# HW4, extra problem

Prove that a 1-dimensional Cellular Automaton is equivalent to a Turing Machine.

A 1-dimensional cellular automaton is defined here as a set $Q$ of states, an accept state $q_a \in Q$, a reject state $q_r \in Q$, a "default" state $q_0$, and a transition function $\delta : Q \times Q \times Q \to Q$ such that $\delta(q_0, q_0, q_0) = q_0$. The global configuration of the automaton at time $i$ is defined by the function $C_i : \mathbb{Z} \to Q$ (where $\mathbb{Z}$ is the set of all integers), which is equal to $q_0$ for all but a finite set of values. Computation over a certain alphabet $\Sigma \subseteq Q$ is defined by setting $C_0(i)$ to the $i$'th character input string (for $i = 0 \ldots n$, where $n$ is the size of the input), and to $q_0$ elsewhere. At any step $j$, then, the new configuration of the automaton is defined by $C_j(i) = \delta(C_{j-1}(i-1), C_{j-1}(i), C_{j-1}(i+1))$. The automaton accepts at step $j$ if there exists an integer $n$ such that $C_j(n) = q_a$; if there isn't, then the automaton rejects if there exists an integer $n$ such that $C_j(n) = q_r$.

Note that the "global configuration" function above can be thought of as an infinite 1-dimensional tape, with blanks ($q_0$'s) everywhere except a finite part of it. Each step changes a cell only based on the previous values of that cell and the two cells adjacent to it.