Kernel-size lower bounds: accompanying exercises

Andrew Drucker*

April 19, 2013

These exercises are intended to aimed at building background to understand recent work on complexity-theoretic evidence for kernel-size lower bounds [BDFH09, FS11, DvM10, Dru12]; in particular, much of the background of [Dru12] is developed in some detail. This includes finite probability, the minimax theorem, and some results from the theory of interactive proofs.

Basic information theory is another central ingredient in [Dru12], and some background is also very helpful (although the necessary facts are all given references in the paper). But to gain proper acquaintance with this powerful and elegant theory, I believe that there is no substitute for working through the introductory portion of a good textbook ([CT06, Chapter 2] is adequate for our purposes, and recommended).

These exercises are meant to accompany a tutorial given at the 2013 Workshop on Kernelization ("Worker"), at the University of Warsaw. I thank the organizers for providing this opportunity.

1 Probability distributions and statistical distance

1.1 Background on probability distributions

In most of complexity theory, and much algorithmic work, it is enough to work with random variables that assume only finitely many possible values. Doing so eliminates the need to use measure theory and makes life easier, so we only develop this fragment of probability theory.

To begin, we review basic notions used in the study of finite probability distributions and fix some notation we'll use. The discussion may seem finicky and cumbersome; isn't finite probability supposed to be easy and intuitive? Yes, and much of the time it can be reasoned about without too much attention to definitions; but tricky situations can arise, particularly when one is *designing* probabilistic experiments for purposes of analysis. In such cases it can be very helpful to have a solid formal understanding.

A *probability distribution* $\mathcal{D}$ over a finite set U is just a mapping $\mathcal{D} : U \to [0,1]$ satisfying $\sum_{u \in U} \mathcal{D}(u) = 1$. (Note that U here may be a finite set of real numbers, or any other finite set.) We define the *support* of $\mathcal{D}$ as $\{u : \mathcal{D}(u) > 0\}$, and for $A \subseteq U$ we write

$$\mathcal{D}(A) \ := \ \sum_{u \in A} \mathcal{D}(u) \ .$$

*Institute for Advanced Study, Princeton, NJ. Email: andy.drucker@gmail.com. Preparation of this teaching material was supported by the National Science Foundation under agreements Princeton University Prime Award No. CCF-0832797 and Sub-contract No. 00001583. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

It is often important to argue that two distributions are "similar" or "different." The most natural way to study this is by considering the *statistical distance* between two distributions $\mathcal{D}, \mathcal{D}'$ over U, defined as

$$||\mathcal{D} - \mathcal{D}'||_{\text{stat}} \ := \ \frac{1}{2} \sum_{u \in U} |\mathcal{D}(u) - \mathcal{D}'(u)| \ .$$

The set of distributions over U form a metric space under this distance measure.

When we are analyzing a probabilistic experiment where multiple quantities of interest are being studied, it is standard to take a particular set U and distribution $\mathcal{D}$ over U as being the fundamental objects defining the experiment. When they play these fundamental roles, U is often called the *universe* and $\mathcal{D}$, the *probability measure* associated to U, and together $(U, \mathcal{D})$ form a *(finite) probability space*. The elements $u \in U$ are then called the *atoms* of U, and an *event* is a subset $A \subseteq U$.

Probability spaces allow us to define *random variables:* a random variable X associated with the probability space $(U, \mathcal{D})$ is a function $X : U \to S$, for some second finite set S, whose elements are often referred to as *outcomes* of X. For $S' \subseteq S$, we define the event $[X \in S']$ as the set of atoms

$$\{u \in U : X(u) \in S'\} \ .$$

The probability associated with this event is $\Pr[X \in S'] := \sum_{u : X(u) \in S'} \mathcal{D}(u)$. If $X^1, \ldots, X^t$ are all random variables on $(U, \mathcal{D})$, with X^i mapping to a set S^i, we can define the *joint random variable* $(X^1, \ldots, X^t) : U \to S_1 \times \ldots \times S_t$ by $(X^1, \ldots, X^t)(u) := (X^1(u), \ldots, X^t(u))$. Then we can define events such as $[X^1 = X^2]$ in the natural way as a subset of outcomes of the joint random variable, and measure their probability according to the previous definition.

