What is an Algorithm?

A fundamental concept in the theory of computation (and thus in the theory of representations) is that of an algorithm. Some procedures are said to be 'algorithmic', some problems have an 'algorithmic' solution, many others have not.

Let us approach the definition of an algorithm by a route that highlights the need for some distinctions the absence of which has caused a lot of confusion about something being 'algorithmic' (and especially confusion in claims of refuting the *Church-Turing-Thesis*).

Suppose a TM M_1 computes the decimal expansion of Π. Should we say that Π is computable?

1. In a *narrow* sense Π is not computable as we never compute Π (in its entirety). We only compute Π up to some digit. Computable in the narrow sense is the nth digit in the decimal expansion of Π. This computation terminates. One may thus say that Π is computable (in principle). [Turing's classic paper deals with irrationals thus computable!]

2. This gives us a notion of algorithm in the *narrow* sense.

   An **algorithm is**:

   a. **Implementation neutral** (abstract)

   b. **Effective** in its individual steps (they do not require ingenuity)

   c. **Finite** (given finite input and a finite number of machine states in a finite time the algorithm terminates with a finite output in finite space if it terminates)

   We see implementation neutrality in one algorithm being carried out by devices that may differ in their architecture and material. If 'program' is understood as 'program in one or the other programming language' then different programming languages provide different ways to implement one and the same algorithm.

   We see effectiveness in the simple basic steps of the different paradigms of computability (e.g. moving the head of a Turing Machine or reading a register in a register/abacus machine). The individual steps are basic mechanical operations.

3. Partial recursive functions are computable in the narrow sense.
   [Partial recursive functions correspond to Turing Machines, *inter alia*.]

4. If one strengthens the finitude constraint to

   c'. **Finite** (given finite input and a finite number of machine states the algorithm terminates in a finite time with a finite output in finite space)

   partial recursive functions are computable in that narrowest sense as well in as much as there is always an extensionally equivalent terminating total recursive function.

5. In a *broad* sense Π may be said to be computable itself as M_1 computes it. M_1 is algorithmic in a broad sense as it is an infinite sequence of terminating sub-computations.

6. This gives a notion of an algorithm in the *broad* sense.

    An **algorithm** is:

    a. **Implementation neutral**

    b. **Effective**

    c. **Finite in its sub-computations** (each of which given finite input and a finite number of machine states in a finite time terminates with a finite output in finite space)

    Halting algorithms then are those that enter after finitely many of such sub-computations in a halting/acceptance state. Non-halting algorithms either stay in a non-halting state or consist of an infinite sequence of sub-computations.

7. Allowing for unbounded sub-computations violates the finitude constraint. In this way – *inter alia* – the *Halting Problem* becomes solvable.

8. If some notional computing device is (said to be) 'hyper-computational' this means that it can solve problems not solvable by Turing Machines. This capacity depends – in the notional machines proposed so far – on: infinitely many states, infinite input, infinite time, space or infinite precision of measurements – all features beyond the intuitive concept of computation and algorithm which involves *finitude*!
    A special case are Oracle Turing Machines, which resort to *ineffective* steps of computation (by consulting the oracle). All these notional machines defy realization. None of them refutes the *Church-Turing-Thesis* or even the *Physical Church-Turing-Thesis* (that a machine beyond the Turing Limit cannot be realized).

Manuel Bremer 2011/2012