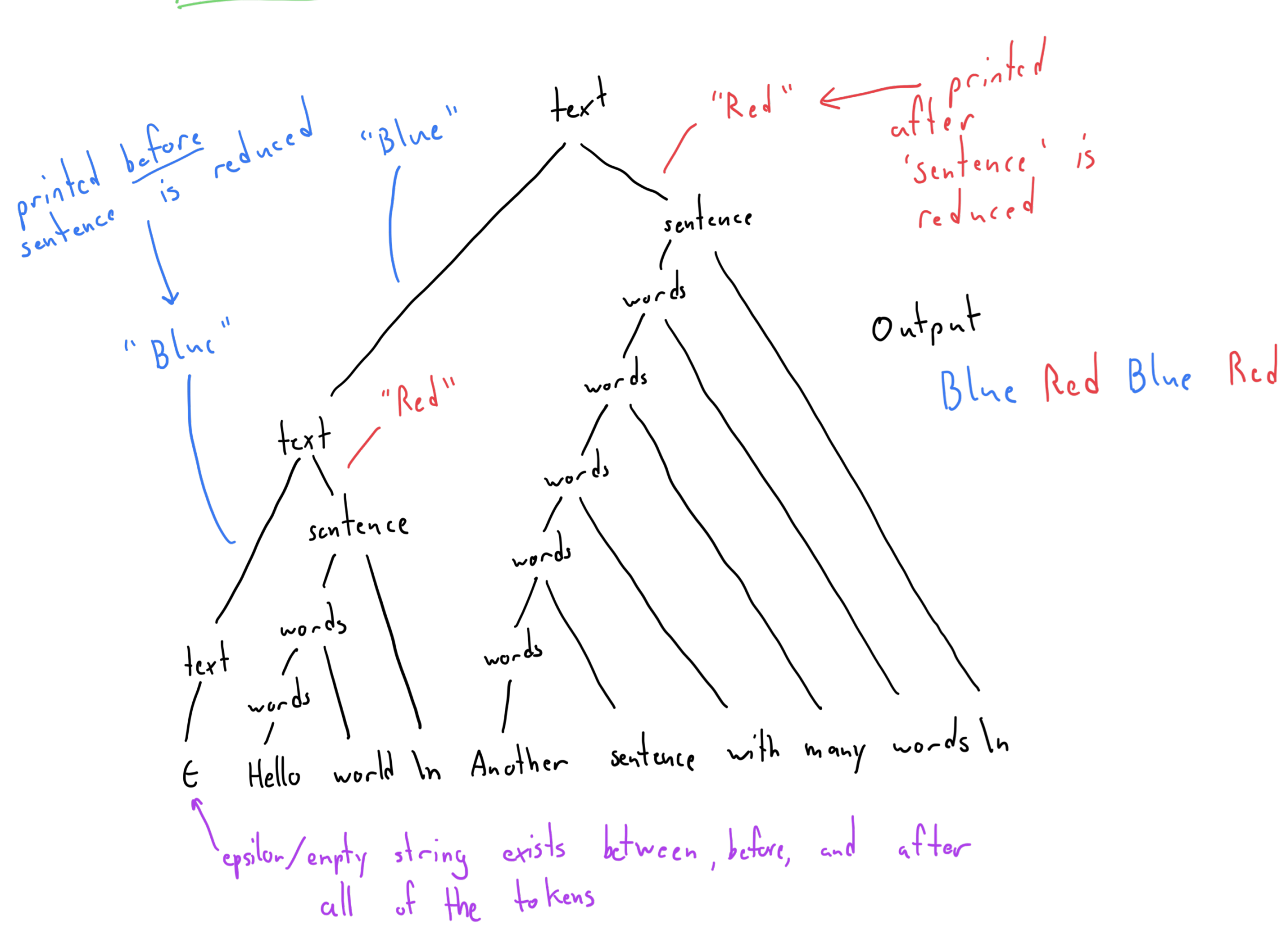


```

text : /* epsilon */
      | text {count << "Blue";} sentence {count << "Red";}

sentence : EOL
          | words EOL

words : WORD
       | words WORD
  
