This book is concerned with abstract models of computation. Several new models of computation have emerged in the last few years (e.g., chemical machines, bio-computing, quantum computing, etc.). Also, many developments in traditional computational models have been proposed with the aim of taking into account the new demands of computer system users and the new capabilities of computation engines. A new model of computation, or a new feature in a traditional one, usually is reflected in a new family of programming languages and new paradigms of software development. Thus, an understanding of the traditional and emergent models of computation facilitates the use of modern programming languages and software development tools, informs the choice of the correct language for a given application, and is essential for the design of new programming languages.

But what exactly is a “model of computation”? To understand what is meant by a model of computation, we briefly recall a little history. The notions of computability and computable functions go back a long time. The ancient Greeks and the Egyptians, for instance, had a good understanding of computation “methods”. The Persian scientist Al-Khwarizmi in 825 wrote a book entitled “On the Calculation with Hindu Numerals”, which contained the description of several procedures that could now be called algorithms. His name appears to be the origin of the word “algorithm”: When his book was translated into Latin, its title was changed to “Algoritmi de Numero Indorum”. The word “algorithm” was later used to name the class of computation procedures described in the book. Roughly, an algorithm is: