

A reprint from

American Scientist

the magazine of Sigma Xi, The Scientific Research Society

This reprint is provided for personal and noncommercial use. For any other use, please send a request to Permissions, American Scientist, P.O. Box 13975, Research Triangle Park, NC, 27709, U.S.A., or by electronic mail to perms@amsci.org.
©Sigma Xi, The Scientific Research Society and other rightsholders

Recreational Computing

Erik D. Demaine

MARTIN GARDNER was a great man of many talents. He was an amateur mathematician, a puzzler, a professional magician, a debunker of pseudoscience, and a popular writer about all of these topics. He wrote more than 65 books and published a column, “Mathematical Games,” in *Scientific American* for 25 years, from 1957 to 1982. Because of his influence on countless readers, Gardner became known as the father of “recreational mathematics”—playful mathematical problems designed and solved purely for fun. Gardner’s accessible, inviting prose and his ability to correspond with impressive numbers of readers gave the general public the opportunity to enjoy mathematics and to participate in mathematical research. Many of today’s mathematicians, including myself, entered the field at least in part due to Gardner’s influence.

Sadly, Gardner died on May 22, 2010, at the age of 95. His death has been sorely felt by mathematicians around the world. But rather than dwell on our loss, I feel compelled to celebrate the tradition that Gardner started. Roughly every two years since 1993, Tom Rodgers has organized a conference in Atlanta called the Gathering for Gardner. It brings together mathematicians, puzzlers, magicians and debunkers who love the work of Martin Gardner and the spirit he embodied—playful intellectual curiosity. Gardner’s own absence from the Gathering since 1996 has not stopped it from continually growing in participation and intensity. The ninth Gathering, held last March, was the most prodigious yet, with 300 participants, a half-day sculpture-building party and two evening magic shows.

Erik Demaine is an associate professor in computer science at the Massachusetts Institute of Technology. Address: MIT CSAIL, 32 Vassar St., Cambridge, MA 02139. Internet: <http://erikdemaine.org/>

Puzzles and tricks from Martin Gardner inspire math and science

I am a theoretical computer scientist, which puts me at the boundary of computer science and mathematics. The goal of the field is to use mathematics to understand computation—what it is and what it can do. Readers of this column already know that computation is extremely powerful, offering new perspectives, approaches and solutions in perhaps every discipline. Computer science is highly unusual in this universality of influence—the only other example I know of is mathematics—and it’s what excites me about the field. The interdisciplinary field of “computational x ” is central to most fields where it has been considered (the “ x ” could be biology, chemistry, neuroscience, geometry, linguistics, finance, and so on)—and for other fields, I believe it is simply yet to be discovered.

What I’d like to show here is that computation is a useful way to think about more recreational pursuits, too—specifically puzzles and magic. Martin Gardner is my inspiration: He did not consider puzzles, magic and mathematics as separate pursuits, but blurred the traditional boundaries between them. He routinely illustrated mathematics using puzzles and magic, and he studied puzzles and magic using mathematics. I like to apply the same spirit to theoretical computer science, where the computational perspective offers new ways to think about puzzles and magic—specifically, how to design chal-