Intuitively, a probability space $(U, \mathcal{D})$ is meant to describe all possible outcomes of some probabilistic process, assigning probabilities to each; an atom $u \in U$ captures "everything that matters to us" in a particular outcome. The random variables associated with our probability space are then interpreted as describing *particular* features of the outcome.

Example 1. *Consider the experiment of tossing two six-sided dice—one red, one black. The underlying probability space may be modeled as having universe $U = [6] \times [6]$, where the first coordinate of an atom $u = (a, b)$ gives the red die's outcome.*

If one is playing Monopoly or many other games, the two dice are regarded as identical, and the relevant information in an outcome is given by the random variable Rolls*, that maps the ordered pair (a, b) to the unordered multiset $\{a, b\}$. Another relevant random variable is the mapping* Sum*, which sends (a, b) to the value $a + b \in \{2, 3, \ldots, 12\}$. Then, for example, we can explicitly write out the events*

$$[\text{Rolls} = \{3, 4\}] \ = \ \{(3, 4), \ (4, 3)\} \ , \qquad [\text{Sum} = 4] \ = \ \{(1, 3), \ (2, 2), \ (3, 1)\}$$

as subsets of U. These events have probabilities $2/36$ and $3/36$ respectively if the underlying distribution $\mathcal{D}$ is taken as uniform over $[6] \times [6]$ (fair dice).

Note that for these variables, we can determine the outcome of Sum *by that of* Rolls*. This property can be useful for analysis purposes.*

Now, a random variable $X : U \to S$ is defined with respect to the underlying measure $\mathcal{D}$ on the probability space; but X also has its own *governing distribution* $\mathcal{D}_X$ over S, namely

$$\mathcal{D}_X(x) \ := \ \Pr[X = x] \ , \qquad x \in S \ .$$

It is often necessary to estimate the statistical distance between the governing distributions of two random variables taking outcomes over the same set X. It is standard to overload notation, using

$$||X - X'||_{\text{stat}}$$

to denote the statistical distance between the governing distributions of two random variables X, X'. That is, we let $||X - X'||_{\text{stat}} := ||\mathcal{D}_X - \mathcal{D}_{X'}||_{\text{stat}}$. We use $X \sim \mathcal{D}_0$ to denote that X has governing distribution equal to $\mathcal{D}_0$.

If the random variable X is *real-valued*, i.e., if the image of the mapping X is a subset S of real numbers, define the *expectation* or *expected value* of X as

$$\mathbb{E}[X] \ := \ \sum_u \mathcal{D}(u) \cdot X(u) \ = \ \sum_{x \in S} x \cdot \Pr[X = x] \ .$$

For a real-valued random variable X, we can also form derived random variables such as X^2, $2X+5$, etc. in the natural way. The k^{th} *moments* $\mathbb{E}[X^k]$ are particularly important quantities for analyzing the behavior of a real-valued random variable X.

A final, vitally important notion is that of *conditioning* on an event. Given a probability space $(U, \mathcal{D})$ and an event $A \subseteq U$, we let the *conditional probability space* defined by A have universe U and probability measure $\mathcal{D}|_A$ given by

$$\mathcal{D}|_A(u) \ := \ \begin{cases} \frac{\mathcal{D}(u)}{\mathcal{D}(A)} & \text{if } u \in A, \\ 0 & \text{otherwise.} \end{cases}$$

Then the *conditional probability* of an event B, denoted $\Pr[B|A]$, is defined as $\mathcal{D}|_A(B)$. A random variable X on $(U, \mathcal{D})$ induces a *conditional* random variable $X|_A$, defined by the same mapping $X : U \to S$, but with respect to the new measure $\mathcal{D}|_A$.

To give a brief illustration, suppose we return to the scenario of Example 1 and consider conditioning on the event $A = [\text{Sum} = 4]$. Then the conditional random variable $\text{Rolls}|_A$ has $2/3$ probability mass on outcome $\{1, 3\}$, and the remaining $1/3$ on $\{2, 2\}$. Usually in discussions this sort of conditioning notation is suppressed; one might simply say that "conditioned on A, Rolls has $2/3$ probability of equaling $\{1, 3\}$."

Two random variables X, Y are *independent* if conditioning on any outcome of X does not change the governing distribution of Y, and vice versa. A *collection* $X_1, \ldots, X_t$ of random variables is called independent if for every $j \in [t]$, X_j is independent from $(X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_t)$. The collection is *pairwise-independent* if each pair $X_j, X_{j'}$ with $j \neq j'$ are independent. (This is a weaker condition.)

