If you ask a computer science major what their job entails, they might say that they solve complex business, scientific, and general computing problems. Another computer science major might tell you that they write software for businesses. Ask another computer science major, and they might tell you that they work on improving hardware for computers. Because the field of computer science is actually made up of so many subfields, it’s difficult to build a curriculum for a computer science degree that teaches about all of these subfields. The curriculum for a B.S. degree in Computer Science at UCF attempts to teach the different sides of computer science by offering classes that teach both the theory and application of computer science. While I think it is good to be exposed to both the theory and the application of computer science, we have to remember that there are only a limited number of classes that students can be required to take. With only a limited amount of required classes, I think that the main focus of teaching should be on material that will help students get jobs after they graduate with their bachelor’s degree. COT 4210 seems to only teach us material about theory that is already presented in other required computer science classes and the theory that it teaches us does not seem to have any practical application for those graduating with only bachelor’s degrees. It is for these reasons that I think Discrete Structures should be removed as a requirement from the computer science curriculum.

As stated before, there are courses that exist in the computer science curriculum that cover some of the material taught in Discrete Structures. Students are first introduced to Automata Theory in Systems Software. This class covers the concepts of DFAs, NFAs, Context
Free Grammars, and Regular Languages which are all taught again in Discrete Structures. In Computer Science 2, students learn about some of the NP Complete problems and are also shown some mapping-reduction proofs which are taught again in Discrete Structures. Although these topics are covered more in depth in Discrete Structures, the point is that the fundamental ideas of the topics are covered in other classes and covering the fundamentals should be enough to give students a basic understanding of computer theory. True, there are topics in Discrete Structures that the other classes do not cover such as PDAs, Turing Machines, and decidability. However, these topics can be considered to be too theoretical and not practical enough to be able to help a new college graduate get a job.

The material taught in Discrete Structures does very little to prepare students for the jobs they will most likely work at when they graduate with their bachelor’s degree. A lot of computer science graduates with bachelor’s degrees will most likely start off working as programmers. However, Discrete Structures does nothing to expand the students’ programming skills. The skills that Discrete Structures teaches such as proving a language is undecidable and converting NFAs to DFAs are interesting to learn about but they have no obvious application in a programming job where the success of your work is mostly dependant on how skillful you are at the programming language you use. The skills learned in Discrete Structures will only help those that decide to make computer theory their main focus of study. For those who want to learn about the theory of computers and do research in that area, they will almost certainly need to go to graduate school. In fact, most computer scientists that focus on the computer theory are required to possess a Ph.D. Considering that not everyone who gets a bachelor’s degree will go on to graduate school to study computer theory, we should not require students to take classes for their bachelor’s degree that are only really useful at the graduate level.
The number of jobs for theoretical computer scientists, while expanding, is small in relation to the number of jobs for application based computer scientists (such as software engineers and programmers). The number of jobs for theoretical computer scientists in 2008 was 28,900 and the number of jobs is expected to grow 24% by 2018. On the other hand, the number of jobs for application computer scientists in 2008 was 1.3 million and the number of jobs is expected to grow 21% by 2018. Based on the number of jobs available for application based computer scientists, shouldn’t we spend more time teaching undergraduate students concepts that will help them land jobs in the practical fields of computer science rather than teach them theory which will only help them if they decide to go into a theoretical field? Discrete Structures would be a good class to get rid of in order to make room for more practical classes.

Some may say that removing the class will have negative effects. By removing Discrete Structures, some people may argue that not as many students will want to study computer theory since they will not get enough exposure to it when they are an undergraduate. However, I think that the amount of theoretical material taught in Systems Software and Computer Science 2 is enough for students to realize if they want to pursue their interest in computer theory further. Some may also argue that learning theory is better than learning specific skills because skills can change through time while theory will remain the same. While this is true, it’s important to remember that the jobs that most college graduates will have will be jobs where they will need to apply their knowledge in programming and not their knowledge in computer theory.

I agree that students should be required to be introduced to theory since it gives them the opportunity to see if it’s a field they would like to go into. However, considering the lack of practical skills it teaches students, I do not think that it is a good idea to make this class a
required class. The fundamental ideas of computer theory that are presented in other computer science classes should be sufficient enough to give students a glimpse of computer theory. For those who want a deeper look into computer theory than what is presented in the other computer science classes, they should be able to take this class as an elective. With the removal of Discrete Structures as a required class, I feel that students will have the opportunity to gain more practical knowledge which should make it easier finding a job after college.