

Problem 1.

You are employed as a programmer, and you are asked to write the given program. For each case, write "Y" or "N" if the program can or cannot be written. Do not concern yourself with memory limitations, that is, assume that the computer used to run your program has a large, effectively unbounded, amount of memory.

Solution**Part (a)**

receives in input a generic C program x , and counts the number of statements in x .

Y

Part (b)

receives in input a generic C program x , and an input string w , and counts the number of statements executed at least once when x runs on input w .

N

Part (c)

receives in input a generic C program x and an input string w , and counts the number of statements never executed when x runs on input w .

N

Part (d)

receives in input a generic C program x and decides whether x is syntactically correct.

Y

Part (e)

receives in input two natural numbers and computes a specific function $f : \mathbb{N}^2 \Rightarrow \mathbb{N}$.

N

Part (f)

receives in input a generic arithmetic expression e composed of integers and the four arithmetic operators, and computes its value.

Y

Part (g)

halts on the empty string

Y

Part (h)

receives in input a generic C program x and decides whether x halts only on the empty string.

N

Part (i)

receives in input two generic regular expressions and decides whether they are equivalent.

Y

Part (j)

receives in input a generic C program x and the name of one of its functions, f , and decides whether x can ever call f .

N

Part (k)

receives in input a generic C program x , an input string w , and the name of one of its functions, f , and decides whether x calls f when running on input w .

N

Part (l)

receives in input two generic C program x_1 and x_2 and an input string w , and decides whether x_1 and x_2 produce the same output when running on input w .

N

Part (m)

receives in input two generic C programs x_1 and x_2 and decides whether x_1 and x_2 produce the same output when running on every possible input.

N

Part (n)

receives in input two generic C programs x_1 and x_2 , and decides whether x_1 and x_2 produce the same output when running on at least one input.

N

Part (o)

receives in input a generic C program x , an input string w , and a natural number n , and decides whether x uses less than n bytes of memory when running on w .

Y

Part (p)

receives in input a generic C program x , an input string w , and decides whether there is an $n \in \mathbb{N}$ such that x uses less than n bytes of memory running on input w .

N

Problem 2.

Use reduction to prove that the language

$$L = \{p(M)p(w) : \text{the TM } M \text{ never enters its initial state again when running on } w\}$$

is undecidable.

Solution

Proof. via contradiction

□