1.2 Cheat sheet: useful probabilistic inequalities

The following is a "who's-who" of basic inequalities that get used over and over in theoretical computer science.

1. **Union bound:** If $A_1, \ldots, A_t$ are events over probability space $(U, \mathcal{D})$, then

$$\Pr[A_1 \cup \ldots \cup A_t] \ \leq \ \sum_j \Pr[A_j] \ .$$

2. **Linearity of expectation:** if $X_1, \ldots, X_t$ are real-valued random variables over a shared probability space, and X is their sum, then

$$\mathbb{E}[X] \;=\; \sum_j \mathbb{E}[X_j] \;.$$

3. **Markov's inequality:** if X is a nonnegative real-valued random variable and $c > 0$, then

$$\Pr[X \geq c] \;\leq\; \frac{\mathbb{E}[X]}{c} \;.$$

4. **Chebyshev's inequality:** if X is any real-valued random variable and $c > 0$, then

$$\Pr[|X - \mathbb{E}[X]| \geq c] \;\leq\; \frac{\mathbb{E}[(X - \mathbb{E}[X])^2]}{c^2} \;=\; \frac{\mathbb{E}[X^2] - \mathbb{E}[X]^2}{c^2}.$$

   This is particularly useful when X is a sum of pairwise-independent random variables $X_1, \ldots, X_t$. Pairwise-independence implies that $\mathbb{E}[X_j X_{j'}] = \mathbb{E}[X_j] \cdot \mathbb{E}[X_{j'}]$ for $j \neq j'$; this helps us to upper-bound $\mathbb{E}[X^2]$. Both Chebyshev's and Markov's inequalities follow directly from the definition of expectation.

5. **Chernoff/Hoeffding inequalities:** Let $X_1, \ldots, X_t$ be a collection of independent random variables with outcomes in the range $[0, 1]$, each with $\mathbb{E}[X_j] = p$. If X is their sum, then for $\varepsilon > 0$,

$$\Pr[X \geq (1+\varepsilon)pt] \;\leq\; e^{-\frac{\varepsilon^2 pt}{3}} \;, \qquad \Pr[X \leq (1-\varepsilon)pt] \;\leq\; e^{-\frac{\varepsilon^2 pt}{2}} \;. \tag{1}$$

   Also,

$$\Pr[X \geq (p+\varepsilon)t] \;\leq\; e^{-2\varepsilon^2 t} \;, \qquad \Pr[X \leq (p-\varepsilon)t] \;\leq\; e^{-2\varepsilon^2 t} \;. \tag{2}$$

Eqs. (1) and (2) above are referred to as the "multiplicative-error" and "additive-error" forms, respectively, of the Chernoff-Hoeffding bounds. The multiplicative-error estimates become stronger when p is close to 0. The bounds above are not tightest-possible, but are chosen to have especially simple form and are often adequate. The Chernoff-Hoeffding bounds can be generalized in several directions.

1.3 Exercises on probability distributions

The first three exercises give three very useful perspectives on our distance measure for distributions. The next one is for fun; the last three are basic facts related to certain steps in [Dru12].

1. Prove the following alternative characterization of statistical distance:

$$||\mathcal{D} - \mathcal{D}'||_{\text{stat}} \;=\; \max_{A \subseteq U} |\mathcal{D}(A) - \mathcal{D}'(A)| \;.$$

2. **Distinguishing experiments:** Let $\mathcal{D}, \mathcal{D}'$ be two distributions over a finite set V, and that Alice and Bob each have descriptions of these distributions. Suppose Alice flips a fair coin, unseen by Bob. If "Heads," she privately samples a value $v \in V$ according to $\mathcal{D}$. If "Tails," she privately samples v according to $\mathcal{D}'$. In either case, she sends the resulting value to Bob.

(a) Under the formal framework provided above, describe a probability space with associated random variables that express what is going on in this experiment. There should be a random variable corresponding to what Bob sees.

(b) Bob's goal is to guess which distribution Alice sampled from. Describe his optimal strategy, and prove that his success probability under this optimal strategy is precisely

$$\frac{1}{2}(1 + ||\mathcal{D} - \mathcal{D}'||_{\text{stat}}) \ .$$

3. **The coupling method:** Let X, Y be random variables over a shared probability space. Suppose that $\Pr[X = Y] \geq 1 - \gamma$. Prove the easy fact that $||X - Y|| \leq \gamma$.

