Computability and Computational Complexity, A.Y. 2019–2020

Guide to the answers

Wednesday, September 2, 2020

Consider the following language on the two-symbol alphabet $\{\mathtt{0},\mathtt{1}\}$:

$$L = \{\mathtt{0}^n\mathtt{1}^m \,|\, n, m \in \mathbb{N} \wedge n > m\}.$$

In plain terms, a string is in L if and only if it starts with a sequence of 0's followed by a (possibly empty) sequence of 1's and nothing else, with strictly more 0's than 1's.
Some examples:

$$\begin{array}{ccc} \mathtt{00011} \in L & \mathtt{00111} \notin L & \mathtt{0} \in L \\ \mathtt{1} \notin L & \mathtt{10} \notin L & \mathtt{11000} \notin L \\ \mathtt{0110100} \notin L & \mathtt{0000} \in L & \mathtt{0011} \notin L \\ & \varepsilon \notin L. & \end{array}$$

Exercise 1

1.1) Write down a one-tape deterministic Turing Machine $\mathcal{M}$ on the three-symbol alphabet $\{\mathtt{0},\mathtt{1},\llcorner\!\lrcorner\}$ that, given an input string $s \in \{\mathtt{0},\mathtt{1}\}^*$, decides $s \in L$.
You may assume that the input string s is surrounded by infinite blank cells $\llcorner\!\lrcorner$ in both directions, and that the initial current position is the leftmost symbol of s.
1.2) What is the time complexity of your machine $\mathcal{M}$?
More precisely: if n is the input size, what is the smallest exponent k such that $\mathcal{M} \in$ DTIME(n^k)? Explain briefly.

Solution 1

1.1) A TM could, for example, keep erasing the leftmost zero and the rightmost one, until only zeroes remain. Any other unexpected condition (zero following a one, no zeroes, and so forth) must cause rejection.
As an example, here is a description of a machine that can be copied and pasted on `http://morphett.info/turing/turing.html`:

```
; Recognize 0^m1^n with m>n.

; Go back and forth, repeatedly removing leftmost 0 and rightmost 1
; until just excess 0's remain. Any other outcome causes rejection.
; The initial state is erase_leftmost_0

; Initially, erase the obligatory leftmost 0
erase_leftmost_0  0 _ r skip_0_right  ; skip all other 0's to the right
erase_leftmost_0  * * r halt-reject   ; Any other symbol, reject

; Keep skipping all 0's to the right
skip_0_right      0 0 r skip_0_right  ; As long as 0, skip it
skip_0_right      1 1 r skip_1_right  ; Upon 1, start skipping 1's
skip_0_right      _ _ l halt-accept   ; All 0's skipped, no 1's: OK

; After all 0's were skipped, start skippping 1's to the right
```

```
skip_1_right      1 1 r skip_1_right  ; Keep moving as long as it's 1
skip_1_right      _ _ l erase_rightmost_1 ; move back to erase the last 1
skip_1_right      0 0 r halt-reject   ; After the 1's, there shouldn't be 0's

; Erase the rightmost 1
erase_rightmost_1 1 _ l skip_1_left   ; then start moving left

; Move to the left skipping all 1's
skip_1_left       1 1 l skip_1_left   ; keep skipping 1's
skip_1_left       0 0 l skip_0_left   ; all 1's were skipped, start with 0's
skip_1_left       _ _ l halt-reject   ; no 0's means error

; Keep moving to the left skipping all 0's
skip_0_left       0 0 l skip_0_left   ; as long as there are 0's
skip_0_left       _ _ r erase_leftmost_0 ; passed the whole string, restart
```

Clearly, any description is acceptable, and some missed halting conditions can be forgiven.

1.2) The machine performs back and forth passes on the n-symbol input string, and removes a symbol with every pass. Therefore, its time complexity is $O(n^2)$.

Exercise 2

Prove that the language L belongs to the complexity class **L**.

Solution 2

A two-tape Turing Machine just needs to initialize a counter to zero on the second tape, then scan the input string; it increments the counter as long as it finds a zero, then decrements it as long as it finds ones, halting in any other case or when the counter returns to zero.
Since the counter never counts more than the number of input symbols, its size is logarithmic with respect to the size of the input.

Exercise 3

Is it always possible for an instructor to correctly evaluate a student's answer to 1.1? Explain.

Solution 3

Giving a positive or negative evaluation to the student's answer amounts to deciding the properties "$\mathcal{M}$ decides L" and "$\mathcal{M}$ doesn't decide L" for the machine $\mathcal{M}$ that he described.
Both properties are semantic, therefore by Rice's Theorem they cannot be decided: there will be some machines for which the instructor won't be able to say whether they correctly answer the question or not.