

Can all RPN expressions be represented such that all operators appear on the left and all operands appear on the right?

Asked 16 years, 3 months ago Modified 12 years ago Viewed 536 times



I've convinced myself that they can't.

1

Take for example:



4 4 + 4 /

stack: 4 stack: 4 4 4 + 4 = 8 stack: 8 stack: 8 4 8 / 4 = 2
stack: 2



There are two ways that you could write the above expression with the same operators and operands such that the operands all come first: "4 4 4 + /" and "4 4 4 / +", neither of which evaluate to 2.

"4 4 4 + /" stack: 4 stack: 4 4 stack: 4 4 4 4 + 4 = 8 stack: 4 8 4 / 8 = 0.5 stack: 0.5

"4 4 4 / +" stack: 4 stack: 4 4 stack: 4 4 4 4 / 4 = 1 stack: 4 1 4 + 1 = 5 stack: 5

If you have the ability to swap items on the stack then yes, it's possible, otherwise, no.

Thoughts?

rpn

Share

Improve this question

Follow

edited Sep 2, 2008 at 9:03



jfs

16.7k ● 13 ● 63 ● 90

asked Sep 2, 2008 at 9:01



kanske

13 ● 1 ● 2

4 Answers

Sorted by:

Highest score (default)



Consider the algebraic expression:

2

$(a + b) * (c + d)$



The obvious translation to RPN would be:



$a b + c d + *$



Even with a swap operation available, I don't think there is a way to collect all the operators on the right:

$a b c d +$

$a b S$

where S is the sum of c and d . At this point, you couldn't use a single swap operation to get both a and b in place for a $+$ operation. Instead, you would need a more sophisticated stack operation (such as roll) to get a and b in the right spot. I don't know whether a roll operation would be sufficient for all cases, either.

Share Improve this answer

answered Sep 2, 2008 at 9:08

Follow



[Greg Hewgill](#)

990k ● 191 ● 1.2k ● 1.3k



1

Actually, you've not only given the answer but a conclusive proof as well, by examining a counter-example which is enough to disprove the assumption implied in the title.



Share Improve this answer

answered Sep 2, 2008 at 9:06

Follow



[Konrad Rudolph](#)

545k ● 139 ● 956 ● 1.2k



1

I know this is a very old thread, but I just found it today and wanted to say that I believe the answer to the original question is YES. I am confident all RPN expressions can be represented such that all operators appear on the left and all operands appear on the right, if in addition to the normal arithmetic operations, we are allowed to include three additional 'navigational' operators in the representation.



Any arithmetic expression can be represented as a binary tree, with variables and constants at the leaf nodes, binary arithmetic operations at the forks in the tree, and unary operations such as negation, reciprocal, or square root along any branches. The three additional operations I suggest represent building a left branch, building a right branch, or reaching a leaf node in a binary tree. Now if we place all the operands to the left of the input string according to the position of their respective leaves in the tree, we can supply the remainder of the input string with operations telling how to reconstruct the appropriate binary tree in memory and insert the operands and mathematical operations into it at the correct points. Finally a depth-first tree-traversal algorithm is applied to calculate the result.

I don't know if this has any practical application. It's probably too inefficient as way to encode and decode expressions. But as an academic exercise, I am sure it is workable.

[Share](#) [Improve this answer](#)

answered Dec 13, 2012 at 4:27

[Follow](#)



[Wayne VanWeerthuisen](#)

11 ● 1



It is enough to show one that can't in order to tell you the answer to this.

0



If you can't reorder the stack contents, then the expression $(2+4)*(7+8)$ can't be rearranged.



$2 \cdot 4 + 7 \cdot 8 + *$



No matter how you reorder this, you'll end up with something that needs to be summed before you go on.

At least I believe so.

[Share](#) [Improve this answer](#)

answered Sep 2, 2008 at 9:05

[Follow](#)



[Lasse V. Karlsen](#)

391k ● 106 ● 646 ● 844
