Electronic Colloquium on Computational Complexity, Report No. 40 (2003)

RP is Small in SUBEXP else ZPP equals PSPACE and NP equals EXP

Philippe Moser*

Abstract

We use recent results from [ABK+02] on the hardness of resource-bounded Kolmogorov random strings, to prove that RP is small in SUBEXP else ZPP = PSPACE and NP = EXP. Along the way we improve a result from [IM03] by reducing the time bounds from exponential to subexponential. More precisely we show that if NP is not small in SUBEXP, then NP = AM.

1 Introduction

An interesting question in Lutz's resource-bounded measure theory is whether the zero-one law holds within EXP, i.e. does every complexity class contained in EXP have either measure zero or one. Although the answer to this question is currently not known, it has been shown in [Mel00] and [IM03] that the zero-one law holds for all three standard probabilistic classes ZPP, RP and BPP, for Lutz's resource-bounded measure in E. These results can be interpreted as "it is likely that RP has measure zero in E, otherwise the unlikely ZPP = EXP is true".

What happens if the resource time bound of the martingales are reduced, by considering for instance the measure of [AS94] in the smaller class SUBEXP instead of E. Unfortunately the proofs of the zero-one laws in [Mel00] and [IM03] do not translate to smaller time bounds. For instance the proof in [Mel00] relies heavily on the following result from [IW98]: if BPP $\neq$ EXP, then every BPP algorithm can be approximated on infinitely many input lengths, by a subexponential time algorithm, which errs only on a small fraction of inputs. Using this result, a martingale succeeding on every BPP language can be constructed. The point is that this martingale needs to bet on *every* string, to ensure that the small fraction of strings on which the subexponential time approximation algorithm errs is too small to prevent the martingale from winning. Since martingales with small time bounds cannot bet on all strings, the proof of [Mel00] does not carry over to measure on small complexity classes.

In this paper we take another path to give some evidences that RP has measure zero in SUBEXP. We use recent results from [ABK+02] on the power of Kolmogorov random strings to prove that RP has measure zero in SUBEXP, otherwise ZPP = PSPACE and NP = EXP.

We also improve a result from [IM03], by reducing the power of the martingales from exponential to subexponential time. More precisely we prove that if NP is not small in SUBEXP , then Arthur-Merlin can be fully derandomized, thus gaining more evidence that NP = AM.

*Computer Science Department, University of Geneva, email: moser@cui.unige.ch

ISSN 1433-8092

2 Preliminaries

We use standard notation for traditional complexity classes; see for instance [BDG95] and [BDG90], or [Pap94]. For $\epsilon > 0$, denote by E_ϵ the class $\mathsf{E}_\epsilon = \bigcup_{\delta<\epsilon} \mathsf{DTIME}(2^{n^\delta})$. SUBEXP is the class $\cap_{\epsilon>0}\mathsf{E}_\epsilon$. Let us fix some notations for strings and languages. Let $s_0, s_1, \ldots$ be the standard enumeration of the strings in $\{0,1\}^*$ in lexicographical order, where $s_0 = \lambda$ denotes the empty string. Denote by $s_0^{(n)}, s_1^{(n)}, \cdots, s_{2^n-1}^{(n)}$ all strings of size n ordered lexicographically. A sequence is an element of $\{0,1\}^\infty$. If w is a string or a sequence and $0 \leq i < |w|$ then $w[i]$ and $w[s_i]$ denotes the ith bit of w. Similarly $w[i \ldots j]$ and $w[s_i \ldots s_j]$ denote the ith through jth bits. We identify language L with its characteristic function χ_L, where χ_L is the sequence such that $\chi_L[i] = 1$ iff $s_i \in L$. If w_1 is a string and w_2 is a string or a sequence extending w_1, we write $w_1 \sqsubseteq w_2$. Denote by $L_{=n}^{(k)} = L \cap \{s_0^{(n)}, \cdots s_{k-1}^{(n)}\}$. A language L is said polynomially dense if there exists a polynomial p, such that $|L_{=n}| \geq 2^n/p(n)$, where $L_{=n}$ denotes the set of strings of size n contained in L.

