CS 6505 - Homework 4

Caitlin Beecham (Worked with Qiaomei Li)

1

Our algorithm is as follows: If a turing machine uses O(log n) space that means that it uses less than $c * log(n)$ space for some sufficiently large value of n. How much time can such a machine use? Well, assuming such a machine halts, it may never repeat a configuration. (Configuration at a certain point is defined as: the contents of the entire tape, the state, and the position of the head at that point in time). How many such configurations may such a turing machine cycle through then? It may cycle through (in some order) at most $|\Gamma|^{clog(n)} \times (clog(n) + n) \times |Q| = A^{clog(n)} * (clog(n) + n) * B$. Since at each time step, the turing machine changes from one configuration to another and it halts, it may use at most $A^{(clog(n))} * (clog(n) + n) * B = 2^{(log(n))(log(A))c} * (clog(n) + n) * B = n^{(clog(A))} * (clog(n) + n) * B = Bn^{(clog(A)+1)} + B(clog(n)) * n^{clogA} \leq Bn^{clog(A)+1} + Bc * n * n^{clogA} = 2Bc * n^{clogA+1} = O(n^{clog(A)+1}) = O(n^{(clog(|\Gamma|)+1)})$ time. So, for each turing machine M in $SPACE(O(logn))$, let the corresponding $k = clog(|\Gamma|) + 1$ where c is a constant such that the space used by M is less than $clog(n)$ for sufficiently large n.