Python3 Constraint Programming Tutorial/Manual

Nov.30.2020 Sugawara Systems

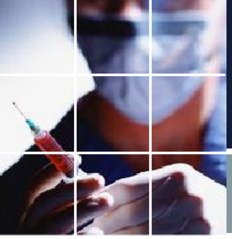
Introduction

We describe how to write constraints using Python 3.

When should we use Python3?

We can write most constraints using GUI. So, we recommend using GUI unless the following cases.

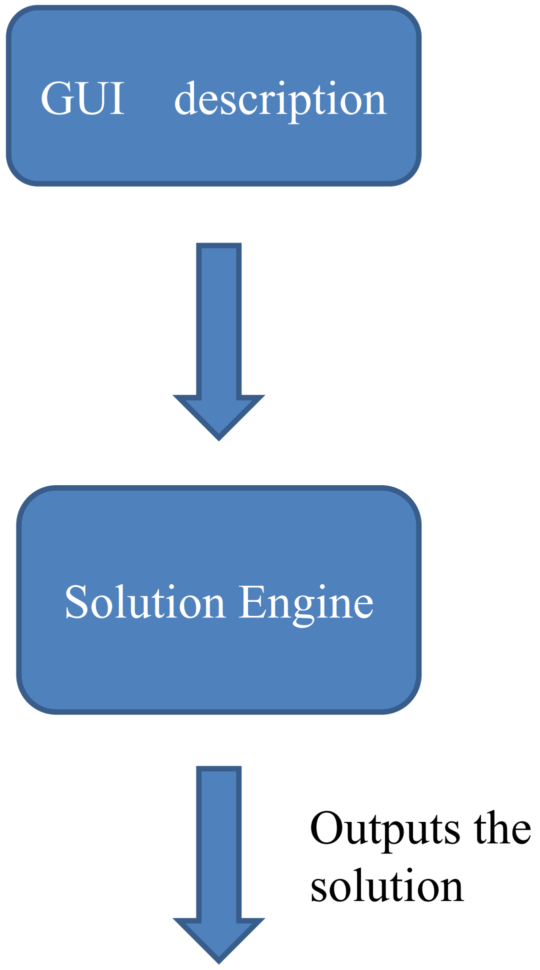
■There are constraints we can not describe using GUI.
■We want to switch the constraints on/off dynamically.
■Dynamic constraints will make the descriptions more maintainable.



A workflow without Python3

■A typical workflow without Python 3 looks like the one on the right.

■The solution engine receives a GUI description and outputs the solution.

