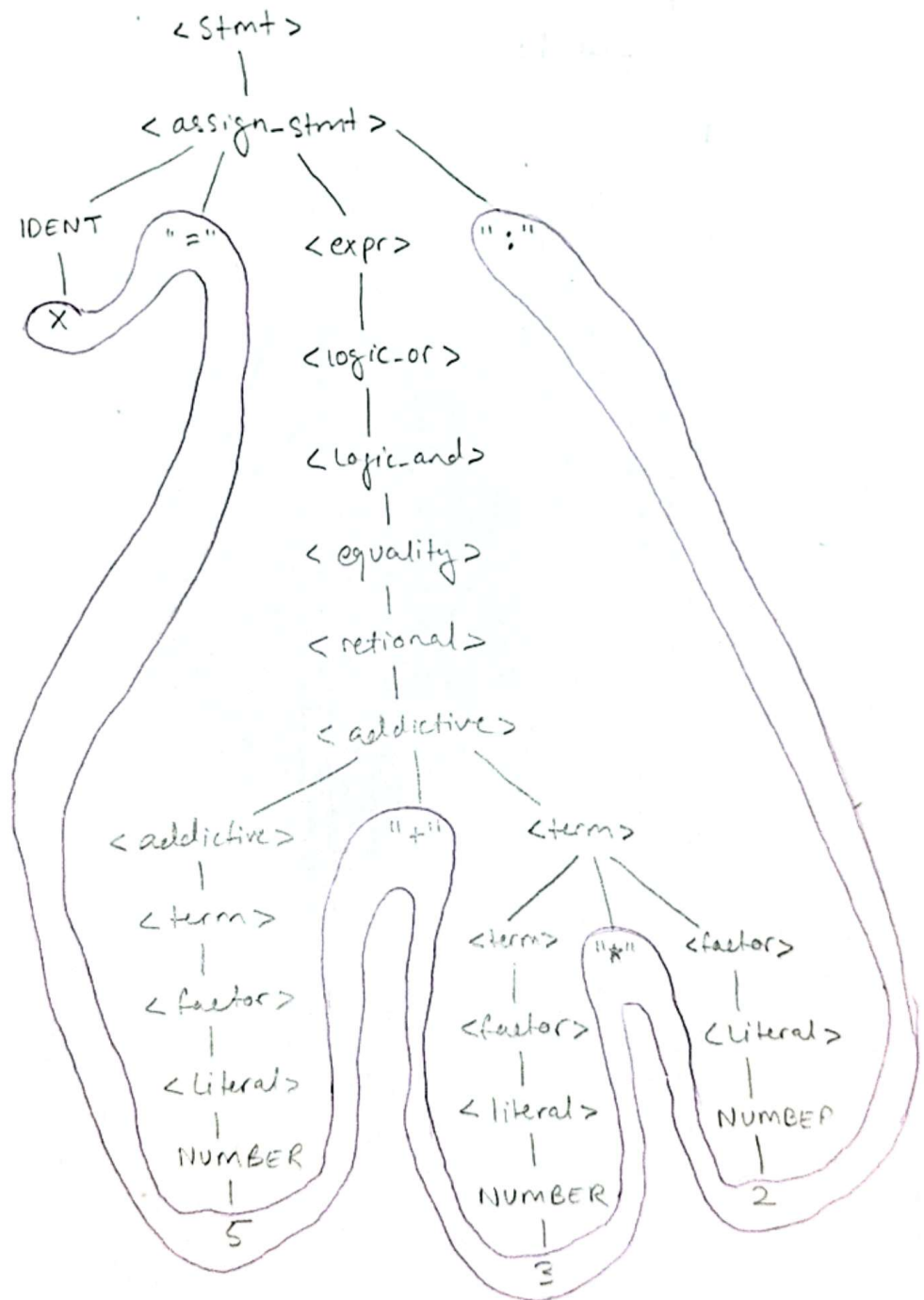Rightmost derivation

Tree # 1 : The Math & precedence Tree

purpose : To show that precedence is correct.

Input :- x = 5 + 3 * 2 ;   (3*2 will happen first).

<stmt>
|
<assign-stmt>

IDENT   " = "   <expr>   " ; "
|
X   <logic-or>
                              |
                          <logic-and>
                              |
                          <equality>
                              |
                          <retional>
                              |
                          <addictive>

<addictive>   " + "   <term>
|
<term>   <term>   "*"   <factor>
| | |
<factor>   <factor>   <literal>
| | |
<literal>   <literal>   NUMBER
| | |
NUMBER   NUMBER   2
| |
5   3

Tree #2: the loop structure explained

purpose : to show how the compiler parses complex statements.

input codes—
```
loop i in 1..9 {
    plot i:
}
```

< Stmt >
|
< Loop-stmt >

"Loop"   IDENT   "in"   <expr>   ".."   <expr>   "{"   <stmt_list>   "}"
          |
          i

<expr>
|
<term>
|
<factor>
|
NUMBER
|
1

<expr>
|
<logic_or>
|
<logic_and>
|
<equality>
|
<relational>
|
<addictive>
|
<term>
|
<factor>
|
NUMBER
|
9

<stmt_list>
|
<stmt>
|
< io_stmt >

"plot"   <expr>   ":"
          |
       <logic_or>
          |
       <logic_and>
          |
       <equality>
          |
       <relational>
          |
       <addictive>
          |
       <term>
          |
       <factor>
          |
       IDENT
          |
        "i"