Stratification module for datalog programs

CONTRERAS Baptiste & CLEMENT Florent

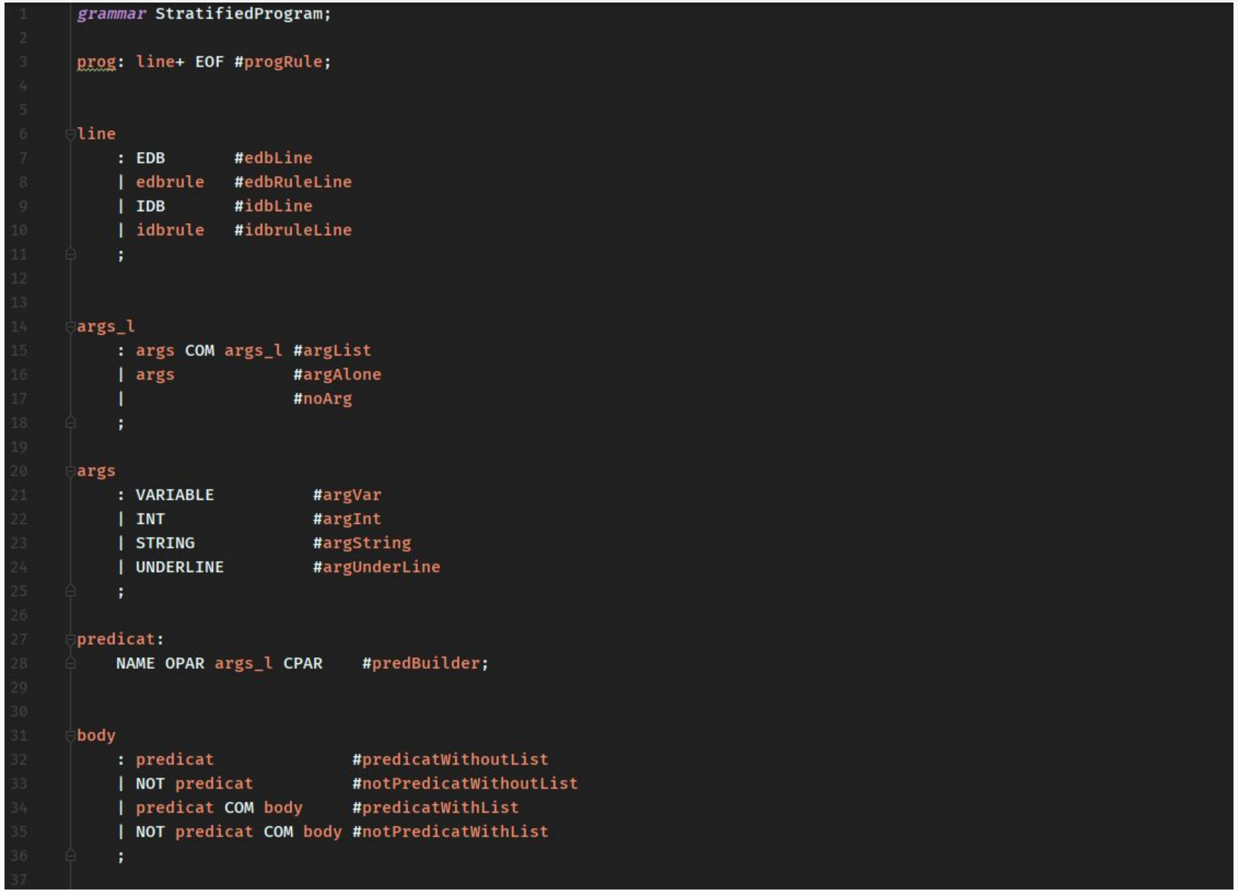
Table of contents

Details of our implementation

Antlr4

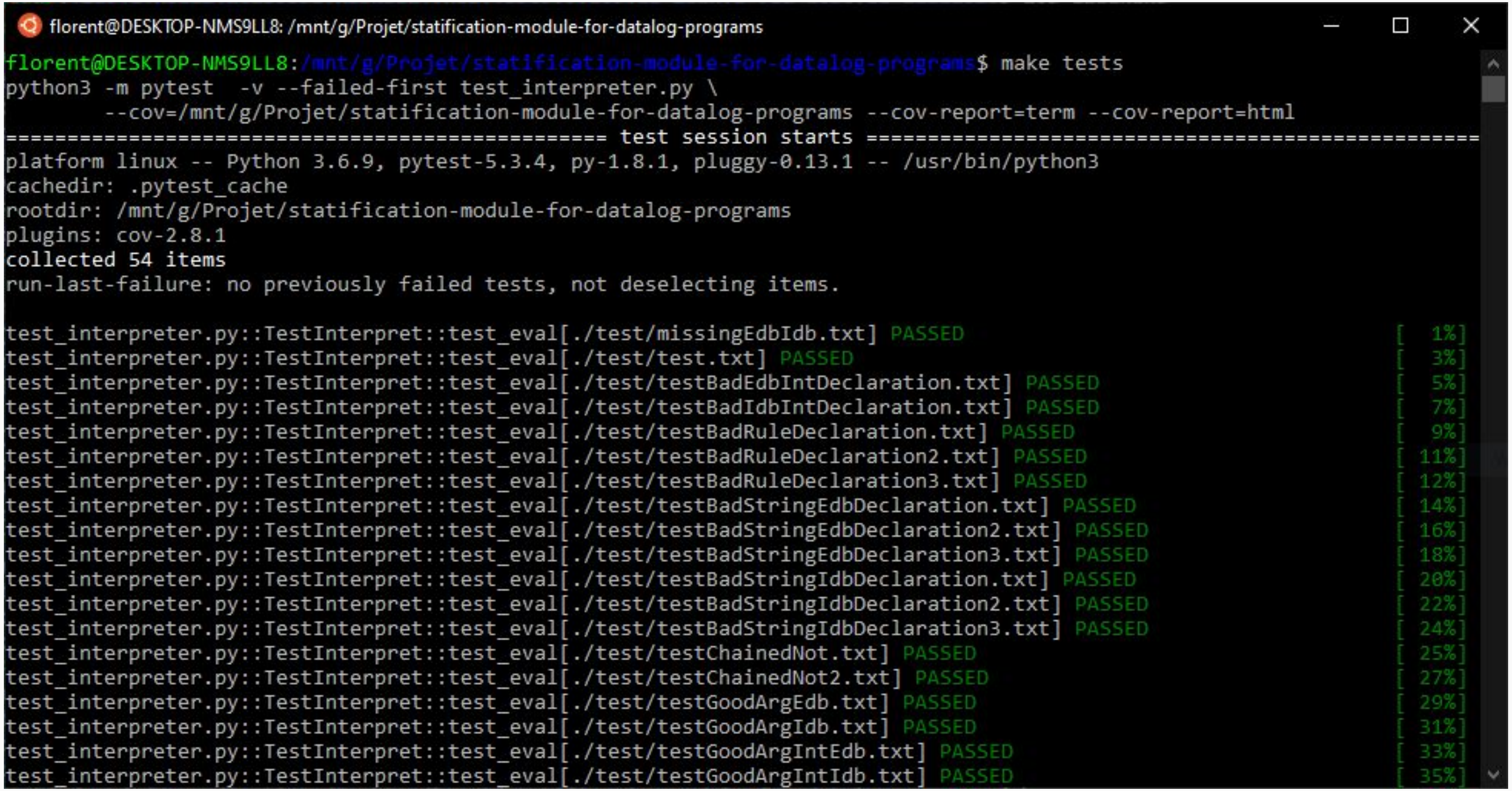
Python 3

Details of our implementation



```
grammar StratifiedProgram;

prog: line+ EOF #progRule;


line
    : EDB       #edbLine
    | edbrule   #edbRuleLine
    | IDB       #idbLine
    | idbrule   #idbruleLine
    ;


args_l
    : args COM args_l #argList
    | args            #argAlone
    |                 #noArg
    ;

args
    : VARIABLE          #argVar
    | INT               #argInt
    | STRING            #argString
    | UNDERLINE         #argUnderLine
    ;

predicat:
    NAME OPAR args_l CPAR    #predBuilder;


body
    : predicat              #predicatWithoutList
    | NOT predicat          #notPredicatWithoutList
    | predicat COM body     #predicatWithList
    | NOT predicat COM body #notPredicatWithList
    ;
```