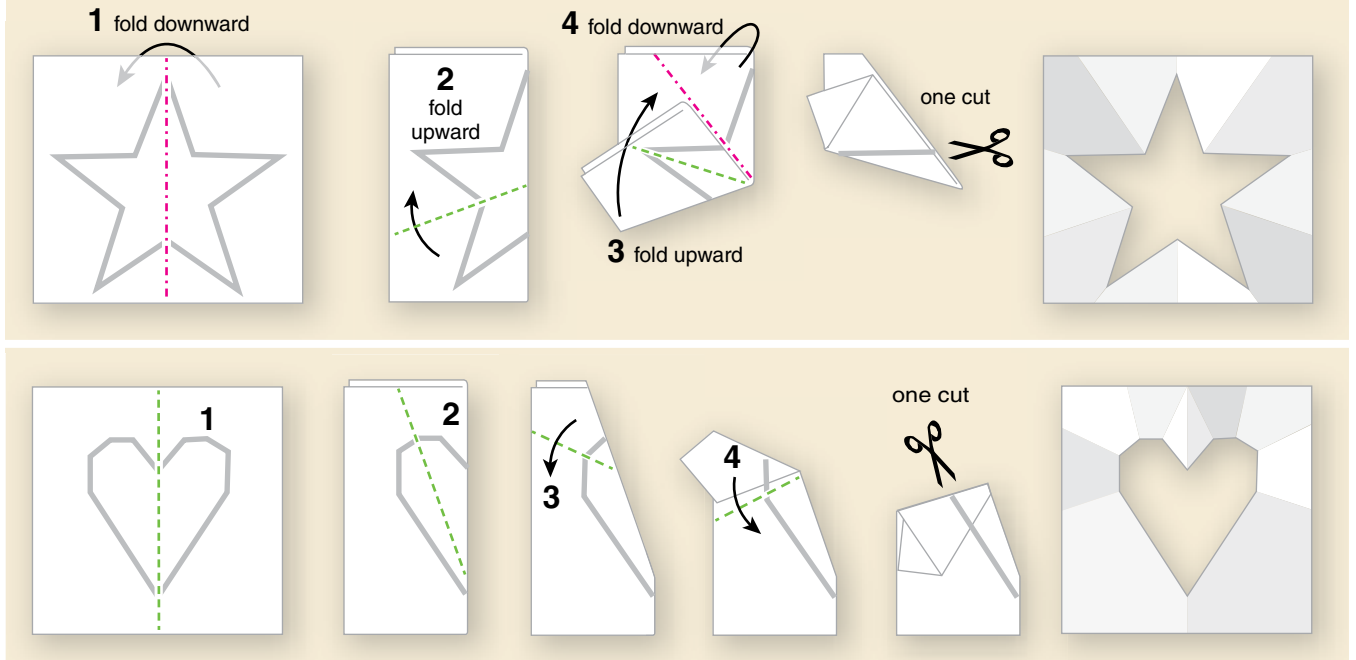
lenges and tricks automatically. Voilà, recreational computer science!

Gardner’s work continues to influence researchers such as myself. The three examples I’ll describe are solutions to problems that Gardner posed—ones he stated explicitly or ones that have been inferred from his work. Throughout Gardner’s writings are countless mathematical questions, puzzles and magic tricks that deserve further research and extension. I encourage everyone to read through his collected works, for the fun this always brings, as well as to help find these seeds for future research. I will collect your suggestions, which you can send to martingardner@csail.mit.edu. Long live the spirit of Martin Gardner!

One-Cut Magic

Our first example of recreational computer science is inspired by a magic trick performed by Harry Houdini before he was an escape artist. In his 1922 book *Paper Magic*, he describes how to take a standard sheet of paper, make a sequence of folds, cut along a straight line, unfold, and obtain a perfect five-pointed star (*see the top of the first figure, next page*). In 1960, Gardner wrote a *Scientific American* column that described a few such magic tricks, producing “simple geometrical figures” by a sequence of folds and one complete straight cut.

Gardner included the tantalizing statement: “More complicated designs ... present formidable problems.” To a theoretical computer scientist such as myself, this screams “Unsolved problem!” Whenever we have several examples of a particular style of magic trick, it is natural to wonder whether it represents a general principle. In this case, we know how to make several simple figures by folding and making one complete straight cut. But what is the complete range of figures that are possible to make in this way, and what sequence of folds will make them? To



Before he was an escape artist, Harry Houdini described how to fold a piece of paper so that a five-pointed star could be made from it with one cut (*top*). Martin Gardner wrote about such one-cut magic tricks in 1960. The author and his colleagues developed a similar series of folds to produce a heart shape with one cut (*bottom*). Red lines indicate where the paper is folded downward so that the creased fold points towards you (called a mountain fold); green lines show an upward fold of the paper, resulting in a creased fold pointing away from you (a valley fold).

put it more computationally, can a computer algorithm tell us whether a particular shape can be made, and if so, tell us where to fold and cut to make it?