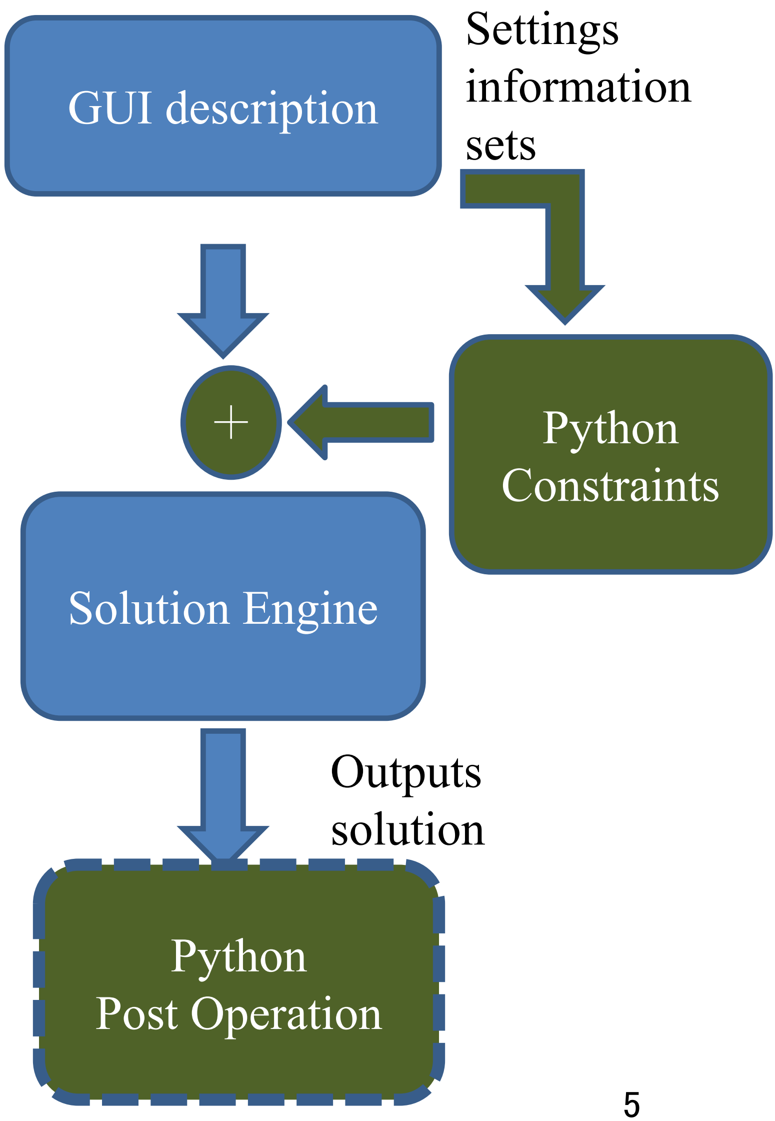


A workflow using Python3

■We describe constraints on Python3, utilizing the set of information of GUI. We can write rules on minimum efforts since GUI already has the necessary settings.

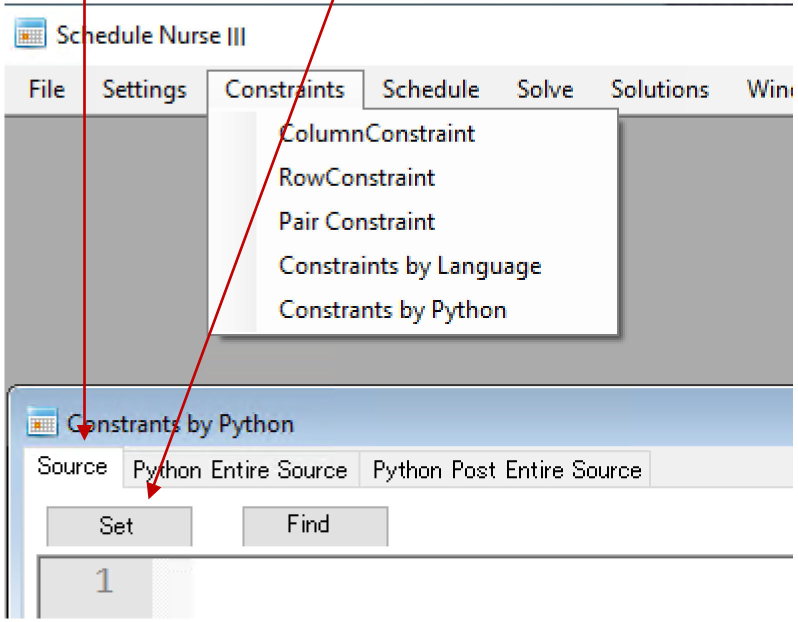
■You can add constraints with add-ons. On the other hand, you can also easily detach them by a checkbox.

■If needed, we can add a post solution operation using shift and task solution.

