Walter Cazzola

Home Page
ADAPT Lab.
Curriculum Vitae
Research Topic

Didactics

Publications

Funded Projects

Research Projects

Related Events

Exam of Advance in Programming

31 January 2013

Disclaimer. Note that to have a running solution for an exercise is not enough: you need a well-cooked solution that proves your ability to use what explained during the classes. The worth of the 2 exercises is the same (15 points). To pass the exam you have to do both exercises. The submissions with only one exercise will not be evaluated at all.

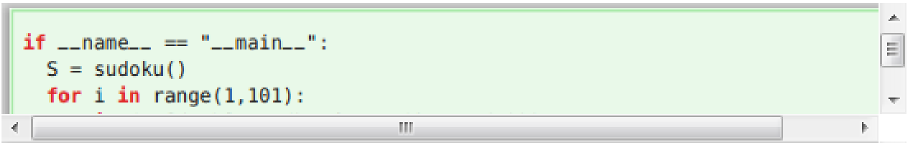
Exercise 1: Sudoku's Generator.

Sudoku is a logic-based, combinatorial number-placement puzzle. The objective is to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 sub-grids that compose the grid contains all of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which typically has a unique solution. There are 6,670,903,752,021,072,936,960 possible sudokus. By reducing the grid size to a 4×4 and considering only figures from 1 to 4 you get a simplified version of the sudoku with only 4 sub-grids 2×2 that admits much less solutions.

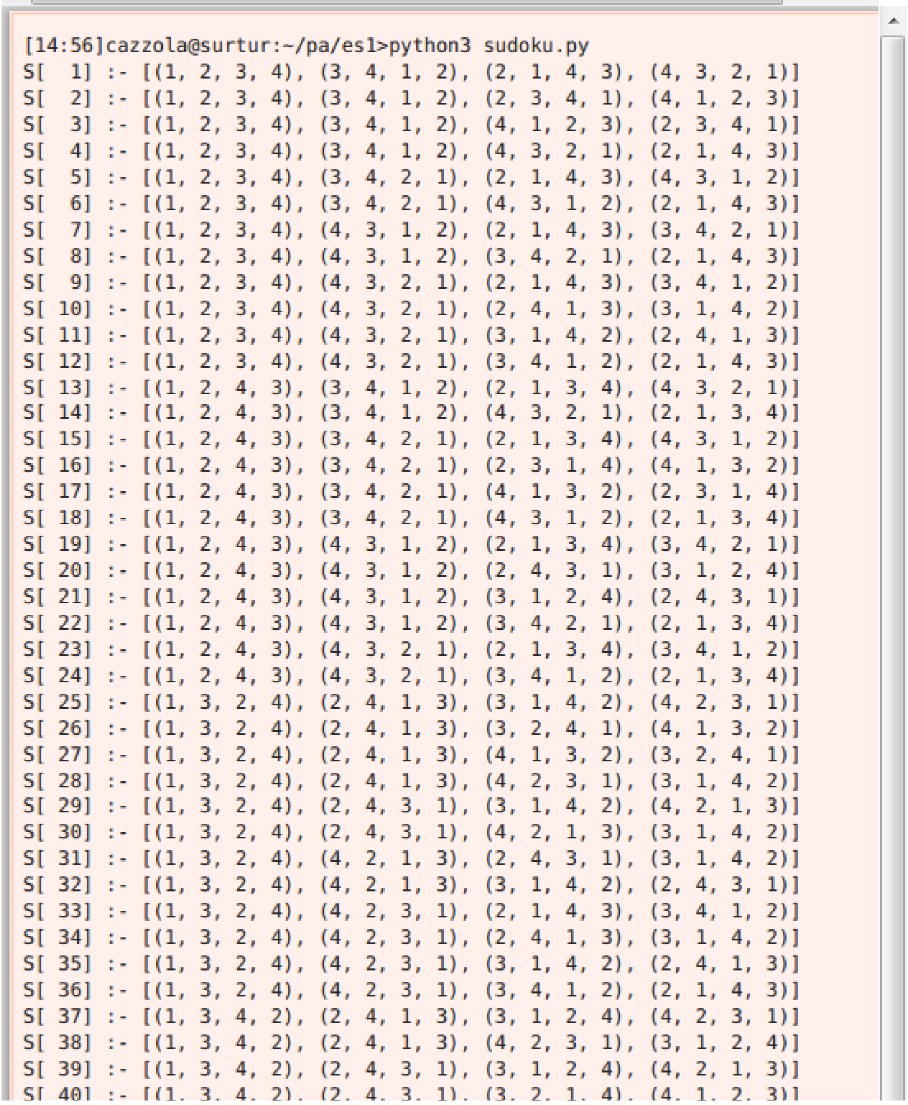
The exercise consists of writing a generator function (`sudoku`) that at each call returns a **valid** (and different) sudoku solutions for the simplified version of the problem. **Note** that not all combinations are valid solutions. The returned combination should be a list of lists where each entry will represent a row of the grid.

