Open Problems Column
Edited by William Gasarch
This Issue’s Column!

Juris Hartmanis, one of the founders of modern complexity theory, passed away on July 29, 2022 at the age of 94. This column is a tribute to him. It is Open Problems by or Inspired by Juris Hartmanis

Request for Columns!

I invite any reader who has knowledge of some area to contact me and arrange to write a column about open problems in that area. That area can be (1) broad or narrow or anywhere in between, and (2) really important or really unimportant or anywhere inbetween.

Open Problems by or Inspired by Juris Hartmanis
Authors:
Eric Allender, Jin-Yi Cai, Lance Fortnow, William Gasarch
Neil Immerman, Stuart Kurtz, James Royer, Ryan Williams

1 Introduction

This column is a collection of open problem that were either by or inspired by Juris Hartmanis. There are many authors.

2 Extremely Spare Sets

by Eric Allender

In a paper by Hartmanis, Immerman, Sewelson [HIS85] they raised the following question:

If determinism and nondeterminism coincide for doubly-exponential time, can there be extremely sparse sets in NP-P?

They proved the answer was no. But there was a bug in the proof [All91]. In fact Allender showed that there exists an oracle where the answer is yes. Hence techniques that relativize will not suffice to resolve the question.

In the mean time, some non-relativizing techniques have been developed, as well as investigations into the limitations of those techniques. This suggests three possible directions:
1. Prove that the answer is **no**, though this will require techniques that do not relativize.

2. Prove that the result algebraizes, as defined by Aaronson and Wigderson [AW09a] and hence may be hard to resolve.

3. Prove that the answer is **yes**. We doubt such a proof will be found soon.

## 3 Algebraic and Transcendental Numbers

### 3.1 The Complexity of Algebraic Numbers

by William Gasarch

In Hartmanis & Stearns’s classic paper [HS65] they defined $\text{DTIME}(T(n))$. This is that part of the paper people usually point to. We will point to a different part: the complexity of real numbers.

Let $\alpha \in \mathbb{R}$. We want to know the complexity of $\alpha$. We say $\alpha \in \text{DTIME}(T(n))$ if there is a Turing machine that will, on input an empty tape, run forever and print the first $n$ digits of $\alpha$ in time $O(T(n))$.

A number is **algebraic** if it is the root of some polynomial in $\mathbb{Z}[x]$. A number is **transcendental** if it is not algebraic. We only deal with real numbers in this section (and the next).

Hartmanis & Stearns did the following:

- Observed that every rational is in $\text{DTIME}(n)$.
- Proved that every algebraic number is in $\text{DTIME}(n^2)$.
- Proved that there exists a transcendental number that is in $\text{DTIME}(n)$.

They asked the question

**Is there an algebraic number that requires $O(n^2)$ time?**

They noted that if there is then there would be an algebraic number that is more complicated than a transcendental number.

We list their question and a few more:

1. Does there exist an algebraic number that requires $\text{DTIME}(n^2)$? $\text{DTIME}(n^{2-\delta})$ for some $0 < \delta < 1$?
2. Is there any relation between the degree of an algebraic number (the min degree of a polynomial over $\mathbb{Z}$ that is satisfies) and its complexity? We suspect not.

3. By an easy diagonalization one can show the following: for all computable $T(n)$, there exists an $\alpha$ that requires time $T(n)$. The $\alpha$ is unnatural. Find concrete examples for such with $T(n) = n^2$ or $T(n) = n^3$ or whatever your favorite function is.

4. There is a vast literature on computing either the first $n$ digits (or bits) of $\pi$ or just the $n$th bit. This is another direction to go.

There has been little progress on their original question or our additions; however, Freivalds [Fre12] have a survey of the work that HAS been done.

3.2 Real Time Computability and Transcendental Numbers

by Jin-Yi Cai

A real number $r$ is said to be real-time computable if there exists a multitape Turing machine (TM) which on blank input, prints the binary expansion of $r$ and gives the first $n$ bits in $O(n)$ steps. P. Fischer, A. Meyer and A. Rosenberg [FMR70] showed that this is equivalent to the following: Some TM on input $1^n$ outputs the first $n$ bits in $O(n)$ steps. The trick is to run two parallel subroutines which take turns to print the first $n = 2^k$ bits for increasing $k$; when one prints (part of) the first $2^k$ bits, the other prepares the (second half of) $2^{k+1}$ bits.