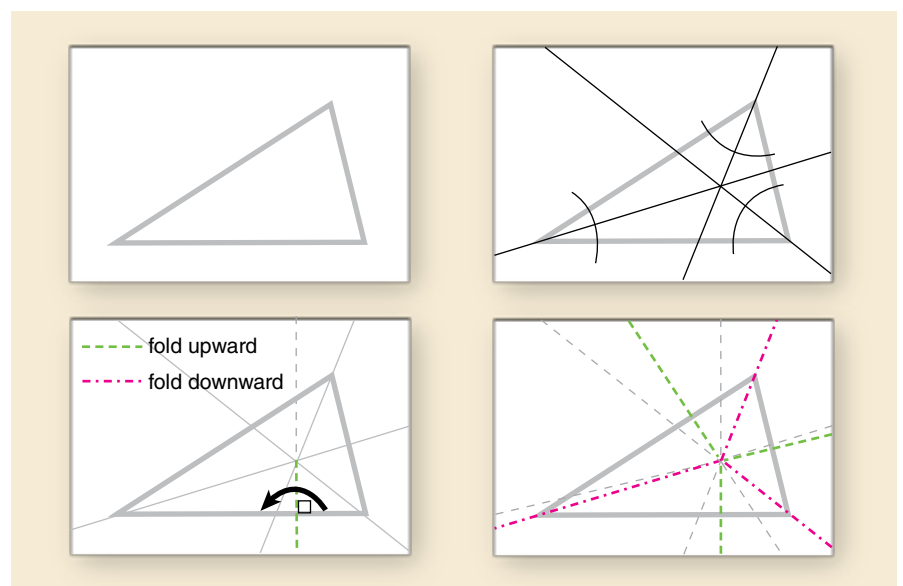
I started thinking about this problem in 1996 when I was a starting Ph.D. student at the University of Waterloo in Canada. My father, Martin Demaine, then an artist and an avid puzzler and now also a mathematician, had read Martin Gardner's article back in 1960, remembered the problem and suggested we work on it. So we tackled it, together with my Ph.D. advisor Anna Lubiw, and essentially solved it two years later.

Initially we experimented, folding and cutting lots of pieces of paper. One of our first recognizable creations was a heart shape (*see the bottom of the first figure, above*). This shape, like the five-pointed star, has a line of reflectional symmetry—the center line along which the pieces on either side are mirror images of each other—which is the first fold to make. Why should we fold along symmetry lines? Our effective goal is to fold the paper to *align* the sides of the shape we want to cut out; once the sides lie along a common line, we simply cut along that line. So if the shape has a line of symmetry, we should fold along it, aligning the two halves of the shape, leaving just one half to align.

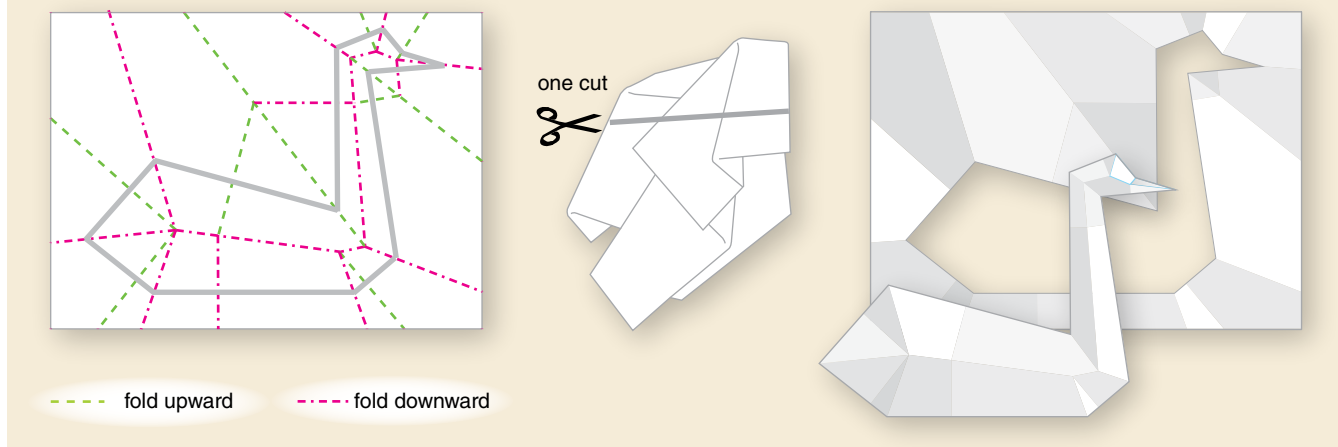
What if the shape we want doesn't have a line of symmetry? This is where the problem gets really interesting. Initially we thought that such shapes were impossible: What else could the first fold be? But as we learned more about origami mathematics, which at the time was just making its debut in the scientific literature, we realized that there were many more types of

folds than just “fold along a line.” Armed with the idea of such complex flat folds, we quickly made progress on solving the problem, building general techniques for producing more and more shapes.

