# Unwed Numbers

Brian Hayes

A few years ago, if you had noticed someone filling in a crossword puzzle with numbers instead of letters, you might well have looked askance. Today you would know that the puzzle is not a crossword but a Sudoku. The craze has circled the globe. It's in the newspaper, the bookstore, the supermarket checkout line; Web sites offer puzzles on demand; you can even play it on your cell phone.

*The mathematics of Sudoku, a puzzle that boasts "No math required!"*

Just in case this column might fall into the hands of the last person in North America who hasn't seen a Sudoku, an example is given on the opposite page. The standard puzzle grid has 81 cells, organized into nine rows and nine columns and also marked off into nine three-by-three blocks. Some of the cells are already filled in with numbers called *givens*. The aim is to complete the grid in such a way that every row, every column and every block has exactly one instance of each number from 1 to 9. A well-formed puzzle has one and only one solution.

The instructions that accompany Sudoku often reassure the number-shy solver that "No mathematics is required." What this really means is that no *arithmetic* is required. You don't have to add up columns of figures; you don't even have to count. As a matter of fact, the symbols in the grid need not be numbers at all; letters or colors or fruits would do as well. In this sense it's true that solving the puzzle is not a test of skill in arithmetic. On the other hand, if we look into Sudoku a little more deeply, we may well find some mathematical ideas lurking in the background.

### A Puzzling Provenance

The name "Sudoku" is Japanese, but the game itself is almost surely an American invention. The earliest known examples were published in 1979 in *Dell Pencil Puzzles & Word Games*, where they were given the title Number Place. The constructor of the puzzles is not identified in the magazine, but Will Shortz, the puzzles editor of *The New York Times*, thinks he has identified the author through a process of logical deduction reminiscent of what it takes to solve a Sudoku. Shortz examined the list of contributors in several Dell magazines; he found a single name that was always present if an issue included a Number Place puzzle, and never present otherwise. The putative inventor identified in this way was Howard Garns, an architect from Indianapolis who died in 1989. Mark Lagasse, senior executive editor of Dell Puzzle Magazines, concurs with Shortz's conclusion, although he says Dell has no records attesting to Garns's authorship, and none of the editors now on the staff were there in 1979.

The later history is easier to trace. Dell continued publishing the puzzles, and in 1984 the Japanese firm Nikoli began including puzzles of the same design in one of its magazines. (Puzzle publishers, it seems, are adept at the sincerest form of flattery.) Nikoli named the puzzle "sūji wa dokushin ni kagiru," (数字は独身に限る), which I am told means "the numbers must be single"—single in the sense of unmarried. The name was soon shortened to Sudoku (数独), which is usually translated as "single numbers." Nikoli secured a trademark on this term in Japan, and so later Japanese practitioners of sincere flattery have had to adopt other names. Ed Pegg, writing in the Mathematical Association of America's *MAA Online*, points out an ironic consequence: Many Japanese know the puzzle by its English name Number Place, whereas the English-speaking world prefers the Japanese term Sudoku.

The next stage in the puzzle's east-to-west circumnavigation was a brief detour to the south. Wayne Gould, a New Zealander who was a judge in Hong Kong before the British lease expired in 1997, discovered Sudoku on a trip to Japan and wrote a computer program to generate the puzzles. Eventually he persuaded *The Times* of London to print them; the first appeared in November 2004. The subsequent fad in the U.K. was swift and intense. Other newspapers joined in, with *The Daily Telegraph* running the puzzle on its front page. There was boasting about who had the most and the best Sudoku, and bickering over the supposed virtues of handmade versus computer-generated puzzles. In July 2005 a Sudoku tournament was televised in Britain; the event was promoted by carving a 275-foot grid into a grassy hillside near Bristol. (It soon emerged that this "world's largest Sudoku" was defective.)

Sudoku came back to the U.S. in the spring of 2005. Here too the puzzle has become a popular pastime, although perhaps not quite the all-consuming obsession it was in the U.K. I don't believe anyone will notice a dip in the U.S. gross domestic product as a result of this mass distraction. On the other hand, I must report that my own motive for writing on the subject is partly to justify the appalling number of hours I have squandered solving Sudoku.

