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$.

HW 4 – Natural proofs, Promise problems, Hierarchies and Counting classes

Out: 2.1.2017
Due: 22.1.2017

1. (Mandatory). Let $H \subseteq F_n$ be the GGM construction with seed length k built using a PRG $G : \{0,1\}^k \to \{0,1\}^{2k}$. Prove that if there exists a distinguisher running in time $2^{O(n)}$ that ε-distinguishes between the uniform distribution over H and the uniform distribution over F_n then there exists a distinguisher running in time $2^{O(n)}$ that $\varepsilon \cdot 2^{-n}$-distinguishes $G(U_k)$ and U_{2k}.

2. (Mandatory). Let $\mathsf{AC}^0[2]$ denote the class of functions computable by a polynomial-size, constant-depth circuits allowing Parity gates.

   - Prove that for any integer d, there exists a family $G_{n,s} \subseteq F_n$, where s is a seed of size polynomial in n, such that every function in $G_{n,s}$ is in $AC_0[2]$ and $G_{n,s}$ looks random for $2^{O(n)}$-size depth-d circuits, i.e., for any polynomial-size (in 2^n) depth d circuit family $C_n : F_n \to \{0,1\}$, $|\Pr[C_n(F_n) = 1] - \Pr[C_n(G_{n,s}) = 1]| < 2^{-\omega(n)}$.
   - Use question 12 from HW2 to prove that there is no lower bound proof which is AC^0-natural and useful against $\mathsf{AC}^0[2]$.

3. (Mandatory). Suppose that the promise problem Π' is Cook-reducible to the promise problem Π and the queries made by the reduction never violate the promise. Then, $\Pi \in \mathsf{Promise\text{-}NP} \cap \mathsf{Promise\text{-}coNP}$ implies $\Pi' \in \mathsf{Promise\text{-}NP} \cap \mathsf{Promise\text{-}coNP}$.

4. (Mandatory). Use the randomness-efficient error amplification to prove that $\mathsf{BPP} \subseteq \mathsf{ZPP}^{\mathsf{NP}}$.

5. (Mandatory). Let $d \geq 1$ be some constant. Prove that if $\mathsf{BPTIME}(n^d) = \mathsf{BPP}$ then

   $$\mathsf{BPTIME}(t(n)) = \mathsf{BPTIME}(t(n)^c)$$

   for every constant $c \geq 1$ and time-constructible function $t(n)$ that satisfies $t(n) \geq n^d$.

6. (Mandatory). Let $t(n)$ and $T(n)$ be time-constructible functions such that there exists a constant k for which $T^{(k)}(t(n)) = 2^{\omega(t(n))}$. Then, $\mathsf{BPTIME}(t(n)) \subsetneq \mathsf{BPTIME}(T(t(n)))$.

7. (Fortnow) (Mandatory). The class GapP is the class of functions f such that for some NP machine M, $f(x)$ is the number of accepting paths minus the number of rejecting paths of M on x. The class FP represent the class of polynomial-time computable functions.

   Prove that for all functions f, the following are equivalent:

   (a) $f \in \mathsf{GapP}$.
   (b) f is the difference of two #P functions.
   (c) f is the difference of a #P function and an FP function.
   (d) f is the difference of an FP function and a #P function.

8. (Fortnow) (Mandatory). Let f be a GapP function and q a polynomial. Prove that the following are GapP functions:

(a) $\sum_{|y| \leq q(|x|)} f(x, y)$.
(b) $\prod_{0 \leq y \leq q(|x|)} f(x, y)$.

References

[1] Adam R Klivans and Dieter Van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31(5):1501–1526, 2002.

[2] Madhu Sudan. Decoding of reed solomon codes beyond the error-correction bound. *Journal of complexity*, 13(1):180–193, 1997.