[Ba] J. Balcázar, Self-Reducibility, *Journal of Computer and System Sciences* **41** (1990) 367-388.

[Bu] S. R. Buss, The Boolean formula value problem is in ALOGTIME, *Proc. 19th ACM Symposium on Theory of Computing*, 1987, pp. 123–131.

[GH] J. Goldsmith, L. Hemachandra, D. Joseph, and P. Young, Near-testable sets, *SIAM J. Comput.* **20** (1991) 506–523.

[HY] J. Hartmanis and Y. Yesha, Computation times of NP sets of different densities, *Theoretical Computer Science* **34** (1984), 17–32.

[He] L. Hemaspaandra, personal communication.

[IN] R. Impagliazzo and M. Naor, Decision trees and downward closures, *Proc. 3rd Structure in Complexity Theory Conference*, pages 29–38, 1988.

[JL] D. Juedes and J. Lutz, The complexity and distribution of hard problems, *Proc. 34th FOCS Conference*, pages 177–185, 1993.

[L92] J. Lutz, Almost Everywhere High Nonuniform Complexity, *Journal of Computer and System Sciences* **44** (1992), pp. 220-258.

[L93] J. Lutz, The quantitative structure of exponential time, *Proc. 8th Structure in Complexity Theory Conference*, pages 158–175, 1993.

[LM] J. Lutz, Measure, stochasticity, and the density of hard languages, to appear in *SIAM J. Comput.* (See also Proc. 10th STACS, Lecture Notes in Computer Science 665, 1993, pp. 38–47.

[M] E. Mayordomo, PhD Thesis, to appear. See also [M2], in which a preliminary version of the PSPACE measure appears.

[M2] E. Mayordomo, Measuring in PSPACE, to appear in *Proc. International Meeting of Young Computer Scientists '92*, Topics in Computer Science series, Gordon and Breach.

6 Sets That Do Not Have Measure Zero

The class of sets with $\Delta_{\rm P}$ measure zero is clearly a very limited class. A concrete example of this limitation is given in the following theorem:

Theorem 11 $\mu_{\Delta_{\rm P}}({\rm ODD}) \neq 0$, *where* ODD *is defined to be* $\{L$ *: for each* n, $L^{=n}$ *has an odd number of elements*$\}$.

Proof.(Sketch) For this sketch, assume that d is equal to the computation $\hat{d}$; the general case is proved in a similar manner.

Suppose $\mu_{\Delta_{\rm P}} = 0$, and let $d_k(w)$ be a null cover of ODD. We will construct an integer k and a language in ODD not covered by d_k. Find l so that $f(2^{l'}) < 2^{l'}$ for $l' \geq l$, where f is the polylog bound for d. Put $n = \sum_{j=0}^{l-1} 2^j$, so n is the number of words of length less than l. Let w_1 be a prefix in ODD of length n and fix $k > n$; note that $d_k(w_1) < 1$. Let $w_2' \sqsupset w_1$ be of length $|w_1| + 2^l$, and such that for each $i \in [|w_1|, |w_2'|)$ bit i of $w_2' = 1$ iff $d_k(w_2'[1..i-1]1) \leq d_k(w_2'[1..i-1]0)$. Since d_k doesn't increase along this path, $d_k(w_2') < 1$. Since $f(2^l) < 2^l$, some bit i_2 corresponding to a word of length l is not queried by d_k; let $w_2 = w_2'$ except that bit i of w_2 is set so that w_2 is a prefix of ODD. Continue, forming w_3' of length $|w_2| + 2^{l+1}$, along the path of nonincreasing d_k, so that $d_k(w_3') < 1$. Some new bit i_3 corresponding to a word of length $l+1$ is not queried by d_k; put $w_3 = w_3'$ except that bit i_3 is set so that w_3 is a prefix of ODD. Continue in this manner constructing w_m for all $m \geq 2$. Finally, the language given by the limit of w_m is in ODD, but not covered by d_k. ∎

We do not view this limitation of $\mu_{\Delta_{\rm P}}$ as a drawback. Indeed, one avenue we intend to pursue is the investigation of whether classes such as Ntime($\log n$) or AC^0 are $\Delta_{\rm P}$-null sets, and we expect that some of these classes do *not* have measure zero. The limitations of the $\Delta_{\rm P}$ measure indicate to us that interesting results along these lines may be obtainable.