Note that the solutions which calculate the combinations via a nested loops are considered **wrong**.

The following is an example of the expected computation.



```
if __name__ == "__main__":
  S = sudoku()
  for i in range(1,101):
```



```
[14:56]cazzola@surtur:~/pa/es1>python3 sudoku.py
S[  1] :- [(1, 2, 3, 4), (3, 4, 1, 2), (2, 1, 4, 3), (4, 3, 2, 1)]
S[  2] :- [(1, 2, 3, 4), (3, 4, 1, 2), (2, 3, 4, 1), (4, 1, 2, 3)]
S[  3] :- [(1, 2, 3, 4), (3, 4, 1, 2), (4, 1, 2, 3), (2, 3, 4, 1)]
S[  4] :- [(1, 2, 3, 4), (3, 4, 1, 2), (4, 3, 2, 1), (2, 1, 4, 3)]
S[  5] :- [(1, 2, 3, 4), (3, 4, 2, 1), (2, 1, 4, 3), (4, 3, 1, 2)]
S[  6] :- [(1, 2, 3, 4), (3, 4, 2, 1), (4, 3, 1, 2), (2, 1, 4, 3)]
S[  7] :- [(1, 2, 3, 4), (4, 3, 1, 2), (2, 1, 4, 3), (3, 4, 2, 1)]
S[  8] :- [(1, 2, 3, 4), (4, 3, 1, 2), (3, 4, 2, 1), (2, 1, 4, 3)]
S[  9] :- [(1, 2, 3, 4), (4, 3, 2, 1), (2, 1, 4, 3), (3, 4, 1, 2)]
S[ 10] :- [(1, 2, 3, 4), (4, 3, 2, 1), (2, 4, 1, 3), (3, 1, 4, 2)]
S[ 11] :- [(1, 2, 3, 4), (4, 3, 2, 1), (3, 1, 4, 2), (2, 4, 1, 3)]
S[ 12] :- [(1, 2, 3, 4), (4, 3, 2, 1), (3, 4, 1, 2), (2, 1, 4, 3)]
S[ 13] :- [(1, 2, 4, 3), (3, 4, 1, 2), (2, 1, 3, 4), (4, 3, 2, 1)]
S[ 14] :- [(1, 2, 4, 3), (3, 4, 1, 2), (4, 3, 2, 1), (2, 1, 3, 4)]
S[ 15] :- [(1, 2, 4, 3), (3, 4, 2, 1), (2, 1, 3, 4), (4, 3, 1, 2)]
S[ 16] :- [(1, 2, 4, 3), (3, 4, 2, 1), (2, 3, 1, 4), (4, 1, 3, 2)]
S[ 17] :- [(1, 2, 4, 3), (3, 4, 2, 1), (4, 1, 3, 2), (2, 3, 1, 4)]
S[ 18] :- [(1, 2, 4, 3), (3, 4, 2, 1), (4, 3, 1, 2), (2, 1, 3, 4)]
S[ 19] :- [(1, 2, 4, 3), (4, 3, 1, 2), (2, 1, 3, 4), (3, 4, 2, 1)]
S[ 20] :- [(1, 2, 4, 3), (4, 3, 1, 2), (2, 4, 3, 1), (3, 1, 2, 4)]
S[ 21] :- [(1, 2, 4, 3), (4, 3, 1, 2), (3, 1, 2, 4), (2, 4, 3, 1)]
S[ 22] :- [(1, 2, 4, 3), (4, 3, 1, 2), (3, 4, 2, 1), (2, 1, 3, 4)]
S[ 23] :- [(1, 2, 4, 3), (4, 3, 2, 1), (2, 1, 3, 4), (3, 4, 1, 2)]
S[ 24] :- [(1, 2, 4, 3), (4, 3, 2, 1), (3, 4, 1, 2), (2, 1, 3, 4)]
S[ 25] :- [(1, 3, 2, 4), (2, 4, 1, 3), (3, 1, 4, 2), (4, 2, 3, 1)]
S[ 26] :- [(1, 3, 2, 4), (2, 4, 1, 3), (3, 2, 4, 1), (4, 1, 3, 2)]
S[ 27] :- [(1, 3, 2, 4), (2, 4, 1, 3), (4, 1, 3, 2), (3, 2, 4, 1)]
S[ 28] :- [(1, 3, 2, 4), (2, 4, 1, 3), (4, 2, 3, 1), (3, 1, 4, 2)]
S[ 29] :- [(1, 3, 2, 4), (2, 4, 3, 1), (3, 1, 4, 2), (4, 2, 1, 3)]
