Ethan Wong
11/11/21

CS 181 HW 4

1. The idea for this TM is similar to MAJ. Instead of looking for a single 1 when we encounter a 0, we have to look for two 1's. This guarantees there are atleast twice as many 1's as 0's.

Pseudocode:

① Scan to the right until you encounter a 0
② If a 0 is not encountered, cleanup the tape and write 1 on the tape (TM returns true)
③ If a 0 is encountered:
  ⓐ Mark to 0 as * (mark it as seen)
  ⓑ Go to the start of the input
  ⓒ Scan to the right until you encounter a 1
    i) If you encounter a 1, mark it as *. Keep scanning the input to find another 1. If you can find it, mark it as * and return to the start state. If the second 1 is not found cleanup the tape and write 0 (TM returns False)
    ii) If the initial 1 was not encountered, cleanup and write 0 to the tape.

Cleanup Pseudocode:

① Keep going left while rewriting each symbol as ϕ
② Once you reach the start of the tape, move one symbol to the right
③ Write 1 if TM returns true and write 0 if if TM returns false, then halt

textbook p278

2. Prove that every function $F: \{0,1\}^* \to \{0,1\}^*$ is only computable by a standard TM if it can be computed by a two-tape TM.

Multiple head / Multiple tape TM denoted by:

$\delta: [K]^l \times \Sigma^l \to [K]^l \times \Sigma^l \times \{L, R, S, H\}^l$

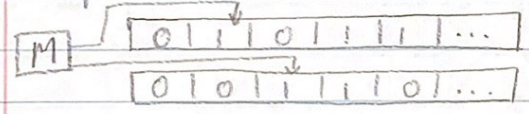
K = number of states l = number of symbols in alphabet

Say that M has l tapes, and S simulates M but has a single tape

We can use '#' as a delimiter to separate the contents of the multiple tapes

S also needs to keep track of the multiple tape heads; this can be done by placing a '*' above the space where the Tape head should be at

Ex:



S can simulate M by:

① Scanning right from the first '#' (start of input) to the final '#' (end of input), S can look at the symbols with * on them to analyze each of the individual tapes in M.

② S can then pass through the selected tape to modify values based on its transition function.

③ If S tries to mark a '#' delimiter with the '*' symbol, it is trying to write in an invalid area. We then insert a '∅' symbol and shift all contents one space to the right.

This shows how a multiple tape, multiple head TM can be simulated by a single tape TM. A single tape TM can be simulated by a multiple tape TM simply by adding a dummy tape that does nothing. Thus, every function $F: \{0,1\}^* \to \{0,1\}^*$ is only computable by a standard TM if it can be computed by a two-tape TM.

1100 If zero, flip, move left, repeat, flip first 1, then halt
1011

1101 If one, flip, halt

0010
0011
0001

1000
1001
1011
1111
0111

3. The idea for this TM is to start from the end of the input. If the final bit is a 1, we simply need to flip it to a 0 then halt. If the final bit is a 0, we flip it to a 1 and move left. This continues as long as there are more 0's. The first 1 we encounter should be flipped to a 0 as well, then we halt. The input will have been decremented by one.

Pseudocode:

① Traverse all the way right so we are at the final bit of the input

② If the final bit is 1, flip it to a 0 and return/halt

③ If the final bit is 0:

ⓐ Flip the 0 to a 1 and move left. Keep doing this until we encounter a 1

ⓑ Once we encounter a 1, flip it to a 0 and return/halt

Example inputs:

Input: 1100 → 1101 → 1110 → 1011 ✓

Input: 0010 → 0011 → 0001 ✓

Input: 1000 → 1001 → 1011 → 1111 → 0111 ✓

Input: 0011 → 0010 ✓

use decbyone on index
go to #
go right that many times
write that value on the tape

11 # [1,0,1,0,1,0]
↓
3

4. The idea for this TM is to use 'DecbyOne' on the counter variable 'i' until it reaches 0. We then create a second tape that only contains x. The tape head is at the start of x. After every decrement we move the head to the right once. When 'i' reaches 0, we cleanup and return that value.

Pseudocode:

① Create a second tape that only contains x. Have the tape head be at the start of x (x[0])
② Call 'DecbyOne' on 'i' in the first tape and move the head of the second tape right by one. Keep doing this until i = 0.
   ⓐ When i = 0, cleanup the second tape and write the result. The result is wherever we stopped once i = 0.

Cleanup Pseudocode:

① Scan all the way to end of input
② Keep going left while rewriting each symbol as ∅
③ Once you reach the start of the tape, move one symbol to the right
④ Write the result