   Next, suppose that there are two *distributions* $\mathcal{D}_1, \mathcal{D}_2$ that we wish to show are at statistical distance at most γ. In applications, this is often difficult to show by direct computation. By our previous observation, it would be sufficient to *construct* a probability space with random variables X, Y, such that $\mathcal{D}_X = \mathcal{D}_1, \mathcal{D}_Y = \mathcal{D}_2$, and $\Pr[X = Y] \geq 1 - \gamma$. You are asked to show that, in principle, this method can always succeed: if $||\mathcal{D}_1 - \mathcal{D}_2|| \leq \gamma$ then X, Y as above can be found.

   (Any "realization" of two distributions $\mathcal{D}_1, \mathcal{D}_2$ as governing distributions over a shared probability space is referred to as a *coupling* of those variables. A coupling X, Y for $\mathcal{D}_1, \mathcal{D}_2$ is called *optimal* if it maximizes the "collision probability" $\Pr[X = Y]$. You are being asked to relate the statistical distance in an exact way to the collision probability of an optimal coupling.)

4. This exercise is not relevant to kernel-size lower bounds; instead it is a fun challenge problem that illustrates the power of the coupling method for the analysis of random walks. The result described here is well-known and often used as an introductory example. Consider an ant taking a "lazy random walk" on the hypercube $V = \{0, 1\}^n$, where at each step the ant remains stationary with probability .5, and otherwise walks to a randomly selected neighboring vertex. You are asked to show that regardless of the starting point, after a sufficiently large $t = O(n \log n)$ steps, the governing distribution of the ant's position at time t is close to the uniform distribution, say, at statistical distance at most .01 from uniform.

   Hint: choose any two starting vertices x, y. Exhibit a process that randomly generates a pair of t-step walks starting from x and y, so that (i) the two walks are *individually* distributed as lazy random walks, and yet (ii) with high probability, the two walks are at the same vertex on the t^{th} step. That is, we want to "couple" the t^{th} steps of these walks. Finally, having exhibited such a construction, what does it imply?

5. **Data processing cannot increase statistical distance:** Suppose X and Y are two random variables both taking values in the set S, and that $f : S \to T$ is some function. Show that for the random variables $f(X), f(Y)$ we have

$$||f(X) - f(Y)||_{\text{stat}} \ \leq \ ||X - Y||_{\text{stat}} \ .$$

   Next, formalize in our language of probability spaces what it should mean for F to be a *randomized function* of its input, so that we obtain the extension

$$||F(X) - F(Y)||_{\text{stat}} \ \leq \ ||X - Y||_{\text{stat}} \ .$$

   (Hint: F's randomness should not be "mixed up" with the randomness in its input, or else the inequality could fail.)

6. Let V be a finite set; let $\{\mathcal{R}_v\}_{v \in V}$ and $\{\mathcal{R}'_v\}_{v \in V}$ be two families of probability distributions, each over some set U; and let $\mathbf{v}$ be a random variable taking values over V. Let $\mathfrak{R}$ be a random variable which observes $\mathbf{v}$ and then outputs a sample according to $\mathcal{R}_{\mathbf{v}}$ (but does not output the value $\mathbf{v}$). Let $\mathfrak{R}'$ be defined analogously for $\{\mathcal{R}'_v\}$.[1] Show that

$$||\mathfrak{R} - \mathfrak{R}'||_{\text{stat}} \leq \sum_v \Pr[\mathbf{v} = v] \cdot ||\mathcal{R}_v - \mathcal{R}'_v||_{\text{stat}} .$$

(Hint: design a pair of random variables whose statistical distance is given by the right-hand side above; then apply the previous exercise.)

2 The minimax theorem and its applications

1. This exercise presumes some familiarity with linear programming. Derive the minimax theorem for 2-player, simultaneous-move, zero-sum games from the *strong LP duality theorem*, which asserts that a linear program has optimal value equal to that of its dual LP.