2.1 Pseudorandom Generators

The hardness of a generator is the size of the smallest circuit which can distinguish the output of the generator from truly random bits. More precisely,

Definition 1 *Let A be any language. The hardness $H^A(G_{m,n})$ of a random generator $G_{m,n} : \{0,1\}^m \longrightarrow \{0,1\}^n$, is defined as the minimal s such that there exists an n-input circuit C with oracle gates to A, of size at most s, for which:*

$$|\Pr_{x\in\{0,1\}^m}[C(G_m(x)) = 1] - \Pr_{y\in\{0,1\}^n}[C(y) = 1]| \geq \frac{1}{s}.$$

The A-oracle circuit complexity of a Boolean function f, denoted $Size^A(f)$ is defined as the size of the smallest circuit computing f. It was discovered in [KvM99] that the construction of pseudorandom generator from high circuit complexity Boolean functions does relativize.

Theorem 1 (Klivans-Melkebeek) *Let A be any language. There is a polynomial-time computable function $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$, with the following properties. For every $\epsilon > 0$, there exists $a, b \in \mathbb{N}$ such that*

$$F : \{0,1\}^{n^a} \times \{0,1\}^{b \log n} \to \{0,1\}^n,$$

and if r is the truth table of a $(a \log n)$-variables Boolean function of A-oracle circuit complexity at least $n^{\epsilon a}$, then the function $G_r(s) = F(r,s)$ is a generator, mapping $\{0,1\}^{b \log n}$ into $\{0,1\}^n$, which has hardness $H^A(G_r) > n$.

2.2 Measure on SUBEXP

In this section we describe the fragment of Lutz's measure theory that we will need. For a more detailed presentation of this theory we refer the reader to the survey by Lutz [Lut97].

The measure on resource-bounded complexity classes is obtained by imposing appropriate resource-bound on a game theoretical characterization of the classical Lebesgue measure.

A martingale is a function $d : \{0,1\}^* \to [0, \infty[$ such that,

$$d(w) = \frac{d(w0) + d(w1)}{2} \tag{1}$$

for every $w \in \{0,1\}^*$.

This definition can be motivated by the following betting game in which a gambler puts bets on the successive membership bits of a hidden language A. The game proceeds in infinitely many rounds where at the end of round n, it is revealed to the gambler whether $s_n \in A$ or not. The game starts with capital 1. Then, in round n, depending on the first $n-1$ outcomes $w = \chi_A[0 \ldots n-1]$, the gambler bets a certain fraction $\epsilon_w d(w)$ of his current capital $d(w)$, that the nth word $s_n \in A$, and bets the remaining capital $(1-\epsilon_w)d(w)$ on the complementary event $s_n \notin A$. The game is fair, i.e. the amount put on the correct event is doubled, the one put on the wrong guess is lost. The value of $d(w)$, where $w = \chi_A[0 \ldots n]$ equals the capital of the gambler after round n on language A. The player wins on a language A if he manages to make his capital arbitrarily large during the game. We say that a martingale d succeeds on a language A, if $d(A) := \limsup_{w \sqsubset A, w \to A} d(w) = \infty$, where we identify language A with its characteristic sequence χ_A. The success set $S^\infty[d]$ of a martingale d is the class of all languages on which d succeeds. Strategies satisfying Equation 1 with the equality sign replaced by "$\geq$" are called supermartingales.

In [AS94], Lutz's resource-bounded measure theory [Lut92] was generalized to small complexity classes. For the sake of completeness, let us recall the definitions of [AS94] adapted to the class SUBEXP. To define a measure on E_α, with $\alpha > 0$, supermartingales computed by Turing machines with random access to their inputs were considered in [AS94], that is on input w, the machine can query any bit of w to its oracle. In order to allow such Turing machines to compute the lengths of their inputs w without querying their oracles, they also have access to $s_{|w|}$; this convention is denoted by $M^w(s_{|w|})$.

It is widely believed that random access subexponential time Turing machines are too strong to define a measure on subexponential time classes, because it is not clear whether the whole class has not measure zero relatively to subexponential-time computed supermartingales. Therefore the concept was weakened in [AS94] by bounding the number of recursive queries such a Turing machine is allowed to make. Let M be a random access Turing machine and $n \in \mathbb{N}$. Define the query set $G_{M,n} \subseteq \{1, 2, \ldots, 2^{n+1}-1\}$ such that for every string $w \in \{0,1\}^*$ coding for words of size up to n, M can compute $M^w(s_{|w|})$ querying only input bits in $G_{M,n}$.

