On the reduction theory for average case complexity[1]

Andreas Blass[2] and Yuri Gurevich[3]

Abstract. This is an attempt to simplify and justify the notions of deterministic and randomized reductions, an attempt to derive these notions from (more or less) first principles.

1. Introduction

Let us review the notion of a decision problem. First of all, one has a set – the set of *instances, the universe of the decision problem. For simplicity, we stick to the Turing machine model and suppose that the universe is always a set of strings in some finite alphabet. Of course, objects of interest are not necessarily strings. They may be graphs for example. But they should be coded as strings.*

Second, the instances of a given decision problem split into positive and negative; this can be formalized by means of a characteristic function from the universe to $\{0,1\}$. *But this is not all. In order to discuss e.g. polynomial time algorithms, we need that instances have sizes. Ordinarily, the size of a string is its length [GJ], but this isn't always convenient. For example, if objects of interest are graphs then one may prefer to define the size of the encoding string as the number of vertices in the graph rather than the length of the encoding string.*

Definition. *A size function on a set* U *is a function from* U *to natural numbers. A size function* $x \mapsto |x|$ *is conventional if the set* $\{x : |x| \leq n\}$ *is finite for each* n.

Typically, the size of an instance is polynomially related to the length, but unconventional size functions turn out to be useful as well. In some situations, it may be convenient to allow non-integer sizes; for simplicity, we shall stick here to integer sizes. The notation $|x|$ *will be used to denote the size of an element* x.

[1]Springer Lecture Notes in Computer Science 533, 1991, 17–30.

[2]Partially supported by NSF grant DMR 88-01988. Address: Mathematics Department, University of Michigan, Ann Arbor, MI 48109-1003, USA; andreas_blass@ub.cc.umich.edu

[3]Partially supported by NSF grant CCR 89-04728. Address: Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI 48109-2122, USA; gurevich@eecs.umich.edu

In the rest of this paper, a decision problem is a set of strings (not necessarily the set of all strings) in some finite alphabet together with a size function and a characteristic function.

In the average case approach, pioneered by Leonid Levin, a decision (resp. search) problem is considered together with a probability distribution on instances. Such a pair is called a randomized, or distributional, decision (resp. search) problem. For simplicity, we restrict attention to randomized decision problems. In order to obtain completeness results, Levin defined reductions among randomized decision problems. His notion of reduction seems ad hoc. In the first part of this paper (Sections 2–4), we attempt to derive a simpler version of it from (more or less) first principles.

Even though that original notion of reduction, deterministic in nature, was sufficient to establish the completeness of a number of natural problems [Le, Gu1, Gu2], it turned out to be too restrictive. Many randomized decision problems of interest are flat in the following technical sense: There exists $\varepsilon > 0$ such that the probability of any instance of sufficiently large size n is bounded from above by 2^{-n^ε}. However, no randomized decision problem with a flat domain is complete unless deterministic exponential time equals nondeterministic exponential time [Gu1]. To overcome this difficulty, Levin suggested more general randomizing reduction. Versions of randomizing reduction were defined and successfully used in [VL] and [BCGL]. A simple version of randomizing reduction was defined in greater detail and proved transitive in [Gu1]. That version was too simple however, and in the second part of the paper (Section 5–6) we attempt to justify a more general version of randomizing reductions.

Remark. *The randomizing reductions of Section 5 can be further generalized. Our goal is not the most general notion of reduction, but rather a simpler notion of reduction sufficient for most applications.*

2. The notion of a domain

Definition [Gu2]. *A domain D is a nonempty set U_D (the universe of D) of strings in some finite alphabet with a size function $|x|$ (or $|x|_D$) and a probability distribution $\boldsymbol{P}_D$ such that there are only finitely many elements of positive probability of any given size.*

Since the elements of a domain are strings, there is a well-defined notion of computability for functions on domains. The requirement that there are only finitely many elements of positive probability of any given size will be relevant in Section 5 where we deal with unconventional size functions.

Remark. *One could work at a more abstract level, where the universes of domains have some prescribed (or assumed) notions of computability and computation time subject to suitable axioms. For example, the size of an output is bounded by the computation time, composing two computable functions yields a computable function, and the computation time is additive for composition. We shall stick here to the simpler definition of domains given above which allows us to use the standard Turing*

machine model. The particular axioms mentioned above are assumed to be true in that model.

It is natural to say that a function T from a set with a size function to nonnegative reals is polynomially bounded if $T(x)$ is bounded by a polynomial of $|x|$.

