

## Recitation 5: Turing Machines

March 8, 2007

Elena Grigorescu

**Problem 1:** Define the following words, phrases and symbols.

1. Turing Machine
2.  $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$
3. (Turing) Decidable Language
4. (Turing) Recognizable Language
5. (Recursively) Enumerable Language

**Problem 2:** Mark each of the following statements either true or false.

1. A Turing machine has a single start state, but may have many accept states.
2. It is possible to make a Turing machine with only one state.
3. A Turing machine halts when its head reaches the end of its input.
4. All decidable languages are regular languages.
5. A nondeterministic TM with  $k$ -heads can recognize more languages than a deterministic TM with  $k$ -tapes.
6. A Turing machine might not halt on a finite input string.
7. A language  $L$  can be both co-decidable and recognizable.

**Problem 3: Implementation-Level (also known as “higher-level” Turing machine description)**

Describe the operation of a basic Turing machine that *decides* the language

$$L = \{w : w \text{ does not contain twice as many 0s as 1s}\}$$

**Problem 4:**

1. (Solved in the textbook). Show that the language

$$EQ_{NFA} = \{\langle N_1, N_2 \rangle : N_1 \text{ and } N_2 \text{ are NFAs and } L(N_1) = L(N_2)\}$$

is decidable.

2. Show that

$$ALL_{NFA} = \{\langle N \rangle : N \text{ is the description of an NFA that accepts all strings in } \Sigma^*\}$$

is decidable. (Hint: Use 1. )

Recall the variants (multitape, multihead, nondeterministic) of TMs discussed in class that are equivalent in power with the single-tape deterministic TM. We discuss next one more such TM model.

**Problem 5: Robustness of the Turing Machine model**

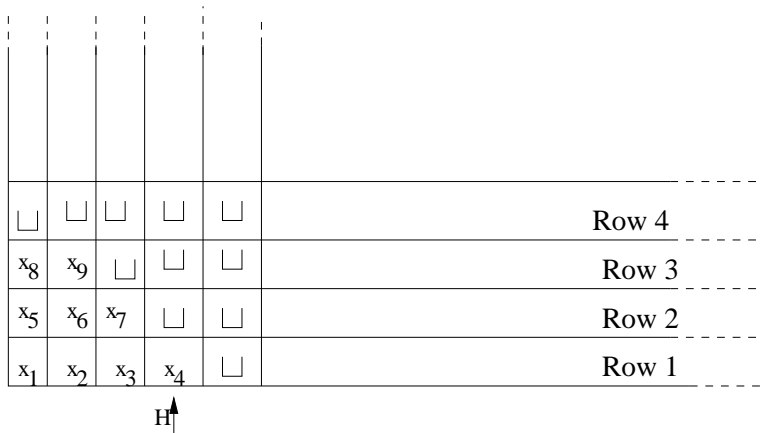
Consider a Turing machine model that uses a 2-dimensional tape, corresponding to the upper right quadrant of the plane. The head of such a Turing machine could move to the right, left, up or down. Sketch a proof that such a model does not add extra computing power; that is, the class of languages recognized by such Turing machines is the same as the class recognized by basic Turing machines.

**Solution 5:** *The key idea is to notice that at any point of time, the TM has looked at only a finite region of the two-dimensional tape.*

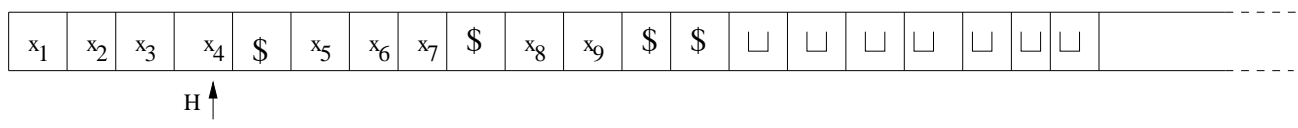
Say the input of the machine is in the first row of the tape, initially the tape head is at the square  $(0, 0)$  and all the other tape squares are blank.

Simulate the 2D tape with a 1D tape as follows:

1. The rows of the 2D tape starting from the “bottom-most” row are written in the 1D tape next to each other, as in the figure below. Note that this is indeed possible, because at any point of time, the TM has looked at only a finite number of rows, and a finite number of tape squares in each row.
2. When the machine tries to move right from a square in the 2D tape, we can simulate it in the 1D tape trivially, except when it tries to move right from the rightmost square in a row. In that case, we “make room” for the new tape square by moving all the symbols to the right. For instance, when the machine tries to move right after reading  $x_7$  in the figure, we simulate the effect in the 1D tape as follows: We move all the symbols starting from the \$ right after  $x_7$  in the 1D tape, one square to the right. In the newly vacant square, we write a blank, and continue simulating.
3. When the machine tries to move left from the leftmost square in a row, we do nothing (*The machine hits the wall*). Otherwise, we can trivially simulate it in the 1D tape.
4. When the machine tries to move up, we move left until we reach a \$ symbol to determine which column we are in. Then we move right beyond the next \$ and into the corresponding column one row up. (To count, we might use an auxiliary tape and later use the 2-tape TM to 1-tape TM conversion). If there is not enough room in the next row, we make room by moving all the tape squares to the right, just as in step 2.
5. The machine tries to move down. We proceed exactly as in Step 4, except that now, we move one row down. In case we are in the bottommost row, we do nothing. (*Again, the machine hits the wall*).



The configuration of the 2D tape at some point of time



The representation of the 2D configuration in the 1D tape