We note in passing that one can also define a measure analogous to $\Delta_{\rm P}$, using *space* bounds, as opposed to time bounds. In this way, one obtains a measure for PSPACE. Mayordomo has previously defined a notion of measure for PSPACE, using polylogspace-bounded machines with one-way access to the input [M]. It is easy to show that any n-DS computable in our formulation is also computable in her formulation, and it is also easy to show that ODD has measure zero using the formulation in [M], whereas the proof of Theorem 11 carries over also in the case of space bounds. Thus the notion of measure on PSPACE defined in [M] is strictly richer than what one obtains with our definitions.

Acknowledgments

We gratefully acknowledge helpful discussions and correspondence with Lane Hemaspaandra, Elvira Mayordomo, Jack Lutz, and David Juedes.

References

[AR] E. Allender and R. Rubinstein, P-printable sets, *SIAM J. Comp.* **17** (1988) 1193–1202.

Theorem 7 *If X is a Δ_{P}-union of Δ_{P}-null sets, then X is Δ_{P}-null.*

Proof. Recall the proof of **M 1**. We have $X = \bigcup X_j$, and $d_{j,k}$ covers X_j with value 2^{-k}. Then $\sum_{j=0}^{\infty} d_{j,k+2^j+1}$ covers X, with computation

$$\hat{d}'_{k,r} = \sum_{j=0}^{\log(r+|w|)} \hat{d}_{j,k+2^j+1,r+2^j+2}.$$

The dependency set $G_{d'_{k,r},|w|}$ satisfies

$$G_{d'_{k,r},|w|} = \bigcup_{j=0}^{\log(r+|w|)} G_{\hat{d}_{j,k+2^j+1,r+2^j+2},|w|}.$$

Thus the size of $G_{d'_{k,r},|w|}$ is at most

$$\log(r+|w|)|G_{\hat{d}_{r+|w|,k+r+|w|+1,2r+|w|+2},|w|}|,$$

which is still polylogarithmic. ∎

Next, **M 2**.

Theorem 8 *All singleton sets of P languages are Δ_{P}-null.*

Proof. Given a language $L \in \mathrm{P}$, we will construct a null cover $d_k(w)$ of the singleton set $\{L\}$ as follows: Decide membership of each word x such that $\mathrm{pos}(x)$ is a power of 2 and is at most $|w|$. Suppose there are s of these. Output 2^{s-k} if bits $1, 2, 4, \ldots 2^s$ of w are all consistent with L, and output 0 otherwise.

It is easy to check that this yields a cover; we show here that $d \in \Delta_{\mathrm{P}}$. Since $L \in \mathrm{P}$ by hypothesis, we can check membership of any single word in polylog time. (This involves no queries to the input w of d). We can in polylog time check all log-many words whose lexicographic position is a power of 2. Also, we query the input at the bits whose positions are powers of 2, generating log-many points in the dependency set. The crucial observation is that the transitive closure of these bits is trivial; no more points are added. ∎