### Hints and Heuristics

If you take a pencil to a few Sudoku problems, you'll quickly discover various useful rules and tricks. The most

*Brian Hayes is Senior Writer for* American Scientist. *Address: 211 Dacian Avenue, Durham, NC 27701; bhayes@amsci.org*

elementary strategy for solving the puzzle is to examine each cell and list all its possible occupants—that is, all the numbers not ruled out by a conflict with another cell. If you find a cell that has only one allowed value, then obviously you can write that value in. The complementary approach is to note all the cells within a row, a column or a block where some particular number can appear; again, if there is a number that can be put in only one position, then you should put it there. In either case, you can eliminate the selected number as a candidate in all other cells in the same neighborhood.

Some Sudoku can be solved by nothing more than repeated application of these two rules—but if all the puzzles were so straightforward, the fad would not have lasted long. Barry Cipra, a mathematician and writer in Northfield, Minnesota, describes a hierarchy of rules of increasing complexity. The rules mentioned above constitute level 1: They restrict a cell to a single value or restrict a value to a single cell. At level 2 are rules that apply to pairs of cells within a row, column or block; when two such cells have only two possible values, those values are excluded elsewhere in the neighborhood. Level-3 rules work with triples of cells and values in the same way. In principle, the tower of rules might rise all the way to level 9.

This sequence of rules suggests a simple scheme for rating the difficulty of puzzles. Unfortunately, however, not all Sudoku can be solved by these rules alone; some of the puzzles seem to demand analytic methods that don't have a clear place in the hierarchy. A few of these tactics have even acquired names, such as "swordfish" and "x-wing." The subtlest of them are nonlocal rules that bring together information from across a wide swath of the matrix.

When you are solving a specific puzzle, the search for patterns that trigger the various rules is where the fun is (assuming you go in for that sort of thing). But if you are trying to gain a higher-level understanding of Sudoku, compiling a catalog of such techniques doesn't seem very promising. The rules are too many, too various and too specialized.

Rather than discuss methods for solving specific puzzles, I want to ask some more-general questions about Sudoku, and look at it as a computational problem rather than a logic puzzle. How hard a problem is it? Pencil-and-paper experience suggests that some instances are much tougher than others, but are there any clear-cut criteria for ranking or classifying the puzzles?

