Fundamental Computer Science
Sequence 1. Turing Machines
The Church-Turing thesis

Denis Trystram

February, 2021

Let us recall what are **computable functions**:

Such functions are defined by a finite set of instructions whose execution is systematic and leads to the solution (when it exists) in a finite number of steps.

Several classes of functions have been proposed in various contexts, namely, λ-calculus, recursive functions, Turing computable, among the most popular ones[1].

[1] at least from the historical perspective

- In 1933, Godel and Herbrand, created a formal definition of a class called general recursive functions.
  The class of general recursive functions is the smallest class of functions (possibly with more than one argument) that includes all constant functions, projections, the successor function.
- In 1936, Alonzo Church created λ-calculus, another method for defining functions.
  He defined an encoding of the natural numbers called the Church numerals. A function on the natural numbers is called λ-computable if the corresponding function on the Church numerals can be represented by a term of the λ-calculus.
- In 1936 (independently) Alan Turing created his *machine*.
  Given a suitable encoding of the natural numbers as sequences of symbols, a function on the natural numbers is called Turing computable if some TM computes the corresponding function on encoded natural numbers.

Thesis.

It is the hypothesis about the nature of computable functions that states that **a function on the natural numbers can be calculated by an effective method if and only if it is computable by a Turing machine**.

It can not be proved since it makes reference to an informal notion (the computable functions).

Thesis.

It is the hypothesis about the nature of computable functions that states that **a function on the natural numbers can be calculated by an effective method if and only if it is computable by a Turing machine**.

It can not be proved since it makes reference to an informal notion (the computable functions).

Church and Turing proved that these three formally defined classes of functions coincide:
a function is λ-computable if and only if it is Turing computable, and if and only if it is general recursive.