# E-mail and the unexpected power of interaction

*László Babai* *
Eötvös University, Budapest
and
The University of Chicago

## Abstract

This is a true fable about Merlin, the infinitely intelligent but never trusted magician; and Arthur, the reasonable but impatient sovereign with an occasional unorthodox request; about the concept of an efficient proof; about polynomials and interpolation, electronic mail, coin flipping, and the *incredible power of interaction.*

About $MIP$, $IP$, $\#P$, $PSPACE$, $NEXPTIME$, and new techniques that do not relativize. About fast progress, fierce competition, and e-mail ethics.

## 1 How did Merlin end up in the cave?

In the court of King Arthur[1] there lived 150 knights and 150 ladies. "Why not 150 married couples," the King contemplated one rainy afternoon, and action followed the thought. He asked the Royal Secret Agent (RSA) to draw up a diagram with all the 300 names, indicating bonds of mutual interest between lady and knight by a red line; and the lack thereof, by a blue line. The diagram, with its $150^2 = 22,500$ colored lines, looked somewhat confusing, yet it should not confuse Merlin, the court magician, to whom it was subsequently presented by Arthur with the express order to find a perfect matching consisting exclusively of red lines.

Merlin walked away, looked at the diagram, and, with his unlimited intellectual ability, immediately recognized that none of the 150! possibilities gave an all-red perfect matching. He quickly completed the 150! diagrams, highlighting the wrong blue line in each, and ordered the servants to carry them into the throne room as evidence that Arthur had asked the impossible.

Of course not even a tiny fraction could fit in the throne room, but Arthur wouldn't even wait till the room filled up. He dismissed Merlin's procedure ("obviously, you overlooked a case") and ordered him to come back with a solution the next day. Arthur's diaries reveal another thought that was on his mind: "The lifetime of the universe wouldn't suffice to check all that crud. That's how the old fox wants to fool me."

Merlin *knew* that he was right, and he knew also that Arthur was reasonable. All Merlin had to do was to convince him, *in five minutes*, that there was no solution.

Fortuitously, in the cafeteria he bumped into an unassuming character dressed in brand new blue jeans. An East Bloc visitor, the man humbly introduced himself as Dénes König, number one expert on perfect matchings. "Frobenius also claims this title," he added without bitterness. "Are you perhaps interested in my mini-max theory?" Having, at last, found a willing listener, the visiting scholar forgot his French fries and the free ketchup, and began a passionate lecture about bipartite graphs, maximum matchings, and minimum covers. His new acquaintance was not the least bothered by his heavy accent and large gestures. Before long, Merlin found out that all he had to look for was a *König obstacle*: a set of $k$ knights all whose hearts burn for only $(k-1)$ ladies. Merlin immediately saw that indeed there was such an obstacle ($k = 79$). With some assistance from the hardly brilliant but quite reliable court astronomer, Arthur managed rapidly to check that the set of 79 knights indeed formed a König obstacle. Being thereby convinced of Merlin's truth, Arthur resigned to the impossibility of a perfect matching and began exploring other avenues to improve society.

The chronicles report that Merlin did not have to wait long for his next call.

One of Arthur's recent innovations had been the in-

*Partially supported by NSF Grant CCR 871007.

[1] For general background we refer to [Ma] and [BaM].

CH2899-3/90/0000/0030$01.00 © 1990 IEEE

troduction of forks at meals. When the Round Table was set for dinner, the noble knights were invited to leave their swords and take their assigned seats. Some of them savored the juicy legs of mutton that they could reach with the sturdy forks. Others, however, discovered more knightly uses of the new utensils, and wasted no time settling scores with their neighbors.

It seemed to Arthur that table manners could improve considerably with the right seating of the knights. As before, he proceeded to call the RSA, who produced a graph indicating who will sit in peace next to whom. Subsequently, the task of finding an appropriate seating arrangement (a Hamilton cycle in this graph, as it came to be called later) was assigned to Merlin.

When Merlin saw that there was no solution, he no longer tried the 150! diagrams trick, he just wrote to König, but either the mail was too slow or there was some other obstacle, König's reply to this letter never arrived. Being unable to produce a solution by the deadline, Merlin was sentenced to the gallows. That sentence was immediately commuted to "eternity in the cave; up for parole when 1/3 served".

## 2 December 13, 1989

The centuries passed in gray monotony. Merlin's only delights were the birds chirping at his window, and an occasional message through e-mail. Meanwhile, he conducted some theoretical studies and became proficient with MACSYMA. He heard about the Cook-Levin theorem through questions posted on a bulletin board, and came to be resigned to the recognition that the conjecture $NP \neq coNP$, if true, would allow the possibility that there was no way for him to ever convince Arthur.

At least so it seemed until December 13, 1989.

On this historic date, the weather was unusually agreeable (maybe a sign of global warming), but this did not prevent Merlin from having a late afternoon nap according to his daily routine. When he awakened, he went to the workstation to check the mail, as had been his custom ever since his murky quarters were first lit by the SUN.

The first five messages, broadcast to the entire Theorynet, raised profound questions of indubitable importance to the community at large. ("I am a graduate student, working on the problem of finding maximal star-like anti-snails in homeotoxal metagraphs in a distributed environment. Would anyone mind proving a theorem I could base my thesis on?" – "I am looking for a *simple* proof that there are infinitely many composite numbers." – Etc.)

But then came a message with a more limited distribution, and Merlin was pleased to see himself listed among the three dozen recipients or so. It took him a little while to appreciate fully *how* pleased he ought to be.

```
Date: Wed, 13 Dec 89 13:38:05 -0800
From: fortnow@gargoyle.uchicago.edu (Lance J.
To:  condon@cs.wisc.edu, jf@coma.att.com, ode
        shafi@theory.lcs.mit.edu, tompa@cs.w
        shamir@wisdom.weizmann.ac.il, sipser
        watanabe@cs.titech.ac.jp, rackoff@th
        merlin@cave.nyneve.gov, avi@hujics.h
                  *  *  *
        boyar@daimi.dk, aiello@flash.bellcor

Subject: IP contains PH

\documentstyle[11pt]{article}
\setlength{\topmargin}{-.5in}
\setlength{\textheight}{9in}
                    *  *  *
\setlength{\leftmargin}{4.5em}}}{\end{list}}
\begin{document}
\begin{center}
{\LARGE\bf The Polynomial-Time Hierarchy has
                 Interactive Proofs}\\
\ \\  \ \\
\rm\normalsize
\begin{tabular}{ccc}
Carsten Lund  & \hspace{1.5in} & \ \\
Lance Fortnow \\
Howard Karloff & & Noam Nisan \\
\\
Department of Computer Science & & Laboratory
for Computer Science\\University of Chicago &
&  Massachusetts Institute of Technology\\
Chicago, IL 60637 & & Cambridge, MA 02139\\
\end{tabular}
\end{center}

\begin{thm}
Every language in $BPP^{\# P}$ has an
  interactive proof system.
\end{thm}

We will show an interactive protocol for
```

```
verifying the permanent. Using the fact that
the permanent is $\# P$-complete (Valiant) we
                    *  *  *
Repeat using $B_3$ and $A_4$ and so on until
we have a single matrix $B_n$. Let
$A^{(n-1)}=B_n$.
\think{V} Reduce $n$ by one and repeat this
process until we have $n=1$ where the
permanent can be easily verified.
\end{protocol}
\end{document}
```

