Order Out of Chaos in Computers

Computer scientists are controlling decision-making with probabilistic methods, which work in cases where determinism does not

For computer scientist Michael Rabin, the moment of truth came about a decade ago when he realized that perhaps one reason computers cannot do all we might want is because we demand perfection. Perhaps, he reasoned, if we allowed computers to make mistakes—as humans do—we might be able to get them to do more.

The idea was disarming, but, as Rabin showed, it works. He and many other investigators have devised extremely powerful probabilistic algorithms—ways of solving problems that almost always work but have a small but definite chance of being wrong (*Science*, 4 June 1976, p. 989). The next step for Rabin, who has a joint position at Harvard University and at Hebrew University in Jerusalem, was to go on to the workings of the computer itself. When computers were simple autonomous units, it was relatively easy to design ways to control which part of the computer was doing what. But with the advent of computer networks and of parallel processing, the problem of keeping order in the computer becomes increasingly complex. Some practical situations, in fact, are so complicated that they simply have no workable, deterministic solution.

Rabin says that the theme of his work is to "achieve order through disorder," in much the same way as occurs in statistical mechanics. There, he notes, a large system, such as the molecules of a gas, behaves in an orderly fashion as a result of randomized behavior of the individual constituents. As examples of the power of this approach, he tells of probabilistic solutions to several well-known problems in computer science that had frustrated investigators who were looking for simple deterministic solutions.

The fanciful problem of the dining philosophers invented by Edwin Dijkstra of Einthoven in Holland is one of these. Says Richard C. Holt of the University of Toronto, who put a picture of the dining philosophers on the cover of his book on computer science, the problem has an intrinsic fascination of its own. "It is easy to state and you can imagine solving it. But there is no easy solution."

The story is that a group of philosophers is sitting around a table, talking and thinking. Between every pair of philosophers is a fork and in the center of the table is a plate of spaghetti. Each philosopher needs two forks to eat spaghetti. From time to time, the philosophers become hungry and want to eat some spaghetti. The system works well if no two philosophers sitting next to each other want to eat at the same time. But suppose, Rabin says, that they all become hungry at once. Each philosopher turns to his right and picks up a fork. Then each turns to his left. All the left forks, however, are now taken. "There is a deadlock. The philosophers never get to eat," Rabin remarks. The difficulty is caused, he says, because the individual philosophers operate sequentially but the group operates concurrently. "If philosopher *a* were the only one who wanted to eat and then philosopher *b* were the only one who wanted to eat, the system would work," he explains.

> The theme of the work is to achieve order through disorder in much the same way as occurs in statistical mechanics.

How can a protocol be devised so that deadlock is impossible? One way, Rabin notes, is to make one of the philosophers king. He tells each of the others when to eat. But if all of the philosophers are equal, Rabin says, "there are some schedules in which the philosophers will starve."

Recently, Rabin and Daniel Lehmann of the Hebrew University proposed a randomized solution. When a philosopher is hungry, he randomly chooses between the left and right fork. The probability that he will choose the left fork, then, is 1/2, as is the probability that he will choose the right fork. Suppose he chooses the fork on the right. Now, he waits for that fork to be available. (Actually, says Rabin, he "busy-waits"—he does something else and from time to time he looks to his right to see if the fork is there.) When the fork on the right is available he picks it up. Next, he looks to his left. If the fork on the left is available, he picks it up, eats, and returns both forks to their places. But if it is *not* available, Rabin emphasizes, "he does not busy-wait but *releases* the first fork and starts the whole process again."

This simple-sounding procedure solves the problem. Through what Rabin describes as a "rather sophisticated proof," he and Lehmann can show that there will never be a deadlock. With an extension of this process they can show that, with probability 1, everyone who is hungry will be able to eat.

The analogy with computer science is to suppose that the philosophers are processes in a computer, communicating in pairs so that each process communicates with its neighbors. Each process operates sequentially but the group operates concurrently. A generalization of the problem is to suppose that the philosophers are communicating sequential processes—a model suggested by Anthony Hoare of Oxford University and one that reflects the practical problems in setting up computer networks. Nissim Frances of the Technion in Haifa and Michael Rodih of IBM in Haifa devised a probabilistic solution for Hoare's problem that is like the solution to the dining philosophers' problem. Their result, Rabin says, "is very simple and you don't have to keep track of each process."

Another notorious computer science problem that can be solved probabilistically is the Byzantine generals' problem, a problem that has plagued computer scientists for nearly a decade. It was originally called "the interactive consistency problem." But, several years ago, Leslie Lamport of SRI in Menlo Park gave it a new name because, he says, "I realized that it is an important problem and I figured that the best way to make people aware of it was to give it a catchy name."