A measure notion on SUBEXP was introduced in [AS94] by considering subexponential-time random access Turing machines with subexponential size query sets.

Definition 2 *Let $\alpha > 0$. A class A of languages is said to have E_α-measure zero if there exists a supermartingale d, such that $d(w)$ is computable in time 2^{n^δ}, with $0 < \delta < \alpha$ and $n = |s_{|w|}|$, by a random access Turing machine with query set of size 2^{n^δ}, and such that $A \subseteq S^\infty[d]$.*

Such a supermartingale is called an E_α-computable supermartingale.

It is implicit in [AS94] that this measure notion satisfies all three measure axioms; i.e. first every single language $L \in \mathsf{E}_\alpha$ has E_α-measure zero, second E_α has not E_α-measure zero, and "easy" infinite unions of sets of E_α-measure zero have E_α-measure zero. More precisely,

Definition 3 *Let $\alpha > 0$. $X = \bigcup_{i \geq 1} X_i$ is an E_α-union of E_α-measure zero sets if there exists an indexed supermartingale d such that $X_i \subseteq S^\infty[d_i]$, where $d_i(w) = d(i,w)$ is computable in time $2^{n^\delta} + q(i)$, for some polynomial q, with $0 < \delta < \alpha$ and $n = |s_{|w|}|$, by a random access Turing machine with query set of size 2^{n^δ}.*

It is implicit in [AS94] that the third axiom holds for the measure notion on E_α .

Theorem 2 *[AS94] Let $X = \bigcup_{i\geq 1} X_i$ be an E_α-union of E_α-measure zero sets. Then X has E_α-measure zero.*

2.3 Resource-Bounded Kolmogorov Complexity

Let us give the basic notions on resource-bounded Kolmogorov complexity that we will need.

Definition 4 *(Levin) Let U be a universal Turing machine. Define $K_t(x)$ to be* $\min\{|d| + \log t : U(d) = x$ *in at most t steps$\}$.

There is a similar definition of space-bounded Kolmogorov complexity.

Definition 5 *Let U be a universal Turing machine. Define $KS(x)$ as* $\min\{|d| + s : U(d) = x$ *in space at most $s\}$.*

It was shown in [ABK$^+$02], that any polynomially dense set of strings with high time-bounded Kolmogorov complexity is complete for EXP under nondeterministic Turing reductions.

Theorem 3 *Let $R \in \mathsf{EXP}$ be a polynomially dense set and $0 < \delta < 1$ such that $r \in R$ implies $K_t(r) \geq |r|^\delta$. Then $\mathsf{NP}^R = \mathsf{EXP}$.*

A similar result for space-bounded Kolmogorov complexity was also shown in [ABK$^+$02].

Theorem 4 *Let $R \in \mathsf{PSPACE}$ be a polynomially dense set and $0 < \delta < 1$ such that $r \in R$ implies $KS(r) \geq |r|^\delta$. Then $\mathsf{ZPP}^R = \mathsf{PSPACE}$.*

3 RP is small else RP = PSPACE and NP = EXP

To prove our main result we will need the following theorem which essentially says that if RP is not small in E_ϵ, then P contains a dense set of strings with great Kolmogorov complexity.

Theorem 5 *Suppose RP has E_ϵ-measure non zero, for some $\epsilon > 0$, then there exists a polynomially dense set $R \in \mathsf{P}$ and $0 < \delta < 1$, such that for every $r \in R$, $K_t(r) > |r|^\delta$ and $KS(r) > |r|^\delta$.*

Proof. To prove Theorem 5, we will need the following lemma.

