The Theory of Computation

Bernard M.E. Moret

Addison-Wesley-Longman, 1998

Solution Manual
Chapter 4, Problem 5.9

The basic idea is quite simple: a write-once machine will simply copy its entire tape over to a new block of fresh tape whenever it needs to simulate a normal machine that alters a tape square. Of course, since the machine writes just once, it will never be able to clean up or reuse old blocks of tape, so that it will use much more tape than the original machine.

It remains to verify that we can copy a block of tape with just one write; our normal copy routine marks copied squares, but that constitutes a *second* writing of the square. So we can do it with a machine that writes each square at most twice. Can we do it with just one write? A bit of thought reveals that we can with a simple trick, but only if we assume that our machine need not write every square that it scans—it is allowed to leave the scanned square unchanged by not writing on it. This is not our standard model, but is needed for our purposes. Our new machine then uses only one out of two tape squares for data used directly in the computation ; the others are used for makers in the copying process. When copying to a fresh block of tape, the machine leaves every other square blank (they are blank to start with) in the new block; when copying that block, it writes a mark in the blank square corresponding to (adjacent to) the square being copied. Fresh blocks of tape are thus separated by at least two blanks or by a special symbol.

When simulating a conventional Turing machine that runs in $f(n)$ time and $g(n)$ space, our new machines takes $O(f(n) \cdot g(n))$ time and space.