

LAB 3 MIF08: a mu evaluator

November 7, 2017

Mu Evaluator

Input: a .mu file:

```
var s:float;  
s=3.14;  
log(s);
```

Output: on std output:

3.14

Code Infrastructure

```
...code/Mu-evalntype$ ls
ex-types/ Makefile  MyMuTypingVisitor.py  README.md
ex/       Main.py    Mu.g4      MyMuVisitor.py      test_evaluator.py
```

- ▶ The grammar of the Mu language is in `Mu.g4`.
- ▶ The main file (command line, driver for the lexer/parser/visitor): `Main.py`.
- ▶ Two visitors: one for typing, the other one to evaluate.
- ▶ A Makefile, a README.
- ▶ Testfiles, and a test script `test_evaluator.py`.

Mu typing, Mu visit

- ▶ A `MyMuTypingVisitor` to type Mu programs given, it rejects programs like:

```
var x:int;  
x="blablabla";
```

⇒ You only have to read the code and play with it to understand how it works.

- ▶ A `MyMuVisitor`, that executes the program. We provide you as an example the arithmetic expression evaluation (and the corresponding test `test00.mu`).
⇒ You have to complete the evaluation for assignments, tests, while.

Test infrastructure (same as in Lab 2)

You write your testcases and expected results:

```
log(3^2+45*(-2/-1));  
log(23+19);  
log(false || 3 != 7)  
log("coucou");
```

```
# EXPECTED  
# 99.00  
# 42  
# 1  
# coucou
```

```
var u:int;  
var b:bool;  
  
u=3; b=true;  
if b {  
    u=u+1;  
} else {  
    u=u-1;  
}  
log(u);  
  
# EXPECTED  
# 4
```

⇒ a helper script (using pytest) will compare the actual and the expected outputs.

Test infrastructure 2/2

```
python3 -m pytest -v test_evaluator.py
===== test session starts =====
platform linux -- Python 3.5.2, pytest-3.2.2, py-1.4.34, pluggy-0.4.0 -- /usr/bin/python3
cachedir: .cache
rootdir: /home/laure/Documents/VCS/Teaching/archico/M1_COMPIL_MIF08/2017_2018/TP/TP03/code/Mu-eval
nifile:
collected 7 items

test_evaluator.py::TestCodeGen::test_expect[ex/test00.mu] PASSED
test_evaluator.py::TestCodeGen::test_expect[ex-types/bad_type04.mu] PASSED
test_evaluator.py::TestCodeGen::test_expect[ex-types/bad_type00.mu] PASSED
test_evaluator.py::TestCodeGen::test_expect[ex-types/bad_type01.mu] PASSED
test_evaluator.py::TestCodeGen::test_expect[ex-types/double_decl00.mu] PASSED
test_evaluator.py::TestCodeGen::test_expect[ex-types/bad_type02.mu] PASSED
test_evaluator.py::TestCodeGen::test_expect[ex-types/bad_type03.mu] PASSED

===== 7 passed in 0.91 seconds =====
```

⇒ Using this test framework is mandatory.