By the time the LaserWriter was done with printing the short document, Merlin had completed some calculations and was ready to send a message to Arthur.

```
Date: Wed, 13 Dec 89 19:42:14 BST
From: merlin@cave
To:  arthur

Sire,
in a separate message I am sending you a
matrix M of order 103,680,300, with entries
from {-1,0,1,2,3}. With reference to
L.G. Valiant, TCS 8 (1979), pp.189-201,
you will easily verify that the permanent
of M is 2^{43,200,000}-times the number of
Hamilton cycles of the ''seating graph''.
Let me know when you have polynomial time.
I will convince you with confidence
1-2^{-1000} that this permanent is zero.
Please review your precalc, especially
Horner's rule, and have your dice ready.
   Yours,          Merlin
P.S. Thanks for the network hookup.
P.P.S. As to reparations, I'll settle for
 a scholarship to Chicago.
```

# 3 The LFKN protocol

Let us recall (cf. [Mi]) that the permanent per$M$ of an $n \times n$ matrix $M = (m_{ij})$ is defined, just as the determinant, as a sum of $n!$ expansion terms

$$\mathrm{per}M = \sum_{\sigma} \prod_{i=1}^{n} m_{i,\sigma(i)},$$

where the summation extends over all permutations $\sigma$ of the set $\{1, 2, \ldots, n\}$. The difference is that no signs are attached to the expansion terms. In spite of the close resemblance of their algebraic expressions, the nature of these two functions could hardly differ more. The determinant is easy to compute, yet the evaluation of permanents even of modest size matrices is beyond hope.

While this did not hinder Merlin in seeing the value of the permanent, his more difficult task was to convince Arthur of the validity of the result. The new protocol devised by Lund, Fortnow, Karloff, and Nisan (LFKN) gave him the clue.

According to the LFKN protocol, Merlin had to state the permanent of the $n \times n$ matrix $M$ first. Subsequently, he had to state the permanents of a sequence of smaller matrices computed using random numbers generated by Arthur. The numbers stated had to meet some consistency criteria always checked by Arthur. The last matrix in the sequence was $1 \times 1$, so the last permanent stated was easily verified by Arthur. The protocol was such that if Merlin always stated the correct values, the consistency criteria were automatically met. On the other hand, if Merlin had cheated once, the consistency criteria would force him with overwhelming probability to continue cheating on smaller and smaller matrices. Eventually, he would be caught red-handed when falsely stating the value of a $1 \times 1$ permanent.

Therefore, if Merlin keeps giving consistent answers and states a correct value for the last $(1 \times 1)$ permanent, this circumstance should be accepted by Arthur as *overwhelming statistical evidence* that the value of the permanent of $M$, initially stated by Merlin, is correct. Indeed, Merlin would have had negligible chance of getting away with a wrong answer.

To describe the method, we first consider the following situation. Suppose $A$ and $B$ are $n \times n$ matrices, and Merlin has stated the values of their permanents. Arthur suspects that at least one of the two values stated is wrong but he does not know which. How can he force Merlin to make a false statement of the permanent of a single, known $n \times n$ matrix?

Here is the solution by Lund et al. Take the line $D(x) = (1-x)A + xB$ through $A$ and $B$ in the space of $n \times n$ matrices. The entries of $D(x)$ are linear functions of $x$; therefore per$D(x)$ is a polynomial $d(x)$ of degree $\leq n$. Arthur requests Merlin to reveal the coefficients of this polynomial. Since $d(0) = \mathrm{per}A$ and $d(1) = \mathrm{per}B$ (a consistency criterion), Merlin *must cheat* if he has cheated on at least one of $A$ and $B$. He is thus forced to state a polynomial $d_1(x)$ which is different from $d(x)$. Now Arthur selects a random member $r$ of a set of $N$ numbers, and records the

fact that Merlin has stated the value $d_1(r)$ for the permanent of the matrix $C = D(r)$. But the chance that this value is right is $\leq n/N$, for the polynomial $d(x) - d_1(x)$ cannot have more than $n$ roots.

In a similar fashion, if Merlin states the permanents of the $n \times n$ matrices $A_1, \ldots, A_k$, and one of these values is incorrect, but Arthur does not know which one, he can force Merlin with large probability to falsely state the value of a single, known $n \times n$ permanent. The procedure consists of replacing two matrices by one as above, and repeating this operation $(k-1)$ times.

Now the protocol to verify the stated value of the permanent per$M$ goes as follows. Let $M_{1j}$ be the minor of $M$ obtained by striking the first row and the $j^{\text{th}}$ column. Then

$$\text{per}M = m_{11}\text{per}M_{11} + \ldots + m_{1n}\text{per}M_{1n}. \quad (1)$$

Merlin has to state the value of each permanent in this equation; and the equation itself is a consistency criterion. It follows that if the stated value of per$M$ is wrong, then the stated value of the permanent of at least one of the $(n-1) \times (n-1)$ matrices $M_{1j}$ must also be wrong. Replacing these $n$ matrices by one as above, Merlin is forced with large probability to produce a single $(n-1) \times (n-1)$ matrix with a falsely stated value of the permanent. Repeating the process, a $1 \times 1$ matrix is reached in $n-1$ rounds. The probability that the value arrived at in the end is correct assuming per$M$ was stated incorrectly and all the consistency criteria along the way were met is less than $n^3/N$, negligible if $N$ is chosen to be large enough.

# 4 Interactive proofs

Merlin is at large again, perhaps sailing toward the New World. He will visit Yale first (he had known someone from that area[2]) before proceeding to the Great Lakes.

Wishing him fair winds and ample random interaction, we leave his tale and turn to a real story.

* * *

Two versions of interactive proof systems have been proposed independently in the mid 80's by Goldwasser, Micali, Rackoff [GMR] and this author [Ba].

[2]See [Cl].

An interactive proof is a game between two players: an all-powerful Prover (Merlin), and a randomizing polynomial time Verifier (Arthur). The players take turns writing strings of polynomial length on a tape. Merlin's strategy is described by a (deterministic) function of the input string $x$ and the strings previously printed on the tape. Arthur's moves are computed in polynomial time from the input, the strings previously printed on the tape, and the random bits currently and previously generated by Arthur. Within a polynomial number of moves, Arthur is required to enter one of two states: "Merlin wins", or "Arthur wins". This terminates the game.

The difference between the two systems is whether Arthur employs private coin flips [GMR] or public ones [Ba]. In the latter case, all Arthur has to do is flip the coins; and at the end, evaluate the game in deterministic polynomial time.

The proof protocol (the set of rules of the game) is correct if winning chances are *bounded away from 1/2:* for every input string $x$, either Merlin has at least $2/3$ chance of winning (if he plays optimally), or he has no more than $1/3$ chance of winning (no matter what his strategy). The *language* defined by such a protocol consists of those strings where Merlin is the favored player. The advantage can be amplified to make the uncertainty exponentially small by playing the game several times on the same input and picking the majority winner.

The class of languages defined by correct interactive proof protocols (the languages for which *membership* has interactive proofs) is denoted by $IP$.

If we interpret $NP$ as the class of languages admitting *efficient formal proofs* of membership (formal in the sense of the *Principia* [WR]), then $IP$ can be viewed as the class of languages admitting *efficient proofs* of membership *by overwhelming statistical evidence.* In this sense, $IP$ seems like a "slight" randomized extension of $NP$.

# 5 More complexity classes

When the game is limited to $t(n)$ moves on inputs of length $n$, we obtain the language class $AM(t(n))$ if Arthur moves first, and $MA(t(n))$ if Merlin moves first. We use the self-explanatory notation $AMA = AM(3)$, $MAMA = MA(4)$, etc. Clearly $A = AM(1) = BPP$, and $M = MA(1) = NP$. It was proved in [Ba] that for any fixed $k \geq 2$, $AM(k) = MA(k+1) = AM$ *(Collapse Theorem),* so the class $AM$ seems particularly robust. The Collapse The-

orem was extended in [BaM] to unbounded Arthur-Merlin alternation: for any (polynomially bounded) function $t(n) \geq 2$, we have $AM(2t(n)) = AM(t(n))$.