Definition. *A function T from a domain D to the set $\bar{\mathcal{R}}^+$ of nonnegative reals augmented with ∞ is AP (or polynomial on average or polynomially bounded on average) with respect to D if, for some $\varepsilon > 0$, $\sum_x (Tx)^\varepsilon |x|^{-1} \boldsymbol{P}_D(x) < \infty$. We will say that T is linear on average if the witness ε can be chosen to be 1.*

Here x ranges over elements of positive size; to avoid the nuisance of dealing with elements of size 0, we restrict attention to domains without elements of size 0 in the rest of the paper. One can, without loss of generality, restrict x to range over elements satisfying $(Tx)^\varepsilon > x$. This is because the omitted terms sum to at most $\sum_x \boldsymbol{P}_D(x) = 1$. The probability $\boldsymbol{P}_D(x)$ of an element x is of course the probability $\boldsymbol{P}_D[\{x\}]$ of the set (or event) $\{x\}$.

The notion of polynomiality on average is due to Levin [Le]; it is motivated and discussed in [Jo, Gu1, BCGL, Gu2]. We have generalized this definition slightly by allowing ∞ as a possible value of the function T in question. In the case when T is the computation time of some algorithm, the infinite value allows us to consider algorithms that may diverge at some inputs: If the algorithm diverges at a point x then $T(x) = \infty$. We suppose that $\infty \cdot 0 = 0$. If the algorithm diverges at some set of probability zero, the computation time still may be polynomial on average. Because we required that there are only finitely many elements of positive probability of any given size n, we have that, for each n, $\sum_{x,|x|=n} (Tx)^\varepsilon |x|^{-1} \boldsymbol{P}_D(x) < \infty$ provided that the probability of the event $T(x) = \infty$ is zero. This property is desirable and consistent with the spirit of the asymptotic approach.

Many complexity experts prefer to deal with instances of a fixed size. Can the definition of AP be reformulated in such terms? For a wide range of domains, the answer is yes; see [Gu1] in this connection.

Definition. *A partial function on a domain D is AP-time (or AP-time computable) if it is computable in time polynomial on average with respect to D.*

The following lemma justifies the use of more convenient size functions:

Lemma 2.1. *Suppose that D_1, D_2 are two domains with the same universe U and the same probability distribution. Let S_i be the size function of D_i and T be a function from U to $\bar{\mathcal{R}}^+$.*

1. *Suppose that S_1 is bounded by a polynomial of S_2 and T is polynomial on average with respect to D_1. Then T is polynomial on average with respect to D_2.*

2. *If S_1, S_2 are bounded each by a polynomial of the other then T is polynomial on average with respect to D_1 if and only if it is polynomial on average with respect to D_2.*

Proof. *It suffices to prove (1). Fix some $k \geq 1$ such that $S_1(x) \leq (S_2 x)^k$. We suppose that ε witnesses that T is polynomial on average with respect to D_1 and prove*

that ε/k witnesses that T is polynomial on average with respect to D_2. Ignoring points x such that $(Tx)^{\varepsilon/k} < S_2(x)$, we have

$$\sum_x (Tx)^{\varepsilon/k} \cdot (S_2 x)^{-1} \cdot \boldsymbol{P}(x) \leq \sum_x (Tx)^{\varepsilon} \cdot (S_2 x)^{-k} \cdot \boldsymbol{P}(x) \leq$$
$$\sum_x (Tx)^{\varepsilon} \cdot (S_1 x)^{-1} \cdot \boldsymbol{P}(x) < \infty. \quad QED$$

The notion of uniform probability distribution plays an important role in the theory of finite probability spaces. On a set of nonzero finite cardinality m, the uniform probability distribution assigns the probability $1/m$ to each element. In order to generalize this notion to infinite probability spaces, one needs to fix a default probability distribution on positive integers; somewhat arbitrarily, we choose the default probability of n to be proportional to $n^{-1} \cdot (\log n)^{-2}$ for $n > 1$. Some comments related to this issue can be found in [Gu1] and [Gu2].

Definition. *A domain D is uniform if elements of the same size have the same probability and $\boldsymbol{P}_D[\{x : |x| = n\}]$ is proportional to $n^{-1} \cdot (\log n)^{-2}$ for $n > 1$.*

Definition. *The direct product $A \times B$ of domains A and B is the domain of pairs (a, b), where $a \in A$ and $b \in B$, such that $|(a, b)| = |a| + |b|$ and $\boldsymbol{P}(a, b) = \boldsymbol{P}_A(a) \times \boldsymbol{P}_B(b)$.*

Given a probability distribution μ on some set U and a subset $V \subseteq U$ of positive probability, we define the restriction $\mu|V$ of μ to V to be the conditional probability $\nu(x) = \mu(x \mid x \in V)$. In other words, $\nu(x) = 0$ for every $x \in U - V$, and $\nu(x) = \mu(x)/\mu[V]$ for every $x \in V$.

