Problem

Let A be the language of properly nested parentheses. For example, (()) and (()(()))() are in A, but)(is not. Show that A is in L.

Step-by-step solution

Step 1 of 2

The class L: L is the class of languages that are decidable in logarithmic space on a deterministic truing machine.

That is $L = SPACE(\log n)$

Given that

'*A*' be the language of property nested parenthesis.

For example, $(( ))$ and $((( )))$ etc are in *A*. But not $)($.

Comment

Step 2 of 2

We have to show that *A* is in *L*.

That means, we have to construct deterministic Turing machine (DTM) that decides *A* in logarithmic space.

Let *M* be the *DTM* that decides *A* in logarithmic space.

The construction of *M* is as follows:

$M =$ "On input *w*:

Where *w* is a sequence of parenthesis.

1. Starting at the first character of *w*, move right across *w*.

2. when left parenthesis '(' is encountered, add 1 to the work – tape and move right.

3. when right parenthesis ')' is encounter and the work tape is blank, then reject.

Otherwise subtract 1 from the work – tape and move right.

4. When the end is reached, accept if the work tape is blank, reject if the work tape in not blank."

• Clearly, the only space used by this algorithm is for the counter on the work tape.

• If this counter is in binary, then the most space used by the algorithm is $O(\ln k)$

Where *k* is the number of $(s$.

• Since the number of $(s$ is less than or equal to *n* (the size of tape), this places the language *A* in *L*.

Thus we proved that $A \in L$

Comment