Lamport stated the Byzantine gener-

Media Center, University of Toronto

The dining philosophers

Each philosopher needs two forks to eat, but there is only one fork between every pair of philosophers. How can they avoid deadlock?

als' problem as a story of an army with many units, each of which is headed by a general. The army is mounting a siege on a city and the generals realize that if they are going to attack, they should all attack at once or else they should all elect to do nothing. They pass a message around, trying to decide on a concerted attack to start, say, at noon the next day. But there are Byzantine generals—traitors—among the generals of the army who are trying to confuse the others and cause a nonconcerted attack. No one has any way of identifying the Byzantine generals and no general has any way of knowing whether a message that another passes to him is true or false. The problem, then, is to create a procedure that will lead to a concerted decision, despite the Byzantine generals' attempts to confuse the others.

This problem, says Lamport, has many analogies in computer science. At SRI, for example, it arose when Lamport, Robert Shostak, and Michael Pease were designing a computer program for NASA that was to provide electronic controls for energy-efficient airplanes. These planes require constant delicate alterations in their surface configurations and must be flown by a computer because, Lamport says, "a human being cannot react fast enough." But the computer must keep the plane flying. It cannot make mistakes. The usual way to provide fail-safe computers is to make them redundant—several processors will make the same calculation and whatever value the majority chooses is assumed to be correct. The NASA system, however, could not work that way. Suppose, says Lamport, that an answer depends on an altimeter reading and two processors read the altimeter at slightly different times. They will get different answers. "The values may be close but since they may be used to decide between continuing an automatic landing or aborting the landing and trying again, a small difference in a value can lead to an entirely different outcome."

One solution to the SRI problem is to connect the processors so that they communicate. But then the investigators found themselves face to face with a Byzantine generals' problem. What if one computer is faulty and gives conflicting information? Which information should be trusted and how should decisions be made?

Initially, the SRI researchers thought that there must be some simple algorithm, some easy way out of this dilemma. Yet every time they attempted to solve the Byzantine generals' problem, they found, says Lamport, "a reasonable type of hardware failure that would defeat the algorithm." Now, after a great deal of research, computer scientists realize just how difficult a problem it is.

The desirable solution to the Byzantine generals' problem, Rabin points out, should result in a decision to attack if all of the loyal generals decide to do so or to do nothing if that is their decision. But in every case, the loyal generals must be coordinated, no matter what confusing messages the Byzantine generals convey. One difficulty, however, is that the generals do not communicate directly. A Byzantine general can tell one general that he has decided to attack and another that he has not. The problem has been extensively studied for the past 6 years.

About $1\frac{1}{2}$ years ago, Nancy Lynch of the Massachusetts Institute of Technology, Michael Fischer of Yale, and Michael Patterson of Warwick University in Coventry, England, got a result indicating that the problem may not even have a deterministic solution. If the length of time it takes to exchange a message is not fixed and if even one general may be Byzantine, it is impossible for the good generals ever to reach a consensus, they found.

There are deterministic solutions if the message time is constant, Lynch and Fischer and also Danny Dolev of Hebrew University and Ray Strong of IBM in San Jose found, but it can take an exhausting amount of time to be sure of reaching a consensus. The generals would have to exchange round after round of messages of the form, "I have decided and I know that you know that he knows that another general knows," and so on. Lynch and Fischer and, independently, Dolev and Strong showed that if there are n participants and up to k may be Byzantine, then as many as $k + 1$ rounds of message exchanges are necessary to reach agreement. So if there are 1000 participants and up to 250 may become unreliable, the participants must go through 251 rounds of exchanging messages to be sure they have reached agreement.

The situation looked hopeless until, about 1 year ago, Michael Ben-Or of MIT discovered a randomized solution that, to everyone's amazement, solved the Byzantine generals' problem whether or not the message time was fixed. Ben-Or's solution, however, was fairly slow. Rabin independently found another randomized solution that also works for variable message times but that is considerably faster than Ben-Or's method.

The idea of randomization, Rabin says, "is to fight fire with fire. The Byzantine generals may be trying to foil agreement, so our countermeasure involves confounding them with surprise random moves so as to foil *their* plan." In each phase of the randomized procedure, every general makes a decision to attack—yes or no. Then each asks every other general what his decision is and determines what the plurality of generals think. Finally, each general has to decide whether to accept the plurality decision as his new decision.

The probabilistic part of the procedure

is the decision that each general makes whether to accept or reject the plurality opinion. There can be no fixed rule, Rabin says, because if there were, the Byzantine generals could exploit it to foil the agreement. The key is to arrange the rule so that it itself is determined by chance. Then the Byzantine generals cannot determine the best strategy to confuse the others.

