CSE 531
Assignment 1

Due October 5, 2000

One of the annoying things about one tape Turing machines is that they take so long to access data. The head has to lumber up and down the tape looking for data. At most $2t + 1$ different cells can be accessed in t steps. Multiple tapes don't help much. Two-dimensional tapes are better because $O(t^2)$ different cells can be accessed in t steps. Back in the 1970s Arnold Schonhage got a little fed up and invented the storage modification machine (SMM) (cf. SIAM J. on Computing 9:3 (1980), 490–508). The SMM has the ability to access $O(c^t)$ different cells in t steps.

A SMM can described briefly as follows.

1. There is a read only *input tape*. When an input symbol from Σ ($m = |\Sigma|$) is read, the machine branches to an instruction based on the symbol and the input head is advanced.
2. There is a write only *output tape*. When a symbol from Σ is output, it is placed on the output tape and the output head is advanced.
3. There is a *storage structure* consisting of finite number of cells that can be accessed by a single pointer called ε (also signifying the empty string). Each cell has k pointers where k is the cardinality of the pointer alphabet Γ. Each pointer either points to a cell or is *nil*. A string in Γ^* identifies a pointer which starts at ε and traverses the cells corresponding to the string. For example, we might have a storage structure which consists of a tree where with ε pointing to its root. With the pointer alphabet $\{\ell, r\}$ we can define ℓ to point to the cell to the left and r to point to the cell to the right. The pointer ℓrr points to left, then right, then right of the root. There is no restriction on the the storage structure. It is up to the SMM program to define it. Initially, the storage structure is empty and the pointer ε points to nil.
4. There is a finite length *program*. Each instruction in the program has a label (σ), so we have essentially a finite state control. Starting with the first instruction, the program instructions are executed in order unless there is a branch. Here is the list of possible instructions:
   (a) Input $\sigma_1, \ldots, \sigma_m$. Read an input symbol and branch according to which input symbol was read.
   (b) Output a. Output the symbol a.
   (c) Goto σ: Branch to the instruction σ.
   (d) Halt.
   (e) New P. Creates a new cell and point P to it. Recall P is a member of Γ^*. The new cell has all it pointers set to nil.
   (f) $P := Q$. Set the pointer P to Q.
   (g) if $P = Q$ then σ_1 else σ_2. Test whether P and Q point to to the same cell. If so go to instruction σ_1 else go to instruction σ_2.

Do the following two problems.

1. Describe how a SMM can be simulated by a multiple tape Turing machine. (Hint: think how an arbitrary data structure would be stored on backup tape.)
2. Describe how a one tape Turing machine can be simulated by a SMM. (Hint: you will have to figure out how to model tape symbols and the head.)