The $AM(t(n))$ notation was originally introduced for the public-coin version, and one can use $IP(t(n))$ for the analogously defined classes with private coins with Arthur moving first. Goldwasser and Sipser [GS] proved the surprising result that the two systems are equivalent for every $t(n)$: $IP(t(n)) = AM(t(n))$. In particular, the class denoted by $AM(\text{poly})$ in [Ba] and defined as $\bigcup_{k\geq 1} AM(n^k)$ is identical with $IP$.

It was noted in [Ba] that $AM \subseteq \Pi_2^P$ and $MA \subseteq \Sigma_2^P \cap \Pi_2^P$. The class $MA$ has a special place in the philosophy of efficient proofs since arguably it represents the languages with efficient *publishable proofs of membership* (no direct interaction between Merlin and Arthur is required; Arthur can flip the coins at any later date).

It was noted by Boppana, Håstad, and Zachos [BHZ], that *$AM$ does not contain $coNP$, unless the polynomial time hierarchy collapses to $PH = \Sigma_2^P = \Pi_2^P = AM$.* Indeed, the proof of this follows immediately from the Collapse Theorem: Assume $AM \supseteq coNP$. Let $L \in \Sigma_2^P$. Then $(\forall x)((x \in L) \Leftrightarrow (\exists^P y)((x,y) \in L_1))$, where $L_1 \in coNP \subseteq AM$. It follows by definition that $L \in MAM$ (Merlin starts with guessing $y$). By the Collapse Theorem, we infer that $L \in AM \subseteq \Pi_2^P$. Summarizing: $\Sigma_2^P \subseteq AM \subseteq \Pi_2^P$, hence $\Sigma_2^P = AM = \Pi_2^P = PH$. Q.e.d.

# 6 Relativized separation: building the emotional barrier

Just what is the power of interactive proofs? An earlier result of Papadimitriou [Pa] implies that $IP \subseteq PSPACE$. In reality, $IP$ looked substantially smaller than $PSPACE$; indeed it seemed plausible to this author that it did not include $coNP$.

At least $AM \not\supseteq coNP$, unless the polynomial time hierarchy collapses. But could more interaction (more rounds) help?

Bounded rounds cannot. But Aiello, Goldwasser, and Håstad [AGH] constructed an oracle $C$ such that $IP^C$ (and actually $AM(t(n))$ for any unbounded, polynomial time computable $t(n)$) did not contain $PH^C$. Comparing this with the inclusion $AM^C \subseteq \Pi_2^{P,C}$, true under every oracle $C$, we see that at least in a relativized world, more interaction may achieve more.

But then, Fortnow and Sipser [FS] came up with an oracle $D$ such that $IP^D \not\supseteq coNP^D$. So, more interaction did not seem to suffice to conquer $coNP$.

Perhaps more *provers* could help. Interactive proofs with *multiple provers* have been introduced by Ben-Or, Goldwasser, Kilian, and Wigderson [BGKW]. In this model, several provers interact with a verifier. The provers are separated and have no knowledge of the interaction of the verifier with other provers. Formally, each verifier is a (deterministic) function which, when applied to the information available to the prover at a given time, outputs that prover's next move. As before, the verifier is a polynomial time bounded randomizing Turing machine. For every input string, the acceptance probability must be bounded away from 1/2; i.e. either at least 2/3 for honest provers, or at most 1/3 for arbitrary provers. The set of languages thus recognized is denoted by $MIP$, a further extension of the concept of what is "efficiently provable".

Only a polynomial number of provers can interact with the verifier. [BGKW] show that actually, two provers always suffice. Fortnow, Rompel, and Sipser [FRS] observe that $MIP \subseteq NEXPTIME$: guess the strategies of the provers (i.e. the functions describing them; these are exponentially long tables), and check for all possible sequences of coin tosses of the verifier. (Here, $NEXPTIME = \bigcup_{k\geq 1} NTIME(2^{n^k})$.)

Can multiple provers really accomplish more than a single prover? In a sense they can. [BGKW] show that in this model, all languages in $NP$ have *zero knowledge* proofs, a statement that is false in the single prover model, unless the polynomial time hierarchy collapses (Fortnow [Fo1]).

But can multiple provers prove membership in a $coNP$-complete language? [FRS] observe that the oracle $D$ of Fortnow and Sipser, separating $coNP$ from $IP$, actually does the same for $MIP$: $MIP^D \not\supseteq coNP^D$.

After this separation, it took considerable courage for anyone to even try an interactive proof for as hard a function as the permanent, which, since last summer, has been known to be at least as hard as the polynomial time hierarchy (Toda [To]).

The person who mustered this courage was Noam Nisan. He surprised a few friends with a November 27, 1989 e-mail announcement that *the permanent had multi-prover interactive proofs.*

# 7 The cat is out of the bag

Nisan's result supported the view that the somewhat esoteric model of multiple provers is, not unexpectedly, vastly more powerful than single provers. But there was a more subtle message in this: a method was there that *did not relativize*, that beat the Fortnow-Sipser oracle. To those familiar with the fact that the oracle $D$, separating $coNP$ from $MIP$, was the very oracle originally constructed to separate $coNP$ from $IP$, the message was even more striking: *maybe* the permanent has a single prover protocol.

Owing to Lance Fortnow's insistent inquiry, this "maybe" turned into the LFKN protocol in Chicago on December 11 and was announced on e-mail two days later. Skepticism about the relevance of multiple provers gave way to great excitement: overnight, $IP$ became known to be enormously more powerful than previously suspected. Why then, would $\#P$ be the limit? Why couldn't $IP$ actually hit the roof, $PSPACE$ ?

I don't know how many people began feverishly working on this problem with the keen sense that there would be just *one* winner: the one who first announces the (hopeful) result on e-mail. That announcement would instantly kill all the competition.

There are indications that a dozen may be a modest underestimate. Of the losers of the race, only those who achieved worthwhile byproducts revealed themselves publicly.

A more modest question to consider was to prove that $IP = coIP$. A more ambitious question: can one reduce the LFKN protocol to bounded rounds (and thus, collapse the polynomial time hierarchy to $AM = \Sigma_2^P = \Pi_2^P$)? We should stress that *the LFKN protocol is the first one in the literature which seems to require an unbounded number of rounds.*

Similar questions regarding multiple provers also became the targets of renewed attacks.

# 8 Arithmetization of Boolean formulas

It was immediately clear that Valiant's theorem was not required for the proof that $IP \supseteq P^{\#P}$. A simple arithmetization of Boolean formulas allows one to adapt the LFKN protocol directly to verify the number of satisfying assignments.