2. Let $\mathbf{G}$ be a 2-player, simultaneous-move, zero-sum game with payoffs to the Player 2, always in the range $[0, 1]$. Let Y, the set of pure strategies for Player 2, be a set of size 2^n. Prove that if optimal play gives an expected payoff of $\alpha \in [0, 1]$ to Player 2, then for all $\varepsilon > 0$ there is a a Player 1 strategy $\mathcal{D}_{\mathsf{sparse}}$ that is a distribution over at most $\text{poly}(n, 1/\varepsilon)$ pure strategies, and that forces Player 2's expected payoff to be at most $\alpha - \varepsilon$. Try to get a reasonable bound; you will want to use an inequality from our "cheat sheet."

   This fact, due to Lipton and Young [LY94] and Althofer [Alt94], is crucial to many complexity applications of the minimax theorem.

3. [FPS08] Here is one interesting complexity application of minimax. Suppose L is a language that does not have polynomial-sized Boolean circuits. Prove that for any $k > 0$, there are infinitely many values n such that there is a set $X_n \subset \{0, 1\}^n$ of size at most $n^{k+O(1)}$, such that: *every* Boolean circuit of size n^k on n input bits fails to compute $L(x)$ correctly on at least one input from X_n.

   Hint: use the minimax theorem and the Lipton-Young-Althofer "sparsification principle." The original proof does not go through minimax, but uses a stage-based process that is structurally similar. This is analogous to the way Fortnow and Santhanam's result can be proved with or without minimax (the minimax version is sketched in our slides).

3 The Fortnow-Santhanam lower bound for OR-compression

These exercises presume basic familiarity with [FS11], and aim to build the reader's understanding of certain points.

In the OR(SAT) problem, we are given formulas $\psi^1, \ldots, \psi^t$, and asked whether at least one of them is satisfiable. As the parameter k we take $k := \max_j |\langle \psi^j \rangle|$ as the maximum description length of any ψ^j. Fortnow and Santhanam [FS11] show that OR(SAT) does not have (deterministic) polynomial kernels, *unless* $\mathsf{NP} \subset \mathsf{coNP/poly}$.

[1] You may wish to define an explicit probability space here.

3.1 Exercises

1. Explain why (as observed in [FS11]) the Fortnow-Santhanam lower bound technique directly applies to rule out *randomized* polynomial kernelization reductions for OR(SAT), as long as the reduction "avoids false negatives."

   (That is, we require that if the input instance to the kernelization reduction is a "Yes"-instance, the output of the kernelization instance is always a "Yes"-instance; but if the input is a "No"-instance, we allow a .1 probability of outputting a "Yes" instance.)

   Also explain what goes wrong if we allow two-sided error, i.e., if we allow both false positives and false negatives in the reduction.

2. Fortnow and Santhanam's method rules out deterministic polynomial kernelization reductions for OR(SAT). It is not hard to see that it also rules out *non-uniform* polynomial kernels, i.e., ones computable in $\mathsf{P/poly}$.

   Now suppose we had a *probabilistic* polynomial kernelization for OR(SAT), with two-sided error. Why can't we just non-uniformly derandomize it to get a deterministic polynomial kernel? After all, in $\mathsf{P/poly}$ we can easily derandomize BPP algorithms, so why can't we just follow along the same lines? What goes wrong?

   You are asked to recall (or rediscover) the construction showing that BPP is contained in $\mathsf{P/poly}$, to propose a natural analogous construction for derandomizing a two-sided error kernelization, and explain why it fails to have the desired properties.

4 Interactive proof systems

Studying the power of nondeterminism has of course been central to the theory of computation since the discovery of NP-completeness. Nondeterminism can be viewed as a source of advice from a *powerful but untrusted prover*; this advice must be checked by a skeptical polynomial-time verifier.

One of the most important themes in complexity theory since the '80s is that adding *randomization* to interactions with a prover can make proof systems more powerful and expressive. (See [Bab85, BM88, GS86] for important early contributions; a full survey is out of scope here.) Results from this theory are important ingredients in [Dru12].