Definition. *A domain B is a subdomain of a domain A if $U_B \subseteq U_A$, and the size function of B is the restriction of the size function of A, and $\boldsymbol{P}_B$ is the restriction of $\boldsymbol{P}_A$ to U_B. Further, A^+ is the subdomain of A comprising the elements of positive probability.*

3. Domain reductions

Consider a function f from a set A of strings in some finite alphabet to a set B of strings in some finite alphabet, and let T range over functions from B to $\bar{\mathcal{R}}^+$. If f is computable and T is bounded by a computable function then the composition $T \circ f$ is bounded by a computable function. Suppose that the sets A and B come equipped with size functions. If f is polynomial time computable and T is polynomially bounded then $T \circ f$ is polynomially bounded. Further suppose that A and B are domains. Now the situation is different. Even if f is AP-time computable and T is AP with respect to B, $T \circ f$ is not necessarily AP with respect to A. The problem is that T may be small on average but very large on the range of f. This problem should not arise if f is to be used as a reduction between randomized decision problems. Thus, one may want to say that a function f from A to B reduces A to B if

(R1) *f is AP-time with respect to A, and*

(R2) *For every AP function T on B, the composition $T \circ f$ is AP with respect to A.*

Before we proceed, let us make a slight generalization by allowing f to be partial provided that it is defined at every element of positive probability in A. We stipulate that $T(fx) = \infty$ if f is undefined at x. The requirements (R1) and (R2) remain meaningful. For brevity, we will say that such an f is an almost total function from A to B.

Unfortunately, the requirement (R2) is difficult to use and a convenient sufficient condition for (R2) is needed. Since we allow ∞ as a possible value of T, (R2) implies

(R0) *If $\boldsymbol{P}_A(x) > 0$ then $\boldsymbol{P}_B(fx) > 0$.*

For, suppose that $\boldsymbol{P}_A(a) > 0$ but $\boldsymbol{P}_B(fa) = 0$. Then the function $T(y) = [$ if $y = f(a)$ then ∞ else $0]$ is AP whereas the function $T \circ f$ has the infinite value at a point of positive probability and therefore is not AP.

The next two definitions lead to a more tractable formulation of (R2).

Definition. *Let A and B be two domains with the same universe U and the same size function. Then B dominates A, symbolically $A \leq B$, if there exists an AP function g on A such that $\boldsymbol{P}_A(x) \leq g(x) \cdot \boldsymbol{P}_B(x)$ for all x in U.*

This concept was discussed in [Gu1] under the name "weak domination"; the term "domination" was restricted in [Le, Gu1] to the case when g is polynomially bounded. Notice that $A \leq B$ if and only if the ratio $\boldsymbol{P}_A(x)/\boldsymbol{P}_B(x)$ is AP with respect to A. It is supposed that $0/0 = 0$ which is consistent with our previous convention $0 \cdot \infty = 0$.

Example. *Let BS1 be the uniform domain of nonempty binary strings where the size of a string is its length. Order binary strings lexicographically (more exactly, first by length and then lexicographically). Let BS2 be the domain of nonempty binary strings where the size of a string is its length and the probability of a string x is proportional to the default probability of the number of x in the lexicographical order. It is easy to check that the two domains dominate each other. This would not be true if the default probability of n were n^{-2} as in [Le].*

Lemma 3.1 [Gu 1, Section 1]. *Let A and B be two domains with the same universe U such that $A \leq B$. For every function T from U to $\bar{\mathcal{R}}^+$, if T is AP with respect to B then it is AP with respect to A.*

Definition. *Suppose that A, B are domains and f is an almost total function from A to B. B dominates A with respect to f, symbolically $A \leq_f B$ if the ratio*

$$\boldsymbol{P}_A[f^{-1}(fx)]/\boldsymbol{P}_B(fx)$$

is AP with respect to A.

Corollary. *If f is one-to-one then $A \leq_f B$ if and only if $\boldsymbol{P}_A(x)/\boldsymbol{P}_B(fx)$ is AP with respect to A.*

Theorem 3.1. *Suppose that A, B are domains and f is an almost total function from A to B such that the function $x \mapsto |fx|_B$ is AP with respect to A. Then (R2)*

holds if and only if $A \leq_f B$. Moreover, let $\mu(x) = \boldsymbol{P}_A(x)$ and let ν be the restriction of $\boldsymbol{P}_B$ to $\{y : \mu[f^{-1}(y)] > 0\}$. Then the condition (R2) is equivalent to each of the following conditions:

(D1) *$A \leq_f B$.*

(D2) *There exists a function g from B to $\bar{\mathcal{R}}^+$ such that*

- *For all $y \in B$, $\mu[f^{-1}(y)] \leq g(y) \cdot \nu(y)$, and*
- *$g \circ f$ is AP with respect to A.*

(D3) *A is dominated by the domain A' obtained from A by replacing μ with the probability distribution*

$$\mu'(x) = [\mu(x)/\mu[f^{-1}(fx)]] \cdot \nu(fx).$$

(D4) *There is a probability distribution $\mu''(x)$ on the universe of A such that*

$$\sum_{x, f(x)=y} \mu''(x) = \nu(y)$$

for every element y of B and the domain A'', obtained from A by replacing μ with μ'', dominates A.

