LAB3 - CS444: an evaluator for mini C

L. Gonnord, C. Deleuze

October 12, 2022

Mu Evaluator

Input: a MiniC .c file:

```
int main() {
float s;
s=3.14;
println_float(s);
return 0;
}
```

Output: on std output:

```
3.14
```

Code Infrastructure

```
ls cs444-labs22/MiniC/
Lib       MiniCC.py  MiniCListener.py  pyproject.toml         test_expect_pragma.py  TP03
Makefile  MiniC.g4   printlib.h        README-interpreter.md  test_interpreter.py
```

- The grammar of the MiniC language is in `MiniC.g4`.
- The main file (command line, driver for the lexer/parser/visitor): `MiniCC.py`.
- In `TP03/` two visitors: one for typing, the other one to evaluate.
- A Makefile, a README.
- Testfiles, and a test script `test_interpreter.py`.

MiniC typing, MiniC visit

in `MiniC/TP03/`

- A `MiniCTypingVisitor` to type MiniC programs given, it rejects programs like:

  ```
  int x;
  x="blablabla";
  ```

  ⇒ You only have to read the code and play with it to understand how it works.
- A `MiniCInterpretVisitor`, that executes the program. We provide you as an example the arithmetic expression evaluation. ⇒ You have to complete the evaluation for assignments, tests, while.

Test infrastructure (same as in Lab 2)

You write your testcases and expected results:

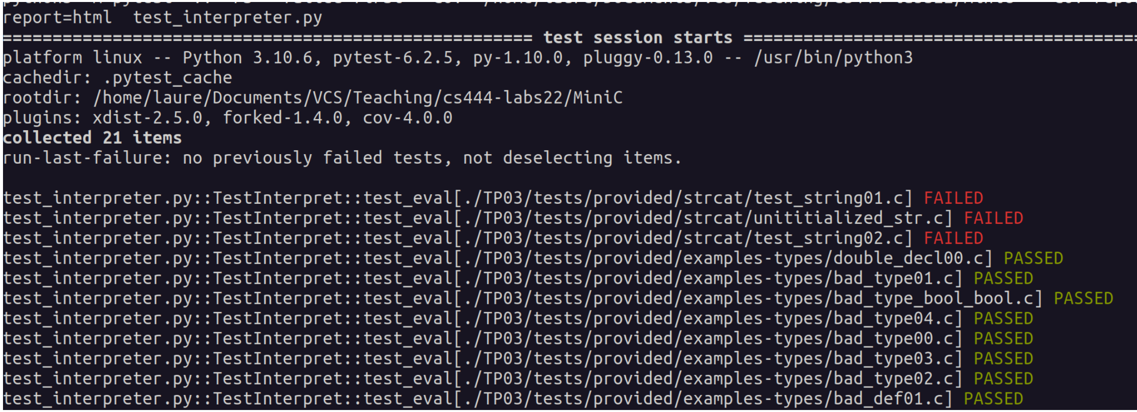
```
#include "printlib.h"
int main() {
  println_float(3^2+45*(-2/-1));
  println_int(23+19);
  println_string("coucou");
}

# EXPECTED
# 99.00
# 42
# coucou
```

```
#include "printlib.h"

int main() {
    if ((1.0 + 2.0) * 3.0 == 9.0) {
        println_string("OK");
    }
    return 0;
}

// EXPECTED
// OK
```

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

Test infrastructure 2/2

Test interpreter rule in Makefile.



```
report=html  test_interpreter.py
=========================================================== test session starts ============================================
platform linux -- Python 3.10.6, pytest-6.2.5, py-1.10.0, pluggy-0.13.0 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /home/laure/Documents/VCS/Teaching/cs444-labs22/MiniC
plugins: xdist-2.5.0, forked-1.4.0, cov-4.0.0
collected 21 items
run-last-failure: no previously failed tests, not deselecting items.

test_interpreter.py::TestInterpret::test_eval[./TP03/tests/provided/strcat/test_string01.c] FAILED
test_interpreter.py::TestInterpret::test_eval[./TP03/tests/provided/strcat/unititialized_str.c] FAILED
test_interpreter.py::TestInterpret::test_eval[./TP03/tests/provided/strcat/test_string02.c] FAILED
test_interpreter.py::TestInterpret::test_eval[./TP03/tests/provided/examples-types/double_decl00.c] PASSED
test_interpreter.py::TestInterpret::test_eval[./TP03/tests/provided/examples-types/bad_type01.c] PASSED
test_interpreter.py::TestInterpret::test_eval[./TP03/tests/provided/examples-types/bad_type_bool_bool.c] PASSED
test_interpreter.py::TestInterpret::test_eval[./TP03/tests/provided/examples-types/bad_type04.c] PASSED
test_interpreter.py::TestInterpret::test_eval[./TP03/tests/provided/examples-types/bad_type00.c] PASSED
test_interpreter.py::TestInterpret::test_eval[./TP03/tests/provided/examples-types/bad_type03.c] PASSED
test_interpreter.py::TestInterpret::test_eval[./TP03/tests/provided/examples-types/bad_type02.c] PASSED
test_interpreter.py::TestInterpret::test_eval[./TP03/tests/provided/examples-types/bad_def01.c] PASSED
```

$\Rightarrow$ Using this test framework is mandatory.