

Q & A

PROJECT

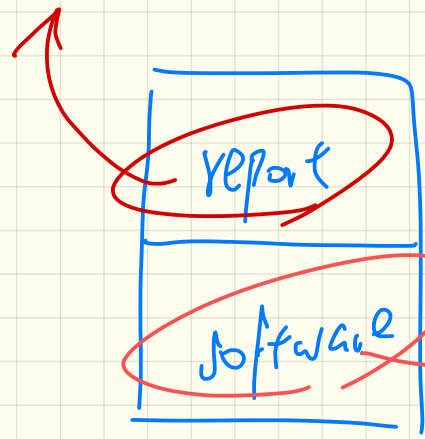
FRIDAY NOVEMBER 29

add\_Command ("C", "cmd",  $\leftarrow$  ["x", "A"],  
["y", "B"],  
["y", "C"],  
["x", "D"])

- nested loop not necessary  $y \rightarrow x$
- find out all the unique names in list

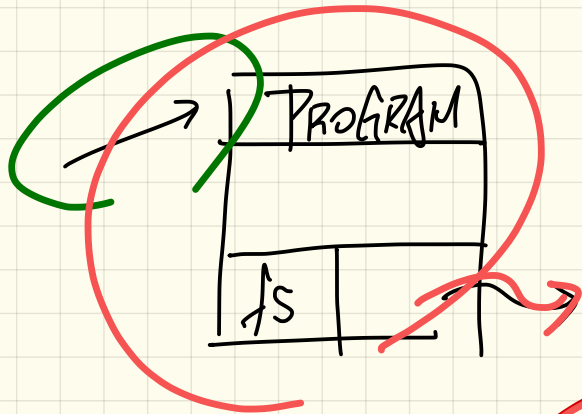
Grammar  
programs  
expressions

java code ←  
type checking ←  
] atol.txt  
oracle.

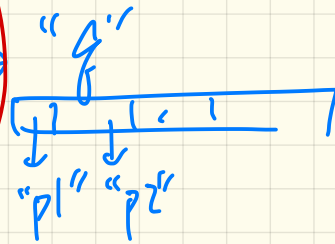
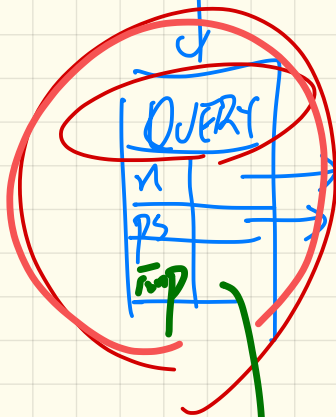


atol.txt  
your own test  
oracle -b atol.txt →  
atol.express.txt

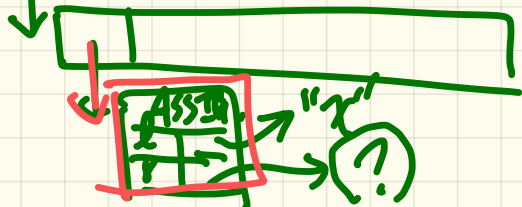
design/structure  
does not  
matter.



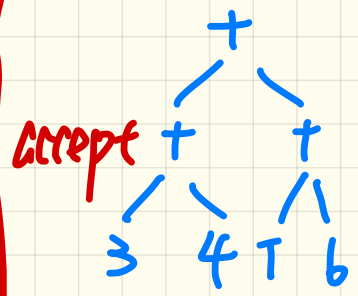
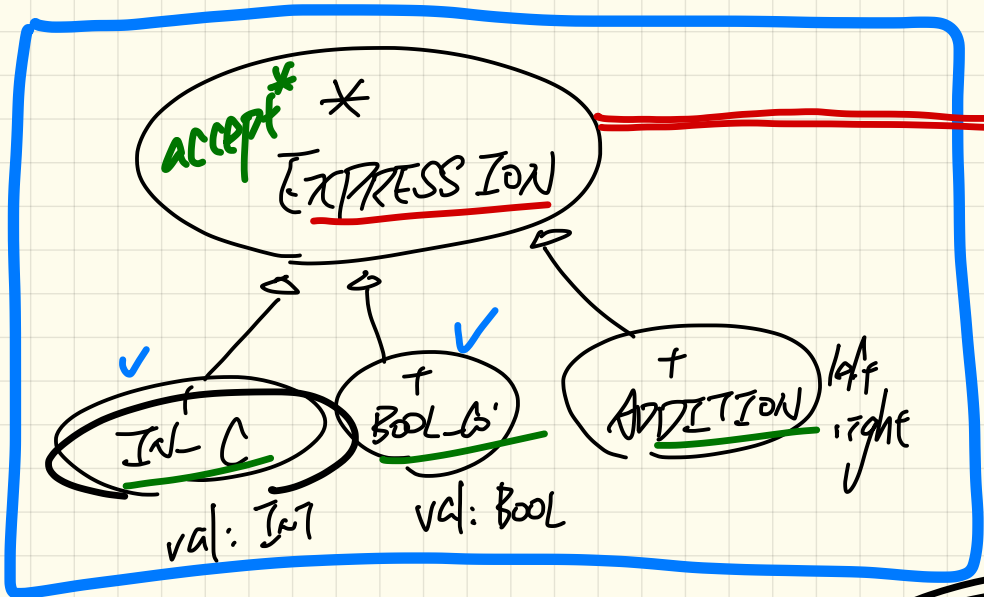
add-assgn. - ms( → → →  
→



