How to Catch a Cheating Computer

A new development in theoretical computer science may make it easier to check the accuracy of supercomputer results

IF YOU'VE DEALT WITH COMPUTERS, you undoubtedly know just how enormous is their capacity for making mistakes. But while it's relatively easy to check the correctness of some computer results, others must seemingly be taken on faith. For example, with megaprograms containing thousands of lines of code and running in complicated operating environments, it's virtually impossible to guarantee the correctness of the results by logically analyzing the program. Beyond testing such a program to see that it runs properly in some sample settings, there often seems to be no alternative to just hitting the return key and crossing your fingers.

But help may be on the horizon. A recent advance in theoretical computer science has opened the door to a "universal" answer checker that can catch mistakes even if a devious, ultra-smart computer tries to cover them up. The breakthrough—little heralded outside the worlds of mathematics and computer science—was made last year by six researchers at four locations in the United States and Israel. Their final conclusion may seem somewhat cryptic. It's IP = PSPACE. But the key is that equal sign. In essence, it says that you don't have to trust what your supercomputer tells you; you can challenge its authority and make it convince you it's got the right answer.

Take the classically hairy "three-coloring" problem which crops up in applications to computer networks and scheduling: Given a roomful of people, some of whom are enemies, is it possible to separate them into three groups (red, blue, and green) such that no pair of enemies winds up in the same group?

At small gatherings, it's relatively easy to tell if the answer is yes or no. Moreover, it's always possible to provide a convincing "yes" answer by actually showing the three groups and checking for animosities. But every time someone new walks into the room, the groups may have to be completely rearranged to accommodate the newcomer. And at some point that may become impossible—although the computer user can find it very difficult to tell when that point's been reached.

Except in certain circumstances (such as a room where everyone hates everyone else) the only obvious way to prove that a "no" answer is correct is to consider each possible "coloring" in turn and show that none succeeds in separating all pairs of enemies. But that's not an efficient method of checking because it takes an undue amount of effort. The number of possible combinations of people grows exponentially, increasing by a factor of 3 with each person who enters the room. So the question is, how do you check a "no" answer efficiently?

That's where the equation IP = PSPACE comes in. PSPACE consists of problems, such as the three-coloring problem, which can, roughly speaking, be solved by exhaustive search.

Julie Cherry

The equation says that PSPACE problems are actually equivalent to another type, the IP problems, which are so called because they can be verified by means of an interactive proof—a procedure in which an "ordinary" computer interrogates an ultra-fast but possibly sloppy or capricious computer for a while and comes away either convinced it's been told the truth or certain that something is wrong. The interrogation consists of posing a series of PSPACE problems, drawn randomly from a large pool of possible questions, to the suspect machine and then using an efficient algorithm to check for consistency in the answers. If the answers are not consistent, then the interactive proof has shown that the ultra-fast computer has made a mistake.

So if the three-coloring problem, and other PSPACE problems, actually belong to IP, then it's possible to check any answer that a program provides. Trust becomes an option rather than a necessity.

A mere year ago, computer scientists familiar with IP problems would never have predicted you could "match" the two very different sorts of problems. That's because interactive proofs were originally devised for a limited class of problems, with applications in cryptography and secure communications.

So the conclusion that they were not so limited came as a surprise even to the people who proved it. Looking back, one of the researchers involved, Lance Fortnow, of the University of Chicago, says: "People didn't expect a proof at all." And Manuel Blum, a computer scientist at the University of California, Berkeley, says, "These new results go against all intuition. It's opening up new possibilities in the mathematical world."

How did the breakthrough occur? The theorem was proved in a flurry of activity late last year, much of it communicated by electronic mail. The key turned out to be a seemingly mundane problem called the matchmaker's problem: Given *n* men and *n* women and information as to which couples are mutually compatible, in how many ways (if any) is it possible to marry everyone off so that all *n* couples are compatible?

While it may seem like little more than a curiosity, the matchmaker's problem is actually representative of a large class of computational problems known in the trade as #P. Every problem in #P can be translated into an instance of the matchmaker's problem. This means, for example, that for each roomful of people in the three-coloring problem, it's possible to find a corresponding collection of men and women in the matchmaker's problem for which the number of compatible marriages is the same as the number of ways the room can be separated into three groups.

Richard Lipton, a computer scientist at Princeton University, got things going last year by showing how any program that solved the matchmaker's problem correctly most of the time could be modified to get the correct answer all of the time. In November, Noam Nisan, then at the Massachusetts Institute of Technology, found that any answer to the matchmaker's problem can be checked by a "multiple" interactive proof—an extension of interaction in which two or more ultra-fast computers are interrogated separately on the same problem, in much the way that police might separate criminal suspects for questioning.

Nisan's announcement, sent out by electronic mail just after Thanksgiving, galvanized action on the problem. Within 3 weeks, Carsten Lund, Lance Fortnow, and Howard Karloff at the University of Chicago were able to show that multiple proofs were unnecessary—the matchmaker's problem could be verified with a single interactive proof. Their verification protocol works by repeatedly reducing the number of couples in the matchmaker's problem in such a way that the computer is forced to give the wrong answer to the reduced problem if it wants to cover up a wrong answer to the original problem. But this eventually forces it to give the wrong answer for just one couple.

