Eric Rothman
Automata and Computation Theory
Writing 3

Quesiton: Suppose you can implement NFAs in practice, what do you think is the advantage of NFAs compared to DFAs? Do you think it is a good idea to introduce non-determinism in computation? Why?

Answer:
Some of the advantages of NFA's are that they can be easier to understand and may be smaller than their DFA counterpart. While it is possible to implement every NFA as a DFA much more room and space would probably be necessary to map out all the different paths the NFA would take in computation. It also probably takes less room since you don't need to remember all the states that are currently acceptable if it is an NFA that is implemented. I think it is a good idea to consider non-determinism when computing, but for actual implementation purposes and always solving with it, I don't think its good. In general I don't think it is good for a computer. The reason being that most programs, except very complicated ones, work in a linear fashion and don't need parallelism. But the easiest way to implement NFA's are using parallelism which is hard to model as a program. On the other hand, the fact that they are smaller and simpler to understand are a huge advantage when they are introduced to computation. So actually I change my mind. I think it is good in general for computation to consider non-determinism since it allows for simpler programs.