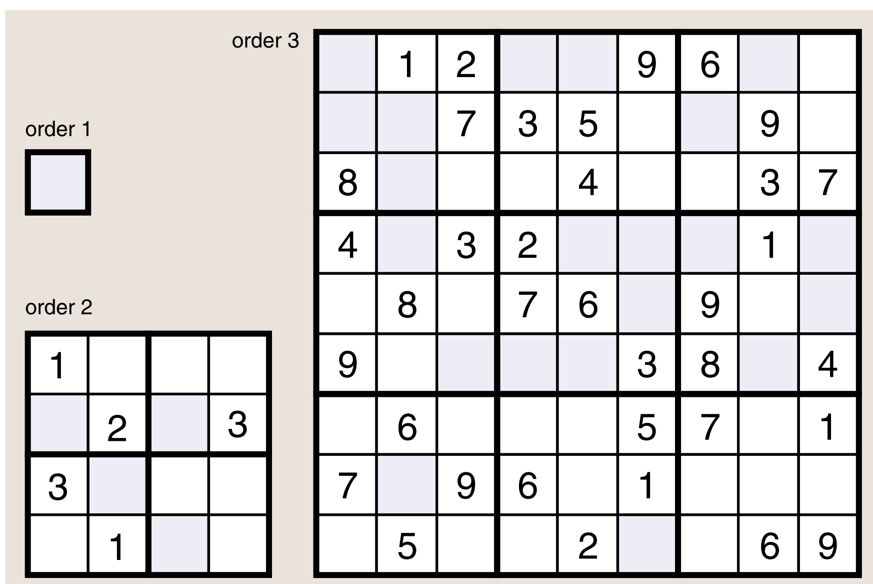
### Counting Solutions

In the search for general principles, a first step is to generalize the puzzle itself. The standard 81-cell Sudoku grid is not the only possibility. For any positive integer $n$, we can draw an order-$n$ Sudoku grid with $n^2$ rows, $n^2$ columns and $n^2$ blocks; the grid has a total of $n^4$ cells, which are to be filled with numbers in the range from 1 to $n^2$. The standard grid with 81 cells is of order 3. Some publishers produce puzzles of order 4 (256 cells) and order 5 (625 cells). On the smaller side, there's not much to say about the order-1 puzzle. The order-2 Sudoku (with 4 rows, columns and blocks, and 16 cells in all) is no challenge as a puzzle, but it does serve as a useful test case for studying concepts and algorithms.

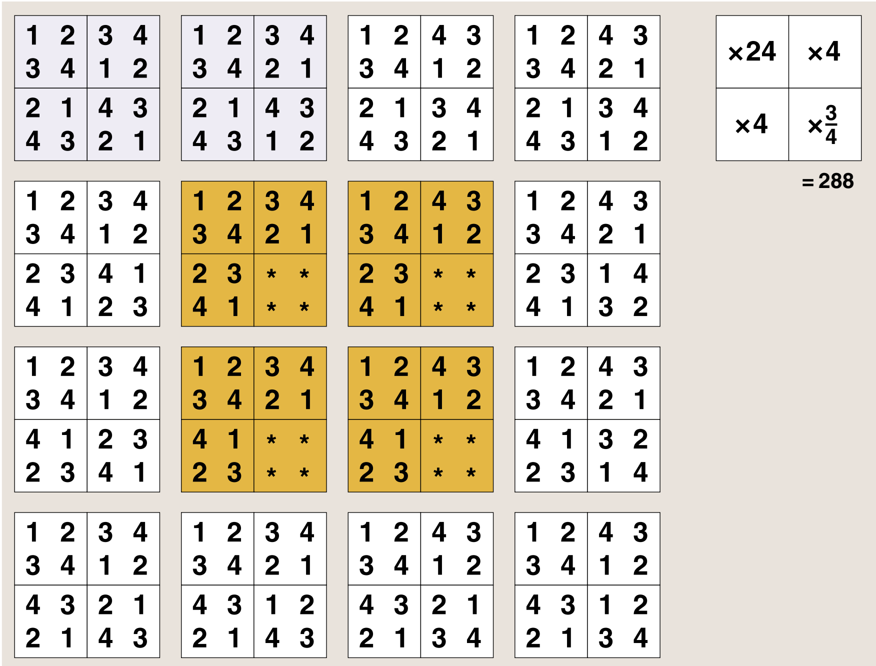
How many Sudoku solutions exist for each $n$? To put the question another way: Starting from a blank grid—with no givens at all—how many ways can the pattern be completed while obeying the Sudoku constraints? As a first approximation, we can simplify the problem by ignoring the blocks in the Sudoku grid, allowing any solution in which each column and each row has exactly one instance of each number. A pattern of this kind is known as a Latin square, and it was already familiar to Leonhard Euler more than 200 years ago.

Consider the $4 \times 4$ Latin square (which corresponds to the order-2 Sudoku). Euler counted them: There are exactly 576 ways of arranging the numbers 1, 2, 3 and 4 in a square array with no duplications in any row or column. It follows that 576 is an upper limit on the number of order-2 Sudoku. (Every Sudoku solution is necessarily a Latin square, but not every Latin square is a valid Sudoku.) In a series of postings on the Sudoku Programmers Forum, Frazer Jarvis of the University of Sheffield showed that exactly half the $4 \times 4$ Latin squares are Sudoku solutions; that is, there are 288 valid arrangements. (The method of counting is summarized in the illustration on the next page.)