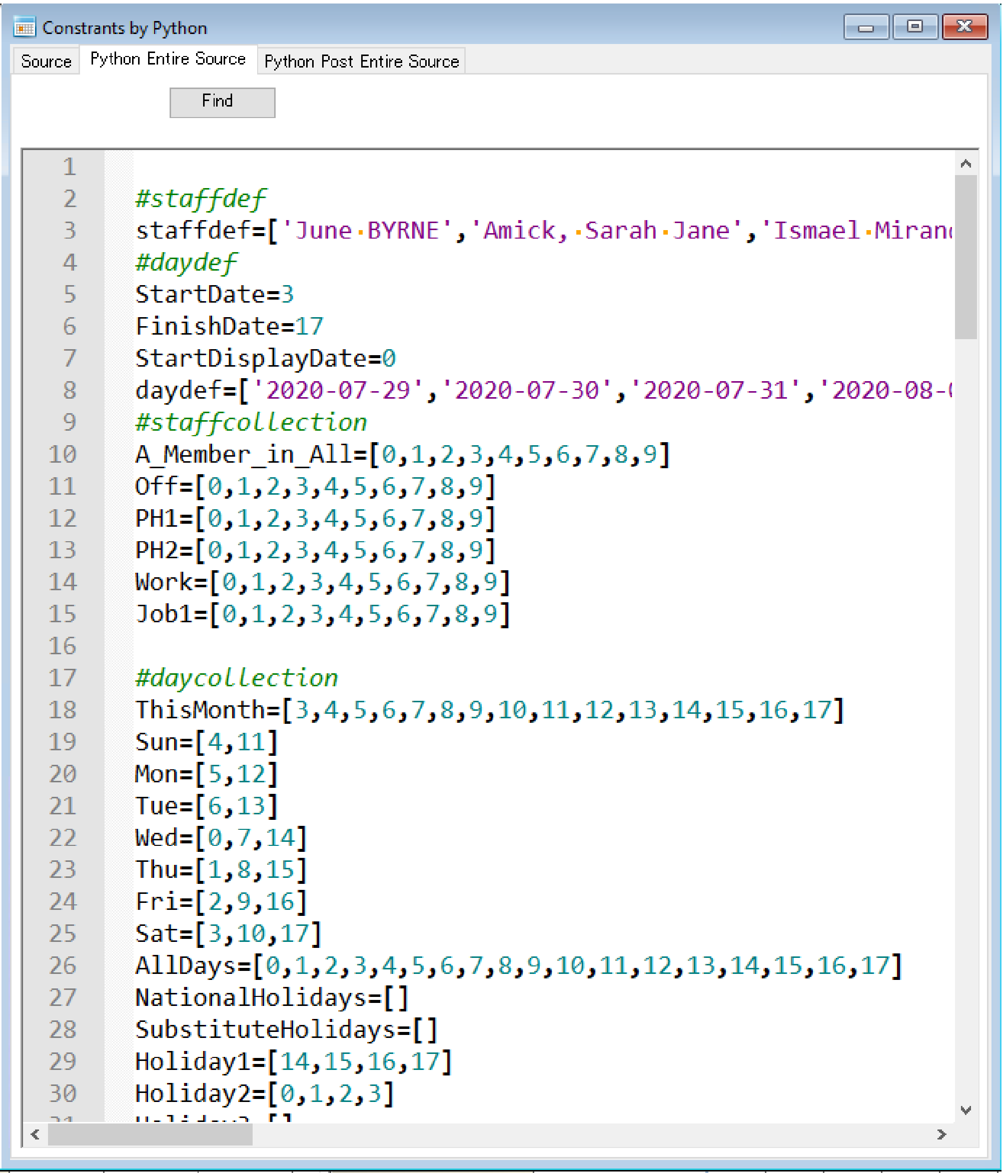


Python Source Edit

- Constraints →Constraints by Python→Source
- Source is the only page where users can write Python code.
- When you click Set, your project will import the source.