Let $\varphi(x_1, \ldots, x_k)$ be a Boolean formula. We may assume it only involves ANDs and NEGATIONs (no ORs). We assign a polynomial $\widetilde{\varphi}(x_1, \ldots, x_k)$ to $\varphi$ inductively. We use the same symbols $x_i$ to denote the arithmetic variables as the Boolean variables. We set $\widetilde{(\neg\varphi)} = 1 - \widetilde{\varphi}$ and $\widetilde{\varphi \wedge \psi} = \widetilde{\varphi} \cdot \widetilde{\psi}$. It is then clear that on any (0,1)-substitution of the variables, the Boolean value of $\varphi$ will agree with the arithmetic value of $\widetilde{\varphi}$. Hence, the number of satisfying assignments of $\varphi$ is

$$\sum_{x_1 \in \{0,1\}} \cdots \sum_{x_k \in \{0,1\}} \widetilde{\varphi}(x_1, \ldots, x_k). \tag{2}$$

We note that the degree of the polynomial $\widetilde{\varphi}$ is not greater than the length of the formula $\varphi$. It is now easy to adapt LFKN to verifying the expression (2). Indeed, let

$$f_i(x_1, \ldots, x_i) = \sum_{x_{i+1} \in \{0,1\}} \cdots \sum_{x_k \in \{0,1\}} \widetilde{\varphi}(x_1, \ldots, x_k). \tag{3}$$

Then

$$f_i(x_1, \ldots, x_i) = f_{i+1}(x_1, \ldots, x_i, 0) + f_{i+1}(x_1, \ldots, x_i, 1), \tag{4}$$

and $f_0$ is the value of (2). Merlin states $f_0$ and the coefficients of the low degree polynomial $f_1(x_1)$; Arthur checks the consistency criterion (4) for $i = 0$ and selects a random number $r_1$ to be substituted for $x_1$. In the general step, Merlin states the coefficients of the polynomial $f_{i+1}(r_1, \ldots, r_i, x_{i+1})$ in the variable $x_{i+1}$ where $r_1, \ldots, r_i$ are random numbers previously selected by Arthur. Arthur checks the consistency (eqn. (4)) of Merlin's claim against the previously stated value of $f_i(r_1, \ldots, r_i)$ and selects the next random number $r_{i+1}$. The protocol ends when Arthur directly checks the value of $f_k(r_1, \ldots, r_k)$. If the length of $\varphi$ is $n$ then the probability that Merlin could get away with a lie is less than $nk/N$ assuming the $r_i$ are selected from a pool of $N$ numbers.

This completes the proof, without using permanents, of the result $IP \supseteq P^{\#P}$. An offshoot of these ideas is a new characterization of $\#P$ via certain straight line programs of polynomials, with consequences not related to interactive proofs.

A *positive polynomial program with binary substitutions* (PPPBS) is a sequence $(p_1, \ldots, p_m)$ of instructions such that for every $k$, one of the following holds:

(a) $p_k$ is 0 or 1 (constant polynomial);
(b) $p_k = x_i$ or $p_k = 1 - x_i$ for some $i \leq k$;
(c) $p_k = p_i + p_j$ for some $i, j < k$;
(d) $p_k = p_i p_j$ where $i + j < k$ *(retarded multiplication)*

(e) $p_k$ is obtained from $p_j$ by substituting 0 or 1 for one of its variables ($j < k$) *(binary substitution)*.

The program defines a sequence $(\widetilde{p}_i)$ of polynomials in several variables. The last of these polynomials, $\widetilde{p}_m$, is said to be computed by the given PPPBS.

Let $P_1, P_2, \ldots$ be a sequence of PPPBS's. We call such a sequence *uniform* if a polynomial time Turing machine, upon input $1^n$, computes the instruction set $P_n$. We also use $P_n$ to denote the polynomial computed by $P_n$. We assume that $P_n$ depends on the variables $x_1, \ldots, x_n$ only.

With this notation, the following characterization of $\#P$ holds [BaF]:

*A function $f : \{0,1\}^* \to \{0,1,2,\ldots\}$ belongs to $\#P$ if and only if there exists a uniform sequence $\{P_n\}$ of PPPBS's such that for every $n$ and every $\alpha_i \in \{0,1\}$,*

$$f(\alpha_1 \ldots \alpha_n) = P_n(\alpha_1, \ldots, \alpha_n).$$

A mod 2 randomized version of this result yields a characterization of the complexity class $BP{\cdot}\oplus{\cdot}P$; and a representation of similar nature exists for the levels of the polynomial time hierarchy. As a consequence, a somewhat simplified version of Toda's proof that $P^{\#P} \supseteq PH$ can be given in the context of polynomial straight line programs [BaF].

These methods were insufficient to overcome the difficulty arising from the exponential growth of the degree of polynomials involved when trying to arithmetize quantified Boolean formulas with unbounded quantifier alternations.

# 9 The game is over

The date was December 26. Christmas just ended, it was the fourth day of Hanukkah. A terse phrase was flashing from the screen.

```
From: shamir%wisdom.weizmann.ac.il@CUNYVM.CU
Subject:  IP=PSPACE
```

To those in Chicago, it was another reminder of how small the globe has shrunk.

It was Adi Shamir, 6000 miles and 8 time zones away, who found the clever trick to remove the obstacle of rapidly growing degrees.

Invoking the classical $PSPACE$-completeness result of Stockmeyer and Meyer [SM], what we need for the proof is an Arthur–Merlin protocol to verify the truth of a quantified Boolean formula (QBF).

Shamir's basic approach to arithmetizing a QBF is to replace, inductively, $(\exists x_1, \ldots, x_s)\varphi$ by $\sum_{x_1,\ldots,x_s \in \{0,1\}} \widetilde{\varphi}$, and $(\forall x_i)\varphi$ by $\prod_{x_i \in \{0,1\}} \widetilde{\varphi}$, where $\varphi$ is a partially quantified Boolean formula, and $\widetilde{\varphi}$ is its arithmetization. (The base step is an arithmetization of a quantifier free formula; the procedure described in the previous section will be adequate, although it differs slightly from Shamir's.) While it is clear that for a fully quantified Boolean formula $\varphi$, the value $\widetilde{\varphi}$ will be zero iff $\varphi$ is false (and positive otherwise), the main difficulty is that the intermediate polynomials may have exponentially large degree: each universal quantifier doubles the degree.

Shamir thus first modifies the QBF such that no variable will survive more than two universal quantifiers. Immediately to the right of each occurrence of a universal quantifier $(\forall x_i)$, he inserts a sequence of existential quantifiers serving to rename the variables quantified to the left of $(\forall x_i)$. Working from left to right, suppose we have already quantified, but not yet renamed the variables $x_1, \ldots, x_{i-1}$. Next comes $(\forall x_i)$. Then we insert the sequence $(\exists x'_1, \ldots x'_{i-1})(x_1 = x'_1) \cdots (x_{i-1} = x'_{i-1})$; and replace each later occurrence of $x_j$ by $x'_j$ $(j < i)$. Then we proceed to the next universal quantifier. (In the next round, the variables to be renamed will of course include the new $x'_j$.) This modified quantified Boolean formula $\varphi'$ is clearly equivalent to the original $\varphi$ and its length is bounded by the square of the length of $\varphi$. The arithmetization of $\varphi'$ goes from right to left exactly as described before, with the added rule that the subformula $(x_i = x_j)\psi$ is replaced by the polynomial $(x_i x_j + (1 - x_i)(1 - x_j))\widetilde{\psi}$. Again, $\widetilde{\varphi}' = 0$ iff $\varphi$ is false; and each intermediate polynomial has degree $O(n)$, where $n$ is the length of $\varphi$.

The only remaining obstacle to applying the modified LFKN protocol, as described in the previous section, is that the numbers involved may blow up. Clearly, there is a $2^{2^n}$ upper bound on the value of $\widetilde{\varphi} = \widetilde{\varphi}'$. Consequently (assuming $n > 5$), $\varphi$ is true iff there exists a prime $p < 2^n$ such that $\widetilde{\varphi}' \not\equiv 0 \pmod{p}$. Merlin should start by announcing $p$ along with a primality certificate, and continue according to the modified LFKN protocol over the field of order $p$.

Curiously, this protocol is *public coin*, thus it directly proves that $AM(\text{poly}) = PSPACE$. This implies $AM(\text{poly}) = IP$ without requiring the hashing technique employed in [GS].

## 10 Program verification

Suppose someone presents a very complex program that is claimed to compute the permanent modulo $p$ where $p$ is part of the input. How can we verify this?

