CS 39: Theory of Computation **Winter '23**

Participation Statement

Prof. Chakrabarti ***Student: Amittai Siavava***

My primary interests in computer science are in systems engineering and in deep learning, and I had seen finite automata show up in both areas.

1. CS-51 (Computer Architecture), one of the projects entailed constructing a CPU using circuits, and to make it work we had to program a finite state machine that performs all the micro-operations For example, a simple instruction such as adding the contents of two memory addresses and storing the result into a third memory address would be broken down into a sequence of transitions, each of which does a single thing (move A, move B, add A and B, store ALU result in memory, loop until the memory write is finished…) This was very similar to the random access machine that we later discussed in CS-39, and this class gave me a better understanding of how such computation works.
2. One of my other areas of interest is deep learning, and One of the current issues is that although we can change parameters, refine the training data, etc., to improve the performance of deep neural networks, most of the internal workings is not fully understood — akin to a black-box that you tune then feed in input and get output, which, usually, happens to have high accuracy. Late last year, a team of German researchers experimented with modeling deep neural networks with finite automata that the input-output behavior of the neural network. Studying these finite automata and how they change when the neural network's training parameters are modified could then give insights into neural network itself.

That aside, decidability and computability were the most intriguing topics to me. They also both showed up in one of my other classes, Math 69 (Mathematical Logic) and it was amazing to see two disparate vantage points converge to the same ideas. One of the ways I engaged with the course content was through weekly discussions I had with Paul Shin, one of my classmates. We would meet on weekends to talk about the problem sets and often spent hours talking about the course content in general and related problems. For example, he was doing a directed study in category theory this term, and I have some experience with functional programming which uses ideas from category theory to model computation. These discussions particularly resonated with me because I do not always feel comfortable speaking up in class where there's multiple other students.

I did not get time to implement the machines we were studying in class due to the volume of work I had this term. It is something I am excited to do over the break.