Remark. *(D4) is an older definition of domination with respect to a given function; see [Gu1].*

Proof. *Without loss of generality, we may suppose that A coincides with its subdomain A^+ comprising elements of A of positive probability. Then f is a total function.*

To prove that (D1) implies (D2), set $g(y) = \mu[f^{-1}(y)]/\nu(y)$. To prove that (D2) implies (D3), notice that if (D2) holds then $g \circ f$ witnesses that A' dominates A. It is obvious that (D3) implies (D4). The implication (D4) $\rightarrow$ (R2) is proved in [Gu1] in the special case when all values of T are finite. We reduce the general case to the special case. Suppose (D4) and let μ'' be an appropriate witness. Let T be an AP function on B and $S(x) = T(fx)$. Then $\nu[\{y : Ty = \infty\}] = 0$, hence $\mu''[\{x : Sx = \infty\}] = 0$, and therefore $\mu[\{x : Sx = \infty\}] = 0$. Let $T'y = [$ if $Ty < \infty$ then Ty else $0]$. Obviously, T' is AP with respect to B and all values of T' are finite; hence the function $S'x = T'(fx)$ is AP with respect to A. But $Sx = S'x$ on every x of positive probability. Hence S is AP with respect to A. Thus, (R2) holds.

Finally, we suppose (R2) and prove (D1). Let $R = \{y : \mu[f^{-1}(y)] > 0\}$. By (R0), $\nu(y) > 0$ whenever $y \in R$. Choose $T(y) = [$ if $y \in R$ then $\mu(f^{-1}y)/\nu(y)$ else $0]$. The function T is linear on average with respect to B:

$$\sum_y T(y)\nu(y)/|y| = \sum_{y \in R} (\mu[f^{-1}y]/\nu(y)) \cdot \nu(y)/|y| \leq \sum_{y \in R} \mu[f^{-1}y] = 1.$$

By (R2), $T(fx)$ is AP with respect to A. Hence (D1) holds. QED

Remark. *The proof of the implication (R2) → (D1) does not use the fact that $|fx|_B$ is AP with respect to A. This hypothesis is used in the part of the proof of (D2) → (R2) cited from [Gu1].*

Now we are ready to define the notion of (deterministic) domain reduction.

Definition. *An almost total function f from a domain A to a domain B deterministically reduces A to B if it has the following two properties:*

Efficiency: *f is AP-time with respect to A.*

Domination: *B dominates A with respect to f.*

We say that a domain A deterministically reduces to a domain B if some almost total function f deterministically reduces A to B.

Theorem 3.2. *If f deterministically reduces a domain A to a domain B and g deterministically reduces B to a domain C then $g \circ f$ deterministically reduces A to C. Thus the deterministic reducibility relation is transitive.*

In essence, Theorem 3.2 is not new [Gu1], but Theorem 3.1 allows us to give a simpler proof.

Proof. *To prove that C dominates A with respect to $g \circ f$, use Theorem 3.1. It remains to check that $g \circ f$ is AP-time computable with respect to A. The time to compute $g(fx)$ splits into two parts: The time to compute $f(x)$ from x, and the time $t(x)$ to compute $g(y)$ from $y = f(x)$. Since f is AP-time with respect to A, it suffices to prove that $t(x)$ is AP. We know that the time $T(y)$ needed to compute $g(y)$ from y is AP with respect to B. Obviously, $t(x) = T(fx)$. By Theorem 3.1, $t(x)$ is AP. QED*

4. Randomized decision problems

A randomized decision problem, in short an RDP, Π may be defined as a domain D together with a function χ, the characteristic function of Π, from D to $\{0,1\}$. Any element x of D is an instance of Π, and the corresponding question is whether $\chi(x) = 1$. The problem Π is AP-time decidable if the characteristic function χ is AP-time computable.

Definition. *Let Π_1 and Π_2 be RDPs with domains D_1, D_2 and characteristic functions χ_1, χ_2 respectively. An almost total function f from D_1 to D_2 deterministically reduces Π_1 to Π_2 if f is a reduction of D_1 to D_2 and f satisfies the following additional requirement:*

Correctness: *For every instance x of Π_1 of non-zero probability, $\chi_1(x) = \chi_2(f(x))$.*

Theorem 4.1. *If f deterministically reduces Π_1 to Π_2 and g deterministically reduces Π_2 to Π_3 then the composition $g \circ f$ deterministically reduces Π_1 to Π_3.*

Proof. *Use Theorem 3.2. QED*

Theorem 4.2. *If f deterministically reduces Π_1 to Π_2 and Π_2 is AP-time decidable then Π_1 is AP-time decidable.*

Proof. *By Theorem 3.1, f satisfies the property (R2). QED*

Deterministic reductions were used to establish the completeness (for an appropriate class of RDPs) of some natural randomized decision problems [Le, Gu1, BCGL, Gu2].