One early step in this progression was making any triangle (*see the second figure, below*). To try this out, draw the three sides of a triangle on a piece



Shapes do not need to have a line of symmetry in order to be folded and produced with one cut. The author and his colleagues were able to demonstrate a method by which any triangle can be produced. Gray and black lines are for visual reference; only colored lines are folded.



The fold-and-one-cut method can be used to produce shapes of theoretically limitless complexity. One such shape is a swan, with fold lines as indicated at left, leading to the collapsed paper with one cut line at center, and resulting in the final figure at right. To download and print this example, go to <http://amsci.org/swan.pdf>. For other examples, see the website <http://erikdemaine.org/foldcut/>.

of paper. Now find the lines that bisect each of the angles of the triangle by starting at one of its corners, folding one edge on top of another, unfolding, and repeating for the two other corners. A classic theorem from high-school geometry is that the three angular bisectors meet at a common point. Now fold along a line through this point so as to bring one of the three triangle sides onto itself, and unfold. The fold will be perpendicular to the triangle side. If you like, you can repeat with the two other sides, though only one perpendicular fold is necessary. The final step, which is the hardest if you've never made a "rabbit ear" in origami, is to fold along all the creases at once, with the angular bisectors folding one way (called "mountain" folds) and the perpendicular ones folding the other way (called "valley" folds). Once you do this, all three triangle sides lie along a line, with the inside and outside sides of the triangle lying on opposite sides of that line. What's cool is that this works for any triangle—no line of symmetry required.

To our surprise, we also found a way to make any polygon with any number of sides, not just triangles. So you can make the silhouette of your favorite shape, such as the swan in the third figure (above), by folding and then making one complete straight cut. The fold is substantially more complicated, but not too hard with practice. I recommend precreasing—folding and unfolding each crease line—before attempting to fold all the creases simultaneously and collapse the swan down to a line.

And there's more: You can make several polygons at once with a single cut. This result is especially useful for spelling initials, such as the ones that I prepared for my first Gathering for Gardner (the fifth one, abbreviated G4G5) in 2002. It's rather difficult to fold anything beyond a few letters, but in principle you could fold a piece of paper, make one complete straight cut, and produce the entire Gettysburg Address.

Although this research was motivated by magic tricks, the theoretical computer science that resulted turns out to have more serious applications as well. A closely related problem is that of compactly folding an airbag flat so that it fits into a small compartment such as your steering wheel. The solution to the fold-and-cut problem leads to a natural way to collapse three-dimensional surfaces such as airbags, and this approach has been applied in some simulations of airbag deployment. So recreational computer science may even save lives.

Coin-Sliding Puzzles

Our next example of recreational computer science is inspired by a collection of classic "penny puzzles," which Gardner wrote about in *Scientific American* in 1966. Helena Verrill, a mathematician now at Louisiana State University, introduced my father and me to puzzles of this type. You are given a pyramid of six coins, shown on the left in part *a* of the fourth figure (next page), and your goal is to make a sequence of moves to arrange the coins into a line. But each move you make must place an existing coin into a position that touches at least two other coins. This

"two-adjacency" constraint is what makes the puzzle challenging. It might initially seem impossible to get rid of all the three-coin triangles, but with a little insight, it is possible.

There are several coin-sliding puzzles of this type, with the two-adjacency constraint on moves. So again we wondered: How does it generalize? The natural computational question here is whether computers can solve this type of puzzle: Given a starting configuration and a goal configuration of coins, can a computer algorithm determine whether the puzzle is solvable under the two-adjacency constraint, and if so, find a sequence of moves to solve it? Even better, can it find the shortest sequence of moves to solve the puzzle?

Both of these questions are actually still unanswered. I suspect that the second question has a negative answer, and it is computationally intractable to determine the fewest moves needed to solve a puzzle, but no one has proved that yet. However, the first question might have a positive answer, which would be much more interesting—it would essentially provide a general theory for this type of puzzle.