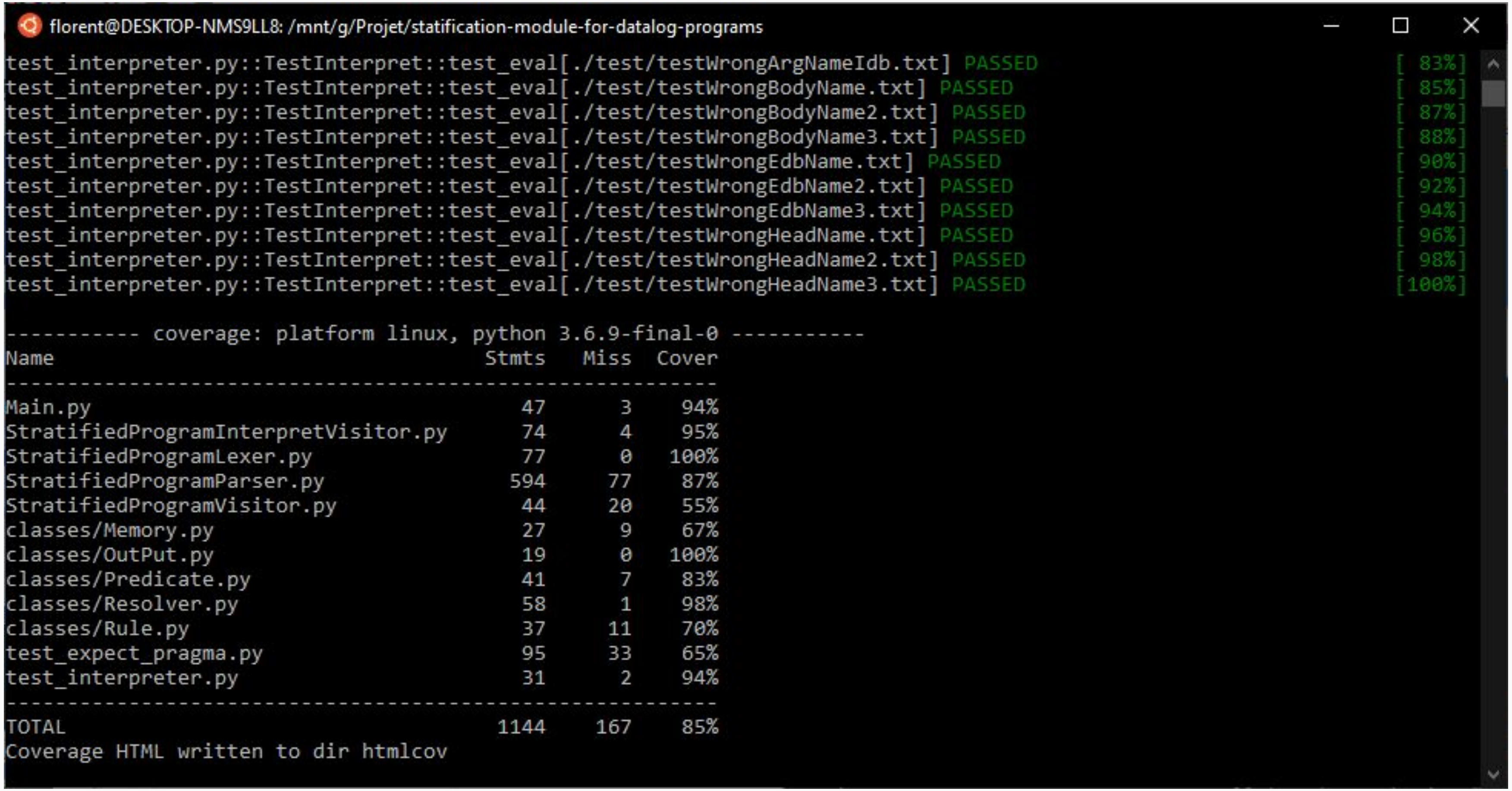
Details of our implementation



florent@DESKTOP-NMS9LL8: /mnt/g/Projet/statification-module-for-datalog-programs

```
florent@DESKTOP-NMS9LL8:/mnt/g/Projet/statification-module-for-datalog-programs$ make tests
python3 -m pytest  -v --failed-first test_interpreter.py \
        --cov=/mnt/g/Projet/statification-module-for-datalog-programs --cov-report=term --cov-report=html
================================================= test session starts ==================================================
platform linux -- Python 3.6.9, pytest-5.3.4, py-1.8.1, pluggy-0.13.1 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /mnt/g/Projet/statification-module-for-datalog-programs
plugins: cov-2.8.1
collected 54 items
run-last-failure: no previously failed tests, not deselecting items.

test_interpreter.py::TestInterpret::test_eval[./test/missingEdbIdb.txt] PASSED                                    [  1%]
test_interpreter.py::TestInterpret::test_eval[./test/test.txt] PASSED                                             [  3%]
test_interpreter.py::TestInterpret::test_eval[./test/testBadEdbIntDeclaration.txt] PASSED                         [  5%]
test_interpreter.py::TestInterpret::test_eval[./test/testBadIdbIntDeclaration.txt] PASSED                         [  7%]
test_interpreter.py::TestInterpret::test_eval[./test/testBadRuleDeclaration.txt] PASSED                           [  9%]
test_interpreter.py::TestInterpret::test_eval[./test/testBadRuleDeclaration2.txt] PASSED                          [ 11%]