Finally,

Theorem 9 *We have $\mu_{\Delta_{\mathrm{P}}}(\mathrm{P}) \neq 0$.*

Proof. Given a cover $d \in \Delta_{\mathrm{P}}$ of value less than 1, we can build a constructor δ for a language not covered by d as in the Section 3. In the proof, we need to compare $d(w)$ with $d(w0)$. Each of these calls takes only polylog time, and it is clear that

$$G_{\delta,|w|} \subset G_{d,|w|} \cup (G_{d,|w0|} \setminus \{|w|+1\})$$

The bound on the size of the dependency sets follows. ∎

Theorem 10 *For all l, $\mu_{\Delta_{\mathrm{P}}}(Dtime(n^l)) = 0$.*

Proof. Exactly as theorem (4). The dependency set is clearly small. ∎

5 A Measure on P

Starting from the framework of [L92], there is a central difficulty in defining a notion of measure on P, as observed in [M] and elsewhere. Given a constructor δ, in order to decide membership of $x \in R(\delta)$, we *a priori* have to compute $\delta^{\mathrm{pos}(x)}(\lambda)$, an exponential time computation. We get around this difficulty simply by not considering such constructors.

Given a machine M and natural number n, define a *dependency set* $G_{M,n} \subset \{1, \ldots, n\}$ to be a set such that for each $i \in G_{M,n} \cup \{n\}$, and each word w of length n, M can, in polylog time, compute $M(w[1..i])$ querying only bits in $G_{M,n} \cap \{1, 2, \ldots i\}$. Note that for all M and n, there is a unique minimal dependency set for M and n, which can easily be computed by expanding the tree of queries that one obtains by assuming both possible values for each queried bit. In what follows, we let $G_{M,n}$ denote this minimal dependency set. Let Δ_{P} consist of the functions that have machines M that run in polylog time and have a polylog-size dependency set. Given a function f computed by machine M, we may abuse notation and speak of $G_{f,n}$ instead of $G_{M,n}$, where this causes no confusion.

Note that in the setting of [L92], requiring that $|G_{M,n}|$ be of size p_i in w is no restriction. Thus Lutz's definitions may be obtained by replacing our polylogarithmic bounds by the corresponding larger bounds.

One technicality that needs to be addressed concerns the manner in which Δ_{P} machines determine the length of their input. If one were to use the usual binary search approach to determine the length of the input, then the dependency sets cannot be kept small while allowing any sort of interesting computation. Our approach to this problem is to have the length of the input be given explicitly as an argument to the machine.[2] If a subscript of a density function is presented in unary, then providing the length of the argument obviates any need to provide the argument itself. Thus if the function $d_{i_1,\ldots,i_n,r}(w)$ is in Δ_{P}, then the running time and the size of the dependency sets are all $(\log(|w| + i_1 + \cdots + i_n + r))^{O(1)}$. Density functions will produce output in $\mathbb{S}$.

Theorem 6 *We have* $\mathrm{P} = R(\Delta_{\mathrm{P}})$.

Proof. Any language in P has a trivial constructor that ignores all but the length of its input, so we have $\mathrm{P} \subseteq R(\Delta_{\mathrm{P}})$.

On the other hand, given such a constructor δ, we can decide if $x \in R(\delta)$ as follows: Start running δ on $w = \delta^{\mathrm{pos}(x)-1}(\lambda)$. When δ needs bit i of w, recursively call δ on $w[1, ..i-1]$ to generate that bit. Unwinding this recursion effectively generates the dependency set $G_{\delta,|w|}$, which by hypothesis is small. Also, the non-query computation of δ is takes time polylogarithmic in $|w|$, and thus polynomial in $|x|$. Thus $R(\Delta_{\mathrm{P}}) \subseteq \mathrm{P}$. ∎

We have to show that the proofs for the measure axioms given in the previous sections satisfy the restrictions on dependency sets.

[2] An alternative approach would be to declare that the input consists of 0's and 1's and is followed by blanks, and to include in the dependency set only queries that ask to distinguish between a 0 and a 1, and not to count those queries that ask only to distinguish between a blank and a nonblank.

If each singleton set in SR is $\Delta_{\rm SR}$-null, then this measure would satisfy the three measure axioms for SR. Thus our goal in this section is to present a condition that has this as a consequence.

First, we need some definitions.