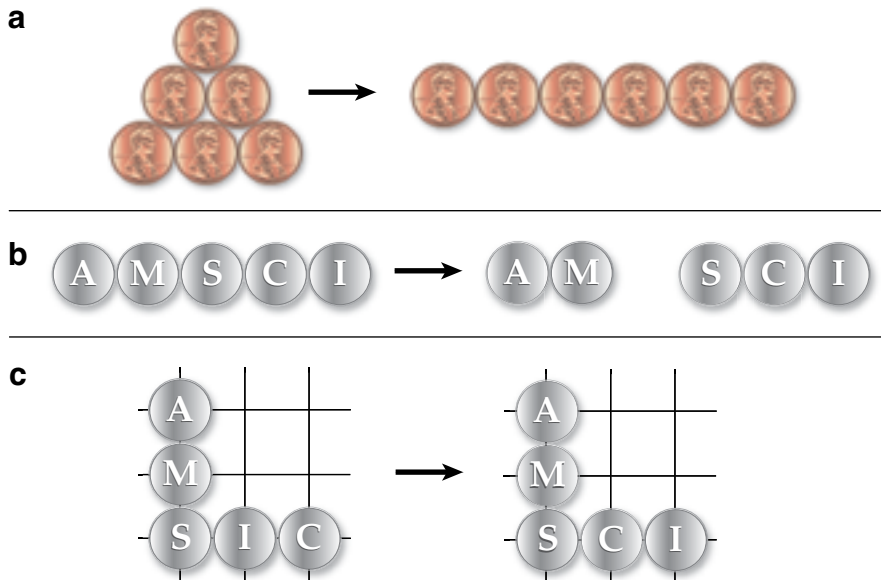
In 1998 Verrill visited my father and me for a couple of days of problem solving, during which we came up with and tackled the first problem. We observed that the puzzle in the fourth figure adheres to a triangular grid: If you draw an equilateral triangular grid where the triangle side length equals the diameter of the coin, then the centers of the coins always remain at grid intersections. This property is a neat consequence of the two-adjacency rule because the coins start on this grid.

For puzzles on the triangular grid, we were able to develop a general computational theory. We found some very simple conditions for when it is possible to transform a starting configuration into a goal configuration via two-adjacency moves. First, the number of coins must be the same in the two configurations, and it must be possible to make at least one two-adjacency move from the starting configuration. Second, and more interesting, the goal configuration must have at least one of three patterns that make it possible to have a last move that solves the puzzle. The patterns are a triangle of three coins, a connected group of four or more coins, and a connected group of three coins along with another connected group of two coins.

As long as the goal configuration has at least one of these patterns, the puzzle is guaranteed to be solvable, and a computer algorithm can tell you how. This result holds even if the coins have different labels (for example, some are heads and some are tails), provided there are more than three coins total.

This kind of simple characterization of solvable puzzles not only helps to find outcomes, but also makes it easy to design new puzzles. Armed with a guarantee about when a puzzle will be solvable, we can design a new one within those constraints that will also be visually interesting. For example, can you solve the puzzle in part *b* of the fourth figure?

Another special type of coin-sliding puzzle arises if we replace the triangular grid with a square grid, requiring every move to place a coin on this grid. These puzzles are substantially harder to solve, both in practice and in theory. We came up with a nearly complete characterization of solvable puzzles, showing how to complete challenges with at least two “extra coins” to help navigate the other pieces. Puzzles without any extra coins are impossible to solve, and puzzles with just one extra are typically either unsolvable or not too engaging—but we still don’t have a computer algorithm to tell us exactly which puzzles can be solved. The situation with two extra coins is quite interesting, however, and has again allowed us to design several puzzles. Try, for instance, the puzzle in part *c* of the fourth figure. To see more coin-sliding puzzles, visit <http://erikdemaine.org/slidingcoins/>.



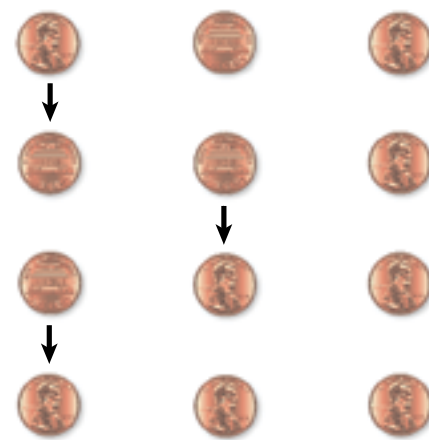
Reconfiguring coins becomes an interesting challenge when there are constraints. For instance, can you transform a pyramid of coins (*a*) into a line using only moves that place coins in contact with at least two other coins? The minimum possible number of moves is seven; a solution is shown on the next page. Similarly, can you split the abbreviation of this magazine’s title (*b*) into two words using only two-adjacency moves? Again, the minimum number of moves is seven. Finally, in a new coin-sliding puzzle, see if you can correct the spelling of the abbreviation (*c*) using only two-adjacency moves on a square grid. This time the minimum number of moves is eight. To see more puzzles of this type, go to <http://erikdemaine.org/slidingcoins/>.