test_interpreter.py::TestInterpret::test_eval[./test/testBadRuleDeclaration3.txt] PASSED                          [ 12%]
test_interpreter.py::TestInterpret::test_eval[./test/testBadStringEdbDeclaration.txt] PASSED                      [ 14%]
test_interpreter.py::TestInterpret::test_eval[./test/testBadStringEdbDeclaration2.txt] PASSED                     [ 16%]
test_interpreter.py::TestInterpret::test_eval[./test/testBadStringEdbDeclaration3.txt] PASSED                     [ 18%]
test_interpreter.py::TestInterpret::test_eval[./test/testBadStringIdbDeclaration.txt] PASSED                      [ 20%]
test_interpreter.py::TestInterpret::test_eval[./test/testBadStringIdbDeclaration2.txt] PASSED                     [ 22%]
test_interpreter.py::TestInterpret::test_eval[./test/testBadStringIdbDeclaration3.txt] PASSED                     [ 24%]
test_interpreter.py::TestInterpret::test_eval[./test/testChainedNot.txt] PASSED                                   [ 25%]
test_interpreter.py::TestInterpret::test_eval[./test/testChainedNot2.txt] PASSED                                  [ 27%]
test_interpreter.py::TestInterpret::test_eval[./test/testGoodArgEdb.txt] PASSED                                   [ 29%]
test_interpreter.py::TestInterpret::test_eval[./test/testGoodArgIdb.txt] PASSED                                   [ 31%]
test_interpreter.py::TestInterpret::test_eval[./test/testGoodArgIntEdb.txt] PASSED                                [ 33%]
test_interpreter.py::TestInterpret::test_eval[./test/testGoodArgIntIdb.txt] PASSED                                [ 35%]
```