Definition 2 *A set S is* P-printable *[HY, AR] if there is a routine that, on input 1^n, outputs $S \cap \Sigma^n$ in time polynomial in n.*

S is a set of easy instances *for L if $S \in$ P and there is a set $A \in$ P such that for all $x \in S$, ($x \in L$ iff $x \in A$).*

Let **(*)** *denote the condition: Every set in SR has an infinite P-printable set of easy instances. (Equivalently,* **(*)** *can be restated as: No set in SR is bi-immune to the P-printable sets.)*

Condition **(*)** is certainly a very strong assumption. Recall that there are sets in SR that are hard for E. Also, Hemaspaandra has noted [He] that there are oracles relative to which there are self-reducible sets (in fact, near-testable sets [GH]) that are bi-immune to P. However, all of our attempts to disprove condition **(*)** have failed, and we would be interested to discover if there are any connections between **(*)** and other complexity-theoretic questions. Since the self-reducible sets constitute a highly-structured subclass of E, we feel that **(*)** is at least plausible.

Theorem 5 *Assuming* **(*)**, *all singleton sets in SR are $\Delta_{\rm SR}$-null.*

Proof. Let L be in SR; consider the following routine for computing the 1-DS d_k: On input w, let $n = |w|$, and let $m = |n|$. Compute the list $\{i_1, \ldots, i_s\}$, which is the list of elements of the P-printable set of easy instances having length at most $m = n^{O(1)} = \log^{O(1)} |w|$. Output 2^{s-k} if bits $i_1, \ldots, i_s$ of w are all consistent with L, and output 0 otherwise.

It is immediate that this defines a $\Delta_{\rm SR}$-null cover of L. ∎

Thus, if **(*)** holds, then $\Delta_{\rm SR}$ does define a reasonable notion of measure on SR. However, it is interesting to note that, even under this strong assumption, results that one would expect to hold (such as an analog of Theorem 4) are still not known to be true. That is, we are unable to prove that the class of languages with self-reductions computable in time n^2 is a measure zero subset of SR. The difficulty lies in the fact that, although a self-reduction, combined with an initial segment of the characteristic sequence of a language, tells us what the next bit needs to be (and thus tells us how we should place our bets if we are betting on the value of the next bit), our current formulation in terms of density functions requires us to output the value of the density function, which seems to require that we know the value of the density function on shorter prefixes, which does not seem to be computable in polynomial time. The next section presents a new formulation of measure, which overcomes this problem.

$$\begin{aligned} d(\delta^k(\lambda)) &< q + 2^{-l}(1 - 2^{-|\delta^k(\lambda)|}) \\ &< q + 2^{-l} \\ &\leq 1. \end{aligned}$$

∎

We do not know if each singleton set in SR is $\Delta_{\rm SR}$-null. On the other hand, it is easy to show that each singleton set in P is $\Delta_{\rm SR}$-null. (This is easy to prove directly; it also follows from the more general Theorem 4.) Thus we are unable to prove that $\Delta_{\rm SR}$ provides a useful notion of measure, either for P or for SR.

However, do note that if $\mu_{\Delta_{\rm SR}}({\rm P}) \neq 0$, then $\mu_{\Delta_{\rm SR}}(\cdot|{\rm P})$ *does* provide a reasonable notion of measure for P, in the sense that the three measure axioms would be satisfied. We believe that this is, in fact, a very plausible hypothesis (and in particular, we believe that it is less controversial than the hypothesis discussed in the next section). Note that Lutz and Mayordomo [L93, LM] have argued that it is plausible that NP is not a measure zero subset of Dtime($2^{n^{O(1)}}$). (See also [JL].) An analog of that same argument states that it is equally plausible that P is not a $\Delta_{\rm SR}$-null set, and that therefore this could in fact be a reasonable notion of measure for P. More precisely, note that if P *is* a $\Delta_{\rm SR}$-null set, then there is a fixed algorithm running in time n^k for some k, such that for all sets L in P (including sets in time $n^{1000000k}$), the algorithm can "bet" successfully on membership of strings in L. This comes close to saying that there is a k such that every set in P has a self-reducibility structure computable in time n^k. Almost nothing is known about the self-reducibility of sets in P, but, to quote [L93], this would "be a very remarkable algorithm!"