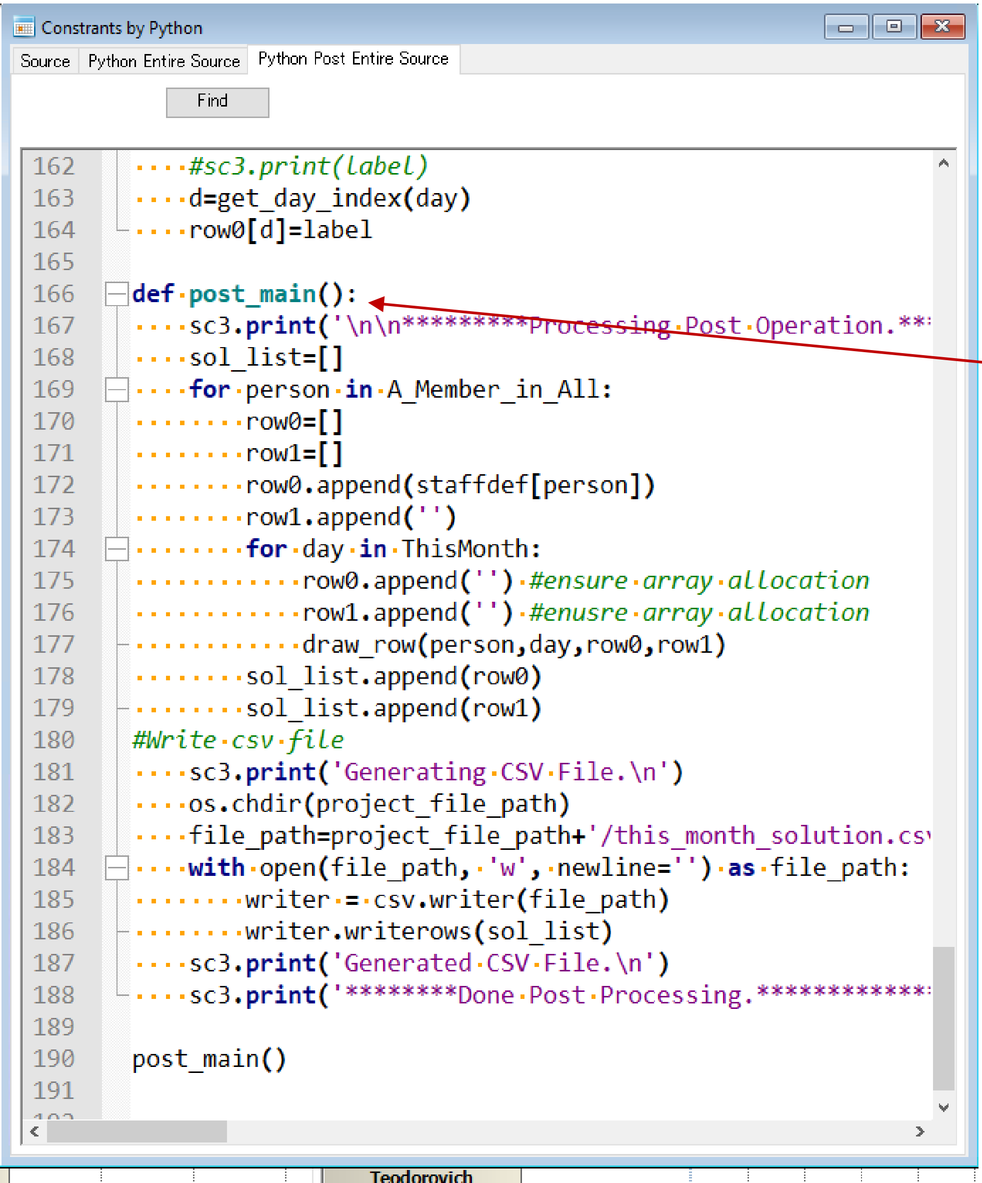
Python Entire Source



```

#staffdef
staffdef=['June BYRNE','Amick, Sarah Jane','Ismael Miran
#daydef
StartDate=3
FinishDate=17
StartDisplayDate=0
daydef=['2020-07-29','2020-07-30','2020-07-31','2020-08-
#staffcollection
A_Member_in_All=[0,1,2,3,4,5,6,7,8,9]
Off=[0,1,2,3,4,5,6,7,8,9]
PH1=[0,1,2,3,4,5,6,7,8,9]
PH2=[0,1,2,3,4,5,6,7,8,9]
Work=[0,1,2,3,4,5,6,7,8,9]
Job1=[0,1,2,3,4,5,6,7,8,9]

#daycollection
ThisMonth=[3,4,5,6,7,8,9,10,11,12,13,14,15,16,17]
Sun=[4,11]
Mon=[5,12]
Tue=[6,13]
Wed=[0,7,14]
Thu=[1,8,15]
Fri=[2,9,16]
Sat=[3,10,17]
AllDays=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17]
NationalHolidays=[]
SubstituteHolidays=[]
Holiday1=[14,15,16,17]
Holiday2=[0,1,2,3]
```

- This page is also Read Only.
- The description is GUI-generated information , followed your written source.
- Python interpreter in the solver interprets and executes the code on this page.
- The purpose of the page is to show the error position and for your reference to GUI objects.

Python Post Entire Source



```
    #sc3.print(label)
    d=get_day_index(day)
    row0[d]=label

def post_main():
    sc3.print('\n\n*********Processing Post Operation.**
    sol_list=[]
    for person in A_Member_in_All:
        row0=[]
        row1=[]
        row0.append(staffdef[person])
        row1.append('')
        for day in ThisMonth:
            row0.append('') #ensure array allocation
            row1.append('') #enusre array allocation
            draw_row(person,day,row0,row1)
        sol_list.append(row0)
        sol_list.append(row1)
#Write csv file
    sc3.print('Generating CSV File.\n')
    os.chdir(project_file_path)
    file_path=project_file_path+'/this_month_solution.cs
    with open(file_path, 'w', newline='') as file_path:
        writer = csv.writer(file_path)
        writer.writerows(sol_list)
    sc3.print('Generated CSV File.\n')
    sc3.print('*********Done Post Processing.*************

post_main()
```

