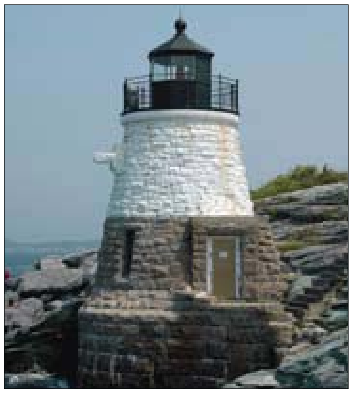
Storytelling About Lighthouses
Criticizing Professor Dijkstra Considered Harmless



By **Sorin Istrail**
Julie Nguyen Brown Professor of Computational and Mathematical Sciences and Professor of Computer Science

Introduction

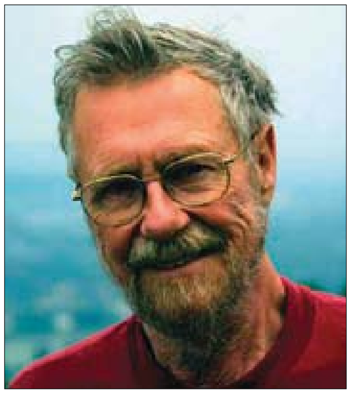
I have a growing sense that scientists of the 1940s, around the time of the Manhattan Project, developed a substantive toughness through the process of critical dialogue. They worked in an environment in which not only did they *not* shy away from colleagues' criticism, they sought it out with the expectation that exposing their ideas to the harsh light of criticism would enhance their scientific survival. Such dialogue made the work of science fun.

This article is about criticism, about personal experiences and observations that lead to the obvious conclusion that criticism should be encouraged and that it could and *should* be taught.

In many ways, I envision collegial criticism serving the same function as lighthouses: On one hand, lighthouses signal safe harbor—sail on. On the other, they warn of rough and hazardous shoals—beware and explore other routes. I am drawn to lighthouses as symbols of how scientific truth is won.

"In many ways, I envision collegial criticism serving the same function as lighthouses: On one hand, lighthouses signal safe harbor–sail on. On the other, they warn of rough and hazardous shoals–beware and explore other routes. I am drawn to lighthouses as symbols of how scientific truth is won."

Such beacons are the motivation for "Storytelling About Lighthouses," a series of articles for *Conduit* about inspiring scientists I've encountered in the random walk of my career. Telling such stories can be difficult, especially if they appear to be self-promoting or of dubious authenticity. Yet stories about these luminaries are priceless—they should be collected and shared, for they inspire long after their first telling. Certainly, one would prefer stories consistent with the following two impressionistic principles: Axiom 0, *Primary source material is prime;* and Axiom 1, taken from the Romanian proverb, *Dupa razboi multi viteji se-arata,* roughly translated as *"After the war, many heroes show up."* I look forward to your feedback and—in the spirit of collegial criticism—intriguing counterarguments, responses to my calls for priceless stories and your solutions to my silly games for consideration in future *Conduit* articles (sorin@cs.brown.edu).



Professor Edsger W. Dijkstra

Perhaps it is fitting to begin with "Criticizing Professor Dijkstra Considered Harmless," prompted by this year's 50th anniversary of *Communications of the ACM*, its new leadership and an exciting renaissance in the journal's next half century. In the January 2008 anniversary issue, the publication, in honor of E.W. Dijkstra, reprinted his paper "Go-To Statement Considered Harmful," the most famous letter to the editor ever published in *Communications of the ACM*. "Considered Harmful" became a Dijkstranian hallmark of critical reflection. [1] My article is about an anniversary of my own: Twenty-five years ago, I wrote my first letter to Professor Dijkstra.

Dijkstra the critic

Donald Knuth put it well in 1974: "A revolution is taking place in the way we write programs and teach programming… It is impossible to read the recent [Dijkstra] book, *Structured Programming*, without having it change your life. The reasons for this revolution and its future prospects have been aptly described by E.W. Dijkstra in his 1972 Turing Award Lecture, 'The Humble Programmer.' "[2]

Indeed, Dijkstra was an outspoken and critical visionary. A prolific writer, he authored more than 1,300 papers, many written by hand in his precise and elegant script. They were essays and parables; fairy tales and warnings; comprehensive explanation and pedagogical pretext. Most were about mathematics and computer science; others were trip reports that are more revealing about their author than about the people and places visited. This "Dijkstranian style" of writing flourished on the frontier between technical computing science and the philosophy substantiating its distinguished development.

It was his habit to copy each paper and circulate it to a small group of colleagues who would copy and forward the papers to another limited group of scientists. I have in my basement a box with several hundred papers from the series. [3] I read them with joy and excitement and my love for mathematics and computer science has been influenced in no small measure by his works. (The University of Texas has since digitized the Dijkstra manuscripts, known as EWDs, and makes them available online at http://www.cs.utexas.edu/users/EWD/. I hope the current generation of students and young scientists enjoys reading some of his papers and gets inspired.)

