The Epic Automata Showdown

Team Cellular Automata

Thus far in class, we have discussed two powerful computational engines: 1D-Cellular Automata and Turing Machines. Neither are bound by the pumping lemma; both operate on (theoretically) infinite data. It is time for the no-holds-barred showdown to determine who the most powerful computational machine is.

# Your Goal

Your goal is to prove that a 1D-cellular automaton can simulate Turing machines. Do this by devising a procedure for converting a Turing machine to an equivalent cellular automaton. Some guidelines:

* Use as many colors and as wide a “looking distance” as you need…you know that anything you build can be converted into a 16-color 3-look automata after all.
* Feel free to make up your own decision for how the automata will signal acceptance or rejection.
* The Turing Machine’s initial tape state can be encoded in the initial state colors of your automata in any way you find convenient.
* You can assume the Turing Machine has only 1 tape.
* You don’t need to spec out every color and transition. Just make a convincing argument for how such cellular automata could be built.



The Epic Automata Showdown

Team Turing Machine

Thus far in class, we have discussed two powerful computational engines: Cellular Automata and Turing Machines. Neither are bound by the pumping lemma; both operate on (theoretically) infinite data. It is time for the no-holds-barred showdown to determine who the most powerful computational machine is.

# Your Goal

Your goal is to prove that Turing machines can simulate 1D-Cellular Automata. Do this by devising a procedure for converting a cellular automaton to an equivalent Turing machine. Some guidelines:

* You can assume the cellular automata has no more than 3 cells of “looking”. The cellular automata can have more than 2 colors however.
* No need to go crazy wondering what an accept state looks like. Just concentrate on finding a way to simulate the way the automata’s cells change over time.
* You can use 2 or more tapes if that helps.
* The automata’s initial state can be represented any way you find convenient as input on your tape.
* You don’t have to specify every state and transition. Just make a convincing argument that such a Turing machine could be built.