S[ 30] :- [(1, 3, 2, 4), (2, 4, 3, 1), (4, 2, 1, 3), (3, 1, 4, 2)]
S[ 31] :- [(1, 3, 2, 4), (4, 2, 1, 3), (2, 4, 3, 1), (3, 1, 4, 2)]
S[ 32] :- [(1, 3, 2, 4), (4, 2, 1, 3), (3, 1, 4, 2), (2, 4, 3, 1)]
S[ 33] :- [(1, 3, 2, 4), (4, 2, 3, 1), (2, 1, 4, 3), (3, 4, 1, 2)]
S[ 34] :- [(1, 3, 2, 4), (4, 2, 3, 1), (2, 4, 1, 3), (3, 1, 4, 2)]
S[ 35] :- [(1, 3, 2, 4), (4, 2, 3, 1), (3, 1, 4, 2), (2, 4, 1, 3)]
S[ 36] :- [(1, 3, 2, 4), (4, 2, 3, 1), (3, 4, 1, 2), (2, 1, 4, 3)]
S[ 37] :- [(1, 3, 4, 2), (2, 4, 1, 3), (3, 1, 2, 4), (4, 2, 3, 1)]
S[ 38] :- [(1, 3, 4, 2), (2, 4, 1, 3), (4, 2, 3, 1), (3, 1, 2, 4)]
S[ 39] :- [(1, 3, 4, 2), (2, 4, 3, 1), (3, 1, 2, 4), (4, 2, 1, 3)]
S[ 40] :- [(1, 3, 4, 2), (2, 4, 3, 1), (3, 2, 1, 4), (4, 1, 2, 3)]
```

```
S[ 41] :- [(1, 3, 4, 2), (2, 4, 3, 1), (4, 1, 2, 3), (3, 2, 1, 4)]
S[ 42] :- [(1, 3, 4, 2), (2, 4, 3, 1), (4, 2, 1, 3), (3, 1, 2, 4)]
S[ 43] :- [(1, 3, 4, 2), (4, 2, 1, 3), (2, 1, 3, 4), (3, 4, 2, 1)]
S[ 44] :- [(1, 3, 4, 2), (4, 2, 1, 3), (2, 4, 3, 1), (3, 1, 2, 4)]
S[ 45] :- [(1, 3, 4, 2), (4, 2, 1, 3), (3, 1, 2, 4), (2, 4, 3, 1)]
S[ 46] :- [(1, 3, 4, 2), (4, 2, 1, 3), (3, 4, 2, 1), (2, 1, 3, 4)]
S[ 47] :- [(1, 3, 4, 2), (4, 2, 3, 1), (2, 4, 1, 3), (3, 1, 2, 4)]
S[ 48] :- [(1, 3, 4, 2), (4, 2, 3, 1), (3, 1, 2, 4), (2, 4, 1, 3)]
S[ 49] :- [(1, 4, 2, 3), (2, 3, 1, 4), (3, 1, 4, 2), (4, 2, 3, 1)]
S[ 50] :- [(1, 4, 2, 3), (2, 3, 1, 4), (3, 2, 4, 1), (4, 1, 3, 2)]
S[ 51] :- [(1, 4, 2, 3), (2, 3, 1, 4), (4, 1, 3, 2), (3, 2, 4, 1)]
S[ 52] :- [(1, 4, 2, 3), (2, 3, 1, 4), (4, 2, 3, 1), (3, 1, 4, 2)]
S[ 53] :- [(1, 4, 2, 3), (2, 3, 4, 1), (3, 2, 1, 4), (4, 1, 3, 2)]
S[ 54] :- [(1, 4, 2, 3), (2, 3, 4, 1), (4, 1, 3, 2), (3, 2, 1, 4)]
S[ 55] :- [(1, 4, 2, 3), (3, 2, 1, 4), (2, 3, 4, 1), (4, 1, 3, 2)]
S[ 56] :- [(1, 4, 2, 3), (3, 2, 1, 4), (4, 1, 3, 2), (2, 3, 4, 1)]
S[ 57] :- [(1, 4, 2, 3), (3, 2, 4, 1), (2, 1, 3, 4), (4, 3, 1, 2)]
S[ 58] :- [(1, 4, 2, 3), (3, 2, 4, 1), (2, 3, 1, 4), (4, 1, 3, 2)]
S[ 59] :- [(1, 4, 2, 3), (3, 2, 4, 1), (4, 1, 3, 2), (2, 3, 1, 4)]
S[ 60] :- [(1, 4, 2, 3), (3, 2, 4, 1), (4, 3, 1, 2), (2, 1, 3, 4)]
S[ 61] :- [(1, 4, 3, 2), (2, 3, 1, 4), (3, 2, 4, 1), (4, 1, 2, 3)]
S[ 62] :- [(1, 4, 3, 2), (2, 3, 1, 4), (4, 1, 2, 3), (3, 2, 4, 1)]
S[ 63] :- [(1, 4, 3, 2), (2, 3, 4, 1), (3, 1, 2, 4), (4, 2, 1, 3)]