```



Forcing Precedence

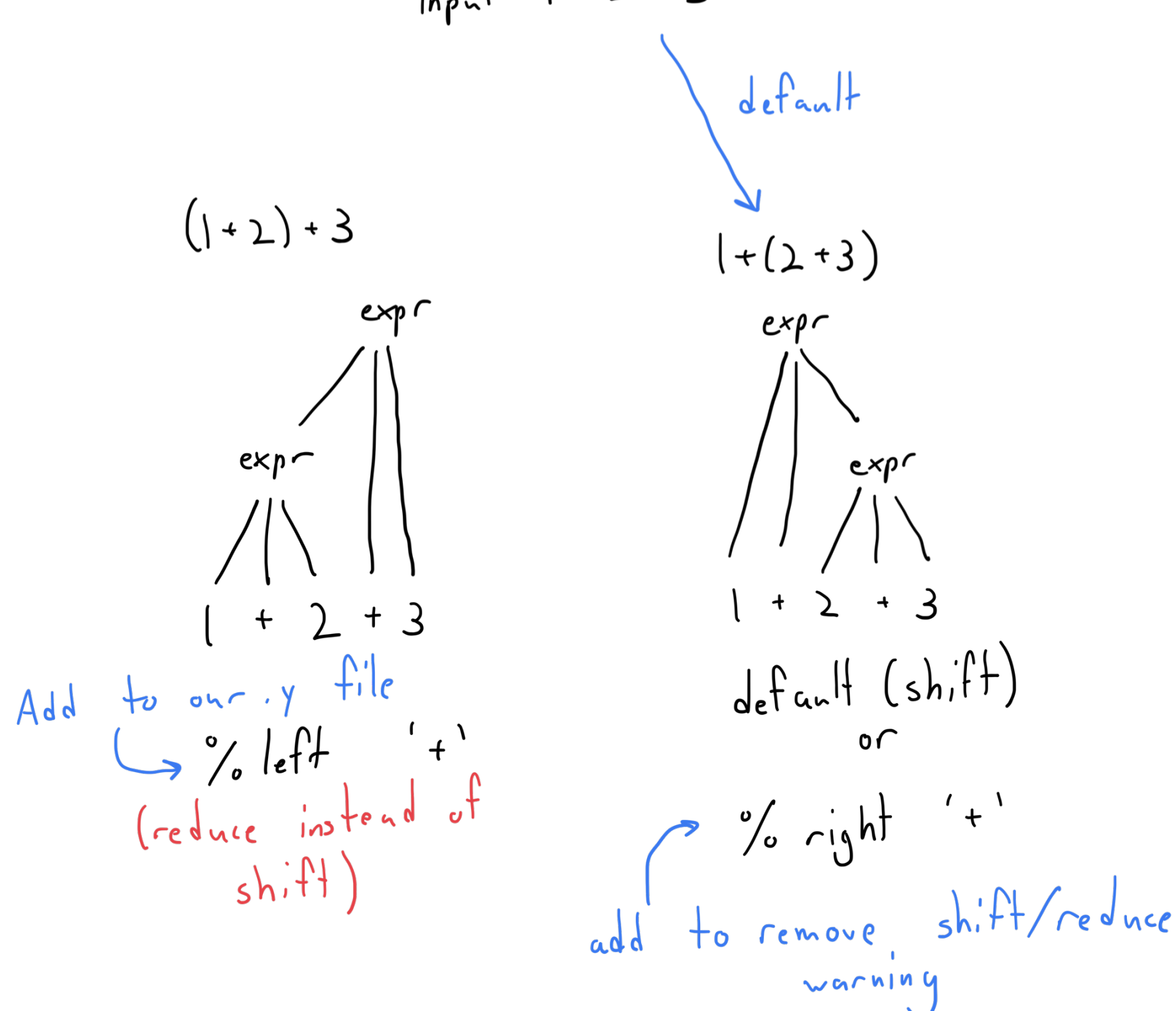
Suppose we have a grammar:

```

expr :
      | expr + expr
      | NUM ← Flex: [0-9]
  
