

EXAMPLE 2.1 *The halting problem*

The problem is to predict whether a given computer program, with given input data, will eventually halt.

This is a very practical problem for us programmers: we all occasionally write a program that gets into a never-ending loop. One of the most famous results in computer science is that this problem cannot be solved by any algorithm. It turns out that any ‘algorithm’ that purports to solve this problem will itself get into a never-ending loop, for at least some programs that might be given to it. As we shall see later in this section, we insist that every algorithm must eventually terminate.

If we can never find an algorithm to predict whether a given program halts with *given* input data, we clearly can never find an algorithm to prove whether a given program behaves correctly for *all possible* input data.

It may still be possible for a human to prove that a *particular* program is correct. Indeed, this has been done for some important small programs and subprograms. But we can never *automate* such proofs of correctness.

In fact, many problems in mathematics and computer science are unsolvable by algorithms. In a way, this is rather reassuring: we can be sure that mathematicians and computer scientists will never be made redundant by machines!

From now on, we shall consider only problems that are solvable by algorithms.

Algorithms

Concerning algorithms themselves, we can state the following principles:

- The algorithm will be performed by some **processor**, which may be a machine or a human.
- The algorithm must be expressed in steps that the processor is capable of performing.
- The algorithm must eventually terminate, producing the required answer.

Some algorithms, as we have already seen, are intended to be performed by humans rather than machines. But no algorithm is allowed to rely on qualities, such as insight and creativity, that distinguish humans from machines. This suggests a definition:

An **algorithm** is an automatic procedure for solving a stated problem, a procedure that could (at least in principle) be performed by a machine.

The principle that the algorithm must be expressed in steps that can be performed by the processor should now be clear. If the processor has to work out for itself what steps to follow, then what we have is not an algorithm.

The principle that every algorithm must eventually terminate should also be clear. If it never terminates, it never produces an answer, therefore it is not an algorithm! So an algorithm must avoid getting into a never-ending loop.