These developments already showed that interactive proofs were more powerful than theorists had anticipated. Then, 2 weeks later, Adi Shamir at the Weizmann Institute in Israel took the final step. Shamir applied the same techniques used by the MIT and Chicago workers to find an interactive proof for a set of PSPACE problems known as Quantified Boolean Formulas.

These problems seek to establish the truth or falsity of complicated logical statements containing multiple users of the quantifiers "for all" and "there exists." Like the interactive proof for the matchmaker's problem, Shamir's approach depends on reducing the number of quantifiers. It was already known that every problem in PSPACE can be translated into a Quantified Boolean Formula problem, so Shamir's result instantly implied that everything in PSPACE has an interactive proof.

While the new results hint at the possibility of computer program answer checkers, don't count on seeing Macintoshes using interactive proofs to check the work of Cray supercomputers anytime soon. The obstacle is that the method assumes that the interrogated computer can instantly solve (or at least claim to solve) the extremely hard matchmaker's problem or the even harder problem of Quantified Boolean Formulas. "In reality we don't have these very powerful [computers] around," Fortnow points out. It would be of interest, he says, to determine exactly how much computing power is required to obtain an interactive proof for a given problem.

There is one other curious caveat on the new excitement. It could conceivably turn out that PSPACE problems aren't inherently unwieldy after all. If an efficient algorithm could be found for Quantified Boolean Formulas, then interactive proofs would be unnecessary. Complexity theorists believe this is an unlikely scenario, but see no way at this point to rule out the possibility. If anything, the unexpected equality of IP with PSPACE indicates that more surprises may yet be in store. ■ Barry Cipra

Identifying Fossils by Computer

BP Research

Clue to oil? *Fossils like these help oil explorers find their targets.*

The paleontologists who classify microfossils for oil drilling companies may soon be able to call upon a new computer program to help with their time-consuming analytical chores. Research presented this month at the conference on Innovative Applications of Artificial Intelligence in Washington, D.C., suggests that computer programs loaded with drawings of fossil parts could help researchers identify the samples taken in exploratory drilling. The program, if it becomes a practical success, may speed up the process by which petroleum companies make multi-million-dollar decisions about drilling at new sites. It promises to expedite a task that Abolfazl Jameossanaie, a fossil expert at Exxon, USA in Houston, now calls "tedious and time-consuming."

In choosing drilling targets, oil companies have come to rely on the advice of a variety of specialists, including a limited supply of fossil experts who study the tiny animals and plants that lived in the ocean hundreds of millions of years ago. The skeletons and shells of these sea creatures rained upon the ocean floors over the millennia and, along with organic material, became preserved in undersea rock. The microfossil composition of the rock layers can thus serve as a guide to their age and geological history and help geologists estimate the likelihood that oil deposits are near.

But this research takes a long time. World authorities who have devoted their careers to such work can identify offhand a few hundred or, by consulting catalogues, papers, or notes, can name a few thousand species—only a small portion of the microfossils commonly encountered.

Oil companies may spend a million dollars a day to keep rigs operating while awaiting word from the experts, and so are eager to speed up microfossil identification. This is where an "expert system" designed by Peter Swaby, a computer scientist at British Petroleum's Research International in Middlesex, England, and his colleagues comes in.

The BP researchers have devised a general computer program that incorporates the tricks, shortcuts, and rules of classification experts. When executed and linked with a full library of microfossil data, the program's graphics package kicks in, displaying pictures, textual descriptions, and command menus all on the same screen. A researcher examining fossil samples with a microscope can thus directly compare what he sees with the computer images.

And that makes the expert system much more efficient than traditional methods of identification—using those dusty tomes that, though supplied with illustrations, are organized by Latin name and written description. Swaby capitalizes on the preferred modus operandi of classification experts—to compare visually first. Written descriptions are often vague and sometimes incomprehensible, Swaby says. In contrast, "The human vision system is very powerful and can compare features quite easily."

Swaby's graphical expert system also has another advantage. It allows a paleontologist to begin describing a fossil with any one of a number of features, thereby breaking out of traditional flow chart schemes that are inherently hierarchical. Established schemes can be bothersome if a key feature can't be discerned because a fossil has been damaged. But with the expert system, users can start their descriptions with any of a variety of features. As the description progresses, the number of possible species becomes small enough so that their images may be perused, on screen, until a match is made.

Using BP's program as a guide, two novices, postgraduate students of geology, successfully classified three samples of conodont microfossils in about 2 hours—a job that is usually not approached before a semester of basic training in the use of reference catalogues. Swaby hopes to see his program, which he plans to link with data on the more commonly encountered Foraminifera microfossils, used in the field in a year or two. "Of course, the ultimate would be to scan an image of a microfossil and let a computer identify microfossils for you," Swaby says. But such a capability is far, far down the road.

Meanwhile, says Alan Higgins, a BP conodont expert, computer systems such as Swaby's "are a way of preserving for the next century a lot of experts' knowledge in a usable form that wouldn't otherwise be accessible." ■ Sarah Williams