Every rational number is real-time computable, as its expansion is eventually periodic—one can use a finite automaton. A more interesting example is the (decimal) Champernowne’s number

$$C_{10} = 0.12345678910111213141516171819202122\ldots$$

A TM $M$ can print the decimal values $k = 1, 2, 3, \ldots$ successively as follows: $M$ uses two tapes holding a counter $k$ on one tape and its head scanning left to right, and a second tape holding $k-1$, to be updated to $k+1$ with its head
going right to left. More interestingly, the following numbers are real-time computable:

\[ \sum_{n \geq 1} \frac{1}{2n^2}, \quad \sum_{n \geq 1} \frac{1}{2n^3}, \quad \sum_{n \geq 1} \frac{1}{2^n!}. \]

These numbers are real-time computable because its nonzero bits occur very sparsely. This latter property implies that they have very good binary rational approximations.

A number is algebraic if it is a root of a polynomial in \( \mathbb{Z}[x] \). It is transcendental if it is not algebraic. Proving specific numbers transcendental is a hard problem, and historically it is intimately related to how close a number can be approximated by rational numbers. Liouville pioneered this line of inquiry and showed that a non-rational algebraic number cannot have rational approximations that are too close. This was used by Liouville, in the 1850’s, to prove that transcendental numbers exist, and numbers such as \( \sum_{n \geq 1} \frac{1}{2^n} \) are transcendental. This method led to the proofs by (1) Hermite, in 1873, that \( e \) is transcendental, and (2) Lindemann, in 1882, that \( \pi \) is transcendental. The latter of course was the negative solution to the ancient Greek problem of squaring the circle. Mahler, in 1937, proved that \( C_{10} \) is transcendental. The transcendence of \( \sum_{n \geq 1} \frac{1}{2^n} \) was only proved in 1996, by Bertrand [Ber97] and Duverney et al. [DNNS96] (independently). The result required deep results about algebraic independence of values of Eisenstein’s series. The transcendence of \( \sum_{n \geq 1} \frac{1}{2^n} \) is still open. Contrast this with Cantor’s proof that transcendental numbers exist because the algebraic numbers are countable.

The question on how well an algebraic number can be rationally approximated culminated in Roth’s theorem, from 1955, that a non-rational algebraic number cannot be approximated better than the order \( \frac{\sqrt{\epsilon}}{q^{2+\epsilon}} \) by rational numbers of the form \( \frac{p}{q} \), for every \( \epsilon > 0 \). (Every real number has such approximations to the order \( \frac{\epsilon}{q^2} \).)

In their Turing award winning paper introducing time complexity classes in 1965, Hartmanis and Stearns proposed the following open problem:

\[ \text{Is every real-time computable number either rational or transcendental?} \]

This has become known as the Hartmanis-Stearns Conjecture. If it is true, it would imply a deep connection between transcendental numbers and computational complexity.
The Hartmanis-Stearns Conjecture is true for finite automata (FA), and in fact true for deterministic pushdown automata (see the paper by Adamczewski-Cassigne-Gonidec [ACG20] and the references therein).

A sequence is \( b \)-automatic if there is a FA, when given \( n \) in base \( b \) expansion, produces the \( n \)-th term at the end. They prove that irrational automatic numbers are transcendental. Their work uses a generalization of Roth’s theorem, and introduced a new criterion for transcendence. On the other hand, the work by Bailey-J. Borwein-Plouffe [BBP97] (see also J. Borwein & J. Borwein [BB87]) show that some transcendental numbers have surprisingly fast computable approximations. E.g., formulae such as

\[
\pi = \sum_{k \geq 0} \frac{1}{16^k} \left( \frac{4}{8k + 1} - \frac{2}{8k + 4} - \frac{1}{8k + 5} - \frac{1}{8k + 6} \right)
\]

can lead to quasi-linear time computation of the bits of \( \pi \). Yap [Yap10] (also see [LR13, Chapter 31] or [LR10]) showed that the digits of \( \pi \) can be computed in logspace. Allender et al. [ABDPar] improved this by showing that the digits of \( \pi \) can be computed in \( \text{TC}_0 \).

