Problem

Let B be the language of properly nested parentheses and brackets. For example, ([()()]()[]) is in B but ([)] is not. Show that B is in L.

Step-by-step solution

Step 1 of 3

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,

‘ B ’ be the language of property nested parenthesis. For example, $[(\ )(\ )]$.

But not $([)]$.

We have to show that B is in L.

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

Comment

Step 2 of 3

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

The construction of *M* is as follows:

M = “On input w:

Where *w* is a sequence of parenthesis and brackets.

1. if $w = \in$ then accept and halt.
2. Otherwise, take a counter $i = 0$.
3. Read the input sequentially from left to right.
4. When $[$ is read from the input, increment *i* by 1.
5. When $]$ is read from the input, decrement *i* by 1.
6. If *i* becomes negative or is not restored to zero at the end of the input, then reject.
7. Otherwise, for each symbol ‘ a ’ in the input sequentially from left to right repeat the following.
8. If a is $)$ or $]$, skip it.
9. Otherwise, ‘ a ’ is a left delimiter (or $[$, using a counter find our the matching right delimiter ‘ b ’ as we done from step 2 to step 6.
10. If b is not found, or if a and b are of different types (parenthesis and bracket or bracket and parenthesis), then reject.
11. If not rejected so for then accept”.

Comment

Step 3 of 3

- Clearly the only space used by this algorithm is for the counter on the work tape.
- If this counters are binary, then the most space used by algorithm is $O(\ln k)$
- Where k is number of brackets and parenthesis
- Since k is always less than or equal to size of the tape this places the language B in L.
- Thus we proved that $B \in L$.So, B is in L.