S[ 64] :- [(1, 4, 3, 2), (2, 3, 4, 1), (3, 2, 1, 4), (4, 1, 2, 3)]
S[ 65] :- [(1, 4, 3, 2), (2, 3, 4, 1), (4, 1, 2, 3), (3, 2, 1, 4)]
S[ 66] :- [(1, 4, 3, 2), (2, 3, 4, 1), (4, 2, 1, 3), (3, 1, 2, 4)]
S[ 67] :- [(1, 4, 3, 2), (3, 2, 1, 4), (2, 1, 4, 3), (4, 3, 2, 1)]
S[ 68] :- [(1, 4, 3, 2), (3, 2, 1, 4), (2, 3, 4, 1), (4, 1, 2, 3)]
S[ 69] :- [(1, 4, 3, 2), (3, 2, 1, 4), (4, 1, 2, 3), (2, 3, 4, 1)]
S[ 70] :- [(1, 4, 3, 2), (3, 2, 1, 4), (4, 3, 2, 1), (2, 1, 4, 3)]
S[ 71] :- [(1, 4, 3, 2), (3, 2, 4, 1), (2, 3, 1, 4), (4, 1, 2, 3)]
S[ 72] :- [(1, 4, 3, 2), (3, 2, 4, 1), (4, 1, 2, 3), (2, 3, 1, 4)]
S[ 73] :- [(2, 1, 3, 4), (3, 4, 1, 2), (1, 2, 4, 3), (4, 3, 2, 1)]
S[ 74] :- [(2, 1, 3, 4), (3, 4, 1, 2), (4, 3, 2, 1), (1, 2, 4, 3)]
S[ 75] :- [(2, 1, 3, 4), (3, 4, 2, 1), (1, 2, 4, 3), (4, 3, 1, 2)]
S[ 76] :- [(2, 1, 3, 4), (3, 4, 2, 1), (1, 3, 4, 2), (4, 2, 1, 3)]
S[ 77] :- [(2, 1, 3, 4), (3, 4, 2, 1), (4, 2, 1, 3), (1, 3, 4, 2)]
S[ 78] :- [(2, 1, 3, 4), (3, 4, 2, 1), (4, 3, 1, 2), (1, 2, 4, 3)]
S[ 79] :- [(2, 1, 3, 4), (4, 3, 1, 2), (1, 2, 4, 3), (3, 4, 2, 1)]
S[ 80] :- [(2, 1, 3, 4), (4, 3, 1, 2), (1, 4, 2, 3), (3, 2, 4, 1)]
S[ 81] :- [(2, 1, 3, 4), (4, 3, 1, 2), (3, 2, 4, 1), (1, 4, 2, 3)]
S[ 82] :- [(2, 1, 3, 4), (4, 3, 1, 2), (3, 4, 2, 1), (1, 2, 4, 3)]
S[ 83] :- [(2, 1, 3, 4), (4, 3, 2, 1), (1, 2, 4, 3), (3, 4, 1, 2)]
S[ 84] :- [(2, 1, 3, 4), (4, 3, 2, 1), (3, 4, 1, 2), (1, 2, 4, 3)]
S[ 85] :- [(2, 1, 4, 3), (3, 4, 1, 2), (1, 2, 3, 4), (4, 3, 2, 1)]
S[ 86] :- [(2, 1, 4, 3), (3, 4, 1, 2), (1, 3, 2, 4), (4, 2, 3, 1)]
S[ 87] :- [(2, 1, 4, 3), (3, 4, 1, 2), (4, 2, 3, 1), (1, 3, 2, 4)]
S[ 88] :- [(2, 1, 4, 3), (3, 4, 1, 2), (4, 3, 2, 1), (1, 2, 3, 4)]
S[ 89] :- [(2, 1, 4, 3), (3, 4, 2, 1), (1, 2, 3, 4), (4, 3, 1, 2)]
S[ 90] :- [(2, 1, 4, 3), (3, 4, 2, 1), (4, 3, 1, 2), (1, 2, 3, 4)]
S[ 91] :- [(2, 1, 4, 3), (4, 3, 1, 2), (1, 2, 3, 4), (3, 4, 2, 1)]
S[ 92] :- [(2, 1, 4, 3), (4, 3, 1, 2), (3, 4, 2, 1), (1, 2, 3, 4)]
S[ 93] :- [(2, 1, 4, 3), (4, 3, 2, 1), (1, 2, 3, 4), (3, 4, 1, 2)]
S[ 94] :- [(2, 1, 4, 3), (4, 3, 2, 1), (1, 4, 3, 2), (3, 2, 1, 4)]
S[ 95] :- [(2, 1, 4, 3), (4, 3, 2, 1), (3, 2, 1, 4), (1, 4, 3, 2)]
S[ 96] :- [(2, 1, 4, 3), (4, 3, 2, 1), (3, 4, 1, 2), (1, 2, 3, 4)]
```

