**Church-Turing thesis**

The notion of computable function can be identified with the class of partial recursive functions is known as Church-hypothesis or Church-Turing thesis. The Turing machine is equivalent in computing power to the digital computer.

The Church-Turing thesis states the equivalence between the mathematical concepts of algorithm or computation and Turing-Machine. It asserts that if some calculation is effectively carried out by an algorithm, then there exists a Turing machines which will compute that calculation.

The notion of algorithm, computation, a step-by-step procedure or a defined method to perform calculations has been used informally and intuitively in mathematics for centuries. However, attempts to formalize the concept only begun in the beginning of the 20th century. Three major attempts were made: λ-calculus, recursive functions and Turing Machines. These three formal concepts were proved to be equivalent; all three define the same class of functions. Hence, Church-Turing thesis also states that λ-calculus and recursive functions also correspond to the concept of computability. Since the thesis aims to capture an intuitive concept, namely the notion of computation, it cannot be formally proven. However, it has gain widely acceptance in the mathematical and philosophical community.

The thesis has been wrongly attributed to many controversial claims in philosophy, that although related are not implied in the original thesis. Some examples are:

1. The universe is equivalent to a Turing Machine and non-computable functions are physically impossible.
2. The universe isn’t equivalent to a Turing Machine and incomputable.
3. The human mind is a Turing Machine, the human mind and/or consciousness are equivalent to and can be instantiated by a computer.
4. The human mind isn’t a Turing Machine, the human mind and/or consciousness emerge due to the existence of incomputable process, such as microtubules performing quantum process in the brain

The Church-Turing thesis (formerly commonly known simply as Church's thesis) says that any real-world computation can be translated into an equivalent computation involving a Turing machine. In Church's original formulation (Church 1935, 1936), the thesis says that real-world calculation can be done using the lambda calculus, which is equivalent to using general recursive functions.

The Church-Turing thesis encompasses more kinds of computations than those originally envisioned, such as those involving cellular automata, combinators, register machines, and substitution systems. It also applies to other kinds of computations found in theoretical computer science such as quantum computing and probabilistic computing.

There are conflicting points of view about the Church-Turing thesis. One says that it can be proven, and the other says that it serves as a definition for computation. There has never been a proof, but the evidence for its validity comes from the fact that every realistic model of computation, yet discovered, has been shown to be equivalent. If there were a device which could answer questions beyond those that a Turing machine can answer, then it would be called an oracle.

Some computational models are more efficient, in terms of computation time and memory, for different tasks. For example, it is suspected that quantum computers can perform many common tasks with lower time complexity, compared to modern computers, in the sense that for large enough versions of these problems, a quantum computer would solve the problem faster than an ordinary computer. In contrast, there exist questions, such as the halting problem, which an ordinary computer cannot answer, and according to the Church-Turing thesis, no other computational device can answer such a question.

The Church-Turing thesis has been extended to a proposition about the processes in the natural world by Stephen Wolfram in his principle of computational equivalence (Wolfram 2002), which also claims that there are only a small number of intermediate levels of computing power before a system is universal and that most natural systems are universal.

Turing’s arguments for the thesis

Turing’s argument I involves a number of plausible assumptions about human computers. These include the following :

1. Instead of using two-dimensional sheets of paper, the computer can do his or her work on paper tape of the same kind that a Turing machine uses—a one-dimensional tape, divided into squares.
2. The computer is not able to recognize more than a finite number of different types of symbol.
3. The computer is not able to observe an unlimited number of tape-squares all at once—if he or she wishes to observe more squares than can be taken in at one time, then successive observations of the tape must be made.
4. When the computer makes a successive observation in order to view more squares, none of the newly observed squares will be more than a certain fixed distance away from the nearest previously observed square. In other words, successive observations do not involve unbounded leaps along the tape.
5. When the computer makes changes to the contents of the tape (e.g., by deleting the symbol written in a particular square and replacing it by a different symbol), no more than one square can be altered at once. If the computer wishes to alter, say, 100 squares then he or she performs 100 successive operations.
6. The computer’s behavior at any moment is determined by the symbols that he or she is observing and his or her ‘state of mind’ at that moment; and the number of ‘states of mind’ that need to be taken into account when describing the computer’s behavior is finite. (Turing noted that reference to the computer’s states of mind can be avoided by talking instead about configurations of symbols, these being “a more definite and physical counterpart” of states of mind.)

Turing argued that, given his various assumptions about human computers, the work of any human computer can be taken over by a Turing machine. Whatever sequence the human computer is computing, a Turing machine “can be constructed to compute the same sequence”, Turing said (1936: 77). Therefore (argument I concludes) any humanly computable number—or, more generally, sequence of symbols—is also computable by Turing machine.