One of the rationales behind our study of computability was to find out exactly what we meant by the term. To do this we looked carefully at several systems of computation and briefly examined the things they could compute. From this study we were able to define the classes of computable functions and computable sets. Then we compared computation via Turing machines and program execution. We found that they were equivalent. Then we examined extensions to Turing machines and found that these added no computational power. After a brief discussion of whether or not Turing machines can perform every computational task we can describe, we came close to assuming that Turing machines (and programs) can indeed compute everything.

Hardly anything is further from the truth! It is not too silly though, for until the 1930's most people (including some very clever mathematicians) felt that everything was computable. In fact, they believed that all of the open problems of mathematics would eventually be solved if someone ingenious enough came along and developed a system in which the problems could be expressed and either verified or refuted mechanically. But there are things which are not computable and now we shall attempt to discover and examine a few of them.

Thus our next step in uncovering the nature of computation shall consist of finding out what we cannot compute!

The sections are entitled:

- Arithmetization
- Properties of the Enumeration
- Universal Machines and Simulation
- Solvability and the Halting Problem
- Reducibility and Unsolvability
- Enumerable and Recursive Sets

- Historical Notes and References
- Problems