Lemma 1 *If RP has E_ϵ-measure non-zero, then there exists $0 < \gamma < 1$, and a language L in RP, and a probabilistic polynomial time Turing machine M deciding L such that: For all but finitely many $m \in \mathbb{N}$, $L^{(m)}_{=m}$ is non-empty* and *for all but finitely many $m \in \mathbb{N}$, for every $x \in L^{(m)}_{=m}$, every random string t such that $M(x,t) = 1$ has great Kolmogorov complexity, i.e. $K_t(t) > |x|^\gamma$ and $KS(t) > |x|^\gamma$.*

Proof. Consider the sum of the following two martingales d_1 and d_2 (enforcing the two conditions above). For every string's size n, d_1 only bets on the n strings $s^{(n)}_0$ to $s^{(n)}_{n-1}$. On input $(w, s_{|w|})$ where $s_{|w|} = s^{(n)}_i$ with $i < n$, d_1 checks that there is no element of L in the

range $s_0^{(n)}$ to $s_{i-1}^{(n)}$, and if so d_1 wagers $2^i/2^n$ that $s_i^{(n)} \notin L$, else d_1 stops betting on n-sized strings. If $s_i^{(n)}$ is the first element of L in the range, d_1 loses

$$|\sum_{j=0}^{i-1} \frac{2^j}{2^n} - \frac{2^i}{2^n}| = 1/2^n.$$

Since $\sum_{n\geq 0} 1/2^n$ converges, by starting with a finite amount of money we ensure that d_1 never goes broke, and its total losses are finite. On the other hand, each time there is no $x \in L$ between $s_0^{(n)}$ and $s_{n-1}^{(n)}$, d_1 wins $2^n/2^n = 1$. Thus, if this happens infinitely often, d_1's wins grow arbitrarily large. It is easy to see that d_1 can be computed in $2^{n^{\epsilon/4}}$ steps, and that the query set is polynomial-sized.

The second martingale d_2 only bets on the n first n-sized strings $s_0^{(n)}$ to $s_{n-1}^{(n)}$, for every length n. Fix an enumeration of polytime probabilistic Turing machines so that M_i always halts within n^j steps where $j \leq \log i$, and a deterministic universal Turing machine U. On input $(w, s_{|w|})$ where $s_{|w|} \in \{s_0^{(n)}, , \cdots, s_{n-1}^{(n)}\}$, d_2 simulates the first $\log n$ probabilistic machines on all random strings for inputs of length at most $\log n$. Let M_i be the first such machine that agrees with L on all such inputs, making errors only when $x \in L$, and then on less than $1/4$ of its random strings, i.e. decides L correctly (RP-wise) on all inputs smaller than $\log n$. d_2 simulates U on all programs d of size smaller than n^γ, during 2^{n^γ} steps, where $0 < \gamma < 1$ will be determined later. For every string t output during U's simulation, check whether $M_i(s_{|w|}, t) = 1$. If there is such a t, d_2 bets half its current capital that $s_{|w|} \in L$. Otherwise, d_2 does not bet.

Claim 1 *Let $L \in \mathsf{RP}$. Let M_{i_0} be the first probabilistic poly-time machine deciding L. Then if there are infinitely many lengths m, such that there exists $x \in L_{=m}^{(m)}$ and a random string t with $K_t(t) < |x|^\gamma$, such that $M_{i_0}(x, t) = 1$, then $L \in S^\infty[d_2]$.*

After a finite amount of time d_2 will always pick $i = i_0$. Thereafter for each length m such that there exists $x \in L_{=m}^{(m)}$ and a random string t with low Kolmogorov complexity, such that $M_{i_0}(x, t) = 1$, d_2 will always bet half its current capital that $x \in L$. Since M_{i_0} has one-sided error, this bet will be correct, and d_2's total stake will be multiplied by $3/2$. Otherwise, d_2 does not bet, so will never lose (once $i = i_0$). So if there are infinitely many such lengths m, d_2's capital grows unbounded.

The running time of d_2 is less than $2^{n^{\epsilon/4}}$, since on input $(w, s_{|w|})$ as above, there are $\log n$ machines to simulate on all inputs of size $\log n$, where each machine runs in time smaller than $(\log n)^{\log\log n}$, i.e. takes time $2^{(\log n)^{\log\log n}}$ to be simulated by trying all random seeds and taking a majority vote. Next d_2 needs to simulate U on 2^{n^γ} programs during 2^{n^γ} steps. Thus $d_2(w)$ can be computed in time $2^{n^{\epsilon/4}}$, by an appropriate choice of γ, with $0 < \gamma < 1$. The query set of d_2 is equal to $Q_{d_2}(n) = \{s_0^{(n)}, \cdots, s_{n-1}^{(n)}\} \cup \bigcup_{j=1}^{\log n} L_{=j}$, hence polynomial-sized.

