LSINF2335 - Programming Paradigms

Romain Capron & Antoine Marchal

April 28, 2014

Python

UCL
Université
catholique
de Louvain

Contents

1 Write a Quine

Write a quine in a programming language of your choice. Can you write one that makes use of reflection ?

To write a quine that makes use of reflection, we will divide our program in two parts :

1. a string definition
2. the program core

The string contains the code of the program core and the program core will print twice this string. Once to print the string, and again to print the program core. The output of this program must its source code.

1.1 Python

This is the quine I wrote :

```
1 _='_=%s; print _%%`_`'; print _%`_`
```

To explain it, I modified the _ symbols by variables a, b, c and d and printed the output. This allows to understand directly that it has the same principle as a Matryoshka doll. The main doll is the source code that can be divided in two parts. In the program core and we can find inside (by printing) a smaller doll ; the string. And there is again a fourth doll in this string. The output of the program is a doll that has the same appearance than the original doll.

```
1 a='b=%s; print c%%`d`'; print a%`a` #source code
2 b='b=%s; print c%%`d`'; print c%`d` #output
```

It is reflective because :

1. the program is divided in two parts; string & core
2. the core will print the string.
3. the string is also divided in two parts; string & core and printed, it will produce a result that equals the source code. It is just like the program was able to print itself.

2 Chosen Language

• What language have you chosen (and why)?
We have chosen Python that is a ... programming language. • What kind of programming paradigm does this language belong to (functional, procedural, logic, object-oriented, multi-paradigm, . . .)?
• Give a brief introduction to the core syntax / semantics / concepts of that language.
• Give an illustrative working code example of a typical program written in that language.

```
a, b = (1, 1)
while b < 10:
    print 'a={0}, b={1} and a+b={2}'.format(a, b, (a+b))
    a, b = (b, a + b)
```

We can run this program by executing the following command :

```
$ python example.py
```

And this is its output :

```
a=1, b=1 and a+b=2
a=1, b=2 and a+b=3
a=2, b=3 and a+b=5
a=3, b=5 and a+b=8
a=5, b=8 and a+b=13
```

• What kind of typical applications is the language targeted at?

3 Reflection and meta-programming

3.1 Reflective features

• What language features for dealing with reflection and meta-programming does the chosen language provide?
• What kinds of reflection and meta-programming features does that language offer?
• What is the MOP (meta-object protocol) for that language?
• What are the limitations of the reflective features provided by this language?
• Illustrate your explanations with working code fragments.

3.2 Applications of reflection

• What are the typical applications that reflection could be used for in this language?
• Can you give a working code example of such a typical problem that requires a reflective solution?
• Does there exist a "killer-app" for this language that has been implemented with reflection?

3.3 Comparison with other languages

• How does this language compare to Smalltalk, Java or Ruby from the point of view of the reflective features it supports, the kinds of reflection it offers, or its MOP?
• What can this language learn from those languages?
• Does it offer some specific reflective features that you do not have in either Smalltalk, Ruby or Java? (Can Smalltalk/Java learn something from reflection in this language?)
• Does it offer some powerful native (non-reflective) features that allow you to express things for which you would need reflection in other languages (like Smalltalk, Ruby or Java)?

and meta-programming

4 Conclusion

In conclusion, how good does this language score as a reflective language?
o Does it provide a very rich, well-structured and well-supported set of reflective features that are supported by the programming environment as well?
o Are there only a few ad-hoc reflective features that are not well supported by the environment?
o What can other (reflective) languages learn from this language?
o What can this language learn from how reflection is dealt with in other languages?