Certain properties of the measures defined in [L92] and [M] for higher complexity classes hold also for this new notion $\Delta_{\rm SR}$. For instance, this next result is typical of the simplest type of result proved for those other measures.

Theorem 4 *For all l, $\mu_{\Delta_{\rm SR}}(Dtime(n^l)) = 0$.*

Proof. It will suffice to express Dtime(n^l) as a $\Delta_{\rm SR}$-union of null sets. But this is easy. Let $M_1, M_2, \ldots$ be a clocked enumeration of Dtime(n^l) machines, and let

$$d_{j,k}(w) = \begin{cases} 0 & \text{if any of bits } \{1, 2, 4, \ldots, 2^k\} \text{ of } w \text{ are inconsistent with } L(M_j) \\ 2^{\lfloor \log |w| \rfloor - k} & \text{otherwise.} \end{cases}$$

The result follows. ∎

4 A Controversial Hypothesis

In the previous section, we discussed hypotheses that would imply that $\Delta_{\rm SR}$ defines a reasonable notion of measure on P. In this section, we present a (controversial) hypothesis that would imply that $\Delta_{\rm SR}$ defines a reasonable notion of measure of SR.

Proof. We show that if d is any Δ_{SR}-computable density function with $d(\lambda) < 1$;, then we can construct a language in SR not covered by d. (This is a weaker assumption than necessary; it is also possible to prove a stronger result analogous to Theorem 3.13 of [L92].)

As in [L92], find q, l such that

$$d(\lambda) < q < q + 2^{-l} \leq 1.$$

Let $a(x) \geq |x| + l + 3$, and define the desired language L by

$$\delta(w) = \begin{cases} 1 & \text{if } \hat{d}_{a(w)}(w0) > \hat{d}_{a(w)}(w) + 2^{1-a(w)} \\ 0 & \text{otherwise.} \end{cases}$$

Note that δ is clearly a constructor in Δ_{SR}, and thus defines a language L in SR.

Finally, we have to show L is not covered by d, i.e., for all n we have $d(\delta^n(\lambda)) < 1$. Again we use the strategy in [L92], an inductive argument. We have

$$\hat{d}_{a(x)}(\delta(x)) \leq \hat{d}_{a(x)}(x) + 2^{1-a(x)} \tag{1}$$

since, if $\hat{d}_{a(x)}(x0) > \hat{d}_{a(x)}(x) + 2^{1-a(x)}$ then

$$\begin{aligned} \hat{d}_{a(x)}(x1) &\leq d(x1) + 2^{-a(x)}, \text{ since } \hat{d} \text{ is a computation} \\ &\leq 2d(x) - d(x0) + 2^{-a(x)}, \text{ by the average condition} \\ &\leq 2\hat{d}_{a(x)}(x) - \hat{d}_{a(x)}(x0) + 2^{2-a(x)}, \text{ a computation} \\ &< \hat{d}_{a(x)}(x) + 2^{1-a(x)}, \text{ by hypothesis.} \end{aligned}$$

Thus, if $d(x) < q + 2^{-l}(1 - 2^{-|x|})$, we get

$$\begin{aligned} d(\delta(x)) &\leq \hat{d}_{a(x)}(\delta(x)) + 2^{-a(x)}, \text{ a computation} \\ &\leq \hat{d}_{a(x)}(x) + 3 \cdot 2^{-a(x)}, \text{ by (1)} \\ &\leq d(x) + 2^{2-a(x)}, \text{ a computation} \\ &< q + 2^{-l}(1 - 2^{-|x|}) + 2^{2-a(x)}, \text{ by hypothesis} \\ &\leq q + 2^{-l}(1 - 2^{-|x|}) + 2^{-(|x|+l+1)}, \text{ by definition of } a(x) \\ &= q + 2^{-l}(1 - 2^{-|\delta(x)|}). \end{aligned}$$

Thus, by induction,

and $d_{j,k}(w_k) \geq 1$. Now note that for all k, $d'_k(w_{k+2^j+1}) \geq 1$, and thus d' is a null cover of X. It remains only to show that d' has Δ_{SR}-computations.

