# Type and Scope checking

We have partially implemented type and scope checking. Scope checking is fully functional, but type checking is only partially functional.

Run the scope/type checker with the QC3 class. It’s output is very verbose at this time and let’s the examiner see which parts have been implemented.

Here is example output on the provided test files. We can see saving and retrieving from the symbol table, correct scoping, and some type checking on assignments.

Test01

Function declaration found. Checking symbol table...

Function has not been declared before. Saving and continuing happily

Saving param

Saving param

j : inti : intintBegan new scope

return i + j>>Incorrect typein Plus expression

;

Exited current scope

Function declaration found. Checking symbol table...

Function has not been declared before. Saving and continuing happily

Saving param

Saving param

j : floati : floatfloatBegan new scope

return i + j>>Incorrect typein Plus expression

;

Exited current scope

Began new scope

Saving new variable s1 to current scope in symbol table as ast.VarDeclNode@67a9b034

s1 : int = Function sum called. Checking if exists in scope

>>Variable/Function sum has not been declared

-1020 + 5>>Incorrect typein Plus expression

Saving new variable s2 to current scope in symbol table as ast.VarDeclNode@356f5b17

s2 : float = Function sum2 called. Checking if exists in scope

>>Variable/Function sum2 has not been declared

10.0-20.0Saving new variable b to current scope in symbol table as ast.VarDeclNode@21c55e69

b : bool = if (s1 < s2 || s1term is ast.AccessorNode@24b950d1 and val is ast.LessThNode@268dc2d

isBool queried. className is AccessorNode

>>Incorrect typein OR expression

== s2){

s1 + s2 / (s1 + s2>>Incorrect typein Plus expression

)>>Incorrect typein DIVIDE expression

>>Incorrect typein Plus expression

>= 30Var for assignment was found

;

} else {

}

Exited current scope

Test02

Saving new variable l1 to current scope in symbol table as ast.VarDeclNode@73a80183

l1 : list = [abcde]>>Incorrect typein VALUE2 of THREEEXPRNODE expression

Saving new variable l2 to current scope in symbol table as ast.VarDeclNode@3aae43b8

l2 : list = [12345]>>Incorrect typein VALUE2 of THREEEXPRNODE expression

Saving new variable s1 to current scope in symbol table as ast.VarDeclNode@6e00321

s1 : string = Saving new variable s2 to current scope in symbol table as ast.VarDeclNode@5ced6f0d

s2 : string = helloBegan new scope

Saving new variable newlist to current scope in symbol table as ast.VarDeclNode@6815ee24

newlist : list = l1::l2;

>>Incorrect typeYou must concat 2 lists of the same type! on line

Saving new variable anotherlist to current scope in symbol table as ast.VarDeclNode@54624a40

anotherlist : list = [s1]>>Incorrect typein VALUE2 of THREEEXPRNODE expression

::[s2]>>Incorrect typein VALUE2 of THREEEXPRNODE expression

;

::newlist;

>>Incorrect typein VALUE of CONCAT expression

Saving new variable thirdlist to current scope in symbol table as ast.VarDeclNode@51f3eab7

thirdlist : list = l2[:-2]::l2;

>>Incorrect typein VALUE of CONCAT expression

>>Incorrect typeYou must concat 2 lists of the same type! on line

[3:]>>Incorrect typein VALUE1 of ENDCOL expression

>>Incorrect typein VALUE2 of ENDCOL expression

Saving new variable b to current scope in symbol table as ast.VarDeclNode@64b6be69

b : bool = len (thirdlist) == len (l2)if (len (newlist) <= len (anotherlist)){

newlist + anotherlist>>Incorrect typein Plus expression

[0]>>Incorrect typein VALUE1 of NOCOL expression

\*\* ast.PlusNode@198f1327

Var for assignment was found

;