First of all, given a matrix $A$, we can verify the value of per$A$, computed by the program, using the LFKN protocol with the program itself playing Merlin's role. (To compute the polynomial per$D(x)$, one uses the program to compute per$D(i)$, $i = 0, \ldots, n$, and interpolates.) This is an example of "instance checking", in the sense of Blum and Kannan [BK].

The next step is to test the program on $n^c$ random input matrices such as to obtain statistical evidence that the program works correctly on a $\geq 1 - n^{-2}$ fraction of the inputs.

Finally, one can modify the program to become *self-correcting* in the sense of Blum, Luby, Rubinfeld [BLR]: the new, randomizing program will be correct on every input with probability $> 1 - 2^{-n}$. This was discovered by Lipton [Li], following ideas of Beaver and Feigenbaum [BeF], and played a material role in guiding Nisan's approach [Ni]. The transformation is very simple. Instead of evaluating the permanent of the input matrix $A$, we compute per$(A + iB)$, $i = 1, \ldots, n + 1$, where $B$ is a random matrix. From this we interpolate the polynomial per$(A + xB)$, and set $x = 0$ to obtain a value for per$A$. If indeed the program failed on at most an $n^{-2}$ fraction of the inputs, then repeating this procedure $k$ times and taking a majority answer for per$A$, that answer will have probability $> 1 - 2^{-k}$ to be correct on $A$.

Shamir's protocol can be used to obtain the analogous results for instance checking, verification, and self-correction of $PSPACE$-complete programs. A byproduct of the $MIP = NEXPTIME$ result to be described in the next section is that the same holds for $EXPTIME$-complete languages [BaFL].

The only nontrivial previously known example of instance checking was graph isomorphism [BK], using the [GMW] interactive nonisomorphism protocol. Instance checking for $NP$-complete languages is an open question. These questions are closely related to the power of the prover(s) in interactive protocols, to be discussed in Section 12 (cf. [BaFL]).

## 11 Multiple provers: "efficient proofs" for $NEXPTIME$

The LFKN announcement prompted immediate progress in the multiprover model as well. Jin-yi Cai announced on e-mail on December 28, 1989, that *multiple provers can implement the LFKN protocol in a single round.* He added at the end of his paper that he had just learned about Shamir's Dec. 26 mailing and that his proof extended to the $PSPACE$ protocol as well.

The actual power of bounded round multi-prover protocols is still open, but without the restriction on the number of rounds the answer is known. Again, the easy upper bound turns out to be tight: $MIP = NEXPTIME$, quite an enormous complexity class for what might still be considered "efficiently provable". The proof of this result is more complex than the previous ones; it wasn't until Jan. 17, 1990, that the twenty-page manuscript hit the network.

Here I'll try to give a brief summary of the ideas. There will be two provers, and the protocol will run in many independent phases. If the provers wanted to cheat, they would have a fair chance ($n^{-c}$) of being caught in any particular phase. So their chance of getting away after $n^{c+1}$ phases would be exponentially small.

In each phase, all questions are directed to Prover 1, except the last one which is randomly selected by the Verifier from all the $\leq n^c$ questions he asked Prover 1. If Prover 2 gives a different answer, the provers lose. So the strategy of Prover 2 can be thought of as an oracle (an answer to all possible questions), and Prover 1 must represent the same oracle if he doesn't want to give a fair chance to being caught.

We can therefore think of the protocol as a single-prover protocol where the Prover is a fixed (but unreliable) oracle. What matters is that his answers do not depend on the Verifier's previous questions.

Let $L \in NEXPTIME$. Following Simon [Si], Peterson – Reif [PR], Orponen [Or] we first derive the $NEXPTIME$ version of the Cook-Levin $NP$-completeness theorem. For every input $x$, we obtain a 3-CNF formula $\Phi_x$ with an exponential number of variables $X(i)$ and an exponential number of clauses $C_i$ ($1 \leq i \leq 2^{n^c}$) such that $x \in L$ iff $\Phi_x$ is satisfiable. Moreover, the expression $C_i$ is polynomial time computable from $i$.

The basic idea is that the Prover-oracle should set his mind on a satisfying substitution $A$ for the vari-

ables $X(i)$ and we should somehow verify that $A$ indeed satisfies $\Phi_x$. Of course we cannot check all the clauses.

First of all we turn the polynomial time computation of the relation "the clause $C_i$ is $t_1X(b_1) \vee t_2X(b_2) \vee t_3X(b_3)$" (the $t_i$ are 0 or 1, expressing the presence or absence of negation in front of $X(b_i)$) into 3-CNF satisfiability via Cook-Levin, i.e. this relation holds if and only if $(\exists^P z)\varphi_x(z, \underline{i}, t, \underline{b}_1, \underline{b}_2, \underline{b}_3)$. We call $z$ a *witness* of this circumstance. (Here, $\underline{j}$ denotes the string of digits of the integer $j$, prefixed by an appropriate number of zeros. Such digits serve as Boolean variables and will subsequently turn into arithmetic variables. We use $t$ to denote the string $t_1t_2t_3$.)

Using the arithmetization described in Section 8, we can thus create a polynomial $F_x(z, \underline{i}, t, \underline{b}_1, \underline{b}_2, \underline{b}_3, v_1, v_2, v_3)$ which is nonzero precisely if $z$ is a witness that $C_i$ has the form described in the preceding paragraph, and the substitution $X(b_i) = v_i$ $(i = 1, 2, 3)$ *does not satisfy* $C_i$. An arithmetic expression for $F_x$ can be computed from $x$ in polynomial time.

Let now $G_x$ be the function obtained from $F_x$ by substituting the values $A(b_i)$ for $v_i$; and let $H_x = \sum G_x^2$, where the summation extends over all (0,1)-strings $z, \underline{i}, t, \underline{b}_1, \underline{b}_2, \underline{b}_3$ of appropriate length. Let moreover $J_x = H_x + \sum(A(i)(1 - A(i)))^2$, where the summation extends over all oracle queries $i$ $(1 \leq i \leq n^c)$. It is now clear that $J_x = 0$ *if and only if all values taken by A are (0,1), and A satisfies* $\Phi_x$. (The terms added last guard against one possible cheating: that the values $A(i)$ may not be Boolean.)

This completes the arithmetization of the question $x \in L$. In order for the "low degree polynomials" technique to apply, we have to turn $A$ into a low degree polynomial. Currently, $A$ is a Boolean function defined over $\{0,1\}^m$ where $m = n^c$. There is a unique way to extend any Boolean function to a multilinear function. We thus extend $A$ to a multilinear function $A : \mathbf{I}^m \to \mathbf{Z}$ where $\mathbf{I} = \{0, 1, \ldots, N-1\}$ for some large value $N$. If now the oracle stores this multilinear function, the LFKN protocol (as modified in Section 8) will work.

One way still remains for the Prover-oracle to cheat: he may store a very complex function, not even remotely multilinear, as $A$. In a separate protocol, of interest in its own right, we *test multilinearity of the oracle A*. The test will certainly accept if $A$ is multilinear, and very likely reject, if there is no multilinear function $g$ such that $A$ and $g$ agree on a $> (1 - m^{-c'})$ portion of their domain $\mathbf{I}^m$.

The multilinearity test will randomly select a polynomial number of lines $\ell$ in each coordinate direction in $\mathbf{I}^m$, and select a polynomial number of sample points on each $\ell$, to determine whether or not $A$ restricted to the sample points on $\ell$ form a linear function.

We need some more notation. Let $\mathbf{I}$ be as before. We call a function $A : \mathbf{I}^m \to \mathbf{Q}$ *$\epsilon$-approximately multilinear* if it agrees with a multilinear function on a $(1-\epsilon)$ fraction of its domain. Let $0 < \epsilon, \delta < (10m)^{-3}$. We call a line $\ell$ in a coordinate direction *$\delta$-wrong* with respect to $A$ if the restriction of $A$ to $\ell$ is not $\delta$-approximately linear.