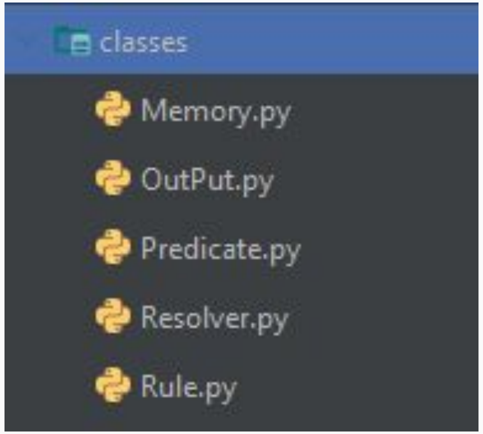
Details of our implementation



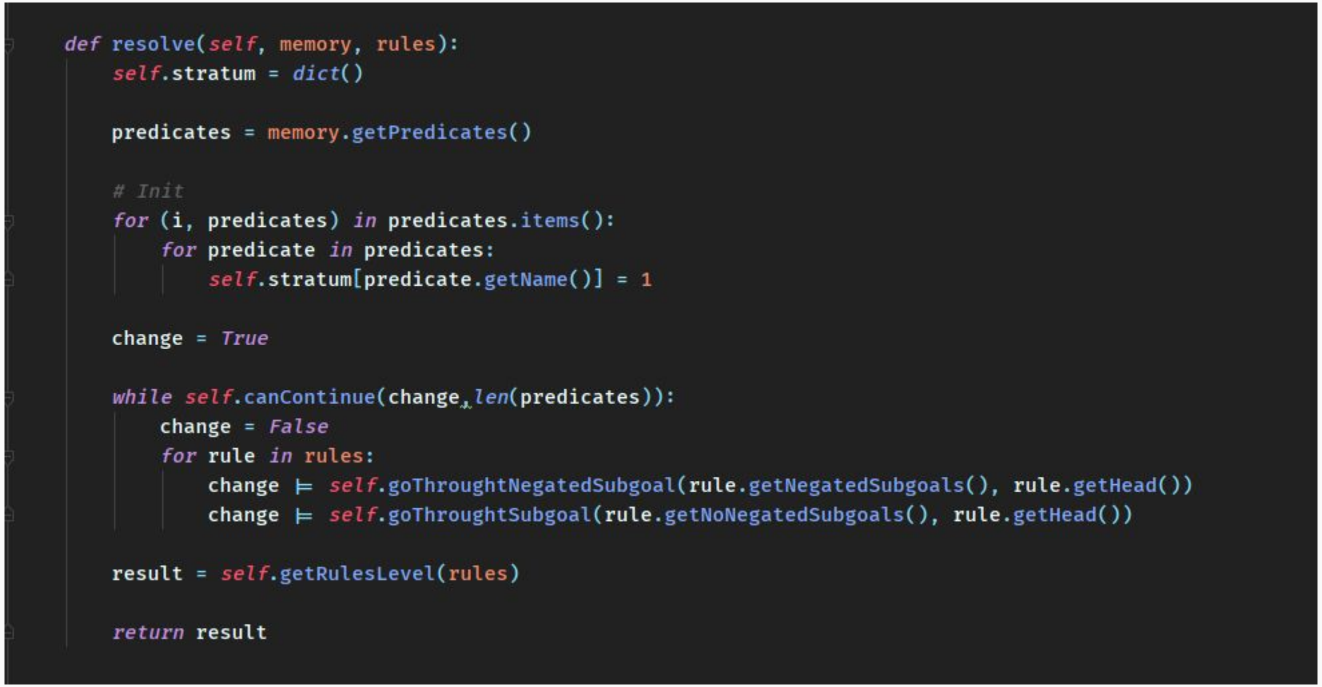
florent@DESKTOP-NMS9LL8: /mnt/g/Projet/statification-module-for-datalog-programs

```
test_interpreter.py::TestInterpret::test_eval[./test/testWrongArgNameIdb.txt] PASSED                 [ 83%]
test_interpreter.py::TestInterpret::test_eval[./test/testWrongBodyName.txt] PASSED                   [ 85%]
test_interpreter.py::TestInterpret::test_eval[./test/testWrongBodyName2.txt] PASSED                  [ 87%]
test_interpreter.py::TestInterpret::test_eval[./test/testWrongBodyName3.txt] PASSED                  [ 88%]
test_interpreter.py::TestInterpret::test_eval[./test/testWrongEdbName.txt] PASSED                    [ 90%]
test_interpreter.py::TestInterpret::test_eval[./test/testWrongEdbName2.txt] PASSED                   [ 92%]
test_interpreter.py::TestInterpret::test_eval[./test/testWrongEdbName3.txt] PASSED                   [ 94%]
test_interpreter.py::TestInterpret::test_eval[./test/testWrongHeadName.txt] PASSED                   [ 96%]
test_interpreter.py::TestInterpret::test_eval[./test/testWrongHeadName2.txt] PASSED                  [ 98%]
test_interpreter.py::TestInterpret::test_eval[./test/testWrongHeadName3.txt] PASSED                  [100%]

----------- coverage: platform linux, python 3.6.9-final-0 -----------
Name                                       Stmts   Miss  Cover
----------------------------------------------------------
Main.py                                       47      3    94%
StratifiedProgramInterpretVisitor.py          74      4    95%
StratifiedProgramLexer.py                     77      0   100%
StratifiedProgramParser.py                   594     77    87%
StratifiedProgramVisitor.py                   44     20    55%
classes/Memory.py                             27      9    67%
classes/OutPut.py                             19      0   100%
classes/Predicate.py                          41      7    83%
classes/Resolver.py                           58      1    98%
classes/Rule.py                               37     11    70%
test_expect_pragma.py                         95     33    65%
test_interpreter.py                           31      2    94%
----------------------------------------------------------
TOTAL                                       1144    167    85%
Coverage HTML written to dir htmlcov
```