The class MA, or *Merlin-Arthur*, consists of those languages L such that there exists a polynomial-time *verifier* algorithm $V(x, y, r)$, such that the following conditions hold:

1. For $x \in \{0,1\}^n$ the algorithm expects auxiliary input strings y, r of lengths $\ell(n), m(n)$ determined by n, of lengths polynomially bounded in n (here, y and r are typically called the *proof string* and *random string* respectively);

2. If $x \in L$,
$$\exists\ y\ :\ \text{for at least } 2/3 \text{ of all } r \in \{0,1\}^{m(n)}, \quad V(x,y,r)\ =\ 1\ ;$$

3. If $x \notin L$,
$$\forall\ y\ :\ \text{for at least } 2/3 \text{ of all } r \in \{0,1\}^{m(n)}, \quad V(x,y,r)\ =\ 0\ .$$

The class AM, or *Arthur-Merlin*, is defined similarly to MA except that we modify conditions 2, 3 as follows:

2'. If $x \in L$,

$$\text{for at least } 2/3 \text{ of all } r \in \{0,1\}^{m(n)}, \exists\ y\ :\quad V(x,y,r)\ =\ 1\ ;$$

3'. If $x \notin L$,

$$\text{for at least } 2/3 \text{ of all } r \in \{0,1\}^{m(n)}, \forall\ y\ :\quad V(x,y,r)\ =\ 0\ .$$

We typically write an AM verifier algorithm as $V(x,r,y)$ to reflect this change in quantification order.

The interpretation is as follows. In an MA protocol, Merlin (the untrusted prover) wants to prove that $x \in L$. To do so, he sends a *proof string* y to Arthur (the skeptical verifier), which Arthur checks probabilistically by randomly generating r and then accepting exactly if $V(x,y,r) = 1$. The correctness requirements 2-3 above say that Merlin should be able to succeed in this task with high probability exactly if $x \in L$ as claimed.

In an MA protocol, the story is similar, except that Arthur chooses a random string first, which he shows to Merlin. (This is a *public-coin* model of interaction, in which Arthur shows all of his random choices to Merlin.)

Note an asymmetry in our definitions: neither type of proof system above gives any obvious way for Merlin to prove that a string x is *not* in L.

It is known [ZF87] that for both MA and AM, we get the same class if we insist on *perfect completeness* in our proof system: that is, if $x \in L$ then there is a Merlin strategy causing Arthur to accept with probability 1. These are fun facts to try to prove, although they are not necessary for [Dru12].

4.1 Exercises

1. Prove the "non-uniform derandomization" result $\mathsf{AM} \subset \mathsf{NP/poly}$. Hint: use random sampling.

   Combining this with the easy result $\mathsf{NP} \subseteq \mathsf{AM}$, conclude that coNP is not contained in AM unless $\mathsf{NP} \subset \mathsf{coNP/poly}$ (this would collapse the Polynomial Hierarchy to Σ_3^p [Yap83]). Thus there is probably no general-purpose way to transform an AM protocol for L into an AM protocol for $\overline{L}$.

2. Show that in either of MA, AM, the "completeness" and "soundness" thresholds $2/3, 1/3$ can be replaced with any other constants c, s with $1 > c > s > 0$; or, more powerfully, even with values $(1 - 2^{-p(n)}, 2^{-p(n)})$ for any polynomial $p(n)$, where $n = |x|$.

3. It was shown in [Bab85] (see also [BM88]) that $\mathsf{MA} \subseteq \mathsf{AM}$. Prove this result. Hint: if a proof string y is a very good choice for Merlin in a MA proof system on input x, then it should work well against *multiple* random strings chosen by Arthur.

   It is open whether these two classes are equal, but is considered likely that both collapse to NP.

4. Working by analogy, define the classes AMA, MAM of languages definable by interactive proofs which involve *three* messages being sent between Arthur and Merlin. (The two classes differ only on who sends the first and last messages.)

   Using the idea from exercise 3, prove that $\mathsf{AMA} = \mathsf{MAM} = \mathsf{AM}$. (More generally, one can show by the same technique that any constant number $k \geq 2$ of rounds of interaction gives the same expressive power as AM [Bab85].)

5. The remaining exercises sketch an important protocol that is used as a building block in designing many interactive proofs: the *Goldwasser-Sipser set-size lower bound* protocol [GS86].

   First, we need a combinatorial definition. Let U be a finite set, and let $\mathcal{H}$ be a finite family of "hash functions" $h : U \to \mathbb{F}_2^k$. We say that $\mathcal{H}$ is a *strongly universal hash family* if, for all $u, u' \in U$ with $u \neq v$, and for all $z, z' \in \mathbb{F}_2^k$, we have

$$\Pr_{h \in_r \mathcal{H}}[h(u) = z \ \wedge \ h(u') = z'] \ = \ 2^{-2k} \ .$$

   In other words, if we restrict our attention to any two particular input values u, u' and select a random hash function from our family, then its distribution is exactly that of a truly random function on those two inputs. This "limited independence" property makes a randomly-selected hash function nearly as good as a truly-random function for many applications.