The correctness of the multilinearity test rests on the following result [BaFL]:

*Assume $A : \mathbf{I}^m \to \mathbf{Q}$ is a function such that the proportion of $\delta$-wrong lines in every coordinate direction is $< \epsilon$. Then $A$ is $\Delta$-approximately multilinear, where $\Delta = 3m^2(\epsilon + \delta + 1/N)$.*

The proof of this result uses combinatorial techniques including simple eigenvalue calculation to estimate the expansion rate of a graph.

# 12 The power of the prover(s)

The LFKN result shows that a $\#P$-powerful (honest) prover is sufficient for an interactive proof of $\#P$ functions. It is an open question whether or not membership in *coNP*-complete languages can be proven with a prover in the polynomial time hierarchy.

The fact that a $PSPACE$-prover suffices for all of $IP$ has been known for some time (Feldman [Fe]); the LFKN-Shamir protocol gives an alternative proof. As a byproduct of the $NEXPTIME$ protocol, we obtain that a pair of $EXPTIME$ provers suffice for any language in $EXPTIME$. However, the exact prover power required for $NEXPTIME$ is not known.

# 13 Circuit reductions and publishable proofs

The multiple prover model has natural applications to circuit reductions. Let $L_1$ and $L_2$ be languages. Nisan observed [Ni]:

*If $L_1$ has a multiple-prover interactive proof system with provers of power $P^{L_2}$ and $L_2$ has polynomial size circuits then $L_1 \in MA$.*

Indeed, Merlin just guesses the circuits that compute the strategy of each prover; then Arthur simu-

lates the verifier using the circuits for provers. (As previously remarked, an $MA$-proof is "publishable".)

It follows that *if $EXP$ has polynomial size circuits then $EXP = \Sigma_2^P = \Pi_2^P = MA$.* The same result with $P^{\#P}$ and $PSPACE$ in the place of $PSPACE$ follows from the LFKN and Shamir protocols, respectively.

A further application concerns deterministic simulation of $BPP$. We say that a machine $M$ *weakly computes* the language $L$ if it computes $L$ for infinitely many input lengths. We say that $L$ admits *weak subexponential simulations* if for every $\epsilon > 0$ there exists an $\exp(n^\epsilon)$-time bounded Turing machine weakly computing $L$. Combining the ideas just described with previous results of Nisan and Wigderson [NW], we obtain [BaFNW]:

*$BPP$ admits weak subexponential simulations unless $EXPTIME = MA = \Sigma_2^p \subset P/poly$.*

# 14 Space bounded interactive proofs

While public coins are as powerful as private coins in polynomial time bounded interactive protocols [GS], this no longer seems to be the case if a space bound is added. Anne Condon was the first to study simultaneous time and space bounds for interactive proofs [Co] and proved, among other things, that with private coins, a polynomial time, log-space verifier can simulate all of $IP$, while his public coin counterpart will be restricted to recognizing languages in $P$ (see also [Fo2]).

In a most recent development (March 1990), a weak converse of the public coin result was derived by Fortnow and Lund [FL]: all languages in $P$ are recognized by a polynomial time, $O(\log^2 n/\log\log n)$-space verifier. Their proof builds on the arithmetization technique and the modified LFKN protocol (Section 8).

Although not related to the LFKN breakthrough, let me briefly mention some other results on space-bounded verifiers.

Interactive proofs with private coins and no time bound become immensely powerful even if the verifier is restricted to be a 2-way probabilistic finite automaton (2PFA). The are two different rejection rules to consider in this context. We always insist that if $x \in L$ then the verifier should accept $x$ with probability $> 2/3$; but if $x \notin L$, we may either require that the verifier reject with probability $> 2/3$ (strong model), or just that it accepts with probability $< 1/3$. Condon and Lipton [CL] show that in the weak model, *all recursively enumerable languages* are recognized by a 2PFA verifier. In the strong model, the languages recognized by a 2PFA verifier are within $ATIME(2^{2^{O(n)}})$ [CL] and include $E = DTIME(2^{O(n)})$ (Dwork, Stockmeyer [DS]).

Multiple provers add a great deal to the power of the strong model. Two-prover (and multi-prover) interactive proof systems with a 2PFA verifier, which halt on every input with probability 1, recognize precisely the recursive languages (Feige, Shamir [FeS], Condon, Lipton [CL]).

# 15 Conclusion

New simulation techniques, based on low degree polynomials, have recently been introduced in structural complexity theory. The first consequences have been startling for their simplicity and unexpectedness. While most of the new results concern various models of interactive proofs, some implications to more classical complexity classes as well as to other areas such as program verification have already been established.

A striking feature of the new techniques is that they *do not relativize*. This fact seems to bring the predictive value of relativized separation results seriously in question and may carry the promise of new ways to attack previously intractable simulation and separation problems such as the long standing open question of $BPP$ vs. $NEXPTIME$ ([HIS], [He]).

# 16 Some problems of e-mail ethics

Mathematicians have always been eager to communicate with one another (for no one else would listen or appreciate the subtleties of their thoughts), but this communication has for times immemorial been marred by rivalry, treachery, and self-protection bordering paranoia. More important than the idea itself has been the ego of the discoverer. Many a mathematician's life was crushed by the real or imagined theft of a treasured thought. And it is often posterity, rather than the actual rival, that does the rudest injustice.

Cardano tricked the formula for the third degree equation out of secretive Tartaglia on oath to keep it secret, yet published it 6 years later (1545). Although he frankly credited Tartaglia [Gi], the formula came to be called Cardano's nonetheless.

János Bolyai discovered the hyperbolic geometry around 1823, at the age of 21, and is likely to have completed its description by 1825. The work did not appear until seven years later, and then as an Appendix (see [Bo]) to a text by his father, Farkas Bolyai, himself a geometer and a professor at the Reformed College of Maros-Vásárhely[3] in Transylvania. Farkas sent a copy of the Appendix to his old friend from college, Carl Friedrich Gauss, just to learn that "all of the paper's contents ... coincide almost completely with my own reflections which I partly carried out thirty to forty-five years ago. ... I had intended to write it down little by little ..."(1832). And, although in a letter to someone else Gauss wrote that "this young geometer Bolyai is a genius of first order", he never accorded him any public recognition. Bolyai's life was shattered. His geometry is most often being referred to as Lobachevsky's, after the simultaneous and independent discoverer, who was later (on Gauss' suggestion) elected to the Göttingen Royal Society (and whom Bolyai also suspected of plagiarism - an unfounded charge; cf. [Leh]).

Mathematicians invented all sorts of devices to fend against such misfortune. Cipher was a favorite. Another method, used by Archimedes, was reminiscent of zero knowledge protocols. Out of Syracuse, he used to entertain his colleagues in Alexandria with lists of his recent discoveries, stated at first without proof. But just to forestall statements that "we had discovered all these ourselves", he inserted an occasional false statement or a practically insoluble problem among them [vdW]. The best known of these traps, remarkable for its computational complexity, was stated in a letter to Eratosthenes of Cyrene. It required to count "the cattle of the sun", based on a system of seven linear equations in eight variables and two quadratic conditions, all with very small coefficients. The system reduces to the Pell equation

$$t^2 - 4,729,494u^2 = 1,$$

of which the smallest positive solution has over 206,500 digits [Ar].