5. Randomizing domain reductions

According to Section 3, a deterministic reduction of a domain A to a domain B is an AP-time computable almost total function f from A to B such that B dominates A with respect to f. The notion of deterministic reduction is generalized in this section by allowing the algorithm that computes f to flip a coin. For simplicity, only fair coins will be flipped.

Define a randomizing Turing machine, in short an RTM, to be a Turing machine that can flip a fair coin. Formally, this may mean that the machine has an additional read-only tape containing a random sequence of zeroes and ones. Thus, an RTM can be seen as a deterministic Turing machine with two input tapes. Our RTMs are transducers, i.e., they compute functions. We say that an RTM halts on an input x and a sequence r of coin flips if it reaches a special halting state; if the machine is stuck because the sequence r happened to be too short, it does not halt on (x, r).

Call a set X of binary strings prefix-disjoint if no element of X is a prefix of another element of X. A prefix-disjoint set is called a barrier if every infinite sequence $b_1 b_2 \ldots$ of bits has a prefix in X. By König's lemma, every barrier is finite.

Lemma 5.1. *For every barrier X, $\sum_{r \in X} 2^{-|r|} = 1$.*

Proof. *Associate the real interval $[0.a_1 \ldots a_k 000 \ldots, 0.a_1 \ldots a_k 111 \ldots)$ of length 2^{-k} with any binary string $a_1 \ldots a_k \in X$. These intervals partition the interval $[0, 1)$. QED*

If M is an RTM, let $\mathcal{B}_M(x)$ be the collection of finite sequences r of coin flips such that the computation of M on (x, r) halts using all bits of r. It is easy to see that $\mathcal{B}_M(x)$ is always prefix-disjoint. We say that M halts on x if $\mathcal{B}_M(x)$ is a barrier.

There is a good justification for us to restrict attention to machines that halt on every input: We deal with problems of bounded complexity, e.g., NP problems; given sufficient time, the reducing algorithm can simply solve the problem that it is supposed to reduce. For consistency with preceding sections, we will make a slightly more liberal restriction on our randomized Turing machines. It will be supposed that inputs come from a certain domain and the machine halts on every input of positive probability. More formally, a domain A is an input domain for a randomized Turing machine M if every $x \in A$ is a legal input for M and M halts on every $x \in A$ with $\boldsymbol{P}_A(x) > 0$.

Definition. *A dilator for a domain A is a function δ that assigns a prefix-disjoint set $\delta(x)$ of binary strings to each $x \in A$ in such a way that if $\boldsymbol{P}_A(x) > 0$ then $\delta(x)$*

is a barrier. Two dilators α and β for A are equivalent if $\alpha(x) = \beta(x)$ whenever $\boldsymbol{P}_A(x) > 0$.

If M is an RTM with input domain A then $\mathcal{B}_M(x)$ is a dilator for A.

Definition. *Let A be a domain with a universe U. If δ is a dilator for A, then the δ-dilation of A is the domain $A * \delta$ such that:*

- *The universe of $A * \delta$ is the cartesian product of U and the set $\{0,1\}^*$ of all binary strings.*
- *$|(x,r)| = |x|$.*
- *If $r \in \delta(x)$ then $\boldsymbol{P}_{A*\delta}(x,r) = \boldsymbol{P}_A(x) \cdot 2^{-|r|}$, and otherwise $\boldsymbol{P}_{A*\delta}(x,r) = 0$.*

The second clause here ensures that the notions like AP refer to the size of the actual input x, not the random string r. Use Lemma 5.1 to check that $\boldsymbol{P}_{A*\delta}$ is indeed a probability distribution. Even though $A * \delta$ may have infinitely many elements of a given size n, only finitely many of them have positive probability. Thus, $A*\delta$ is indeed a domain though its size function is not conventional. Notice that equivalent dilators give the same dilation. We will ignore the distinction between equivalent dilators.

Definition. *Let A be a domain with a universe U. A random function on A is a partial function on $U \times \{0,1\}^*$ such that the function*

$$\delta_f(x) = \{r : f \text{ is defined at } (x,r)\}$$

is a dilator. Two random functions f and g on A are equivalent if they have equivalent dilators and they coincide on all elements of the dilated domain which have positive probability.

