Compilers – Fall 2022

HW2: Syntax-directed translation scheme Total: 5 points + 1 for bonus

[What you need to know, Hints or Course Supplemental before doing this HW2]

We have covered in the classroom, when a grammar, especially in prefix or postfix order to be translateed into infix order. We need to think about its “precedence”

Grammar:

P → + P P

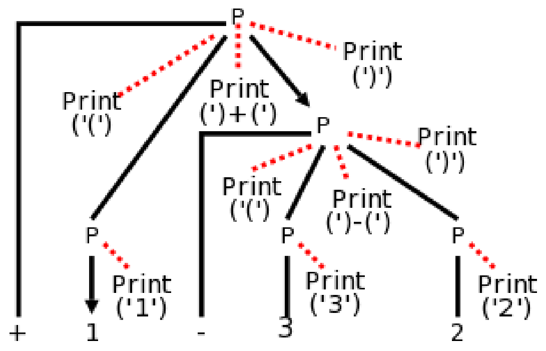
| - P P

| 1 | 2 | 3

For example, the input string +1-32 is in prefix but when we translate that into the infix, we need to think about its precedence. By following the way called “symantic actions” we introduced in the classroom. Remember the red-dotted lines we introduced in the classroom? For this +1-32, the table and the parsing tree are given as follows.

Production with Semantic Action	Semantic Rule
P → + { print('(') } P_1 { print(')+(') } P_2 { print(')') }	P.t := '(' || P_1.t || ')+(' || P.t || ')'
P → - { print('(') } P_1 { print(')-(') } P_2 { print(')') }	P.t := '(' || P_1.t || ')-(' || P.t || ')'
P → 1 { print('1') }	P.t := '1'
P → 2 { print('2') }	P.t := '2'
P → 3 { print('3') }	P.t := '3'

Prefix to infix translator



So, in this way, we can have our input string. If we only trace red-dotted actions, +1-32 can be translated into: (1)+((3)-(2))

See? We only need to deal with red-dotted actions, left to right. This is where our "print()" goes. I mean, the red-dotted actions

Q1. Given an input string in this format, it can be either 9-5+2, or 9-5*2, in the infix form, construct a syntax-directed translation scheme that translates arithmetic expressions from infix notation into prefix notation in which an operator appears before its operands

The productions are given. As you can see, it can support +, -, * and /

```
expr -> expr + term
      | expr - term
      | term
term -> term * factor
      | term / factor
      | factor
factor -> digit | (expr)
```

Yur job is to print() something, according to the table we had shown in the previous page. But the locations of the print() could be different (because of different "fix"). We only need something similar to the LHS of the table in the previous page.

For 8 productions here, you need to write out the semantic action for 8 productions (5%), and draw the red-dotted actions on the parsing tree for bonus credit (1%)