} else {

newlist - newlist>>Incorrect typein MINUS expression

[0]>>Incorrect typein VALUE1 of NOCOL expression

\*\* ast.MinusNode@32728d

Var for assignment was found

;

}

Exited current scope

Test03

person:age : intsurname : stringname : string;

family:children : listfather : idmother : id;

Began new scope

Saving new variable m to current scope in symbol table as ast.VarDeclNode@4e4d1abd

m : id = aaaaAAAbbBB0\_i40Saving new variable p to current scope in symbol table as ast.VarDeclNode@28169674

p : id = aaabAAAbbBB0\_i35Saving new variable c1 to current scope in symbol table as ast.VarDeclNode@62efae3b

c1 : id = aaabAAAbbBB0\_i1Saving new variable c2 to current scope in symbol table as ast.VarDeclNode@6597d63b

c2 : id = aaadAAAbbBB0\_i2Saving new variable c3 to current scope in symbol table as ast.VarDeclNode@5371ef98

c3 : id = aaaeAAAbbBB0\_i3Saving new variable f to current scope in symbol table as ast.VarDeclNode@5d290ef4

f : id = mp[c1c2]>>Incorrect typein VALUE2 of THREEEXPRNODE expression

>>Variable/Function f.children has not been declared

f.children::[c3]>>Incorrect typein VALUE2 of THREEEXPRNODE expression

;

>>Incorrect typein VALUE of CONCAT expression

>>Variable/Function Assignment error : has not been declared

;

Exited current scope

Test 04

Function declaration found. Checking symbol table...

Function has not been declared before. Saving and continuing happily

Saving param

t : tupletupleBegan new scope

Saving new variable t2 to current scope in symbol table as ast.VarDeclNode@5371ef98

t2 : tuple = []Saving new variable i to current scope in symbol table as ast.VarDeclNode@5d290ef4

i : int = 0return t2 ;

if (len (t) > 0){

repeat {i + 1>>Incorrect typein Plus expression

Var for assignment was found

;

t[i]>>Incorrect typein VALUE1 of NOCOL expression

\*\* ast.AccessorNode@198f1327

+ t2>>Incorrect typein Plus expression

Var for assignment was found

;

} until (i < len (t));} else {

}

Exited current scope

Saving new variable a to current scope in symbol table as ast.VarDeclNode@32728d

a : tuple = [123aabb]Began new scope

Saving new variable b to current scope in symbol table as ast.VarDeclNode@6ffe8714

b : tuple = Function invert called. Checking if exists in scope

has been defined. Continuing happily

aExited current scope

Test 05

Saving new variable s1 to current scope in symbol table as ast.VarDeclNode@73a80183

s1 : string = Alice in WonderlandSaving new variable s2 to current scope in symbol table as ast.VarDeclNode@1afb7ac7

s2 : string = GilgameshSaving new variable s3 to current scope in symbol table as ast.VarDeclNode@7439aca7

s3 : string = One Thousand and One NightsBegan new scope

Saving new variable key to current scope in symbol table as ast.VarDeclNode@7676438d

key : string = icSaving new variable books to current scope in symbol table as ast.VarDeclNode@4e4d1abd

books : list = [s1s2s3]>>Incorrect typein VALUE2 of THREEEXPRNODE expression

Saving new variable found to current scope in symbol table as ast.VarDeclNode@54624a40

found : bool = falseSaving new variable i to current scope in symbol table as ast.VarDeclNode@f8db08

i : int = 0Saving new variable tmp to current scope in symbol table as ast.VarDeclNode@51f3eab7

tmp : string = while (i + 1>>Incorrect typein Plus expression

Var for assignment was found

;

if (key in tmpisBool queried. className is AccessorNode

>>Incorrect typein VALUE of IN expression

isBool queried. className is AccessorNode

>>Incorrect typeString only takes characters in IN

){

trueVar for assignment was found

;

} else {

}

books[i]Var for assignment was found

;

) do {

i < len (books)}Exited current scope