Questions

Bookmark this page

String to AST 1

1/1 point (graded)

If the input string is "19+23+18", which abstract syntax tree would be produced by `buildAST` above?

- [ ] Plus(Number(19))
- [ ] Plus(19, 23, 18)
- [ ] Plus(Plus(19, 23), 18)
- [x] Plus(Plus(Number(19), Number(23)), Number(18))
- [ ] Plus(Number(19), Plus(Number(23), Number(18)))

Submit

Show Answer

Correct (1/1 point)

String to AST 2

1/1 point (graded)

Which of the following input strings would produce:

```
Plus(Plus(Number(1), Number(2)),
     Plus(Number(3), Number(4)))
```

- [x] "(1+2)+(3+4)"
- [ ] "1+2+3+4"
- [ ] "(1+2)+3+4"
- [x] "(((1+2)))+(3+4)"

Explanation

"(1+2)+(3+4)" directly corresponds to the AST value's structure.
"1+2+3+4" would be Plus(Plus(Plus(Number(1), Plus(2)), Number(3)), Number(4)) as explained in the previous answer.
"(1+2)+3+4" first walks over the sum 1+2 which recursively produces Plus(Number(1),Number(2)). It then creates a new Plus with that result and Number(3) resulting in Plus(Plus(Number(1),Number(2)), Number(3)); it then continues adding Number(4) to the expression resulting in the same expression as the previous choice Plus(Plus(Plus(Number(1), Plus(2)), Number(3)), Number(4)).
"(((1+2)))+(3+4)" corresponds to the desired AST in structure, just with extra parentheses. Parentheses aren't represented in the AST, so the extras have no effect on the AST value.

Submit

Show Answer

Answers are displayed within the problem

Previous Next