He offered criticism with a combination of dramatics and humor—an approach I liken to Don Quixote tilting at windmills. (Imagine my surprise when I met Dijkstra not far from a lighthouse in Newport and discovered that he resembled Peter O'Toole's Quixote in the movie *Man of La Mancha*.)

Take for example EWD498, "How Do We Tell Truths that Might Hurt?" In it, Dijkstra wrote that "the use of COBOL cripples the mind; its teaching should, therefore, be regarded as a criminal offense" and "it is practically impossible to teach good programming to students that have had prior exposure to BASIC. As potential programmers they are mentally mutilated beyond hope of regeneration." (I am curious whether he commented about C++ or Java. Might you have a story to share about this?) Other titles hint at the passionate arguments of his favorite themes. [4]

Nothing and absolutely no one was safe, not the "real" programmer, the "real" mathematician, the electrical engineer, the industrial manager, the "systems people" nor American computing science. [4] Not even von Neumann or Turing. "The fathers of the field had been pretty confusing: John von Neumann speculated about computers and the human brain in analogies sufficiently wild to be worthy of a medieval thinker and Alan M. Turing thought about criteria to settle the question of whether Machines Can Think, which we now know is about as relevant as the question of whether Submarines Can Swim." [5]

But whether he was lecturing on algorithm design, writing an essay on the need for rigorous mathematical thought or taking programmers to task, elegance and simplicity were Dijkstra's common denominators. His demand for elegance was based on his essential formation as a "pragmatic industrial mathematician." As he wrote in EWD538, *A Collection of Beautiful Proofs,* "we have to fight chaos, and the most effective way of doing that is to prevent its emergence."

One of Dijkstra's core beliefs was in mathematical rigor as the foundation for reliable software design. It was a philosophy he outlined in "Why Correctness Must be a Mathematical Concern," an inspiring keynote address (later published as EWD720) presented at the University de Liege, Belgium, in 1979. It was this presentation—and what he called a "silly game" played by one person with an urn and as many white balls and black balls as needed—that emboldened me to contact him.

(What is a "silly game"? What are the axioms for it? We'll save that discussion for another time.)

The first letter

It was 1983 and I was a junior researcher at the Computing Center of the University of Iasi, Romania, just four years out from my Ph.D. I had read Dijkstra's paper on correctness and after years of dreaming about corresponding with this inspiring and mesmerizing man, I felt I finally had something to say.

In my letter dated January 19, 1983, I solicited his comment and guidance on a technical report I had sent him previously. It contained two programming puzzles: "The Father-in-Law vs. the Pajamas" and "On a Chinese Olympiad Problem." The technical report was inspired by Dijkstra's art of problem solving—his "silly games." I wanted so much to master his style—striving for elegance in defining new puzzles, especially in the mathematical derivation of the algorithms that solved them.

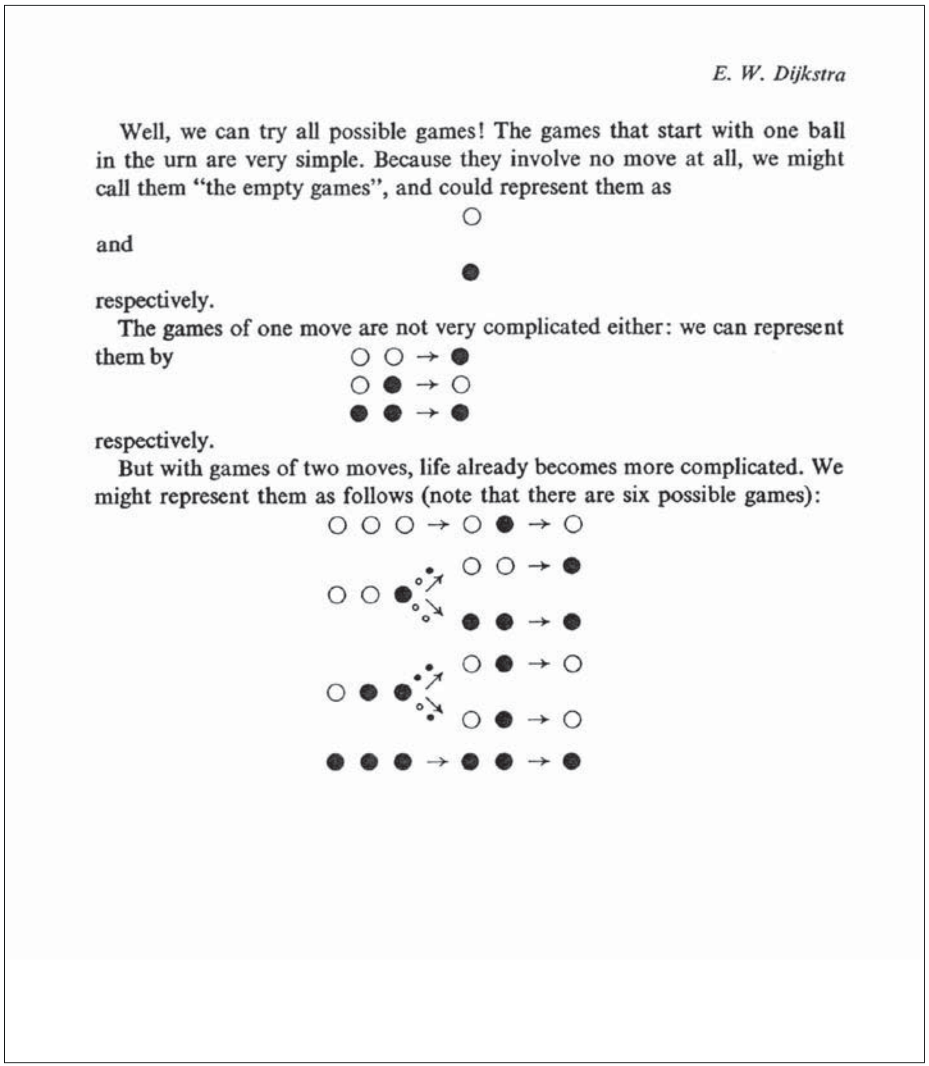
"I would be very much obliged if you could have a look at the problems… and if they deserve such a favor, please give me a reply," I wrote. "My deep hope is that you like these problems, and maybe use them in your celebrated conferences."

