A Short Review of Theorem Proving

Whenever Sherlock Holmes solves a case in a Conan Doyle book, the suspect always seems to get killed trying to escape or else confesses to everything, handily supplying all the gory details. Just once I would like to see the suspect say "Alright, it's time to get the lawyers involved - I'll see you in court". If this happened, it's possible that the suspect would win, as the case put forward by Holmes would be quite unsound in places.

Holmes uses various forms of reasoning to solve his cases. These include *induction, abduction,* and occasionally his trademark: *deduction*. Induction is the process by which we generalise rules from examples. So, for instance, if a silk scarf was left at each of ten murder sites, Holmes might induce the fact that this is something to do with the murders themselves, and expect to see one at the site of the next murder. Abduction is the process whereby we assign explanations to observations. In the case of the silk scarf, based on previous experience, Holmes might abduce the explanation that the killer leaves the scarf as a calling card. Deduction is the process whereby we derive new facts from old ones in such a way that the new facts must follow from the old ones. In the case of the silk scarf, if Holmes narrowed it down to two suspects, one of whom had an iron -clad alibi, then he could deduce that the other suspect was the killer.

Induction and abduction are unsound procedures because they are based on guesswork. For example, if all the ten murder victims were members of the silk scarf appreciation society, then the scarves would possibly have no connection with the murderer. Deduction, on the other hand, is a sound procedure. If we know that some old facts are indeed true, then any new facts deduced from them are guaranteed to be true also. Deduction can be used opportunistically to find new facts from old ones. Often, however, we want to do things the other way around: we start with some things we know to be true (called axioms) and something we suspect to be true (called a conjecture). If we can show that the conjecture follows deductively from the axioms, then we have proved it to be true and we would upgrade the conjecture to a theorem. More importantly, because we have used deduction, we know that our reasoning is sound and that the theorem really is true. Getting computers to perform such theorem proving is a well established and very successful area of Artificial Intelligence.

Proving that something is genuinely correct is a very powerful tool to have in your toolbox. For instance, suppose that, through extensive testing, a company is absolutely positive that certain electronic components perform in specific ways. This information can be used as axioms. When they combine these components in a circuit board, however, things get more complicated. While the company might be quite sure that the circuit board works as they want it to, they would probably want proof of this. Such a proof can be deduced from the axioms about the individual components using automated theorem proving software. The verification of hardware and software is an important application of AI techniques.

In many cases, it wouldn't be too bad if the hardware or software didn't quite perform as specified. However, there are many situations which are safety critical, and it is absolutely essential to prove that the hardware and software perform as we want them to. Next time you are in an aeroplane, you can rest assured that the computers helping fly the plane are using hardware and software which have been automatically verified to perform as they should. The same is true for power stations, medical equipment, and even your humble home computer.

Automated theorem proving is enabled by our understanding of logic. The word 'logic' can be interpreted in many ways. When we talk about a particular logic, we are really describing a language which enables us to express certain things in a very constrained way. If you think of how many times you have misunderstood someone over something quite simple, you can see the advantage of having a very restricted language which everyone agrees upon. This is especially true of computers, as they aren't so good at interpreting sentences. There are many different logics that we can pick and choose from to fit the nature of the theorem we want to prove. The simplest ones such as propositional and first order logic have been studied for centuries and are extremely useful. However, they have the drawback that certain things we might want to say in English cannot be expressed in these logics. Other logics are said to be more expressive if we can say more with them. For instance, if we need to express the fact that certain things change over time, we would need a temporal logic. Similarly, if we need to express the fact that certain statements are probably true, we would need a probabilistic logic. Probably. If we restrict our language to a logic, we can employ rules to perform deduction. These are called rules of inference, because we use them to infer something new. The simplest such rule is called Modus Ponens. As an example of Modus Ponens in action, suppose Sherlock knows for sure that silk scarves can only be bought from one shop in London. On discovering the first scarf at a murder site, he can deduce that it must have been purchased from that shop - a valuable lead for him to follow.

There are many similar rules of inference and the goal in automated theorem proving is to find a route from the axioms to the conjecture using only such rules. There are many ways to do this, and there is continued research into making the process better. One important method is proof by contradiction: pretend that the conjecture is false and deduce something silly as a result (i.e., something which contradicts the axioms, which we know are true). If the axioms are contradicted, then the conjecture can't be false, so it must be true. Automated theorem proving has great potential for an area of human endeavour where deduction rules supreme: mathematics. Unfortunately, as with many things in AI, we underestimated how difficult it is to prove mathematical theorems, and AI has not had a big impact here. Very occasionally, though, theorem provers have beaten mathematicians. In particular, as reported on the front page of the New York Times, a theorem prover developed by researchers in Chicago managed to deductively prove the Robbins conjecture, which had eluded mathematicians for 70 years. Sherlock would be proud.

Simon Colton, Imperial College, London
sgc@doc.ic.ac.uk
http://www.doc.ic.ac.uk/~sgc/