Moving to higher-order Sudoku and larger Latin squares, the counting gets harder in a hurry. Euler got only as far as the $5 \times 5$ case, and the $9 \times 9$ Latin squares were not enumerated until 1975; the tally is 5,524,751,496,156,892,842,531,225,600, or about $6 \times 10^{27}$. The




**Sudoku puzzles have to be filled in so that each number appears exactly once in each column, each row and each of the blocks delineated by heavier lines. The order-1 puzzle is a trivial 1×1 grid; the order-2 Sudoku is a 4×4 grid to be filled with integers from 1 to 4; the order-3 puzzle is a 9×9 grid where the allowed numbers are 1 through 9. Some useful terminology: The individual compartments are *cells*; the *n×n* groups of cells are *blocks*; the cells are arranged in horizontal *rows* and vertical *columns*; the blocks likewise are organized in horizontal *bands* and vertical *stacks*; the union of a cell's row, column and block is called its *neighborhood*; the numbers supplied in the initial state are *givens*. The order-3 Sudoku shown here is a variation on the very first puzzle published, in 1979 in *Dell Pencil Puzzles & Word Games*; by present-day standards it is quite easy. Cells marked in blue are fully determined by the givens alone.**




**Order-2 Sudoku are small enough that all possible configurations can be conveniently enumerated. When the block in the upper-left quadrant of the grid is held fixed, the rules of Sudoku allow four variations in the upper-right quadrant and four more in the lower-left, generating the 16 grids shown here. For 12 of these, the lower-right quadrant can be filled in in just one way; in the remaining four cases *(orange)*, no completion of the grid is possible. Thus there are a total of 12 states for the given configuration of the upper-left quadrant. But that quadrant can actually have any of 24 permutations, and so the total number of grids is 12×24, or 288. In another sense, there are only two distinct grids. The entire set of 288 solutions can be generated from the two arrangements at the upper left *(blue)*.**

order-3 Sudoku must be a subset of these squares. They were counted in June 2005 by Bertram Felgenhauer of the Technical University of Dresden in collaboration with Jarvis. The total they computed is 6,670,903,752,021,072, 936,960, or $7\times10^{21}$. Thus, among all the 9×9 Latin squares, a little more than one in a million are also Sudoku grids.

It's a matter of definition, however, whether all those patterns are *really* different. The Sudoku grid has many symmetries. If you take any solution and rotate it by a multiple of 90 degrees, you get another valid grid; in the tabulations above, these variants are counted as separate entries. Beyond the obvious rotations and reflections, you can permute the rows within a horizontal band of blocks or the columns within a vertical stack of blocks, and you can also freely shuffle the bands and stacks themselves. Furthermore, the numerals in the cells are arbitrary markers, which can also be permuted; for example, if you switch all the 5s and 6s in a puzzle, you get another valid puzzle.

When all these symmetries are taken into account, the number of essentially different Sudoku patterns is reduced substantially. In the case of the order-2 Sudoku, it turns out there are actually only two distinct grids! All the rest of the 288 patterns can all be generated from these two by applying various symmetry operations. In the order-3 case, the reduction is also dramatic, although it still leaves an impressive number of genuinely different solutions: 3,546,146,300,288, or $4\times10^{12}$.

Does the large number of order-3 Sudoku grids tell us anything about the difficulty of solving the puzzle? Maybe. If we set out to solve it by some kind of search algorithm, then the number of patterns to be considered is a relevant factor. But any strategy that involves generating all 6,670,903,752,021,072, 936,960 grids is probably not the best way to go about solving the puzzle.

### NP or Not NP, That Is the Question

Computer science has an elaborate hierarchy for classifying problems according to difficulty, and the question of where Sudoku fits into this scheme has elicited some controversy and confusion. It is widely reported that Sudoku belongs in the class NP, a set of notoriously difficult problems; meanwhile, however, many computer programs effortlessly solve any order-3 Sudoku puzzle. There is actually no contradiction in these facts, but there is also not much help in dispelling the confusion.