   The great virtue of selecting a function from a strongly-universal hash family instead of a truly-random function is that the former can be much more succinctly described. The following useful explicit construction provides a good example. Given $k, \ell > 0$, consider $U := \mathbb{F}_2^\ell$, and define the family of functions

$$\mathcal{H}^{\ell,k} \ := \ \left\{ \ h_{A,v} : \mathbb{F}_2^\ell \to \mathbb{F}_2^k \ \right\}_{A \in \mathbb{F}_2^{k \times \ell}, v \in \mathbb{F}_2^k}$$

   given by

$$h_{A,v}(x) \ := \ Ax + v \qquad \text{(with addition over } \mathbb{F}_2^k) \ .$$

   You are asked to prove that $\mathcal{H}^{\ell,k}$ is a strongly universal hash family. Note that this construction is completely explicit even when ℓ, k are large values.

6. Suppose that $k, \ell > 0$, that $U' \subseteq U = \mathbb{F}_2^\ell$, and that $|U'| = \theta \cdot 2^k$, for some $\theta > 0$. Say we select a random hash function from $\mathcal{H}^{\ell,k}$ as defined previously; then,

$$1 - \theta^{-2} \leq \ \Pr[\ 0^k \in h_{A,v}(U') \ ] \ \leq \ \theta \ .$$

   Hint: we are just using the strongly-universal property of our hash family. For the lower bound, use Chebyshev's inequality.

7. Suppose B is a polynomial-time decidable language, and for $n > 0$ let

$$\beta_n \ := \ \frac{|B \cap \{0,1\}^n|}{2^n} \ .$$

   Assume that for every $n > 0$, we have *either* $\beta_n \geq .63$, *or* $\beta_n \leq .61$.

   Give an AM protocol for Merlin to prove that we are in the first case. That is, prove that the language

$$L \ = \ \{1^n \ : \ \beta_n \geq .63\}$$

   is in AM. Hint: you will want to perform a random experiment as in the previous exercise. However, you will need a way to boost our confidence in the outcome of this experiment.

   It is not hard to significantly generalize the result of this exercise. For instance, if we replaced the values $(.63, .61)$ in our assumption with $(.61 + n^{-100}, .61)$, we could obtain the same conclusion.

8. Following [GMR89], consider a *private-coin variant* of AM proof systems, in which Arthur hides the outcome his random string r from Merlin, and sends only some *challenge string* w that is determined as $w = F(x, r)$ for some polynomial-time computable F.

   Formally define the class of languages definable by such protocols; this class is usually denoted IP[2]. Goldwasser and Sipser [GS86] used the set-size lower bound in a clever way to prove that IP[2] is equal to AM; that is, private coins don't increase expressive power. They proved this holds more generally for any finite number of rounds of interaction.

   There are two steps in their proof. First, they demonstrate how to convert private-coin protocols to public ones, at the cost of increasing the number of rounds of interaction by 2. Second, they use the previously-mentioned result of [Bab85] to reduce the rounds of interaction and obtain an AM protocol. This is not an easy result, but you are encouraged to think about it or read the paper.

9. A *promise problem* is a pair (Π_Y, Π_N) of disjoint subsets of $\{0,1\}^n$ (the "yes" and "no" instances). Any such pair defines a computational problem: we are given a string that is *promised* to lie in one of these two sets, and must decide which one.

   Most (all?) complexity classes have natural promise-problem analogues. Please define, for example, the classes promise-AM and promise-IP[2]. The result of [GS86] carries over to promise classes: these two classes are equal.

10. Given a Boolean circuit C with $m \geq 1$ output bits in designated order, let $\mathcal{D}_C$ denote the output distribution of C when C is fed uniformly random inputs.

    Define the *statistical distance problem* $\mathsf{SD}^{\geq .9}_{\leq .1}$ as the following promise problem:

    - **Input:** a description $\langle C, C' \rangle$ of a pair of circuits with the same number of output gates;
    - Π_Y equals
    $$\{ \langle C, C' \rangle \ : ||\mathcal{D}_C - \mathcal{D}_{C'}||_{\text{stat}} \ \geq \ .9 \ \} \ ;$$
    - Π_N equals
    $$\{ \langle C, C' \rangle \ : ||\mathcal{D}_C - \mathcal{D}_{C'}||_{\text{stat}} \ \leq \ .1 \ \} \ .$$

    Prove that $\mathsf{SD}^{\geq .9}_{\leq .1}$ lies in promise-IP[2].