This proves Lemma 1, since if RP does not have E_ϵ-measure zero, there must be a language $L \in \mathsf{RP}$ such that $d_1 + d_2$'s capital does not grow unbounded on L.

The proof of Theorem 5 then follows. Let L and M be as in Lemma 1, and suppose M runs in time n^k. Consider the set R of random strings t such that $M(x, t) = 1$ for some $x \in L_{=n}^{(n)}$, where $n^k = |t|$. Because $L \in \mathsf{RP}$, R is polynomially dense. Moreover if $t \in R$, then $M(x, t) = 1$ for some $x \in L_{=n}^{(n)}$, therefore $K_t(t) > |x|^\gamma > |t|^{\gamma/k}$. Putting $\delta := \gamma/k$ ends the proof. □

We are now ready to prove our main result stating that if RP is not meager in E_ϵ, then ZPP = PSPACE and NP = EXP.

Theorem 6 *For every $\epsilon > 0$,* RP *has E_ϵ-measure zero, else* ZPP = PSPACE *and* NP = EXP.

Proof. Suppose RP has E_ϵ-measure non-zero, for some $\epsilon > 0$, then by Theorem 5, P contains a polynomially dense set R of strings with high K_t and KS complexity. Since $\mathsf{NP} = \mathsf{NP}^R$ and $\mathsf{ZPP} = \mathsf{ZPP}^R$, Theorem 3 and 4 end the proof. □

4 Derandomization of AM if NP is not small

The following result says that if NP is not small in E_α, then NP = AM.

Theorem 7 *Suppose* NP *has E_α-measure non zero, for some $\alpha > 0$. Then* NP = AM.

Proof. In order to prove Theorem 7, we will need the following lemma.

Lemma 2 *If* NP *has E_α-measure non-zero then there is a language L in* NP *and a nondeterministic polynomial time Turing machine M deciding L such that: For all but finitely many $m \in \mathbb{N}$, $L^{(m)}_{=m}$ is non-empty* and *for almost every $m \in \mathbb{N}$, for every $x \in L^{(m)=m}$, every nondeterministic witness t such that $M(x,t) = 1$, has great* SAT*-oracle circuit complexity, i.e. $Size^{\mathrm{SAT}}(t) > |x|^\gamma$.*

Proof. The first property is easily verified by constructing the appropriate martingale d_1 similarly to Lemma with $K_t(t) < |x|^\gamma$ or $KS(t) < |x|^\gamma$, 1. For the second property consider the following martingale d, which only bets on the n first n-sized strings $s_0^{(n)}$ to $s_{n-1}^{(n)}$, for every length n. Fix an enumeration of nondeterministic Turing machines so that M_i always halts within n^j steps where $j \leq \log i$. On input $(w, s_{|w|})$ where $s_{|w|} \in \{s_0^{(n)}, , \cdots, s_{n-1}^{(n)}\}$, d simulates the first $\log n$ nondeterministic machines on all nondeterministic witnesses for inputs of length at most $\log n$. Let M_i be the first such machine that agrees with L NP-wise on all such inputs, i.e. decides L correctly (NP-wise) on inputs smaller than $\log n$. Next d constructs all circuits with oracle gates for SAT of size smaller than $|x|^\gamma$ on $\log i \log |x|$ inputs, where $0 < \gamma < 1$ will be determined later, and computes the truth table t for each. If $M_i(x,t) = 1$ for any such t, d bets half its current capital that $x \in L$. Otherwise, d does not bet.

Claim 2 *Let $L \in$ NP. Let M_{i_0} be the first nondeterministic poly-time machine deciding L. Then if there are infinitely many lengths m, such that there exists $x \in L^{(m)}_{=m}$ and a random string t, such that $M_{i_0}(x,t) = 1$, and t viewed as a function of $\log |t|$ inputs has circuit complexity less than $|x|^\gamma$, then $L \in S^\infty[d]$.*