The French Academy of the 19th century created the institution of depositing "secret packets" with the Academy, allowing members later to resolve disputes over priority without forcing them to come out in the open too early. The protection provided by the secret packets clearly enhanced the privileged position of the Academy. – But publication in an obscure place was almost as good as a secret packet. The overconfident gentlemen of Paris (Cauchy in the first place) were dumfounded when they found out on May 24, 1847 that all their recent brilliant advances and ambitious claims (whether packeted or not) regarding Fermat's Last Theorem had been greatly surpassed and largely devastated in an 1844 memoir by E. E. Kummer, published in a Festschrift dedicated by the University of Breslau[4] to the anniversary of the University of Königsberg[5] [Ed].

One may ask whether there are any norms or moral principles to follow in the area of scientific communication, but the answer would probably just be volumes of merry or bitter stories.

Clearly, the speed of the medium is of prime importance. The fact that the records of the French Academy were published instantly gave the members some advantage over a German college teacher (which they were unable to exploit). At the same time, the members of the Academy seemed to live in a perpetual competition.

How is this all relevant to the effect of e-mail on the way the Theory of Computing is done?

During the past twenty years, the Theory of Computing has become one of the most intensely competitive areas of mathematics. This is plainly demonstrated by the FedEx bill this community generates at FOCS/STOC deadlines. The publicity gained by those who get the privilege of instant FOCS/STOC publication puts them by two years ahead of the competition, and in spite of all the worldwide democracy offered by the community, this advantage is extremely difficult to compensate for from outside.

Here are some of the concerns and dilemmas raised by the year-ends story of quick proofs via e-mail interaction. The *nature* of most problems is as old as science. It is the *dimension* that is new and may call for rethinking of some ideals.

- *The ideal (?) condition for scientific inquiry.*

  E-mail is capable of creating an *ultracompetitive atmosphere* on a much grander scale than any medium before. Full documents are being transferred at no cost to any number of addresses around the globe in minutes or hours at worst. No labels needed, no overhead, just add one more alias to your *.mailrc* file and hit a key. – Those favoring quiet work, pull the plug.

- *Who receives the privileged information?*

[3] Tirgu Mureş.

[4] Wrocław.

[5] Kaliningrad.

Such a mailing may give unprecedented information advantage to a well chosen, sizable, and consequently extremely powerful elite group. The group of recipients, as the events described exemplify, may be fully capable of making rapid advances before others would even find out that something was happening. Although such elite groups belong to the very nature of the hierarchy of scientific research (and the elites in question are among the most tolerant and open bunches in history and, I believe, even among current scientific communities), their sheer intellectual force combined with the information advantage makes them look from outside like an impenetrable fortress. Among those who did not receive any of the mailings were Toda, Razborov, all of East Europe, ...

- *Age of global communication: the age of a global intellectual supermarket?*

  Hungary has been accessible by e-mail since March 1990 (3 months too late for this competition), although for short messages only. We have seen e-mail from India, and even from the vicinity of Tienamnen square.

  This seems like a welcome development. But will the *diversity of thought* that now exists be preserved in the era when intellectual fashions are dictated by the strongest? Shall we see more Levins and Razborovs come out of the Kolmogorov school, bringing in so prominently different, yet profoundly relevant ideas?

- *Age discrimination?*

  It is surely coincidental that no more than two of the recipients were age 40 or over. Of course, kindhearted students and postdocs on the mailing list are likely to have passed on the information to some of their (former) advisors. – Teachers, beware: if your life (or grant) depends on timely information, you may be at the mercy of your (former) students.

- *Does an e-mail announcement to a substantial portion of the immediately affected experts count as publication?* Should one refer to them the way one does to tech reports?

  Dates and exact texts are documented with greater accuracy than would be with many journals[6] ([Ni], [LFKN], [Sh], [Ca], [BaFL]) – yet it is impossible to subscribe for such announcements, even though some of them may have wide circulation.

- When events proceed at such pace, one idea is built upon another well before the other had the slightest chance of being published, *whom will posterity credit with authorship of the final product?*

  Some believe that those who let the cat out of the bag prematurely can only blame themselves. Maybe so. After all, their move did not only serve the noble purpose of promoting the common good by sharing their joy over the beauty of the new ideas – they also instantly killed the unknown competitor, for this is an unconcealed and recognized purpose of such large mailings.

  But one of the remarkable features of our story is how relatively small each step was along the way to the striking final result. So, gentle reader, also as part of posterity, how will *you* refer to the $IP = PSPACE$ theorem ?

The race told in our story was triggered by the most gentle giant in the field [Ni]. Clearly, all he meant was to share a new and surprising insight, and he did so by sending separate communications to a presumably modest number of colleagues. I feel fortunate to have been one of the addresses, and I hope he'll keep me on his mailing list.

Nisan's mailing was not meant to dispose of competitors; it seems likely (although it cannot be said for certain), that he had no competition at the time. He could have continued quietly and conceivably achieved much of what has subsequently been done by a series of authors.

Instead, he chose to invite an undisclosed list of researchers to join. But then, those joining in had no option anymore but to compete and announce.

Here is the dilemma. If the initiator tells his ideas to his immediate colleagues only, others won't even have a chance to join in. But if a critical mass of recipients is believed to have been reached, the race is called automatically.

*E-mail is there,* for better or for worse. There is no way to slow it down. The question is, what to mail, whom to send it to. Maybe the longer the list, the better. Science is likely to benefit from wider communication.

[6] With blatant disregard for scholarly documentation, some journals, including *Advances in Mathematics* and *Advances in Applied Mathematics*, do not print the dates of submission.

But even at this breathtaking pace, one might take a minute's break once in a while, and think hard, how to be considerate. A tall order perhaps, but it might be worth a try.

**Acknowledgments.** I am grateful to Lance Fortnow for his penetrating insights and an exciting collaboration on the subject of this survey. I owe particular gratitude to Gene Luks for his comments on the fable; and to Noam Nisan, Adi Shamir, Jin-yi Cai, and all the others who have kindly kept me on their mailing lists.

# 17 Epilogue

1989 was an extraordinary year. A curtain ascended[7], a wall went down, and the dominoes fell in rapid succession. Arguably, the information revolution contributed to a real revolution.

# References

[AGH] Aiello, W., Goldwasser, S., Håstad, J.: On the power of interaction, in: *Proc. 27th IEEE FOCS*, 1986, pp. 368-379. Journal version to appear in *Combinatorica*.

[Ar] Archibald, R.A.: The Cattle Problem, *Amer. Math. Monthly* **25** (1918), 411-414.

[Ba] Babai, L.: Trading group theory for randomness, *Proc. 17th ACM Symp. on Theory of Computing* (1985), pp. 421-429.

[BaF] Babai, L., Fortnow, L.: A characterization of $\#P$ by straight line programs of polynomials, with applications to interactive proofs and Toda's theorem, Tech. Rep. CS90-02, Dept. Comp. Sci., University of Chicago, 1990.

[BaFL] Babai, L., Fortnow, L., Lund, C.: Nondeterministic exponential time has two-prover interactive protocols, University of Chicago Tech. Rep. CS90-03, Jan. 1990. (E-mail: Jan. 17, 1990.)

[BaFNW] Babai, L., Fortnow, L., Nisan, N., Wigderson, A.: $BPP$ has weak subexponential simulations unless $EXPTIME$ has publishable proofs, University of Chicago Tech. Rep. CS90-14, April 1990.

[7] Cf. [Sc].

[BaM] Babai, L., Moran, S.: Arthur–Merlin games: a randomized proof system, and a hierarchy of complexity classes, *J. Comp. Sys. Sci.* **36** (1988), 254-276.

[BeF] Beaver, D., Feigenbaum J.: Hiding Instances in Multioracle Queries, *Proc. 7th Symp. on Theoretical Aspects of Comp. Sci.* , *LNCS* 415 (1990), pp. 37-48.