But my January 19 letter also included a manuscript I had written titled, "On the Facets of a Jewel." In it, I pointed out a certain mathematical difficulty concerning the game with the urn and balls described in EWD720. "My remarks point to some nice mathematical properties underlying the game and adding to its beauty. Shall I turn it into a publishable form?" I asked.

I never shared the manuscript with anyone, but in discussing my findings with colleagues, I told them I was considering sending the manuscript to Dijkstra.

Some colleagues suggested otherwise. Pointing out "some nice mathematical properties" and actually meaning "I found a certain difficulty with your problem" could be construed by this exceedingly tough perfectionist as a severe critique. I'd be committing professional suicide, they said.

I did not consider my manuscript a critique at all. Instead, I was eager to demonstrate to Dijkstra



E. W. Dijkstra

Well, we can try all possible games! The games that start with one ball in the urn are very simple. Because they involve no move at all, we might call them "the empty games", and could represent them as

○

and

●

respectively.

The games of one move are not very complicated either: we can represent them by

○ ○ → ●
○ ● → ○
● ● → ●

respectively.

But with games of two moves, life already becomes more complicated. We might represent them as follows (note that there are six possible games):

○ ○ ○ → ○ ● → ○

$$\begin{tikzcd} & \circ\ \circ \to \bullet \\ \circ\ \circ\ \bullet \arrow[ur, "\circ\bullet"] \arrow[dr, "\circ\circ"'] & \\ & \bullet\ \bullet \to \bullet \end{tikzcd}$$

$$\begin{tikzcd} & \circ\ \bullet \to \circ \\ \circ\ \bullet\ \bullet \arrow[ur, "\bullet\bullet"] \arrow[dr, "\circ\bullet"'] & \\ & \circ\ \bullet \to \circ \end{tikzcd}$$

● ● ● → ● ● → ●

Dijkstra's "silly game" of the urn and the balls, above, illustrates his teaching philosophy while offering insight into his patterns of thought.

how to enhance the beauty of his game. I thought he would like it. Rather than being a show of bravery, it was a show of excitement—or perhaps the stubbornness of a young man ignoring senior colleagues' advice. (Not that bravery is unrelated to criticism, but if it was then present in any small measure it was because of his inspiration.)

The "silly game"

Dijkstra's game of the urn and the balls magnificently illustrates his teaching philosophy while offering insight into his patterns of thought.

"You cannot expect me to explain in a few words what mathematics is all about… but I would like to show you one simple argument in order to give you in a nutshell some of the flavors of mathematics," he wrote in EWD720, "Why Correctness Must Be a Mathematical Concern."

"Consider the following silly game to be played by a single person with an urn and as many white balls and black balls as he needs. To begin with, an arbitrary positive number of balls is put into the urn and as long as the urn contains two or more balls, the player repeats the following moves: he shakes the urn and, without looking, he takes two balls from the urn; if those two balls have the same color he throws one black ball into the urn, otherwise he returns one white ball into the urn. Because each move decreases the total number of balls into the urn by 1, the game is guaranteed to terminate after a finite number of moves and it is not difficult to see that the game ends with exactly 1 ball in the urn. The question is: 'What can we say about the color of the final ball when we are given the initial contents of the urn?'"

On the facets of a jewel

The manuscript I sent offered my view that the problem statement was vague and imprecise. It also vindicated, through my mathematics, that the vagueness cannot be removed; a sort of "incompleteness."

The following excerpt from my paper conveys the key to my reasoning:

I have read for the first time your problem with the urn and the balls, in David Gries' monograph. [6] By following his advice, I spent 10 minutes on the problem. But neither did a solution come nor did I really start to solve it. In fact, I spent these 10 minutes trying to convince myself that having started with an initial content of the urn, the color of the final ball would be unique, i.e., it would not depend on the sequence of used rules …. It was clear to me that due to non-determinism, there are many ways to follow, but it was unclear whether all the ways led to Rome!