    Hint: if C, C' define distributions that are far apart, Merlin should be able to distinguish one from the other. Test his ability to do so.

    By [GS86], it follows that this problem is in promise-AM. Then, by non-uniform derandomization, it follows that this problem is in the promise version of NP/poly. This is a key fact for [Dru12]. The values b, a here used to define $\mathsf{SD}^{\geq b}_{\leq a}$ in this result are fairly arbitary; all that matters is that $b - a \geq \frac{1}{\text{poly}(n)}$ where $n = |\langle C, C' \rangle|$.

11. The problem $\mathsf{SD}^{\leq a}_{\geq b}$ is defined similarly by reversing the "yes" and "no" cases:

    - **Input:** a description $\langle C, C' \rangle$ of a pair of circuits with the same number of output gates;
    - Π_Y equals
    $$\{ \langle C, C' \rangle \ : ||\mathcal{D}_C - \mathcal{D}_{C'}||_{\text{stat}} \ \leq \ a \ \} \ ;$$

- Π_N equals
$$\{\langle C, C' \rangle \ : ||\mathcal{D}_C - \mathcal{D}_{C'}||_{\text{stat}} \ \geq \ b \ \} \ .$$

This problem is *also* known to lie in promise-AM [For87, SV03], under more restrictive conditions: we require that $b^2 - a \geq \frac{1}{\text{poly}(n)}$. This more difficult result is an important ingredient in the results of [Dru12] on probabilistic OR-compression.

The usefulness of promise problems (in the study of ordinary decision problems) is as a target for *reductions*. For instance, if you are familiar with the PCP Theorem, you will see that it can be viewed as giving a polynomial-time reduction from an arbitrary instance of an NP problem to an equivalent instance of a certain problem in promise-NP. The problems $\mathsf{SD}^{\geq b}_{\leq a}$, $\mathsf{SD}^{\leq a}_{\geq b}$ play a similar role in [Dru12]; one difference is that the reductions given to these promise problems are non-uniform.

As the exercise, you are just asked to check that this reduction scheme can imply upper bounds for the decision problems being studied. Suppose, for example, that there is a polynomial-time computable reduction from L to the promise problem (Π_Y, Π_N)—where "yes" instances map to "yes" instances and vice versa. Suppose also that (Π_Y, Π_N) is in promise-AM. Prove that then we have $L \in \mathsf{AM}$.

References

[Alt94] Ingo Althöfer. On sparse approximations to randomized strategies and convex combinations. *Linear Algebra and its Applications*, 199, Supplement 1(0):339 – 355, 1994.

[Bab85] László Babai. Trading group theory for randomness. In Robert Sedgewick, editor, *STOC*, pages 421–429. ACM, 1985.

[BDFH09] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009. Earlier version in ICALP '08.

[BM88] László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

[Dru12] Andrew Drucker. New limits to classical and quantum instance compression. In *53rd IEEE FOCS*, pages 609–618, 2012.

[DvM10] Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *42nd ACM STOC*, pages 251–260, 2010.

[For87] Lance Fortnow. The complexity of perfect zero-knowledge (extended abstract). In Alfred V. Aho, editor, *19th ACM STOC*, pages 204–209, 1987.

[FPS08] Lance Fortnow, Aduri Pavan, and Samik Sengupta. Proving SAT does not have small circuits with an application to the two queries problem. *J. Comput. Syst. Sci.*, 74(3):358–363, 2008.

[FS11] Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[GS86] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *18th ACM STOC*, pages 59–68, 1986.

[LY94] Richard J. Lipton and Neal E. Young. Simple strategies for large zero-sum games with applications to complexity theory. In *26th ACM STOC*, pages 734–740, 1994.

[SV03] Amit Sahai and Salil P. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2):196–249, 2003.

[Yap83] Chee-Keng Yap. Some consequences of non-uniform conditions on uniform classes. *Theor. Comput. Sci.*, 26:287–300, 1983.

[ZF87] Stathis Zachos and Martin Furer. Probabalistic quantifiers vs. distrustful adversaries. In *FSTTCS*, pages 443–455, 1987.