[BGKW] Ben-Or, M., Goldwasser, S., Kilian, J., Wigderson, A.: Multi-prover interactive proofs: How to remove the intractability assumptions, *Proc. 20th ACM Symp. on Theory of Computing* (1988), pp. 113-131.

[BK] Blum, M., Kannan, S.: Designing Programs that Check Their Work, *Proc. 21st ACM Symp. on Theory of Computing* (1989), pp. 86-97.

[BLR] Blum, M., Luby, M., Rubinfeld, R.: Self-testing and Self-correcting programs, with Applications to Numerical Programs, *Proc. 22nd ACM Symp. on Theory of Computing* (1990), to appear.

[Bo] Bolyai, J.: *The Science Absolute of Space / Independent of the Truth or Falsity of Euclid's Axiom XI (which can never be decided* a priori), translated from Latin by G.B. Halsted, The Neomon, Austin TX 1896.

[BHZ] Boppana, R., Håstad, J., Zachos, S.: Does $coNP$ have short interactive proofs?, *Info. Proc. Letters* **25/2** (1987), 127–132.

[Ca] Cai, Jin-yi: The polynomial time hierarchy is provable by two provers in one round, *e-mail announcement*, Dec. 28, 1989.

[Cl] Clemens, Samuel Langhorne: *A Connecticut Yankee in King Arthur's Court*, 1889.

[Co] Condon, A.: Space bounded probabilistic game automata, in: *Proc. 3rd Ann. Conf. Structures in Complexity Theory*, IEEE, 1988, pp. 162-174.

[CL] Condon, A., Lipton, R.J.: On the complexity of space bounded interactive proofs, in: *Proc. 30th IEEE FOCS*, 1989, pp. 462-467.

[Coo] Cook, S. A.: The complexity of theorem proving procedures, *Proc. 3rd ACM Symp. on Theory of Computing* (1971), pp. 151-158.

[DS] Dwork, C., Stockmeyer, L.: Interactive proof systems with finite state verifiers, Tech. Rep. RJ6262, IBM Almaden, 1988.

[Ed] Edwards, H.M.: *Fermat's Last Theorem: A Genetic Introduction to Algebraic Number Theory*, GTM 50, Springer 1977.

[FeS] Feige, U., Shamir, A.: Multi-oracle interactive protocols with space bounded verifiers, *Proc. 4th Ann. Conf. Structures in Complexity Theory*, IEEE 1989, pp. 158-164.

[Fe] Feldman, P.: The Optimum Prover lives in $PSPACE$, manuscript, M.I.T.(1986).

[Fo1] Fortnow L.: The Complexity of Perfect Zero-Knowledge, In S. Micali, ed., *Randomness and Computation*, Advances in Computing Research Vol. 5 (1989), JAI Press, pp. 327-343.

[Fo2] Fortnow L.: Complexity-Theoretic Aspects of Interactive Proof Systems, Ph.D. Thesis, Massachusetts Institute of Technology, Laboratory for Computer Science, Tech Report MIT/LCS/TR-447 (1989).

[FL] Fortnow, L., Lund, C.: Interactive proof systems and alternating time-space complexity, University of Chicago Tech. Rep. CS90-11, March 1990.

[FRS] Fortnow, L., Rompel, J., Sipser, M.: On the power of multi-prover interactive protocols, *Proc. 3rd Structure in Complexity Theory Conf.* (1988), pp. 156-161.

[FS] Fortnow, L., Sipser, M.: Are there interactive protocols for $co-NP$ languages?, *Inf. Proc. Letters* 28 (1988), pp. 249-251.

[Gi] Gindikin, S.G.: *Tales of Physicists and Mathematicians*, Birkhäuser, Boston 1988.

[GMW] Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity and a methodology of cryptographic protocol design, in: *Proc. 27th IEEE FOCS*, 1986, pp. 174-187.

[GMR] Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems, *Proc. 17th ACM Symp. on Theory of Computing* (1985), pp. 291-304.

[GS] S. Goldwasser and M. Sipser: Private coins versus public coins in interactive proof systems, in: *Proc. 18th ACM STOC*, Berkeley CA 1986, pp. 59-68; final version appeared in: *Randomness in Computation* (S. Micali, ed.), Advances in Computing Research Vol. 5 (1989), JAI Press, pp. 327-343.

[HIS] Hartmanis, J., Immerman, N., Sewelson, V.: Sparse sets in $NP - P : EXPTIME$ versus $NEXPTIME$, *Inf. and Control* **65** (1985), pp. 158-181.

[He] Heller, H.: On Relativized Exponential and Probabilistic Complexity Classes, *Information and Computation* **71** (1986), 231-243.

[Leh] Lehrer, Tom: Lobachevsky.

[Le] Levin, L.: Universal'nyĭe perebornyĭe zadachi (Universal search problems : in Russian), *Problemy Peredachi Informatsii* 9: 3, pp. 265-266.

[Li] Lipton, R. J.: New directions in testing, manuscript (1989).

[LFKN] Lund, C., Fortnow, L., Karloff, H., Nisan, N.: The polynomial time hierarchy has interactive proofs, *e-mail announcement*, Dec. 13, 1989.

[Ma] Malory, Sir Thomas: *Le Morte Darthur*, William Caxton, 1485.

[Mi] Minc, H.: *Nonnegative Matrices*, Wiley 1988.

[Ni] Nisan, N.: co-SAT has multi-prover interactive proofs, *e-mail announcement*, Nov. 27, 1989.

[NW] Nisan, N., Wigderson, A.: Hardness vs. randomness, in: *Proc. 29th IEEE FOCS*,1988, pp. 2-11.

[Or] Orponen, P.: Complexity Classes of Alternating Machines with Oracles, *Proc. 10th ICALP* (1983), *Lecture Notes in Computer Science* 154, pp. 573-584.

[Pa] Papadimitriou, C.: Games against Nature, *Proc. 24th IEEE Symp. on Foundations of Computer Science* (1983), pp. 446-450.

[PR] Peterson, G., Reif, J.: Multiple-person alternation, *Proc. 20th IEEE Symp. on Foundations of Computer Science* (1979), pp. 348-363.

[SFM] Seiferas, J., Fischer, M., Meyer, A.: Separating Nondeterministic Time Complexity Classes, *JACM* **25** 1 (1978), pp. 146-167.

[Sc] Schmemann, Serge: Hungary Allows 7000 East Germans To Emigrate West / Scenes of Joy and Relief / Tough Decision by Budapest Severely Strains Solidarity of Soviet Bloc Allies, *The New York Times*, September 11, 1989, p. A1.

[Sh] Shamir, A.: IP=PSPACE, *e-mail announcement*, Dec. 26 1989.

[Si] Simon, J.: On Some Central Problems in Computational Complexity, Ph.D. Thesis, Cornell University, Computer Science Tech Report TR 75-224, 1975.

[SM] Stockmeyer, L.J., Meyer, A.R.: Word problems requiring exponential time, *Proc. 5th ACM STOC*, 1973, pp. 1-9.

[To] Toda, S.: On the computational power of $PP$ and $\oplus P$, in: *Proc. 30th IEEE FOCS*, 1989, pp. 514-519.

[Va] Valiant, L. G.: The complexity of computing the permanent, *Theoretical Computer Science* **8** (1979), 189-201.

[vdW] van der Waerden, B.L.: *Erwachende Wissenschaft*, Birkhäuser, Basel 1966.

[WR] Whitehead, A.N., Russell, B.: *Principia Mathematica*, Cambridge Univ. Press, London, 1910, 1927.

*Mailing address*

Department of Computer Science
University of Chicago
1100 E 58th Street
Chicago, IL 60637
E-mail: laci@cs.uchicago.edu

April 1990