```

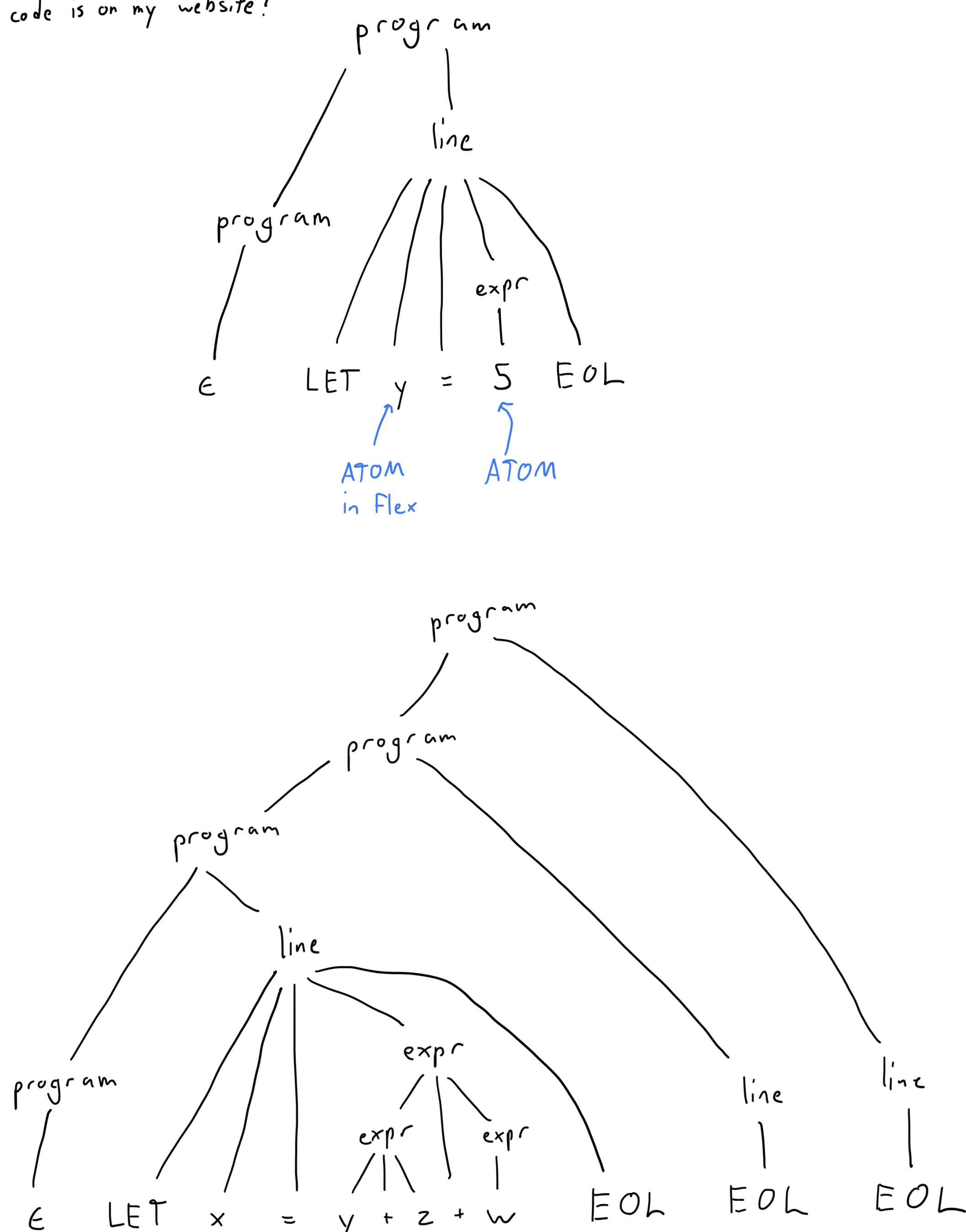
This is ambiguous!

input: 1 + 2 + 3



basic.l / basic.y

• code is on my website!

