Overview and Description

This course studies the fundamental capabilities and the fundamental limitations of computers. While this is an essential topic in computer science, the methods and style of the course are much more like that of a mathematics course. We will define mathematical objects and prove theorems, just as in a 300-level math course. No programming experience is needed, and writing programs is not a part of this course. The course counts as an elective toward the major or minor in Mathematics, and toward the minor and major in Computer Science. It is highly recommended for students considering graduate study in Computer Science. This course will include the study of the following questions.

1. What kinds of problems can be solved on a computer? Are there problems for which no algorithm exists for solving them?

2. Are there problems that are solvable yet truly intractable, that is, so difficult that no computer can possibly solve them in, say, the lifetime of the universe?

3. What does it really mean for a problem to be NP complete? You may have heard this term in other math classes or in a math coffee. In this class, we will study this idea in detail.

4. Can you name a million-dollar problem? We’ll describe a problem in this course that someone will really pay a million dollars for its solution!

In the first part of the course, we examine several different models of computation, beginning with finite automata, and we study the capabilities and limitations of each. This goes hand in hand with the study of the complexity of grammars and languages. This part of the course culminates in the study of the Turing machine, which is an idealized mathematical model of a computer. This allows us to prove theorems on what can and cannot be computed on any computer. The second part of the course deals with the question of what problems can be solved quickly on a computer, which ones definitely cannot be, and an important class of problems for which this answer is not known. In the last part of the course we will consider some related ideas, like space requirements, in addition to time requirements.

This is all very recent mathematics, compared with most undergraduate mathematics courses. In this subject you will see major contributions made in mathematics by such twentieth-century luminaries as Kurt Gödel, Alan Turing, and Noam Chomsky.

Prerequisites and Textbook

- You should have already completed one of the introductory proof courses in math: CSC/MAT 220, MAT 230, or MAT 255.

- The textbook is *Introduction to the Theory of Computation*, third edition, Michael Sipser, Cengage Learning, 2013. (I recommend looking into the electronic version of the text, which
is much less expensive than the print version. Alternatively, you could look into text rental, or the used book market.)