Concerning transcendental numbers, mystery persists. E.g., it is still unknown whether \( e + \pi, e\pi \) and \( \pi^e \) are transcendental, but \( e^\pi \) is. Hilbert’s 7th problem asks for a proof that if \( \alpha \neq 0, 1 \) and non-rational \( \beta \) are both algebraic, then \( \alpha^\beta \) is transcendental. This was proved by Gel’fond and Schneider (independently in 1934). It was generalized to the famous Baker’s theorem (1966): If \( \alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_k \) are nonzero algebraic numbers such that there are no non-trivial relations of the form \( \alpha_1^{n_1} \cdots \alpha_k^{n_k} = 1 \), then \( \log(\alpha_1^{\beta_1} \cdots \alpha_k^{\beta_k}) \) is transcendental. Relations of the form above also have connections to complexity theory lately, e.g., such relations—called lattice conditions—are used in the proof of dichotomy theorems for counting problems by Cai, Guo and Williams [CGW16].

4 Sizes of DPDA’s and PDA’s

by William Gasarch

Convention 4.1 A device will either be a recognizer (e.g., a DFA) or a generator (e.g., a regular expression). We will use \( \mathcal{M} \) to denote a set of devices (e.g., DFAs). We will refer to an element of \( \mathcal{M} \) as an \( \mathcal{M} \)-device. If
\( P \) is an \( \mathcal{M} \)-device then let \( L(P) \) be the language recognized or generated by \( P \). Let \( L(\mathcal{M}) = \{ L(P) : P \in \mathcal{M} \} \).

**Def 4.2** Let \( \mathcal{M} \) and \( \mathcal{M}' \) be two sets of devices such that \( L(\mathcal{M}) \subseteq L(\mathcal{M}') \). (e.g., DFAs and DPDAs). A *bounding function for (DPDA,PDA)* is a function \( f \) such that for all \( A \in L(\text{DPDA}) \), if \( A \in L(\text{PDA}) \) via a PDA of size \( n \) then \( A \in L(\text{DPDA}) \) via a device of size \( \leq f(n) \).

Valiant [Val76], and later Hartmanis [Har80] with an easier proof, showed the following:

**Theorem 4.3** If \( f \) is a bounding function for (DPDA, PDA) then \( \text{HALT} \leq_T f \).

Theorem 4.3 shows that there is a large difference between sizes of DPDA’s and PDA’s for the same language. However, the theorem does not give us a concrete example of a DPDA language which has a much smaller PDA than DPDA.

Beigel & Gasarch [BG16], drawing heavily on Filmus [Fil11], showed the following.

**Def 4.4**

1. \( W_n = \{ww : |w| = n\} \).
2. \( A_n = \overline{W_n} \).

**Theorem 4.5** For almost all \( n \):

1. There is a PDA of size \( O(n) \) for \( A_n \).
2. Any DPDA that recognizes \( A_n \) requires size \( \geq 2^{\Omega(n)} \).

Hence we have a double-exp gap between DPDA’s and PDA’s. Can we do better. Yes and No. Beigel & Gasarch [BG16] proved the following, drawing heavily on Hartmanis [Har80].

**Def 4.6** INF is the set of all Turing machines (actually their indices) \( M \) such that \( M \) halts on an infinite number of inputs.
Theorem 4.7 Let $f$ be such that $\text{INF} \not\leq_T f$. For infinitely many $n$ there exists a language $B_n$ such that:

1. There is a PDA of size $n$ that accepts $B_n$.
2. Any DPDA that accepts $B_n$ is of size $\geq f(n)$.

(The set $B_n$ is contrived. It involves Turing machines.)

Realize that for any computable $f$, including (say) Ackerman’s function, $\text{INF} \not\leq_T f$. So Theorem 4.7 seems like a much bigger separation than Theorem 4.5. But wait! Theorem 4.5 is an almost-all-$n$ result, whereas Theorem 4.7 is an infinitely-many-$n$ result. So they are actually incomparable. We would like to have the best of both worlds.

Open Question For some $f$ such that $2^n \ll f(n)$ show the following:

For almost all $n$ there exists a language $C_n$ such that:

1. There is a PDA of size $O(n)$ for $C_n$.
2. Any DPDA that recognizes $C_n$ requires size $\geq f(n)$.
3. Bonus points if $C_n$ is not constructed by diagonalization and does not involve Turing machines.