… What seemed clear was that the question demanded the final color as a function of the initial content only; the sequence of applied rules did not matter. My initial feeling was that the function might be undefined for some values of the arguments...

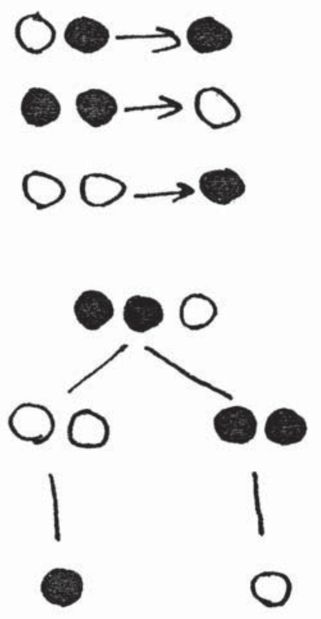
However, the question captured this case too–"what can we say" was: Nothing!

I cried out when I saw the solution: Extraordinary!!![7] I realized that the invariant pointed out by the solution assured the uniqueness of the final color—but somewhat a posteriori. *I felt then that "proving uniqueness" and "solving" were somewhat inseparable: a feeling close enough to the truth!*

Though I saw the solution, I couldn't explain in a transparent way why the color was unique; what was at hand seemed to be only an *a fortiori* proof.

Indeed, here is another game with non-unique final ball:

Rule0: W,B ->B, rule1: B,B->W, rule2:W,W->B for which {B,B,W} –r1->{W,W} –r2->{B} and {B,B,W}-r0->{B,B}-r1->W.



Clearly there must be a property that distinguishes the two games, assuring for the first the "uniqueness" property.

My paper concluded sharply:

"*The question of the problem contains a vague tone which cannot be made more precise. My initial desire of adding the statement '… it is simple to observe that the game has a uniqueness property, so find the* function *the game describes' is not advisable, and this is so because proving uniqueness is a task nearly as difficult as solving the problem. So, vagueness is the best possible form, giving at the same time a certain flavor to the question…*"

Programs and games

Let us think that the black ball is represented by 0 and the white ball by 1. Then the three rules of the game define a binary function from f:{0, 1}➔{0,1}. The function is commutative as indeed we pick the two balls together not in any particular order f(x,y)=f(y,x). With this notation, the move of the game becomes: "take two balls from the urn, {b0, b1} and return in the urn the ball f(b0, b1)." To see how this notation works, suppose that we have an urn with initial content {b0, b1, b2}. If the first pick is {b0, b1} then after the first move the urn has content{b2, f(b0,b1)}. After the second move then the urn will contain the ball f(b2,f(b0,b1)). Thinking this way, if all the plays starting from the initial contents of the urn end up with the same color for the final ball, i.e., is completely predictable, this is equivalent to the fact that all the f-expressions evaluate to the same value. This insight led me to the proof of the following:

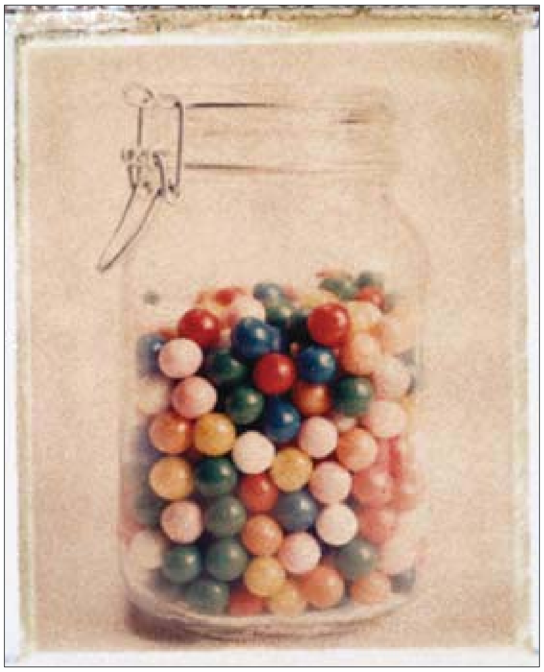
Theorem 0. A Dijkstra f-game with f commutative is completely predictable if and only if the function f is commutative and associative.