Define the computation

$$\hat{d}'_{k,r}(w) = \sum_{j=0}^{\log(r+|w|)} \hat{d}_{j,k+2^j+1,r+2^j+2}(w).$$

This is polylog-time computable. There are only polylog many terms to add. Computing the sum of two addends amounts to sorting all the powers of 2 and performing carries (carries correspond to ties in the sort), and carries can propagate at most polylog positions. We can perform log-many binary additions in polylog time total.

Let $\sigma = \sum_{j=0}^{\log(r+|w|)} d_{j,k+2^j+1}(w)$. Then (as in [L92], and letting $j' = 2^j$) :

$$\begin{aligned}
|d'_k(w) - \sigma| &\leq \sum_{j=\log(r+|w|+1)}^{\infty} d_{j,k+2^j+1}(w) \\
&\leq \sum_{j=\log(r+|w|+1)}^{\infty} 2^{|w|} d_{j,k+2^j+1}(\lambda) \\
&\leq \sum_{j=\log(r+|w|+1)}^{\infty} 2^{|w|} 2^{-(k+2^j+1)} \\
&\leq \sum_{j'=r+|w|+1}^{\infty} 2^{|w|} 2^{-(k+j'+1)} \\
&= 2^{-(k+r+1)} \\
&\leq 2^{-(r+1)}.
\end{aligned}$$

Also,

$$\begin{aligned}
|\sigma - \hat{d}'_{k,r}(w)| &\leq \sum_{j=0}^{\log(r+|w|)} |d_{j,k+2^j+1} - \hat{d}_{j,k+2^j+1,r+2^j+2}(w)| \\
&\leq \sum_{j=0}^{\log(r+|w|)} 2^{-(r+2^j+2)} \\
&\leq \sum_{j=0}^{\infty} 2^{-(r+j+2)} \\
&= 2^{-(r+1)}
\end{aligned}$$

as in [L92]. ∎

Next, we show that SR has nonzero measure by following the diagonalization in [L92].

Theorem 3 *$\mu_{\Delta_{\mathrm{SR}}}(\mathrm{SR}) \neq 0$.*

where the subscripts $i_1, \ldots, i_n, r$ are represented in unary $\mathbb{U}$. In order to ease the size restriction of the output somewhat, we interpret the output to be encoded in another representation, denoted $\mathbb{S}$: a formal sum of decreasing powers of 2. If there are only polylog many powers of 2 in x and y (this is the case if x and y are the output of a polylog time machine), then $x+y$ and xy can be computed and output in $\mathbb{S}$ format, and the result will have only polylog-many powers of 2.

Note that a Δ-computation does not have to satisfy the average law (item 3 above), though it has to be close, depending on r.

For any class of functions Δ, we say that a set X is Δ-null if there is a Δ-computation of a 1-DS $d_k(w)$ covering X such that for all k, $d_k(\lambda) \leq 2^{-k}$. In this case we write $\mu_\Delta(X) = 0$.

If a class X has nonzero measure, we can talk about another set Y having "measure 0 in X," and we write $\mu_\Delta(Y|X) = 0$, if $\mu_\Delta(Y \cap X) = 0$ or "measure 1 in X," $\mu_\Delta(Y|X) = 1$, if $\mu_\Delta(X \setminus Y) = 0$.