Details of our implementation

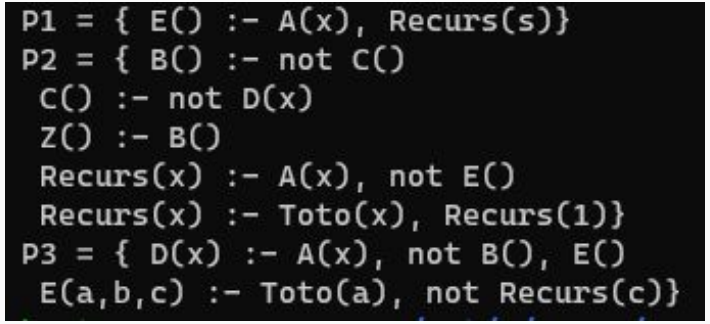


```
classes
    Memory.py
    OutPut.py
    Predicate.py
    Resolver.py
    Rule.py
```

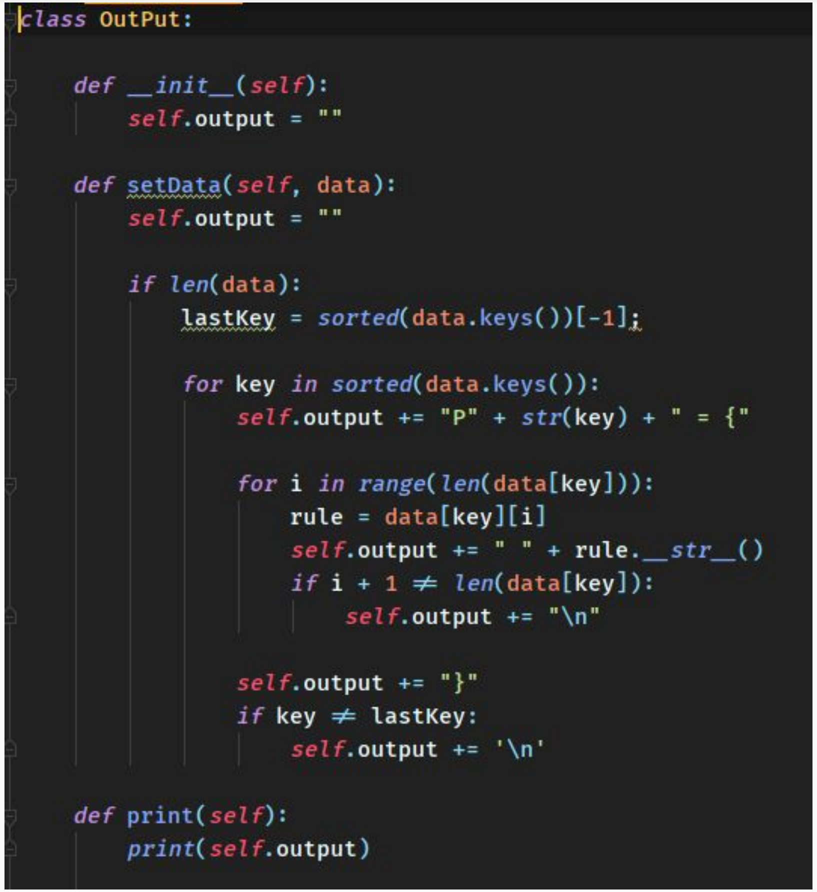


```python
def resolve(self, memory, rules):
    self.stratum = dict()

    predicates = memory.getPredicates()

    # Init
    for (i, predicates) in predicates.items():
        for predicate in predicates:
            self.stratum[predicate.getName()] = 1

    change = True

    while self.canContinue(change,len(predicates)):
        change = False
        for rule in rules:
            change |= self.goThroughtNegatedSubgoal(rule.getNegatedSubgoals(), rule.getHead())
            change |= self.goThroughtSubgoal(rule.getNoNegatedSubgoals(), rule.getHead())

    result = self.getRulesLevel(rules)

    return result
```

Details of our implementation



```
P1 = { E() :- A(x), Recurs(s)}
P2 = { B() :- not C()
 C() :- not D(x)
 Z() :- B()
 Recurs(x) :- A(x), not E()
 Recurs(x) :- Toto(x), Recurs(1)}
P3 = { D(x) :- A(x), not B(), E()
 E(a,b,c) :- Toto(a), not Recurs(c)}
```