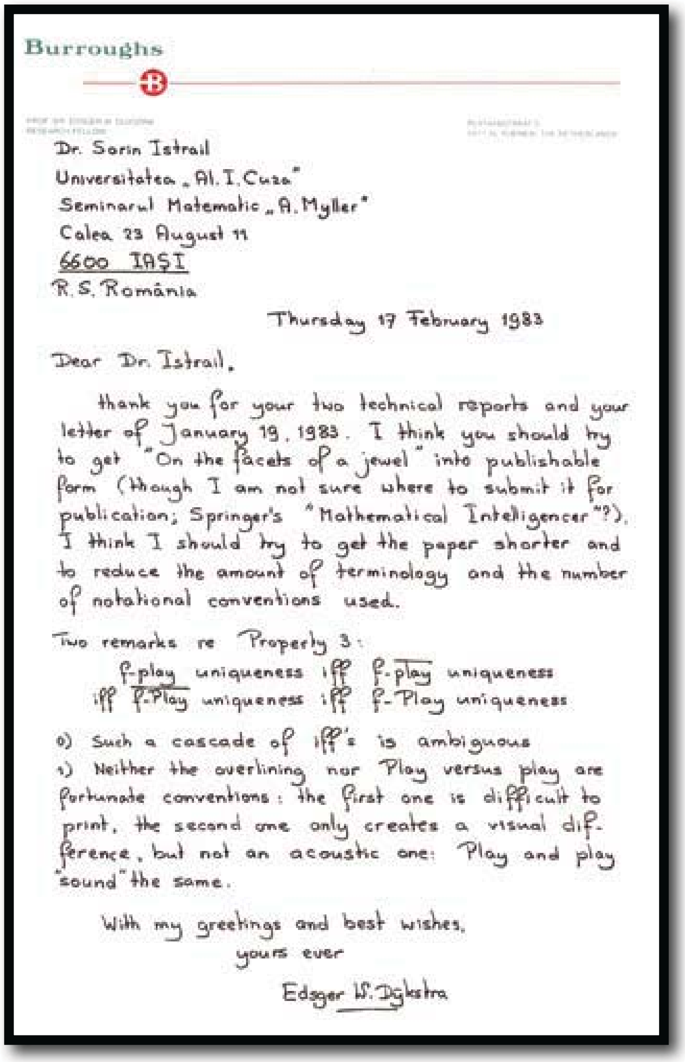
To understand the role of commutativity, I considered a new type of game, this time played with balls arranged in a sequence. (The distance between consecutive balls does not matter.) Pick two adjacent balls and return the resulting ball to the middle point of the removed two. This is equivalent to having an f-game that is no longer necessarily commutative. It turns out that we can prove the following:

Theorem 1. A Dijkstra f-game is completely predictable if and only if the function f is associative.

Anyone for a game on a Conway's Monster group Co1 with 4,157,776,806,543,360,000 elements? Any finite group would do. We start by bringing a contestant and do not share with her our secret from Theorem 0. We put in the urn a multi-set of elements of the group and we bet on the "color" of the final element in the urn. We can easily compute it (most of the time) and always win!

I sent my manuscript along with a letter seeking Dijkstra's forgiveness for its "somewhat sentimental" tone, a function of what I said was my "heartfelt desire" to correspond with him. But would that desire backfire as my colleagues suggested?





Burroughs

Dr. Sorin Istrail
Universitatea „Al. I. Cuza"
Seminarul Matematic „A. Myller"
Calea 23 August 11
6600 IAŞI
R. S. România

Thursday 17 February 1983

Dear Dr. Istrail,

thank you for your two technical reports and your letter of January 19, 1983. I think you should try to get "On the facets of a jewel" into publishable form (though I am not sure where to submit it for publication; Springer's "Mathematical Intelligencer"?). I think I should try to get the paper shorter and to reduce the amount of terminology and the number of notational conventions used.

Two remarks re Property 3:

f-play uniqueness iff $\overline{\text{f-play}}$ uniqueness
iff $\overline{\text{f-Play}}$ uniqueness iff f-Play uniqueness

0) Such a cascade of iff's is ambiguous
1) Neither the overlining nor Play versus play are fortunate conventions: the first one is difficult to print, the second one only creates a visual difference, but not an acoustic one: Play and play "sound" the same.

With my greetings and best wishes,
yours ever

Edsger W. Dijkstra

It's just business

My answer arrived from the Netherlands a month later in an envelope bearing Dijkstra's unmistakable handwriting.

Dear Dr. Istrail,

Thank you for your two technical reports and your letter of January 19, 1983. I think you should try to get "On the Facets of a Jewel" into publishable form (though I am not sure where to submit it for publication; Springer's "Mathematical Intelligencer"?) I think I should try to get the paper shorter and to reduce the amount of terminology and the number of notational conventions used."

He offered two excellent comments regarding one of the properties that I had noted and signed the letter, *"With my greetings and best wishes, yours ever, Edsger W. Dijkstra."*

There it was. A simple, elegant and generous response, scientist to scientist. Perhaps in reading the hundreds of EWDs I had discovered a message between the lines that my colleagues—concerned for my professional reputation—had not seen: Criticism is as fundamental to science as asking questions and Dijkstra was unafraid of honest, intellectual exchanges. As Michael Corleone said in *The Godfather*: It's nothing personal; just business.

But did he hold a grudge? We ultimately met face-to-face in Newport, R.I., in 1986. I call the episode "When Professor Dijkstra Slapped Me"—another story for another time.

What are the principles of criticism?

Clearly this is a difficult topic, yet it is important—criticism can and *should* be taught. But how? We should follow Dijkstra's lead and be substantively critical—verbally, by injecting tough questions at a technical talk, and in written analysis. Each has different challenges and inhibitions.