3 Properties of a Measure

For the notion of measure on a class X to make sense, it should satisfy the following measure axioms:

M 1 *Easy unions of null sets are null.*

M 2 *Easy singleton sets are null.*

M 3 *X does not have measure zero.*

In this section, we begin to show that the notions defined in the previous section satisfy these axioms. To begin, we need to define more precisely what we mean by "an easy union of null sets".

Definition 1 *Let Δ be a class of functions. A set X is a* Δ-union of Δ-null sets *if $X = \bigcup_{j=0}^{\infty} X_j$, and there is a 2-DS $d_{j,k}$ so that $d_{j,k}$ covers X_j with value 2^{-k}, and $d_{j,k}$ has a Δ-computation $\hat{d}_{j,k,r}$.*

Theorem 2 *If X is a Δ_{SR}-union of Δ_{SR}-null sets, then X is Δ_{SR}-null.*

Proof. Let $d_{j,k}$ be a 2-DS witnessing that X is a Δ_{SR}-union of Δ_{SR}-null sets. Let

$$d'_k(w) = \sum_{j=0}^{\infty} d_{j,k+2^j+1}(w).$$

We will show d' is a Δ_{SR}-null cover of X.

It is immediate that d' is a 1-DS, and that $d'_k(\lambda) \leq 2^{-k}$. Now, to see that d' covers X, note that if ω is any sequence in X, then ω is in some X_j, and for all k there is a $w_k \sqsubseteq \omega$

A density-system is called an n-DS according to the number of subscripts. A 0-DS is a density function. Note there is no mention here of resource bounds.

Let $y \sqsubseteq w$ indicate that y is an initial substring of w, and let $C_w = \{\omega \sqsupseteq w\}$ denote the cylinder at w (thus C_w is the set of languages with the $|w|$ initial words according to w and the other words arbitrary). A density function covers X if $X \subset \bigcup\{\, C_w \mid d(w) \geq 1 \,\}$.

In defining notions of measure for E, ESPACE, and other large complexity classes, Lutz defines a Δ-measure for various classes of functions Δ. For the remainder of this paper, let $\Delta_{\rm SR}$ refer to the class of functions computable in polylogtime. Deterministic polylogtime computation has been studied by several authors; see, for example, [IN, Bu]. In order for sublinear-time machines to perform interesting computation, we follow the usual convention of providing these machines with random access to their input; that is, the machines have an "address tape," and if i is written in binary on the address tape, then the machine can in unit time move its read/write head to bit position i of the input. Among other things, such machines can, in logarithmic time, compute the length of their input ([Bu]). In order to avoid uninteresting technicalities regarding encoding of pairing functions, we adopt the convention that a machine computing a k-ary function is provided with k input tapes (with an "address tape" for each input tape). The running time of such a machine must be polylogarithmic in m, where m is the sum of the lengths of its k inputs. (Note that by choosing a suitable encoding, such a machine can be simulated by a machine with a single input tape.) The $\Delta_{\rm SR}$ machines we consider will write their output on a write-only output tape; thus the output is restricted to be of polylogarithmic length.[1]

We will actually need two types of polylogtime-computable functions. If f is a function in $\Delta_{\rm SR}$, where $f : \{0,1\}^* \to \{0,1\}$, then f defines a *constructor* δ, where $\delta(w) = wf(w)$. A constructor specifies the sequence that is the limit as $j \to \infty$ of $\delta^j(\lambda)$. This gives rise to the class $R(\Delta_{\rm SR})$, which is the class of languages whose characteristic sequences are given by some constructor in $\Delta_{\rm SR}$. Following similar observations in [M], it is easy to show:

Proposition 1 $R(\Delta_{\rm SR}) = {\rm SR}$.

Here, SR is the class of word-decreasing self-reducible sets. SR was studied in [Ba], where it was shown that SR $\subseteq$ E, and SR contains sets that are many-one complete for E (and thus P is properly contained in SR). It follows easily from [L92] that SR is a measure zero subset of E.