In terms of Section 3, a random function f is an almost total function on $A * \delta_f$. Every RTM M with input domain A computes a random function on A which will be called F_M. The corresponding dilator is $\mathcal{B}_M$. We will ignore the distinction between equivalent random functions.

One may object to the term "random function" on the grounds that a random function on A should be a randomly chosen function on A. We fashioned the term "a random function" after well accepted terms like "a real function". A real function on a set A assigns real numbers to elements of A. A random function on a domain A can be seen as a function that assigns random objects to (almost all) elements of A.

Definition. *A random function T from a domain A to $\bar{\mathcal{R}}^+$ is AP if the following function from $A * \delta_T$ to $\bar{\mathcal{R}}^+$ is AP:*

$$(x,r) \mapsto [\text{ if } T \text{ is defined at } (x,r) \text{ then } T(x,r) \text{ else } \infty].$$

Definition. *A randomized Turing machine M with input domain A is AP-time if the computation time of M is AP.*

A randomized Turing machine M with input domain A can be viewed as a deterministic machine with input domain $A * \mathcal{B}_M$. It is easy to see that M is AP-time if and only if the corresponding deterministic machine is AP-time.

Definition. *A random function f on a domain A is AP-time if there exists an AP-time randomized Turing machine M that computes f, i.e., A is an input domain for M and $f = F_M$.*

We proceed to define a composition of random functions.

Definition. *Suppose that f is a random function from a domain A to a domain B and g is a random function on B. The composition $g \circ f$ of f and g is the random function h on A such that, for every $x \in A$,*

- $\delta_h(x) = \{rs : \ r \in \delta_f(x) \text{ and } s \in \delta_g(f(x,r))$,
- *if $r \in \delta_f(x)$ and $s \in \delta_g(f(x,r))$ then $h(x,rs) = g(f(x,r),s)$.*

Lemma 5.2. *The composition of random function is associative.*

Proof. *Suppose that f is a random function from a domain A to a domain B, g is a random function from B to a domain C, and h is a random function from a C to a domain D. It is easy to see that both $h \circ (g \circ f)$ and $(h \circ g) \circ f$ are equivalent to a random function k on A such that if $x \in A^+$ then $\delta_k(x)$ comprises strings rst with $r \in \delta_f(x)$, $s \in \delta_g(f(x,r))$, $t \in \delta_h(g(f(x,r),s))$ and $k(x,rst) = h(g(f(x,r),s),t)$. QED*

Definition. *Let f be a random function from a domain A to a domain B. Then B dominates A with respect to f, symbolically $A \leq_f B$, if B dominates $A * \delta_f$ with respect to f.*

Theorem 5.1. *Let f be a random function from a domain A to a domain B such that the random function $x \mapsto |f(x,r)|$ is AP with respect to A. Then the following statements are equivalent:*

- *$A \leq_f B$,*
- *For every AP random function T from B to $\bar{\mathcal{R}}^+$, the composition $T \circ f$ is AP.*

Proof. *Let $\alpha = \delta_f$ so that f is an almost total function on $A * \alpha$. Let β range over dilators for B and γ be the dilator for A such that*

$$\gamma(x) = \{rs : \ r \in \alpha(x) \text{ and } s \in \beta(f(x,r))\}.$$

Let x range over elements of A, r range over $\alpha(x)$ and s range over $\beta(f(x,r))$. Define an almost total function $g(x,rs) = (f(x,r),s)$ from $A * \gamma$ to $B * \beta$; $g(x,t)$ is undefined unless t has the form rs. The function $|g(x,t)|$ is AP on $A * \gamma$. For, let ε witness that $|f(x,t)|$ is AP on $A * \alpha$; then

$$\sum_{x,r,s} |g(x,rs)|^\varepsilon \cdot \boldsymbol{P}_{A*\gamma}(x,rs)/|x| = \sum_{x,r} |f(x,r)|^\varepsilon \cdot \boldsymbol{P}_{A*\alpha}(x,r)/|x| < \infty.$$

If $\delta_T = \beta$ then $\delta_{T \circ f} = \gamma$. The composition $T \circ f$ of the random function f from A to B and a random function T on B is the composition $T \circ g$ of the (deterministic) almost total function g from $A * \gamma$ to $B * \beta$ and the (deterministic) almost total function T on $B * \beta$. It suffices to prove that, for every β, the following statements are equivalent:

1. *$A * \alpha \leq_f B$,*
2. *$A * \gamma \leq_g B * \beta$,*
3. *For every almost total AP function T on $B * \beta$, $T \circ g$ is an almost total AP function on $A * \gamma$.*

(2) and (3) are equivalent by Theorem 3.1. It remains to prove that (1) and (2) are equivalent. It is easy to see that if a function π witnesses (1) in the sense of Theorem 3.1(D2) then any function ρ from $A * \gamma$ to $\bar{\mathcal{R}}^+$ such that $\rho(x, rs) = \pi(x, r)$ witnesses (2) in the sense of Theorem 3.1(D2). To prove the other direction, fix a function S that assigns an element of $\beta(x, r)$ to each pair (x, r). If a function ρ witnesses (2), then any function π from $A * \alpha$ to $\bar{\mathcal{R}}^+$ such that $\pi(x, r) = \rho(x, rs)$, where $s = S(x, r)$, witnesses (1). QED.

