Assignment 4 SOLUTIONS **SE4I03 Fall 2002** **due on Nov. 15**

1. The class of languages recognizable by this model is bigger. The reason is that with two stacks we can simulate a Turing Machine M as follows: first, the input that would be presented to M on its tape, is pushed on the first stack of the two-stack push-down automaton P, and then all but the last symbol (which is the first symbol of the input) are pushed onto the second stack.

   Let (q,u,v) be a *configuration* of M; that is, (q,u,v) means that M is in state q, uv is contained on the tape, and the head is reading the last symbol of u. This completely describes M at any point of its computation. The configuration (q,u,v) is simulated by P as follows: P is in state q, u is in the first stack, and v in the second.

   Suppose now that the next configuration of M is (q',u',v'). Then the next state of P is q', and depending on u',v', P pushes/pops or does nothing with the two stacks. For example, if $u'=uv_1$ and $v'=v_2\dots v_k$ (that is, the head of M did not change the last symbol of u, and moved one square to the right), then P would pop the top of the second stack, and push it on top of the first stack.

2. Regular languages. We show how such a Turing Machine can be simulated by a DFA. First of all, since the work tape is bounded by c in both directions, for any given alphabet Σ, the contents of the work tape can be kept track with $|\Sigma|^{2c}$ many states. Thus the states of the DFA are going to be pairs (p,q), where p is a state of the TM, and q is one of the $|\Sigma|^{2c}$ states that describe the contents of the work tape. Now, as the TM reads its input left to right, so does the DFA, but instead of writing on the work tape, it keeps track of the contents of the work tape by changing states.

3. We show that (a) and (b) are polynomially equivalent, and (a) and (c) are polynomially equivalent; this will be enough. A TM with an infinite tape in one direction can be trivially simulated by a TM with a tape infinite in both directions (by using just the right half of the tape). In the other direction, we have to show how to simulate a TM with a TM where the tape is infinite only in the left direction. Furthermore, the simulation has to be such that the increase in the number of steps in the computation is small.

   The idea is simple: let M be a TM with a tape infinite in both directions, let M' be the restricted TM. Let the positive squares of M correspond to the even squares of M', and the negative squares of M to the odd squares of M'. Let the states of M' be pairs $(q,e/o)$, where q is a state of M and e,o are two states that inform M' whether it is in an even or odd square. Suppose that M' is in e. Then, when the computation takes it back to the first square, and at that point M would go over to the left side of the tape, M' changes from e to o and continues simulating M on odd squares. The slow down is a factor of 2, since for every step of M, M' has to do 2 steps.

   To show the polynomial equivalence of (a) and (c), just note that the contents of the k tapes of M can be stored on the single tape of M', by giving M' special symbols to separate the contents of the k tapes on its single tape (for example, $\Box$ could signal that the contents of a tape end, so the single tape of M' would contain $x_1\Box x_2\Box\ldots\Box x_k$). One step of M would have to be implemented on each of the x_i separately on M'. The crucial observation is that if M takes $f(|x|)$ many steps, then none of its tapes can have more than $f(|x|)$ symbols, i.e., at all times $|x_i| \leq f(|x|)$. So each of the $f(|x|)$ steps of M, can be simulated on M' with $O(kf(|x|))$ many steps, so M' takes in total $O((f(|x|))^2)$, so the increase is polynomial.