-2

Turing machine for adding numbers

algorithms turing-machines arithmetic

I'm having trouble in a certain problem.

I have to write a TM that gets as an input strings $\#x\#y$ such that $x, y \in \{0,1\}^*$, and writes to the tape the output $z\#x\#y$ where $z = x + y$, all in binary base 2.

The addition, and knowing where to place it is rather hard. Any tips on getting started?

share improve this question

Pants
1 • 2

asked
Nov 18 '15 at 0:24

Raphael ♦
47.5k • 16 • 115 • 250

edited
Nov 18 '15 at 12:11

1 What have you tried? Where did you get stuck? What approaches have you considered? Is there any simplified version of this that you *are* able to solve? Just solving your exercise for you is probably of limited value. – **D.W. ♦** Nov 18 '15 at 2:18

add a comment

1 Answer

order by votes

1

Here's a pseudo-algorithm for a possible solution. I assume you know how to implement any of the sub-tasks (as they are very common and usually given as basic examples)

1. Add a $ to the right of the input.
2. Duplicate the content of the tape $\$x\#y \to \$x\#y\$x\#y$
3. Add a 0 at the very start $\to 0\$x\#y\$x\#y$
4. Loop until the first x is all-zero string:
   4.1 Subtract 1 from the first x, and add 1 to the first field, moving the input if necessary.
   4.2 remove the all-zero field that is now where x was, and repeat step 4 with y.
   [Note: a better algorithm will skip 4.1/4.2 for x, and immediately start with subtracting from y, where the counter z starts from x rather than 0, but this is equivalent.]
5. Clean the $ and # as necessary, move the head to the first blank, and halt.

So the only non-trivial step is how to add 1 or subtract 1 from a given number x given in a binary base 2 form. I assume you know how to perform binary arithmetic: for instance, for binary addition you add 1 to the LSB; if it was 0 you are done; if it was 1 you move on to the bit next to it ("carry") until you find a 0 or until you have passed the MSB (then you assume the digit after the MSB should be 0, move the input to accommodate that zero and add 1 to it). It is a bit of a hassle to convert the algorithm into the form of a TM, but this is basically what you are requested to do. It is possible, since TM can compute any (well-defined) algorithm that a CPU-computer can do. I'll leave you to filling up the details.

share improve this answer

Ran G.
15k • 2 • 38 • 82

answered
Nov 18 '15 at 0:40

edited
Nov 18 '15 at 3:03

Your Answer

Body

Add picture

Log in

OR

Name

Email

By posting your answer, you agree to the privacy policy and terms of service.

Post Your Answer