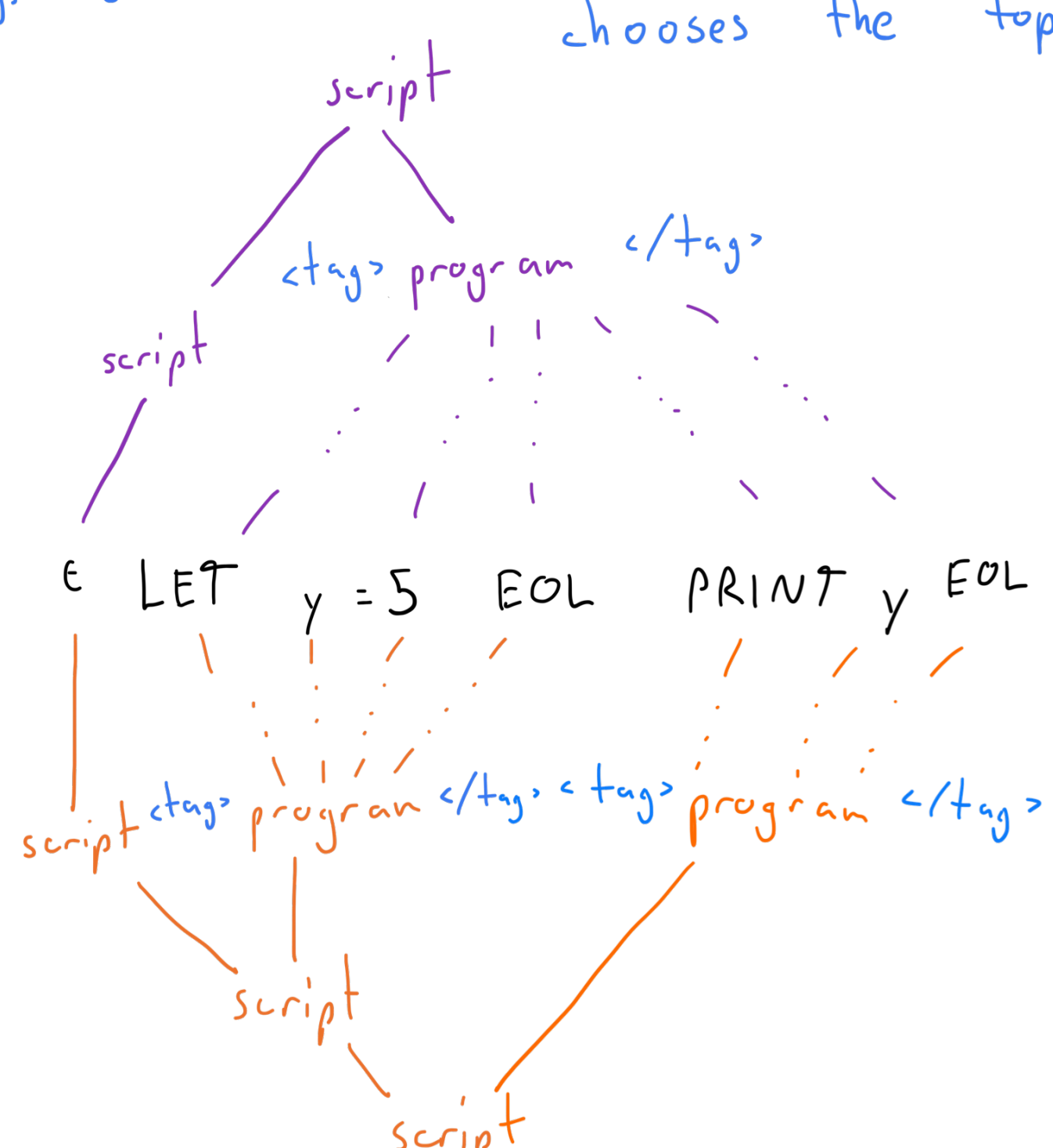


Add a new rule to .y

```

script : /* empty */
        | script {printf("<tag>");} program {printf("</tag>");};
  
```

The grammar has shift/reduce conflicts now!
The <tags> show 2 valid parse trees (but Bison chooses the top one)



Modified BASIC to Javascript transpiler

- code is also on canvas
- added HTML support
- High-level parse tree