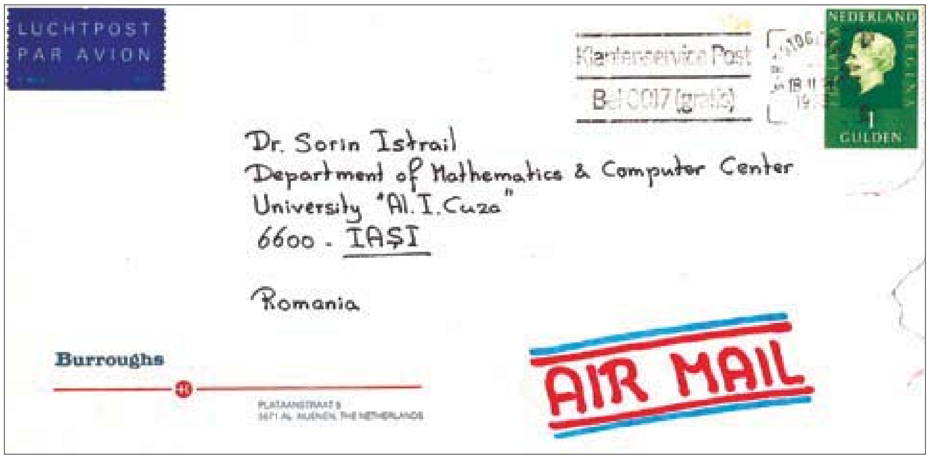
Why be critical at all? Clearly, it is easier to remain noncommittal. Obviously, there is resistance to opening your big mouth and asking a difficult question. You are moving from a state of equilibrium—of somewhat disengaged listening—to a state of non-equilibrium, on alert in dangerous territory. You would be making a statement, a public evaluation, perhaps pompous self-promotion—"Do you know who I am?"—in which your personal scientific weight is not unrelated to the seriousness of the answer from the speaker you critique.

What if this backfired? Are you prepared to clearly restate your point if the exchange becomes heated? Can you summarize eloquently and concisely the deep belief that triggered the comment without diluting the scientific integrity of the dialog?

In the end, substantive criticism says more about the critic than the critiqued. The unwritten rules of giving scientific talks are such that it is okay to ask tough questions; this is part of being alive scientifically. It is a lot of fun and to experience such rare and inspiring exchanges offers important lessons.

As for being on the receiving end of a tough question, how do you react? After all, it is not easy to receive criticism, especially in real time when you must respond coherently, defend your work and present counterarguments. On the plus side, being criticized means that the inquirer is so stimulated by your talk, she willingly leaves her equilibrium state to venture a question in order to learn more about your work.

I was privileged to write papers with Eric Davidson at the California Institute of Technology, Albert Meyer at MIT and Craig Venter at Celera Genomics—famously tough scientists who are legendary in their fields. Criticism offered by Davidson and Venter in the biological sciences bore a pronounced sense of urgency for the speed of discovery. Meyer, in computer science, delivered his criticism with intimidating, mathematically deep power. Venter's dramatic delivery was designed as a "poke in the eye" of deadlocked researchers. In sounding an alarm to leaders of the Human Genome Project about the "genome sequencing crisis," he echoed Dijkstra's



LUCHTPOST
PAR AVION

Klantenservice Post
Bel 0017 (gratis)

NEDERLAND
JULIANA REGINA
1
GULDEN

Dr. Sorin Istrail
Department of Mathematics & Computer Center
University "Al. I. Cuza"
6600 - IAŞI

Romania

Burroughs

PLATAANSTRAAT 5
5671 AL NUENEN, THE NETHERLANDS

AIR MAIL

alarm to the computing community about the "software crisis."

Through my years of close collaboration, I learned that their criticism, though passionate, pointed and pronounced, was nothing personal; it's just business. Criticism is essential.

But sometimes being the critic has its price.

In 1994, while working at Sandia National Labs, I had the pleasure of hosting David Botstein of Stanford University (now at Princeton), inventor of the RLFP molecular biology procedure that revolutionized forensic analysis.

Our meeting occurred during the O.J. Simpson trial and Botstein, an outspoken and eloquent critic, had remarked that "biological data has the O.J. Simpson problem: No matter how good the data looks, it is full of errors!" I told Botstein that because of the O.J. Simpson trial, society would have a better understanding of his discovery, which, in my view, would lead him to winning a Nobel Prize. He disagreed. Big awards have components of popularity contests and political games, he said, and being bluntly honest and critical would not always win brownie points. I know that my three distinguished and exceedingly critical collaborators are only too aware of this. But as Cervantes' novel was revolutionary in discussing the distinctions of class and worth, I hope that (as we will see from our criticism equation, "Responsibility" cancels out "Inconvenience") the Nobel Prize and Turing Award committees are hard at work to include my three collaborators and Botstein—lighthouses worthy of highest distinction in their classes.