5 The Berman-Hartmanis Conjecture

5.1 The Berman-Hartmanis Isomorphism Conjecture: Origins

by Stuart A. Kurtz and James S. Royer

The Berman-Hartmanis Isomorphism Conjecture [BH77] posits that all NP complete sets are isomorphic under polynomial time isomorphisms.\footnote{Polynomial time computable 1-1 and onto functions whose inverse is also polynomial time computable} As evidence for their conjecture, they showed that SAT is paddable,\footnote{There is a polynomial time computable and invertible pairing function $p$ such that $p(x, y) \in A \iff x \in A$.} and isomorphic to all paddable NP complete sets, including all of the then-known “natural” NP complete sets.
Analogs to the Berman-Hartmanis Conjecture include the Cantor-Bernstein
Theorem of set theory\(^3\) and Myhill’s Theorem of computability theory\(^4\),
which are both true, and indeed the proof of Myhill’s theorem can be adapted
to the complexity-theoretic context, albeit at the cost of a slightly weaker
result, which we’ll call

**Theorem A:** If \( f : A \rightarrow B \) and \( g : B \rightarrow A \) are polynomial time computable,
1-1, *invertible, and length-increasing*, then \( A \) and \( B \) are polynomial-time
isomorphic.

Deborah Joseph and Paul Young observed that the gap between Theorem
A and the Isomorphism Conjecture is uncomfortably large. In particular, if
\( f \) is a polynomial-time 1-way function (in this case, meaning a function such
that the only polynomial-time computable sets contained in its range are
sparse), then \( f(\text{SAT}) \) will be \( \text{NP} \) complete, but non-paddable, and so not
isomorphic to \( \text{SAT} \), contracting the conjecture.

The Isomorphism Conjecture and related work were a strong drivers in
the emergence of *structural complexity*, which emphasized reductions and
degree structure as an approach to studying complexity theory. There are
oracles known relative to which the conjecture holds, relative to which it
fails, and (perhaps most surprisingly) relative to which it holds, even though
certain sorts of 1-way functions exist.

### 5.2 The Cylinder Conjecture

**by Lance Fortnow**

Suppose we wanted to find a counterexample to the Berman-Hartmanis
Isomorphism Conjecture [BH77]. Consider the following family of languages.

\[
f(\text{SAT}) = \{ f(x) : x \in \text{SAT} \}
\]

If \( f \) is polynomial-time computable, injective and length non-decreasing
then \( f(\text{SAT}) \) is \( \text{NP} \)-complete. For the rest of this section we will limit our-
selves to \( f \) that have these properties.

The idea is to find an \( f \) *complicated enough* so that \( f(\text{SAT}) \) is not iso-
morphic to \( \text{SAT} \). Deborah Joseph and Paul Young [JY85] first considered
this approach in their study of \( k \)-creative sets.

\(^3\)If \( f : A \rightarrow B \) and \( g : B \rightarrow A \) are 1-1, then there exists a bijective \( h : A \rightarrow B \).

\(^4\)If \( A \) and \( B \) are c.e. complete under computable 1-reductions, then \( A \) and \( B \) are
computably isomorphic.
In 1995 Stuart Kurtz, Steve Mahaney and Jim Royer [KMR95] define the notion of a scrambling function, a function $f$ such that the range of $f$ does not contain a non-empty paddable language, i.e., where there is a polynomial-time computable and invertible injective function $g$ such that for all strings $y$, $x$ is in $L$ if and only if $f(x, y)$ is in $L$. They show

1. For any scrambling function $f$, $f$(SAT) is not isomorphic to SAT.

2. Relative to a random oracle, scrambling functions exist.

As an immediate corollary, the Berman-Hartmanis conjecture fails relative to a random oracle.

Based on this work, the Berman-Hartmanis conjecture was generally considered likely false as scrambling or other similar functions seemed reasonably likely to exist in the unrelativized world. Or so we thought until 2009 when Manindra Agrawal and Osamu Watanabe [AW09b] showed that for the known one-way function candidates $f$, $f$(SAT) is isomorphic to SAT, at least non-uniformly.