Exercise 2: Magic Inheritance.

Inheritance is a nice construct, that as you know, permits to enrich a class with methods/fields defined in their superclasses.

In python a similar behavior can be yielded via decorators and metaclasses. A decorator can be used to link one or more classes as referents to another and via a sort of osmosis the linked class will gain only those methods used by instances of the referent classes.

The exercise consists in implementing such a decorator (name `@magic`) that applied to a class definition, links such a class as a referent to the one passed as a parameter to the decorator. The osmosis only regards methods, i.e., fields are anchored to the last instance using one method and the anchor moves each time a new instance of the class invokes a method (not necessarily always the same). Basically the methods do not belong to the linked class until they are used by the real owner. If the decorator is used on many class definitions all these classes are referents of the linked class; if used several times but with different classes as parameter the osmosis happens towards all such classes.

The following is an example of the expected behavior:

```python
class Empty: pass

@magic(Empty)
class Person:
  def __init__(self, name, gross, netp):
    self.gross_salary = gross
    self.netpercentage = netp
    self.name = name
  def who(self): return self.name
  def salary(self): return self.gross_salary*self.netpercentage/12

@magic(Empty)
class Exam:
  def __init__(self, title, n, ne):
    self.title = title
    self.students = n
    self.exams = ne
  def todo(self):
    return "still {} students should pass the {} exam".format(
                        self.students-self.exams, self.title)

if __name__ == "__main__":
  m = Empty()
  x = Exam("PA", 100, 15)
  y = Exam("TSP", 50, 45)
  p = Person("Bob", 100000, .6)

  try:
    print("m salary :- ",m.salary())
  except Exception as e: print("*** {0}: {1}".format(type(e).__name__, e
  print("p salary :- ", p.salary())
  print("m salary :- ", m.salary())
  try:
    print("m todo :- ", m.todo())
  except Exception as e: print("*** {0}: {1}".format(type(e).__name__, e
  print("x todo :- ", x.todo())
  print("m todo :- ", m.todo())
  p.netpercentage=.45
  print("m salary :- ", m.salary())
  print("y todo :- ", y.todo())
  print("m todo :- ", m.todo())
  print("m students :- ",m.students)
  print("x todo :- ", x.todo())
  print("m students :- ",m.students)
  try:
```

```
[16:51]cazzola@surtur:~/pa/es2>python3 magic-inheritance.py
*** AttributeError: Attribute «salary» doesn't exist
p salary :-  5000.0
m salary :-  5000.0
*** AttributeError: Attribute «todo» doesn't exist
x todo :-  still 85 students should pass the PA exam
m todo :-  still 85 students should pass the PA exam
m salary :-  3750.0
y todo :-  still 5 students should pass the TSP exam
m todo :-  still 5 students should pass the TSP exam
m students :-  50
x todo :-  still 85 students should pass the PA exam
m students :-  100
```

Last Modified: Tue, 19 Feb 2013 16:10:06 ADAPT Lab.