The axioms

"Chivalry is only a name for that general spirit or state of mind which disposes men to heroic actions and keeps them conversant with all that is beautiful and sublime in the intellectual and moral world."[8]

As knight-errant, Don Quixote tried bravely to force his contemporaries to face a crisis in chivalric code. Similarly, Dijsktra fought forcefully to have the computer programming community face a crisis in software code. Dijkstra's criticism was the analogue of Quixote's lance. Honor was the founding and guiding principle of chivalry and of Don Quixote, leading to battles in honor's name. Likewise, Dijkstra's approach to programming as a high intellectual challenge was the founding and guiding principle of his battles against anti-intellectual solutions to program construction. "Real programmers don't reason about their programs, for reasoning isn't macho. They get their substitute intellectual satisfaction from not quite understanding what they are doing in their daring irresponsibility and from the subsequent excitement of chasing the bugs they should not have introduced in the first place."[9] Don Quixote's belief in enchantment parallels Dijkstra's belief in mathematical beauty and simplicity, always the ultimate goal of reliable software design.

The Association of Computing Machinery's 1972 citation for Dijkstra's Turing Award reads not only like an induction as the Knight of Programming, but also as the Spiritual Leader of the Software Code. "The working vocabulary of programmers everywhere is studded with words originally or forcefully promulgated by E.W. Dijkstra… but his influence on programming is more pervasive than any glossary can possibly indicate. The precious gift that this Turing Award acknowledges is Dijkstra's *style,* his approach to programming as a high, intellectual challenge… and his illuminating perception of problems at the foundations of program design… his memorable indictment of the go-to statement… We have come to value good programs in much the same way as we value good literature. And at the center of this movement, creating and reflecting patterns no less beautiful than useful, stands E.W. Dijkstra." The ACM-EATCS Edsger W. Dijkstra Prize in Distributed Computing recognizes that "no other individual has had a larger influence on research in principles of distributed computing."

His silly games are not just elegant mathematical puzzles. They go to the heart of computer science. They are *simplest but not simpler* about the exceedingly difficult task of writing reliable large programs. They are unique in highlighting subtle points mathematicians often miss. His urn and balls game is included in programming textbooks [6] as an example of a problem where design and testing would not quite do the job; it is the discovery of program invariants that holds the key.

Dijkstra's mathematical beauty axioms

In these axioms, "mathematics" and "computer science" are referred to especially in the context of "mathematical arguments relevant to automatic computing." And "mathematical beauty" is especially about the elegance of solutions and of proofs.

Axiom 0: Mathematical beauty is more important for computer science than for mathematics

Axiom 1: Proofs are more important than Theorems

Axiom 2: Mathematical beauty could and should be taught

Dijkstra's criticism axioms

How do we teach criticism? Here's one way: The NSF recently funded our proposal, "Sweatbox Q&A Boot Camp at Brown: Asking Tough Scientific Questions." I admiringly borrowed the concept from the Marine Biological Laboratory at Woods Hole, where legend says visiting speakers at its famous embryology course were brought to a warm room for a so-called sweatbox Q&A session.

Eric Davidson, for many years the course's teacher-in-chief, told me the story; our proposal also was inspired by his beacon of critical discourse. At the boot camp, Dijkstra's papers will be a must-read.

In talking about criticism, there are a several impressionistic quantities: authority (a), inconvenience (I), bravery (B), responsibility (R), substance (s) and energy of criticism (C). We have that C is proportional to R and B and that B and R are proportional to a, while R is proportional to I and s, and B is inverse proportional to I. It makes sense to define then $B = \frac{a}{I}$, $R = sIa$, and $C = RB$. It follows that "I" cancels and we get the *criticism equation* $C = sa^2$.

Axiom 0: It's nothing personal; just business [10]

Axiom 1: Principles only mean something if you stick to them when it's inconvenient [11]

Axiom 2: Authority is the speed of criticism

Though I've already stated my view that criticism is essential, I should mention that I also admire luminaries who have the opposite view. In fact, my hero-in-chief, John von Neumann, has put forward what we can call the von Neumann's criticism axiom, formulated by his daughter Marina von Neumann Whitman, who pointed out that he showed an impressive adherence to the old adage: *if you can't say something good about someone, don't say anything at all.*

The last letter

I did not share with Dijkstra, before his untimely death from cancer in 2002, my second set of results concerning the computational complexity of his urn and balls game. I would have enjoyed writing to him again about them. I probably would have written:

Dear Professor Dijkstra,

I have not written to you in a while. In 1983, I did more work on your urn and balls problem from "Why Correctness ..." but then lost the manuscript (or so I thought). Earlier this year, I rediscovered it in a box in my basement. I am now finally writing "On the Facets of a Jewel" and intend to submit it, as you advised, to Mathematical Intelligencer.

I am including a second manuscript, "On the Facets of a Jewel II," containing several results on computational complexity that are related to your game and generalizations. It is quite interesting that they recapitulate some of the deepest concepts of computing science, such as Chomsky grammars, graph theory, NP-completeness and the UNSOLVABLE. So much for silly games!