Complexity classes such as NP do not measure the difficulty of any specific problem instance but rather describe the rate at which difficulty grows as a function of problem size. If we can solve an order-$n$ Sudoku, how much harder will we have to work to solve a puzzle of order $n+1$? For problems in NP, the effort needed grows exponentially.

Most discussions of the complexity of Sudoku refer to the work of Takayuki Yato and Takahiro Seta of the University of Tokyo, whose analysis relates the task of solving Sudoku to the similar problem of completing a partially specified Latin square. The latter problem in turn has been connected with others that are already known to be in NP. This process of "reduction" from one problem to another is the standard way of establishing the complexity classes of computational problems. Yato and Seta employ an unusual form of reduction that addresses the difficulty of finding an additional solution after a first solution is already known. In Sudoku, of course, well-formed puzzles are expected to have only one solution. Yato and Seta say their result applies nonetheless. I don't quite follow their reasoning on this point, but the literature of complexity theory is vast and technical, and the fault is likely my own.

When you lay down your pencil on a completed Sudoku, the thought that you've just dispatched a problem in the class NP may boost your psychological wellbeing, but the NP label doesn't say anything about the relative difficulty of individual Sudoku puzzles. For that, a different kind of hierarchy is needed.

Many publishers rank their Sudoku on a scale from easy to hard (or from gentle to diabolical). The criteria for these ratings are not stated, and it's a common experience to breeze through a "very hard" puzzle and then get stuck on a "medium."

One easily measured factor that might be expected to influence difficulty is the number of givens. In general, having fewer cells specified at the outset ought to make for a harder puzzle. At the extremes of the range, it's clear that having *all* the cells filled in makes a puzzle very easy indeed, and having none filled in leaves the problem underspecified. What is the minimum number of givens that can ensure a unique solution? For an order-$n$ grid, there is a lower bound of $n^2-1$. For example, on an order-3 grid with fewer than eight givens, there must be at least two numbers that appear nowhere among the givens. With no constraints on those symbols, there are at least two solutions in which their roles are interchanged.

Can the $n^2-1$ bound be achieved in practice? For $n=1$ the answer is yes. On the order-2 grid there are uniquely solvable puzzles with four givens but not, I think, with three. (Finding the arrangements with just four givens is itself a pleasant puzzle.) For order 3, the minimum number of givens is unknown. Gordon Royle of the University of Western Australia has collected more than 24,000 examples of uniquely solvable grids with 17 givens, and he has found none with fewer than 17, but a proof is lacking.

Published puzzles generally have between 25 and 30 givens. Within this range, the correlation between number of givens and difficulty rating is weak. In one book, I found that the "gentle" puzzles averaged 28.3 givens and the "diabolical" ones 28.0.

### Logic Rules

Many puzzle constructors distinguish between puzzles that can be solved "by logic alone" and those that require "trial and error." If you solve by logic, you never write a number into a cell until you can prove that only that number can appear in that position. Trial and error allows for guessing: You fill in a number tentatively, explore the consequences, and if necessary backtrack, removing your choice and trying another. A logic solver can work with a pen; a backtracker needs a pencil and eraser.

For the logic-only strategy to work, a puzzle must have a quality of progressivism: At every stage in the solution, there must be at least one cell whose value can be determined unambiguously. Filling in that value must then uncover at least one other fully determined value, and so on. The backtracking protocol dispenses with progressivism: When you reach a state where no choice is forced upon you—where every vacant cell has at least two candidates—you choose a path arbitrarily.

The distinction between logic and backtracking seems like a promising criterion for rating the difficulty of puzzles, but on a closer look, it's not clear the distinction even exists. Is there a subset of Sudoku puzzles that can be solved by backtracking but not by "logic"? Here's another way of asking the question: Are there puzzles that have a unique solution, and yet at some intermediate stage reach an impasse, where no cell has a value that can be deduced unambiguously? Not, I think, unless we impose artificial restrictions on the rules allowed in making logical deductions.