After a finite amount of time d will always pick $i = i_0$. Thereafter we have that for each length m such that there exists $x \in L^{(m)}_{=m}$ and a nondeterministic witness t with small circuit complexity, such that $M_{i_0}(x,t) = 1$, d_2 will always bet half its current capital that $x \in L$. Since M_{i_0} decides NP-wise, this bet will be correct, and d's total stake will be multiplied by 3/2. Otherwise, d does not bet, so will never lose (once $i = i_0$). So if there are infinitely many such lengths m, d's capital grows unbounded. The running time of d_2 is less than $2^{n^{\alpha/4}}$, since on input $(w, s_{|w|})$ as above, there are $\log n$ machines to simulate on all inputs of size $\log n$, where each machine runs in time smaller than $(\log n)^{\log\log n}$, i.e. takes time $2^{(\log n)^{\log\log n}}$ to be simulated by trying all nondeterministic witnesses and taking a majority

vote. Next d needs to construct $2^{n^{2\gamma}}$ circuits of size at most n^γ and construct their truth table. Since evaluating a SAT gate of size at most n^γ takes time $2^{O(n^\gamma)}$, $d(w)$ can be computed in time $2^{n^{\alpha/4}}$, by an appropriate choice of γ, with $0 < \gamma < 1$. The query set of d is equal to $Q_{d_2}(n) = \{s_0^{(n)}, \cdots, s_{n-1}^{(n)}\} \cup \bigcup_{j=1}^{\log n} L_{=j}$, hence polynomial-sized. □

This proves Lemma 2, since if NP does not have E_α-measure zero, there must be a language $L \in \mathsf{NP}$ such that $d_1 + d$'s capital does not grow unbounded on L. □

The proof of Theorem 7 then follows, let L and M be as in Lemma 2, where M runs in time n^d. Let $L' \in \mathsf{AM}$ be any language and N a probabilistic nondeterministic Turing machine deciding it, and assume that on input of size n, N runs in time n^c. For $\epsilon = \gamma/d$ let a and b be as in Theorem 1 and pick m so that $m^\gamma > n^e$, where $e = \epsilon ac$. For every x among $s_0^{(m)}, \cdots, s_{m-1}^{(m)}$ nondeterministically guess a witness t for machine M on input x. Check whether $M(x,t) = 1$ – we know that there is at least one such witness for at least one such x, and every such witness has circuit complexity at least $m^\gamma = n^{\epsilon ac}$ – if so use Theorem 1 to construct from t a pseudorandom generator from $O(\log n)$ bits to n^c bits secure against circuits of size n^c with oracle gates to SAT. Use the outputs of this pseudorandom generator to simulate the nondeterministic Turing machine N for L'. This has expected nondeterministic polynomial time, and never errs for sufficiently large inputs, so $L' \in \mathsf{NP}$. □

References

[ABK+02] E. Allender, H. Buhrman, M. Koucky, D. van Melkebeek, and D. Ronneburger. Power from random strings. *Proc. IEEE Symp. on Found. of Comp. Sci. (FOCS)*, pages 669–678, 2002.

[AS94] E. Allender and M. Strauss. Measure on small complexity classes, with application for BPP. *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 807–818, 1994.

[BDG90] J. L. Balcazar, J. Diaz, and J. Gabarro. *Structural Complexity II.* EATCS Monographs on Theorical Computer Science Volume 22, Springer Verlag, 1990.

[BDG95] J. L. Balcazar, J. Diaz, and J. Gabarro. *Structural Complexity I.* EATCS Monographs on Theorical Computer Science Volume 11, Springer Verlag, 1995.

[IM03] R. Impagliazzo and P. Moser. A zero-one law for RP. *to be published in computational complexity conference*, 2003.

[IW98] R. Impagliazzo and A. Widgerson. Randomness vs time de-randomization under a uniform assumption. *Proceedings of the 39th IEEE Symposium on Foundations of Computer Science*, pages 734–743, 1998.

[KvM99] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial hierarchy collapses. *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 659–667, 1999.

[Lut92] J.H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Science*, 44:220–258, 1992.

[Lut97] J.H. Lutz. The quantitative structure of exponential time. In L.A. Hemaspaandra and A.L. Selman, editors, *Complexity Theory Retrospective II*, pages 225–260. Springer, 1997.

[Mel00] D. Melkebeek. The zero one law holds for BPP. *Theoretical Computer Science*, 244(1-2):283–288, 2000.

[Pap94] C. Papadimitriou. *Computational complexity.* Addisson-Wesley, 1994.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'