The restriction on the size $|f(x, r)|$ is not needed to deduce the first statement of the Theorem 5.1 from the second one.

Definition. *A random function f from a domain A to a domain B reduces A to B if it has the following two properties:*

Efficiency: *f is AP-time computable, and*

Domination: *B dominates A with respect to f.*

We say that a domain A reduces to a domain B if some random function f reduces A to B.

Theorem 5.2. *The reducibility relation is transitive.*

Proof. *Suppose that a random function f reduces a domain A to a domain B, and a random function g reduces B to a domain C. By Theorem 5.1, C dominates A with respect to the composition $h = g \circ f$. It remains to check that h is AP-time computable with respect to A. Let x range over A, r range over $\delta_f(x)$, $y = f(x, r)$ and s range over $\delta_g(y)$. The time to compute $h(x, rs)$ splits into two parts: The time to compute $f(x, r)$ from (x, r), and the time $t(x, rs)$ to compute $g(y, s)$ from (y, s). Since f is AP-time with respect to A, it suffices to prove that t is AP with respect to A. We know that the time $T(y, s)$ needed to compute $g(y, s)$ from (y, s) is AP with respect to B. Obviously, $t(x, rs) = T(y, s)$. Viewing t and T as random functions, we have $t = T \circ f$. By Theorem 5.1, $t(x)$ is AP. QED*

6. Randomizing reductions of problems

It turns out to be useful to weaken the correctness property of (randomizing) reductions of decision and search problems [VL, BCGL, IL]. Here is one possible definition of reductions of RDPs.

Definition. *Let Π_1, Π_2 be randomized decision problems with domains A, B and characteristic functions χ_1, χ_2 respectively. A random function f from A to B is a reduction of Π_1 to Π_2 if f reduces A to B and satisfies the following additional requirement:*

Correctness *There exists a number $a > 1/2$ such that, for every $x \in A^+$, the probability of the event $\chi_1(x) = \chi_2(f(x,r))$ is at least a.*

The notion of AP-time decidability does not fit well the notion of randomizing reductions with a correctness guarantee $a < 1$. If we combine in the obvious way a randomizing reduction of Π_1 to Π_2 and a decision algorithm for Π_2, the result is a randomizing algorithm for Π_1 which is not guaranteed to be correct all the time. Accordingly, we generalize the notion of AP-time decidability. Say that a randomizing Turing machine M solves an RDP Π with a correctness guarantee a if, for every $x \in D^+$, the probability that M computes $\chi(x)$ is at least a.

Definition. *A randomized decision problem Π is RAP-time decidable if there exists a, $1/2 < a \leq 1$, such that some randomizing Turing machine solves Π with correctness guarantee a.*

Lemma 6.1. *Let $1/2 < a < b < 1$ and let Π be a randomized decision problem. If there exists a randomizing AP-time Turing machine that solves Π with correctness guarantee a then there exists a randomizing AP-time Turing machine that solves Π with correctness guarantee b.*

Proof. *Consider Bernoulli trials with probability a for success in a single trial and let k be the least number such that the probability of $> k/2$ successes in k trials is $\geq b$. Given an instance x of Π, repeat the a-correct procedure k times and output the majority answer (in the case of a tie, output any answer). QED*

Remark. *The situation is even better for search problems, where one needs only one successful attempt [VL, BCGL, IL]. The inequality $a > 1/2$ can be replaced by an inequality $a > 0$ in the case of search problems.*

Theorem 6.1. *If f reduces an RDP Π_1 to an RDP Π_2 and Π_2 is RAP-time decidable then Π_1 is RAP-time decidable.*

Proof. *Suppose that the correctness guarantee of f is a_1 and the correctness guarantee of the given RAP-time decision procedure for Π_2 is a_2. By Lemma 6.1, we may assume that a_2 is sufficiently large so that $a_1 a_2 > 1/2$. Consider the randomizing algorithm which, given an instance x of Π_1, first applies f to x, and then – if and when some instance $y = f(x)$ of Π_2 is obtained – applies the a_2-correct decision algorithm to y; of course, the two subcomputations use independent sequences of coin flips. This composite algorithm solves Π_1 with correctness guarantee $a_1 a_2$. QED*

