Iniziato	venerdì, 26 giugno 2020, 12:20
Stato	Completato
Terminato	venerdì, 26 giugno 2020, 13:20
Tempo impiegato	59 min. 49 secondi
Valutazione	Non ancora valutato

Domanda **1**
Completo
Punteggio max.: 7,00

Construct a deterministic TM of the kind you prefer, which decides the following language:

$$\mathcal{L} = \{w \in \{0,1\}^* \mid \text{if } 0^k \text{ is a subsequence of } w, \text{ then } k \leq 2\}.$$

Study the complexity of TM you have defined.

1-tape TM.

states Q = {q_init, q0, q1, q2, q_res, q_halt}
alphabet A={0,1, start, blank}

Transition function:
(q_init, start) -> (q0, start, R)
(q0, 1) -> (q0, 1, R) # If we meet a 1, we keep scanning the string for 0s
(q0, 0) -> (q1, 0, R) # We found a zero
(q0, blank) -> (q_res, blank, S)
(q1, 0) -> (q2, 0, R) # Found a second zero
(q1, 1) -> (q0, 1, R) # found a 1, get back to search 0s
(q2, 1) -> (q0, 1, R) # after 2 zeros, we can only accept if we find a 1 or blank
(q1, blank) -> (q_res, blank, S)
(q2, blank) -> (q_res, blank, S)
(q_res, blank) -> (q_halt, blank, S)

TM = (Q, A, delta)

Time complexity:
The complexity is linear w.r.t. input string size because we only pass the string once.

Domanda **2**

Completo

Punteggio max.: 7,00

You are required to prove that the following problem $\mathcal{L}$ is in $\mathbf{NP}$. To do that, you can give a NTM or define some pseudocode. The problem is one that, given a natural number n, checks whether n is a sum of powers of 3.

We have to check that any solution for the problem can be verified in polynomial time. We do it by defining a pseudocode.
Intuition: Powers of 3 are all dividable by 3. The sum of multiple powers of 3 is also dividable by 3.
So we can just check whether a number "n" is dividable by 3, since we can always express this number "n" as a multiple sum of 3s (3 multiplied by the result of n/3).
This is correct because 3 is 3^1, i.e. a power of 3.

Algorithm:
```
 INPUT: n
 if (n % 3 == 0) then:
  return True
 else:
  return False
```

Time complexity:
The division can be easily computed in polynomial time by a TM w.r.t. to the length of the encoding of n.
So the algorithms runs in polytime.
Since the algorithm checks whether a solution belongs to the language L, we have proved that L is in NP.

Domanda **3**

Completo

Punteggio max.: 6,00

The most popular textbook on AI is the one by Russell And Norvig, called "Artificial Intelligence: a Modern Approach". The algorithms presented in it includes many algorithms used in various branches of AI. Let $\mathtt{RN}$ be the set of algorithms explained in the book, appropriately encoded in a succinct way as finite binary strings. To which complexity class does $\mathtt{RN}$ belong? Prove your claim.

We can say that RN belongs to NP class.
In fact we can easily verify that an algorithm belongs to the book in polynomial time, by looping all the algorithms one-by-one in the book and say if it's equal to the one taken into consideration.

We could now ask if it's also in P.
We can express a define rule to generate any string of RN set. This rule is used to generate an encoding of an algorithm. (after defining for example a mapping from set of algorithms in the book to natural numbers, and from natural numbers to their binary encoding)
So after defining a way to generate a binary encoding which maps to a certain algorithm of RN set, we can easily generate a string belonging to the set of encoding of the RN set using the same rule.
Thus we can say that RN is also in P.

Commento:
Just overly complicated.

◄ LAAI - 26062020 - Questions

Vai a...