The other type of functions in $\Delta_{\rm SR}$ that we need are the functions that compute approximations to density systems (where density systems are as defined above). A $\Delta_{\rm SR}$-computation of an n-DS $d_{i_1,\cdots,i_n}$ is an $(n+1)$-subscripted function $\hat{d}_{i_1,\cdots,i_n,r}$ satisfying

- $\hat{d}_{i_1,\cdots,i_n,r}(w)$ is computable in the resource bound $\Delta_{\rm SR}$, (and thus has at most a polylogarithmic number of bits) and
- $|d_{i_1,\cdots,i_n}(w) - \hat{d}_{i_1,\cdots,i_n,r}(w)| \leq 2^{-r}$

[1] This is different from the usual notion of polylog-time computable functions, which requires only that the language $\{(x,i,b) \mid$ bit i of $f(x)$ is $b\}$ be computable in polylog time.

1 Introduction

Lutz has defined a notion of measure on complexity classes E = DTIME($2^{O(n)}$) and higher, where languages are the analogs of the classical points, and complexity subclasses of E are the sets to be measured [L92]. This notion of measure has been investigated in a series of related papers, and many of its properties are now understood. Unfortunately, Lutz's definitions are meaningful only for "large" complexity classes; they provide no notion of measure for P or PSPACE, etc. More recently, Mayordomo defined an extension of Lutz's notions, and provided a notion of measure for PSPACE [M].

Many of the most interesting questions and theorems in complexity theory involve subclasses of P; Lutz's notion of measure is of no use in investigations of these subclasses. Our goal in beginning this research was to define a suitable notion of measure that would allow one to investigate the question of whether or not AC^0 is a measure-zero subset of P, whether the set of languages complete for P under projections is a measure zero set, and other related questions. As it turns out, merely formulating definitions that give rise to a meaningful notion of measure for P was itself a time-consuming undertaking. In this paper, we present two definitions:

1. a simple and natural definition that in all likelihood provides a measure on P, but we cannot prove this is the case, and

2. a somewhat more complicated definition, for which we can prove that all of the "measure axioms" are satisfied.

We show that, for all k, Dtime(n^k) is a measure-zero subset of P. Future work will investigate the use of this notion of measure to study relations among subclasses of P.

2 Preliminaries

Much of our notation and our definitions come from [L92]. Due to space limitations, we refer the reader to [L92] for more precise definitions.

Fix an enumeration $\{s_i\}_{i=0}^{\infty}$ of all words, and let pos(x) denote the position of x in this ordering. A language L is identified with its characteristic sequence: an infinite sequence $\omega = \langle a_i \rangle$ with $a_i = 1$ iff $s_i \in L$. A complexity class is a set of languages. Let $w[i..j]$ denote the string of the i^{th} through j^{th} bits of w. Concatenation of strings and bits denotes itself.

A density-system is a function $d_{i_1,\ldots,i_r}\{0,1\}^* \to [0,\infty)$, where –

1. The subscripts are natural numbers.

2. The argument is considered a partial characteristic sequence of a language.

3. $d_{(\cdot)}(w) \geq \frac{d_{(\,)}(w0)+d_{(\,)}(w1)}{2}$.

ABSTRACT

We investigate the issues and obstacles involved in extending Lutz's notion of measure to provide a measure for P. We provide one natural definition that, under a plausible but unproven assumption, provides a reasonable notion of measure for P. We also provide a more complicated definition that does provide a measure for P (with no unproven assumption).

DIMACS Technical Report 94-14
May 1994

Towards a Measure for P

by

Eric Allender[1,2]
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

Martin Strauss[3,4]
Department of Mathematics
Rutgers University
New Brunswick, NJ 08903

[1]Permanent Member
[2]Research supported by NSF grant CCR-9204874.
[3]Affiliated Graduate Student Member
[4]Research supported by NSF grant CCR-9204874.

DIMACS is a cooperative project of Rutgers University, Princeton University, AT&T Bell Laboratories and Bellcore.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.