Intuitively, Agrawal and Watanabe show that if $f$ has an easy cylinder than $f$(SAT) is isomorphic to SAT. A cylinder is a way to embed $\Sigma^*$ into an invertible range of $f$, informally two easy to compute functions $e$ and $g$ such that for all $x$, $g(f(e(x))) = x$. The formal definition they need is a bit more technical [AW09b, Definition 3]. Agrawal and Watanabe made the following conjecture:

**Conjecture** (Cylinder Conjecture) If $f$ is easy to compute then $f$ has a non-uniform easy cylinder.

Shortly after Agrawal and Watanabe made their conjecture, Oded Goldreich published [Gol11] a potential counterexample one-way function based on expander graphs. Goldreich’s function composes a function with itself several times depending on the input length. Agrawal and Watanabe counter that if one iterates a function with an easy cylinder in this manner, the iterated function should also have an easy cylinder, though they can’t prove this point.

The cylinder conjecture remains open and is likely the key to whether the isomorphism conjecture is true in the unrelativized world. While it can’t be settled with a relativizing proof, more evidence of the conjecture holding such as a proof that Goldreich’s function has an easy cylinder, or a more convincing counterexample would help us better conjecture whether the isomorphism conjecture may be true.
5.3 The BH Conjecture with Weaker Reductions

by Neil Immerman

Hartmanis and his student, Len Berman, considered the question of
whether all NP complete problems are the same or if they vary. Myhill had
proved that all r.e. complete problems are recursively isomorphic [Myh55].
Berman and Hartmanis made the following conjecture:

**Berman-Hartmanis Isomorphism Conjecture** [BH77]: If $A, B$ are NP
complete via ptime many-one reductions, then $A$ and $B$ are ptime isomorphic
($A \cong_p B$).

The Berman-Hartmanis Isomorphism Conjecture remains open. In par-
ticular, it implies that $P \neq NP$, but even the weaker conjecture – If $P \neq NP$
Then the Berman-Hartmanis Isomorphism Conjecture holds – is open.

In the late 1970’s, when Steve Mahaney and I were his grad students,
Hartmanis was very interested in proving structural properties of NP-complete
sets. Mahaney succeeded in doing just that, proving

**Mahaney’s Theorem** [Mah82]: If $P \neq NP$ then all NP-complete problems
are dense.

**Reductions**

Hartmanis was also interested in the fact that complete problems seem to re-
main complete via surprisingly weak reductions. Initially, Cook proved that
SAT is NP complete via ptime Turing reductions [Coo71]. When Karp pro-
duced many other important NP-complete problems, he used ptime, many-
one reductions [Kar72]. Jones showed that they stay complete via logspace
reductions [Jon75]. Hartmanis, Immerman and Mahaney showed that one-
way logspace reductions suffice [HIM78], in this model the transducer reads
its input once from left to right.

When I introduced Descriptive Complexity, I was pleased to see that
first-order reductions – which are the natural way to translate one logical
problem to another – preserve the completeness properties for all natural
complete problems for all complexity classes [Imm99]. When I mentioned
this to Mike Sipser, he pointed out that Valiant’s projections are weaker
and still preserve natural complete problems [Val82]. Projections are non-
uniform reductions which perform no computations: the $i$th output bit is
either always 0 or always 1, or it is a fixed bit, $b_{f(i)}$, of the input, or the
negation of a fixed bit, $\neg b_{f(i)}$. 

10
I was pleased to observe that there is a natural restriction of first-order reductions making them (first-order uniform) projections. I call these first-order projections (fops) and observe that natural problems stay complete via fops \[\text{Imml99}\]. A bonus is that fops are so well-behaved that we can prove an isomorphism theorem:

**Fops Isomorphism Theorem [ABI97]** For all important complexity classes \(C\), (this includes L, NL, P, NP, PSPACE, EXPTIME) every two problems complete for \(C\) via fops are first-order isomorphic.

I felt that this was a positive answer to the Berman-Hartmanis Conjecture, and Hartmanis agreed with me. Later, Agrawal strengthened our theorem to the following:

**First-Order Isomorphism Theorem [Agr01]** For all important complexity classes, \(C\), every two problems complete for \(C\) via first order reductions are first-order isomorphic.

**Dichotomy Phenomenon**

We are familiar with the fact that “natural” computational problems tend to be complete (and in fact complete via fops) for one of our favorite complexity classes. Furthermore, from the Fops Isomorphism Theorem, we can conclude that these natural problems are really a very small number of problems – only one for each of our favorite complexity classes.