The 2007 Turing Awards for Model Checking, given to Professors Edmund Clarke, E. Allen Emerson and Joseph Sifakis, are a splendid tribute to "Dijkstra's dream"—an era when designing programs and their mathematical proof of correctness go hand in hand. As you wrote in "Why Correctness...," "The most general topic... of the widest significance could be called "the scaling up of mathematics." As far as the traditional mathematician is concerned, "there is a big, big difference: never in his life has [he] encountered such big formulae." The hard-won battles of so many around logic, automata and graph theory led to the discovery of these two beautiful islands of practical tractability: LTL and CTL (You may call it a case of "after-war heroes," but I would like to have seen cited the 1995 paper "Bisimulation Can't Be Traced," which Albert Meyer, then-student Bard Bloom and I published in the Journal of the ACM. *We showed, conceptually, the above "2" by proving the impossiblity of axiomatizing Bisimulation within the axiom systems of linear processes.[12]).*

As Johnny von Neumann pointed out, "The very concept of 'absolute' mathematical rigor is not immutable"(see

[13]). Nor is "program correctness proof," as we can see from the spectacular achievements of model checking, which could aptly be called, using Lewin's quote, "There is Nothing So Practical as a Good Theory" or—even better—"Practical Theory Considered Beautiful."

The breakthroughs we are witnessing in computer science in the 21st century, including those of the Turing awardees and of Dijkstra Prize-winner Maurice Herlihy, my next-door colleague at Brown, are clear indications that the era of scaling up of mathematics has arrived.

To dream the impossible dream
To fight the unbeatable foe
To bear with unbearable sorrow
To run where the brave dare not go

To right the unrightable wrong
To love pure and chaste from afar
To try when your arms are too weary
To reach the unreachable star

This is my quest
To follow that star
No matter how hopeless
No matter how far

To fight for the right
Without question or pause
To be willing to march into Hell
For a heavenly cause

And I know if I'll only be true
To this glorious quest
That my heart will lie peaceful and calm
When I'm laid to my rest

And the world will be better for this
That one man, scorned and covered with scars
Still strove with his last ounce of courage
To reach the unreachable star [14]

Thank you for everything.

Yours ever,
Sorin Istrail

Looking at the three single moves possible, we observe that the last two (○ ● → ○ and ● ● → ●) leave the number of white balls in the urn unchanged, while the first move (○ ○ → ●) reduces the number of white balls in the urn by two. In other words, each move leaves the so-called "parity" of the number of white balls in the urn unchanged: an even number of white balls in the urn remains even, and an odd number of white balls in the urn remains odd. In short: if the initial number of white balls is even, the final ball is black, and if the initial number of white balls is odd, the final ball is white. And that answers the question!

References

[0] I would like to thank Tracie Sweeney, my editor and writing partner, for her wonderful work on this article. I would also like to thank Erin Klopfenstein for her many contributions to this article. Thanks go also to Shriram Krishnamurthi for an insightful discussion about model checking and to Franco Preparata who inspired my search for one of the axioms. Copies of original documents related to this article can be found at http://www.cs.brown.edu/~sorin/dijkstra

[1] http://en.wikipedia.org/wiki/Considered_harmful

[2] E.W.Dijkstra, *"The Humble Programmer"* Communications of the ACM, Vol. 11, No. 3, pp. 147–148, 1968
D.E. Knuth," *Structured Programming* with go to Statements", Computing Surveys, Vol. 6, No.4, pp. 261–301, 1974

[3] *I would receive once a month from Teleprocessing Inc. an envelope with the latest EWD writings—about 10–15 of them.*

[4] EWD340: "The Humble Programmer"; EWD473: "On the Teaching of Programming i.e., on the Teaching of Thinking"; EWD480: "Craftsman or Scientist?"; EWD709: "My Hopes for Computing Science"; EWD898: "The Threats to Computing Science"; EWD920: "Can Computing Science Save the Computer Industry?"; EWD1036: "On the Cruelty of Really Teaching Computing Science"; EWD1095:"Are 'Systems People' Really Necessary?"; EWD 1209: "Why American Computing Science Seems Incurable"; EWD1304: "The End of Computing Science?"

[5] EWD898, "The Threats to Computing Science"

[6] D. Gries, *The Science of Computing*, Springer, 1981

[7] The solution of the silly game is in on one of the pages of this article. However, the reader must discover it.

[8] K. Henry Digby, *The Broad-Stone of Honour*

[9] EWD1012: "Real mathematicians don't prove"

[10] Mario Puzo, *The Godfather*, 1962

[11] A line from the 2000 movie *The Contender*, directed by Rod Lurie, about the strength and triumph of a male Democratic U.S. president who refused to be intimidated in his selection of a powerful woman as vice president.

[12] B.Bloom, S. Istrail, A. Meyer, *Bisimulation Can't Be Traced*, Journal of the ACM, Vol. 42, No. 1, pp. 232–268, 1995.

[13] Sorin Istrail and Tracie Sweeney, "Randomness is Beautiful: In Search of von Neumann," *Conduit* Magazine (Brown University, Computer Science Research and Alumni News Magazine, Spring/Summer 2006, p. 10–15)

[14] "The Impossible Dream" from *Man of La Mancha*, music by Mitch Leigh, lyrics by Joe Darion ■