\* feature  
↳ attribute  
↳ command  
↳ query

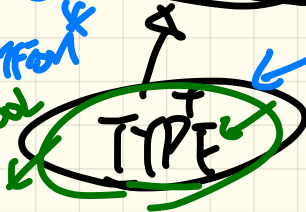


$(3+4) * (tmp+6)$

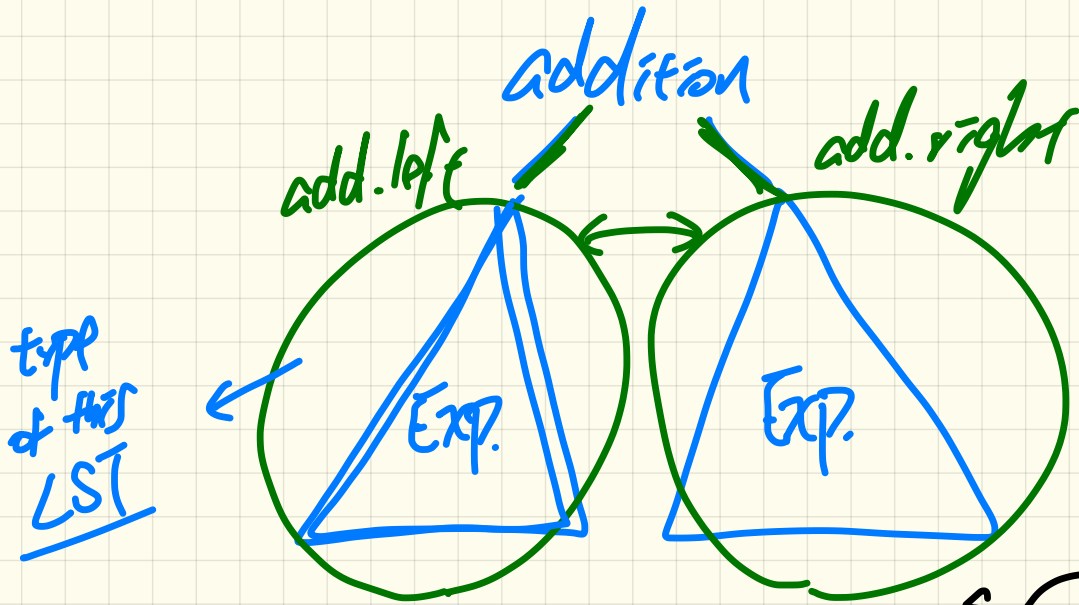


$\rightarrow 3 \checkmark$   
 $tmp \checkmark$   
 $3 + 4 \checkmark$   
 $tmp + 6 \checkmark$

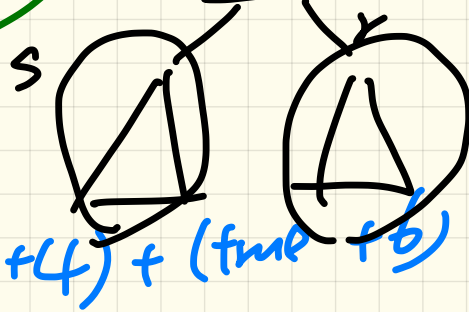
$\rightarrow visit\_int\_C^*$   
 $\rightarrow visit\_bool\_C^*$   
 $\rightarrow visit\_addition^*$   
 $result: bool$



$visit\_int\_C$   
 $do$   
 $result := T$   
 $ad.$



S + " + " v  
 ↓  
 Addition



visit\_addition (add: ADDITION)

local  
 t\_l, t\_r: TYPE  
 do  
 and

# call chain

add-call-chain  
( << "pl" >> )

```

class A {
  B b;
  C c;
  void cmd ( B pl, C pz ) {
    c := pl.a.pz
  }
}

```

1. Current class
2. parameters
3. Result.

```

class B {
  A a;
  C c;
}

```

```

class C {
  A a;
  B b;
}

```

c := pl.a.pz ~~X~~

b := pl B ✓  
 c := pl.a.c  
 B A C

c := pl.a A X

add\_assignment\_instruction ("l", "end",  
"x")