Backtracking itself can be viewed as a logical operation; it supplies a proof by contradiction. If you make a speculative entry in one cell and, as a consequence, eventually find that some other cell has no legal entry, then you have discovered a logical relation between the cells. The chain of implication could be very intricate, but the logical relation is no different in kind from the simple rule that says two cells in the same row can't have the same value. (David Eppstein of the University of California at Irvine has formulated some extremely subtle Sudoku rules, which capture the kind of information gleaned from a backtracking analysis, yet work in a forward-looking, nonspeculative mode.)

### A Satisfied Mind

From a computational point of view, Sudoku is a constraint-satisfaction problem. The constraints are the rules forbidding two cells in the same neighborhood to have the same value; a solution is an assignment of values to cells that satisfies all the constraints simultaneously. In one obvious encoding, there are 810 constraints in an order-3 grid.

It's interesting to observe how differently one approaches such a problem when solving it by computer rather than by hand. A human solver may well decide that logic is all you need, but backtracking is the more appealing option for a program. For one thing, backtracking will always find the answer, if there is one. It can even do the right thing if there are multiple solutions or no solution. To make similar claims for a logic-only program, you would have to prove you had included every rule of inference that might possibly be needed.

Backtracking is also the simpler approach, in the sense that it relies on one big rule rather than many little ones. At each stage you choose a value for some cell and check to see if this new entry is consistent with the rest of the grid. If you detect a conflict, you have to undo the choice and try another. If you have exhausted all the candidates for a given cell, then you must have taken a wrong turn earlier, and you need to backtrack further. This is not a clever algorithm; it amounts to a depth-first search of the tree of all possible solutions—a tree that could have $9^{81}$ leaves. There is no question that we are deep in the exponential territory of NP problems here. And yet, in practice, solving Sudoku by backtracking is embarrassingly easy.

There are many strategies for speeding up the search, mostly focused on making a shrewd choice of which branch of the tree to try next. But such optimizations are hardly needed. On an order-3 Sudoku grid, even a rudimentary backtracking search converges on the solution in a few dozen steps. Evidently, competing against a computer in Sudoku is never going to be much fun.

Does that ruin the puzzle for the rest of us? In moments of frustration, when I'm struggling with a recalcitrant diabolical, the thought that the machine across the room could instantly sweep away all my cobwebs of logic is indeed dispiriting. I begin to wonder whether this cross-correlation of columns, rows and blocks is a fit task for the human mind. But when I *do* make a breakthrough, I take more pleasure in my success than the computer would.

### Bibliography


Bammel, Stanley E., and Jerome Rothstein. 1975. The number of 9×9 Latin squares. *Discrete Mathematics* 11:93–95.

Eppstein, David. Preprint. Nonrepetitive paths and cycles in graphs with application to sudoku. http://arxiv.org/abs/cs.DS/0507053

Felgenhauer, Bertram, and Frazer Jarvis. Preprint. Enumerating possible Sudoku grids. http://www.shef.ac.uk/~pm1afj/sudoku/sudoku.pdf

Pegg, Ed. 2005. Math Games: Sudoku variations. *MAA Online*. http://www.maa.org/editorial/mathgames/mathgames_09_05_05.html

Royle, Gordon F. 2005. Minimum sudoku. http://www.csse.uwa.edu.au/~gordon/sudokumin.php

Simonis, Helmut. 2005. Sudoku as a constraint problem. In *Proceedings of the Fourth International Workshop on Modelling and Reformulating Constraint Satisfaction Problems*, pp. 13–27. http://www.icparc.ic.ac.uk/~hs/report.pdf

Sudoku Programmers Forum. 2005. Discussion thread, May 5, 2005, through June 11, 2005. http://www.setbb.com/phpbb/viewtopic.php?t=27&mforum=sudoku

Wikipedia. 2005. Sudoku. http://en.wikipedia.org/wiki/Sudoku

Yato, Takayuki, and Takahiro Seta. 2002. Complexity and completeness of finding another solution and its application to puzzles. http://www-imai.is.s.u-tokyo.ac.jp/~yato/data2/SIGAL87-2.pdf