Unfortunately, our partially correct reductions of RDP's do not compose in a satisfactory way: If f reduces Π_1 to Π_2 with correctness guarantee a_1 and g reduces Π_2 to Π_3 with correctness guarantee a_2 then the correctness guarantee of $g \circ f$ is only $a_1 a_2$ which may be well below 1/2. (This difficulty does not arise for search problems.) The repetition technique for boosting the probability of correctness, which we applied to randomizing decision (and search) algorithms above, is not directly applicable to our many-one randomizing reductions. Repeating a randomizing reduction k times results in k outputs in the target domain, not one as in the definition of reduction. In other words, such a repetition is a version of Turing (or truth-table) reduction, not a many-one reduction. For simplicity, we spoke about constant correctness guarantees. This restriction can and should be relaxed [VL, BCGL, IL]. We hope to address elsewhere the issues arising from this.

Some randomized decision and search problems complete for RNP with respect to partially correct reductions can be found in [VL, BCGL]. Partially correct reductions play an important role in [IL].

Appendix. On deterministic domain reductions

Return to the motivation of deterministic domain reductions in the beginning of Section 3. The fact that functions T were allowed to have the value ∞ simplified the situation somewhat. It enabled us to derive (R0) from (R2). In this appendix, we give a version of Theorem 3.1 covering the case when functions T have only finite values. We use the notation of Section 3.

Let A, B be domains and f be a function that assigns an element of B to every element of A (including elements of zero probability). Consider the following version of the requirement (R2):

(R3) *For every AP function T on B that takes only finite values, the composition $T \circ f$ AP on A.*

Obviously, (R3) does not necessarily imply (R0). It turns out that (R3) does not necessarily imply (D1) either. Here is counterexample. Pick any domain A and any element $a \in A$ of positive probability. Let B be the domain such that (i) B has the same universe and the same size function as A, and (ii) $\boldsymbol{P}_B(a) = 0$ is and $\boldsymbol{P}_B(x)$ is proportional to $\boldsymbol{P}_A(x)$ for $x \neq a$. Finally, let f be the identity function. Obviously, (R3) holds and (D1) fails.

Let $\mu(x) = \boldsymbol{P}_A(x)$, $\nu(y) = \boldsymbol{P}_B(y)|range(f)$, $R = \{y : \nu(y) > 0\}$, $E = \{y : \nu(y) = 0$ but $\mu[f^{-1}(y)] > 0\}$, and consider the following version of deterministic domain domination:

(D0) *E is finite and there exists a function g from B to $\bar{\mathcal{R}}^+$ such that*

- *For all $y \in R$, $\mu[f^{-1}(y)] \leq g(y) \cdot \nu(y)$, and*
- *$g \circ f$ is AP with respect to A.*

Theorem. *If $|f(x)|$ is AP on A then the condition (D0) is necessary and sufficient for (R3).*

Proof. *First suppose (R3). Then E is finite: Otherwise, choose T such that T is zero outside E and $T(fx)$ is not AP, and get a contradiction. The desired $g(y) =$ [if $y \in R$ then $\mu[f^{-1}y]/\nu(y)$ else 0]. To check that $g \circ f$ is AP with respect to A, notice that g is AP with respect to B and use (R3).*

Next suppose (D0) and let T be an AP function on B taking only finite values. Let m be the maximal value of the function $S(x) = T(fx)$ on E. For any ε,

$$\sum_{x \in E} (Sx)^{\varepsilon} \mu(x)/|x| \leq \sum_{x} m^{\varepsilon} \mu(x) < \infty.$$

Therefore, we have to prove only that, the restriction of S to $f^{-1}(R)$ is AP with respect to A. Without loss of generality, we can suppose that E is empty. Then (D0) implies (D2). By Theorem 3.1, (D2) implies (R2). By (R2), S is AP with respect to A. QED

References

[BCGL] Shai Ben-David, Benny Chor, Oded Goldreich and Michael Luby, "On the Theory of Average Case Complexity", Symposium on Theory of Computing, ACM, 1989, 204–216.

[Gu1] Yuri Gurevich, "Average Case Complexity", J. Computer and System Sciences (a special issue on FOCS'87) to appear.

[Gu2] Yuri Gurevich, "Matrix Decomposition Problem is Complete for the Average Case", Symposium on Foundations of Computer Science, IEEE Computer Society Press, 1990, 802–811.

[GJ] Michael R. Garey and David S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", Freeman, New York, 1979.

[IL] Russel Impagliazzo and Leonid A. Levin, "No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random", Symposium on Foundations of Computer Science, IEEE Computer Society Press, 1990, 812–821.

[Le] Leonid A. Levin, "Average Case Complete Problems", SIAM Journal of Computing, 1986.

[VL] Ramarathnam Venkatesan and Leonid Levin, " Random Instances of a Graph Coloring Problem are Hard", Symposium on Theory of Computing, ACM, 1988, 217–222.