Coin-Flipping Magic

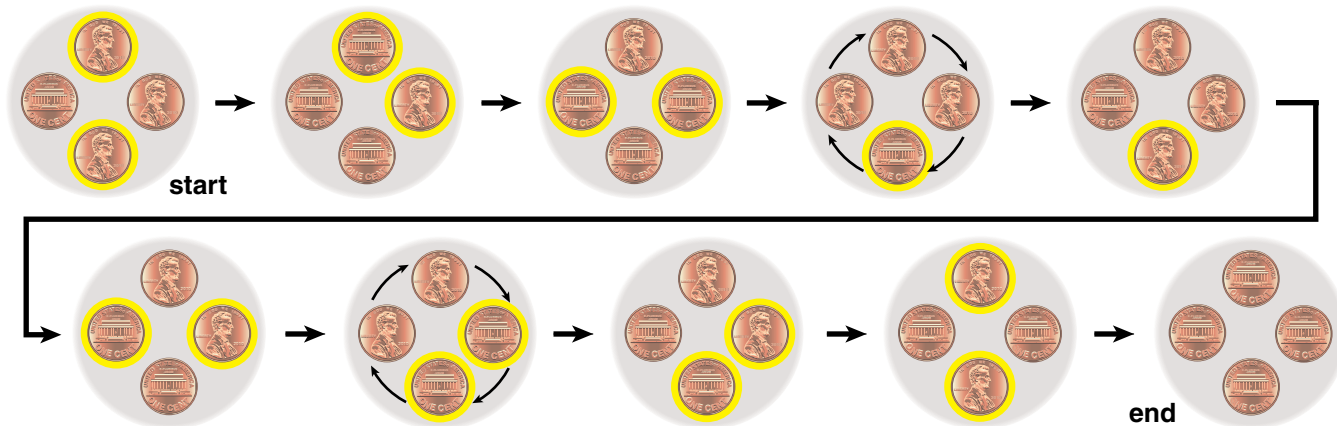
Our last example of recreational computer science is inspired by two tricks with a blindfolded magician, which involve flipping coins into a desired configuration. Because a volunteer manipulates the coins, these tricks have the distinction of being performable over the telephone, on radio or on television, thus being performed “personally” for many people at once.

The first trick, independently invented by Martin Gardner and Karl Fulves before 1980, involves three coins arranged in a line. The spectator arranges the coins as heads or tails, in any combination they like. The magician’s goal—without seeing the coins—is to make them all the same—all heads or all tails. Naturally, the spectator should not choose this outcome as the starting configuration, or else the trick will be over rather quickly. The magician now gives a sequence of instructions: Flip the left coin, flip the middle coin, and flip the left coin again. In between, the magician asks whether the coins are yet all the same, and continues to the next instruction only if the trick is not yet over. It’s no surprise that the magician can eventually equalize all the coins, but it’s impressive that it always takes at most three moves (see the fifth figure, at right).

Why these three moves work is closely linked to a set of codes in computer science that are widely used today to reduce errors when representing digital data with analog signals. These so-called Gray codes, named after Frank Gray from Bell Labs, who patented the system in 1947, have the feature that every two successive binary values differ by only one bit. A con-



Three coins are arranged in a line (*top row*) in any starting sequence. A blindfolded magician can make them all heads or tails in at most three moves (*black arrows*). Here the instructions are to flip the left coin (*second row*), then the middle coin (*third row*), then flip the left coin again (*fourth row*), resulting in all heads.



Four coins arranged in a circle are a variation on the previous coin-flipping trick. The magician tells a volunteer which coins to flip, but this time, before each move, the volunteer can rotate the circle of coins however he or she likes. The puzzle starts at the top left then continues on the second line. Yellow outlines indicate the coins to be flipped in the next move. In this case the volunteer rotates the coins only between steps two and three, and between steps four and five. The magician still accomplishes the trick in seven moves.

figuration of three coins can be seen as a corner of a three-dimensional cube: There are three coins, and each can be either heads or tails, making $2^3=8$ corners. But the cube is effectively folded in half because the all-heads configuration is just as good as the all-tails configuration, leaving just $2^2=4$ “double configurations” arranged in a two-dimensional square. The Gray code tells us how to traverse all these nodes by changing one coin at a time, without ever repeating a configuration. Because we are counting moves instead of configurations visited, we get to subtract 1, for a total of 3 moves. More generally, if we had n coins, we’d have 2^n configurations, 2^{n-1} double configurations, and

$2^{n-1}-1$ moves in the worst case. This study makes it clear why the trick is for three coins: Four coins would already require $2^3-1=7$ moves.