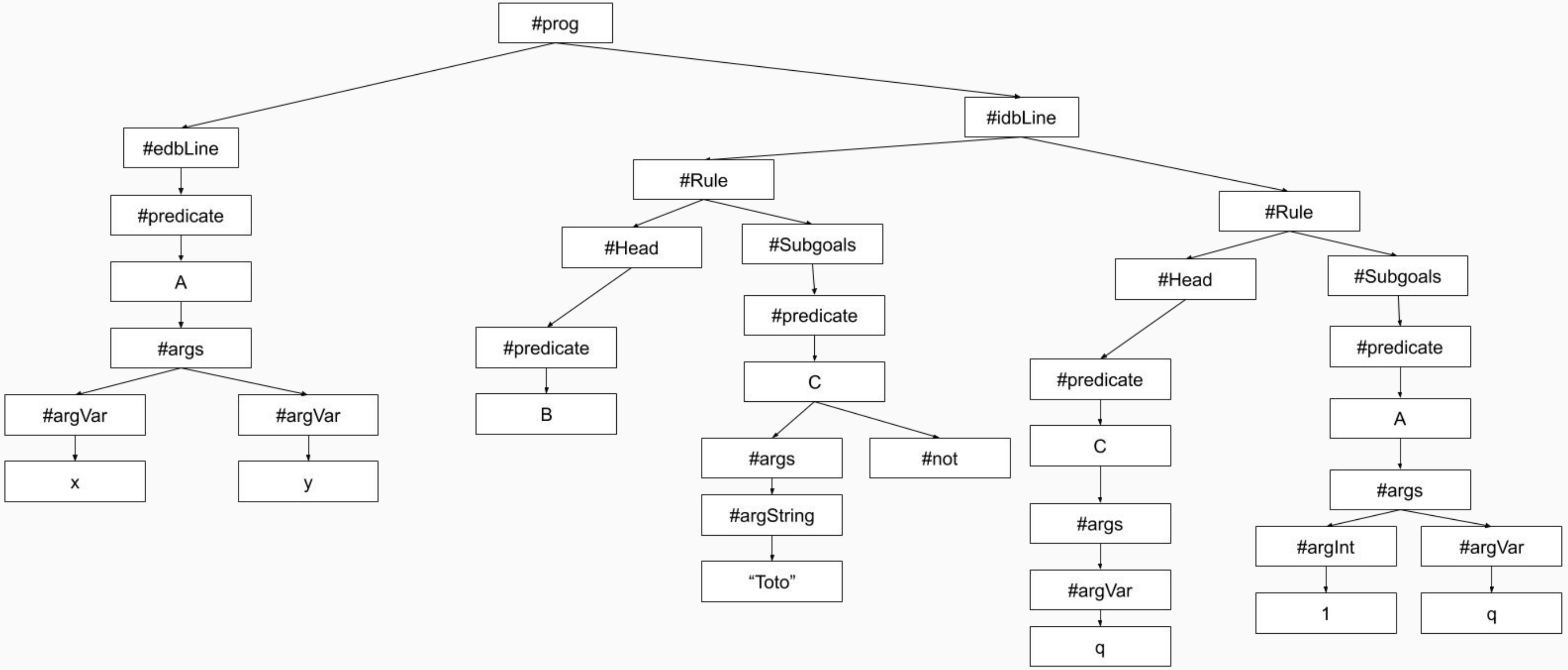


```python
class OutPut:

    def __init__(self):
        self.output = ""

    def setData(self, data):
        self.output = ""

        if len(data):
            lastKey = sorted(data.keys())[-1];

            for key in sorted(data.keys()):
                self.output += "P" + str(key) + " = {"

                for i in range(len(data[key])):
                    rule = data[key][i]
                    self.output += " " + rule.__str__()
                    if i + 1 ≠ len(data[key]):
                        self.output += "\n"

                self.output += "}"
                if key ≠ lastKey:
                    self.output += '\n'

    def print(self):
        print(self.output)
```

Execution step by step of an example

```
%edb
A(x, y).


%idb
B() :- not C("Toto").
C(q) :- A(1, q).
```

Execution step by step of an example



Execution step by step of an example

Memory object	Rules list
• A (x) • B() • C(q)	• B() :- not C("Toto") • C(q) :- A(1)

Execution step by step of an example

```
P1 = { C(q) :- A(1,q)}
P2 = { B() :- not
C("Toto")}
```

Possible evolutions

- Change the syntax
- Add more validation
- Modify display

Known limits of our program

```
%edb
A(x).
B(y).

%idb
C(x,y) :- A(x), B(y).
Recur(x,y) :- A(x), B(y).
Recur(x,y) :- C(x, z), not Recur(x,y).
```

To conclude

- Works well
- Flexible
- New display format