**Open Problem**: Understand, explain and make use of this phenomenon.

A related and equally hard problem was frequently proposed by Hartmanis: separate complexity classes by proving that certain weak reductions do not exist. Here is one example. The problem \(\text{REACH}_d\) is the set of directed graphs of outdegree one having a path from \(s\) to \(t\). A quantifier-free projection (qfp) is a fop that happens to be quantifier-free.

**Theorem [Imml99]** \(\text{REACH}_d\) is complete for \(\text{DSPACE}[\log n]\) via qfps.

**Corollary** \((\text{NP} = \text{DSPACE}[\log n]) \iff 3\text{-COLOR} \leq_{\text{qfp}} \text{REACH}_d\)

**Open problem**: develop techniques towards proving that there is no qfp from 3-COLOR to \(\text{REACH}_d\).

6 Amplification/Magnification/Bootstrapping for SAT

by Ryan Williams
Juris and I discussed many problems, but one stands out for me as particularly prescient, given about 20 years of hindsight. We start with the following curious observation.

**Theorem.** There exists a fixed constant $c$ such that $P = NP$ if and only if $SAT$ is in $O(n^c)$ time.

The proof is trivial but nastily non-constructive. There are two cases. First, if $P = NP$ then $SAT$ is in $O(n^k)$ time for some $k$, so we may set $c = k$. Second, if $P \neq NP$ then the statement is true for every $c$.

**Open Problem:** Find an *explicit* constant $c$ for which the above theorem holds. (For example, does the theorem hold for $c = 10$?)

Juris told me a result like this would truly convince him that we’ve made progress on $P \neq NP$: from (say) an $n^{10}$-time lower bound for $SAT$, we could conclude a super-polynomial time lower bound. This came up while we were discussing Fortnow’s journal paper on time-space tradeoffs for $SAT$ [For00], and the possibility of improving the time lower bound beyond the golden ratio exponent established by Fortnow and Van Melkebeek in CCC’00 [FLvMV00], which I eventually did [Wil08].

I had forgotten about this conversation with Juris until recently. At the time, it felt like an impossible problem to me. However, several years after our discussion, Allender and Koucky [AK10] released their paper on *Amplifying lower bounds by means of self-reducibility*, showing how $n^{1+\varepsilon}$-type $TC^0$ lower bounds on (for example) $FORMULA$ $EVALUATION$ would imply $NC^1 \neq TC^0$ outright (you cannot compute $FORMULA$ $EVALUATION$ on $TC^0$ circuits of any polynomial size). It is possible that he was also aware of Aravind Srinivasan’s STOC 2000 paper [Sri00] showing that *weak-looking* time lower bounds on approximating $CLIQUE$ would imply $P \neq NP$ (I was unaware of it, at the time).

There are now many *amplification* results of a similar flavor, sometimes also called *hardness magnification* or *bootstrapping*, where one shows that a fixed-polynomial lower bound for one problem implies a super-polynomial lower bound for (possibly a different problem), by taking advantage of problem structure [AAW10, LW10, OS18, OPS19, CILM18, MMW19, CT19, CMMW19, CJW19, Hir20, MP20, CJW20, Fu20, CHO+22, CLY22]. (I have tried to be exhaustive, but there are many recent papers! I hope I didn’t leave yours out.)

---

5Sometimes when I’m feeling blue, I pull out an old email from Juris, documenting his response: *My very sincere congratulations!!! This is GREAT!!* May he rest in peace.
Maybe now, this open problem is not quite as impossible as it used to be. Variants of the problem are also just as interesting. Could it be that SAT is not solvable by an algorithm running in both cubic time and logspace if and only if NP $\neq \text{LOGSPACE}$? That would partially explain why it seems so difficult to prove a super-quadratic time lower bound for SAT against logspace machines (for both the decision version of SAT, and the search version).

References


https://doi.org/10.1007/978-3-642-22670-0_16.


https://doi.org/10.1109/SFCS.1978.31.

https://drops.dagstuhl.de/opus/volltexte/2020/12572/.


https://doi.org/10.1016/S0022-0000(75)80050-X.


https://theoryofcomputing.org/articles/v017a011/v017a011.pdf.


https://doi.org/10.1016/S0019-9958(76)90173-X.


[Yap10] Chee Yap. $\pi$ is in log space, 2010.  