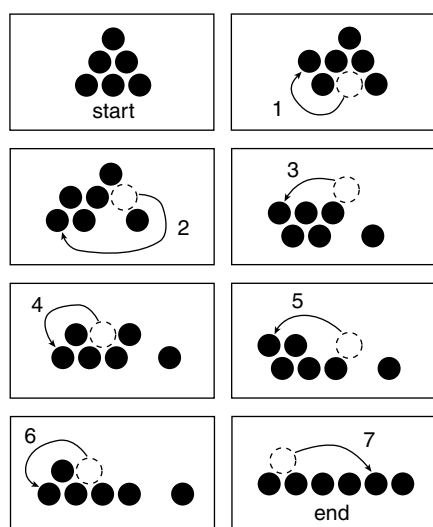
To make the trick more impressive, we can give the spectator more freedom. A 1979 letter from Miner Keeler to Gardner, which Gardner wrote about in *Scientific American* in the same year, describes a trick involving four coins arranged in a circle. The setup and goal of the trick are the same as before, but this time the spectator can rotate the circle of coins however he or she likes before following each of the magician’s instructions. The magician follows these seven steps: Flip the top and bottom coins (after rotation), flip the top and right coins (after rotation), flip the left and right coins (after rotation), flip the bottom coin (after rotation), flip the left and right coins (after rotation), flip the right and bottom coins (after rotation) and flip the top and bottom coins (after rotation). Despite the spectator’s apparent flexibility, the magician equalizes all the coins in no more moves than the original four-coin trick (see the sixth figure, above).

ber of coins must be a power of 2 (as in the four-coin trick above). But it is also possible, for example, to allow the table to be flipped over, because this precisely doubles the number of possible spectator moves.

Like many people, I love puzzles and magic. I also love theoretical computer science. It is wonderful to combine these loves, following in the footsteps of Martin Gardner, and I hope that more computer scientists will consider the recreational side as a good source of fun problems to solve, which may also lead to practical research. Let’s keep carrying the torch that Gardner left burning.

Bibliography

- Benbernou, Nadia, Erik D. Demaine, Martin L. Demaine and Benjamin Rossman. 2008. Coin-flipping magic. Presented at Gathering for Gardner 8, March. <http://erikdemaine.org/papers/CoinFlipping>.
- Cipra, Barry, Erik D. Demaine, Martin L. Demaine and Tom Rodgers (eds). 2004. *Tribute to a Mathemagician*. Natick, Mass: A. K. Peters, pp 23–30 and 63–72.
- Demaine, Erik D., Martin L. Demaine and Helena A. Verrill. 2002. Coin-moving puzzles. In *More Games of No Chance*. Cambridge, U.K.: Cambridge University Press, pp 405–431.
- Demaine, Erik D., and Joseph O’Rourke. 2007. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge, U.K.: Cambridge University Press.
- Gardner, Martin. 1989. Penny puzzles. In *Mathematical Carnival*, chapter 2. Washington, D.C.: Mathematical Association of America. Originally published in *Scientific American*, February 1966.
- Gardner, Martin. 1995. Paper cutting. In *New Mathematical Diversions*, chapter 5. Washington, D.C.: Mathematical Association of America. Originally published in *Scientific American*, June 1960.



A solution for the first coin-sliding puzzle, shown on the previous page, transforms a pyramid of coins into a line. With the rule that each move must place a coin so it touches at least two others, seven moves are required.

What makes this trick possible? A recent paper by two MIT students, Nadia Benbernou and Benjamin Rossman, along with my father and myself, analyzes what type of spectator moves such as “rotate the table” still let the magician equalize the coins with a clever choice of moves. The solution is closely tied to group theory, a field crucial to modern cryptography. The key requirement turns out to be that the number of different moves that a spectator can make is a power of 2. In the case of a rotating table, the num-