Rabin's procedure for the generals is that, after polling each other on their decisions, the participants communally toss a coin. If the coin comes up heads and the plurality count was bigger than $n - k$ (the total number of generals minus the number that may be Byzantine), each accepts the plurality decision as his own. If it is tails and the plurality count was greater than $n/2$, each adopts the plurality decision as his own. They continue this procedure a fixed number of times, say 20. For each round of decision-making, the probability that all the good generals will end up with the same wish is greater than 1/2. After 20 rounds, the probability that the good generals will have failed to reach agreement is less than 1 in 2^{20}, or less than 1 in a million. If the generals repeat the decision-making process 30 times, there is less than 1 in 2^{30} or less than 1 in a billion chance that the good generals will fail to reach agreement. Depending on the degree of certainty they require, the generals can choose to go through more rounds of decision-making. For each round p, the probability of failing to agree is less than 1 in 2^p.

The random approach, says Rabin, cuts through the confusing rounds of polling that are required by the deterministic approach. With the deterministic methods, Dolev and Strong have shown that everyone has to know that everyone else knows that everyone else knows that everyone else knows, and so on, repeated k times. "The probabilistic method," says Rabin, "trades a perfect notion of what everyone knows for a small measure of uncertainty and a great deal of simplification." Rabin's solution, says Lamport, "is very neat."

Rabin predicts that probabilistic approaches will be increasingly important in computer science. "We are entering an era of very large conglomerates of computers, of computing units that are intended to act in parallel," he says. "If we want them to work in unison on some common problem, their activities must be coordinated and synchronized. The advantage of the probabilistic approach is that it achieves order through disorder and is guaranteed to work with extremely high probability."—GINA KOLATA

Fly Antibodies Mark Human Brain

A new class of markers for human brain cells has recently been reported by Seymour Benzer, a neurogeneticist at the California Institute of Technology and Carol Miller, a neuropathologist at the University of California School of Medicine in Los Angeles.* The two researchers find that monoclonal antibodies to the nervous system of the fruit fly *Drosophila* can also bind to specific groups of cells in the brains of humans. Miller and Benzer anticipate that their findings may help to map the human brain according to the biochemical roles of its cells and to study the molecular basis of certain neurological diseases.

Miller and other neurologists have tried other methods of identifying subsets of brain cells, but they were beset with problems, caused in part because there are so few good ways of marking the cells to separate. Most brain markers that are currently used are for known substances such as neurotransmitters and receptors, which may be found in many different cell types, she points out.

The idea of using monoclonal antibodies to identify subsets of brain cells is not new, of course, and researchers have made monoclonal antibodies to brain cells of animals such as the mouse or the chick. "However," says Miller, "no one has ever systematically studied human brains with a large panel of these antibodies."

"The advantage of using *Drosophila* as the source of antigens," says Benzer, "is that it is genetically easy." After having spent the past two decades studying how genes influence behavior in the fruit fly, Benzer and his associates have isolated a large number of mutations affecting almost every aspect of behavior. Some are reminiscent of human neurological disorders, including learning defects and epilepsy. Others have brain, retina, or muscle degeneration, abnormal nerve membrane channels, or disturbed circadian rhythms. With the genetic methods available for *Drosophila*, he and others can make mosaic flies and thereby determine where the primary defect is.

When monoclonal antibodies were discovered a few years ago, Benzer recalls, it was a turning point for him. "Monoclonals provide the first real opportunity to put the *chemo* into the postulated chemospecificity of neurons," he remarks. "The beauty of monoclonals is that one does not have to purify a specific antigen. One can use a shotgun approach, using ground-up fly brains to immunize mice and make a collection of hybridomas, each of which produces an antibody to a single molecule of the mixture. By screening each antibody on a slice of fly brain, one can see at a glance exactly where the corresponding antigen is located."

Using this strategy, Benzer's group found 146 monoclonal antibodies that react with various regions of the *Drosophila* brain and other tissues. Then Miller, who is interested in isolating subsets of human brain that are functionally and, presumably, chemically distinct, decided to try the *Drosophila* antibodies on human brains. To her amazement, they worked. Sixty-one of them were specifically stained human brain cells, and some were specific for neuronal and glial subsets. Since the *Drosophila* brain is so much simpler than the human brain, neither Benzer nor Miller expected that so many of the antibodies would bind so specifically to the human cells.

The *Drosophila* antibodies, Miller points out, are "a very useful tool." For example, she says, they label the pyrimidal neurons of the hippocampus, a cell type that deteriorates in Alzheimer's disease. Some of the other antibodies identify neurons of the dentate gyrus in the hippocampus. Another monoclonal antibody identifies motor neurons. Still others bind to glial cells. These findings, Miller says, promise to be useful for the study of certain human neurological diseases, such as Alzheimer's, Huntington's, and Parkinson's, in which specific neuronal subsets die. Monoclonal antibodies, she predicts, should enable researchers to isolate the classes of cells affected in these diseases and, perhaps, to identify molecules that are missing from, or altered in, these target cells.—GINA KOLATA

*Miller and Benzer discussed their findings at a workshop sponsored by the Hereditary Disease Foundation in Santa Monica on 8 and 9 January 1984, and they published a description of it in the December issue of the *Proceedings of the National Academy of Sciences*.