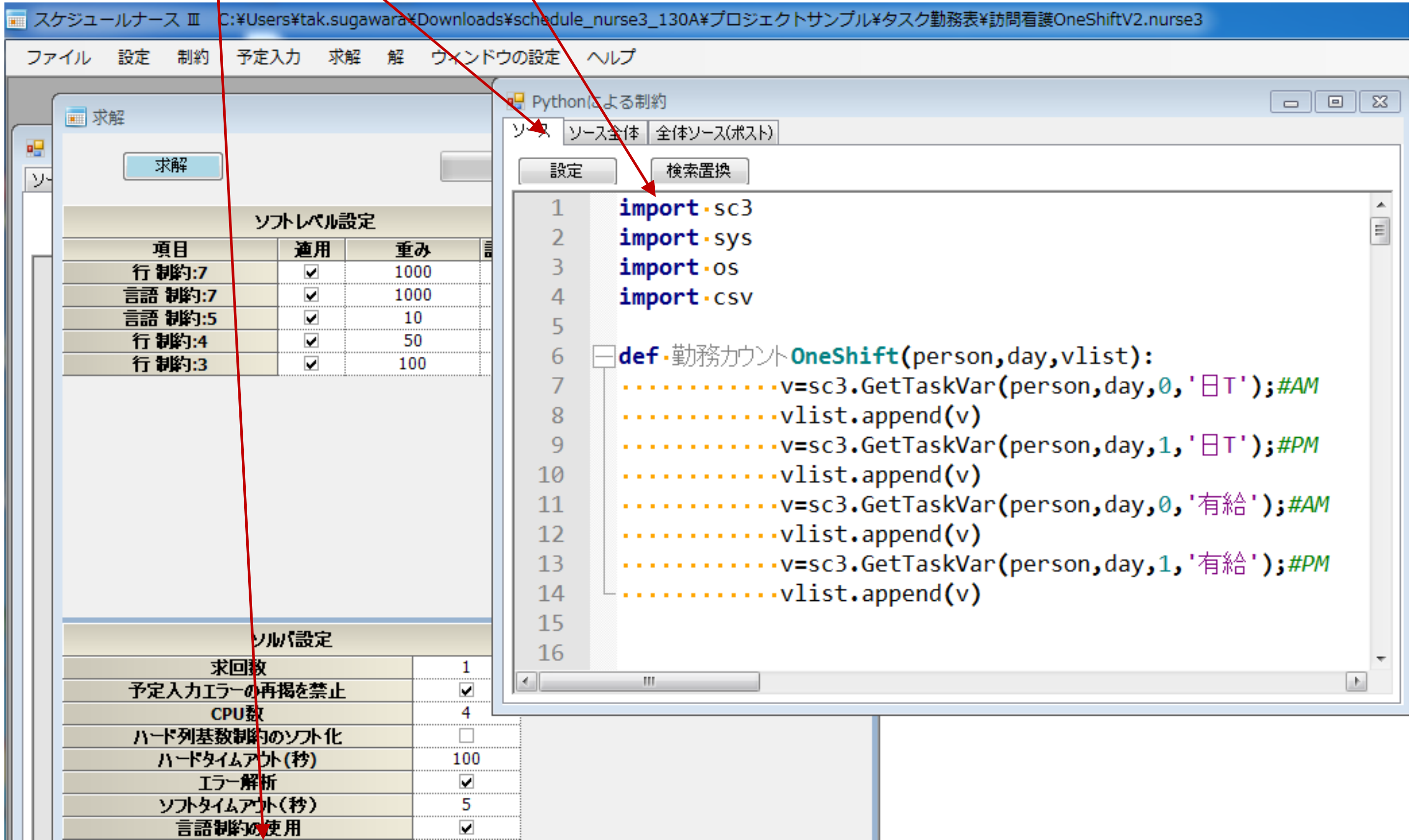
■This page is also Read Only.

■The description is GUI-generated information and the solution information, followed your written source.

■def post_main() is a dedicated function for post-processing. If you write it in the source, python will automatically invoke post_main() as its main routine after you solve the problem.

How to run Python

- First, declare import sc3.
- Click on Set.
- Check the Use Language Constraints checkbox.
- Click on Solve.


スケジュールナース Ⅲ C:¥Users¥tak.sugawara¥Downloads¥schedule_nurse3_130A¥プロジェクトサンプル¥タスク勤務表¥訪問看護OneShiftV2.nurse3
ファイル 設定 制約 予定入力 求解 解 ウィンドウの設定 ヘルプ
求解
求解
Pythonによる制約
ソース ソース全体 全体ソース(ポスト)
設定 検索置換

ソフトレベル設定

項目	適用	重み
行 制約:7	✓	1000
言語 制約:7	✓	1000
言語 制約:5	✓	10
行 制約:4	✓	50
行 制約:3	✓	100

ソルバ設定

求回数	1
予定入力エラーの再掲を禁止	✓
CPU数	4
ハード列基数制約のソフト化	☐
ハードタイムアウト(秒)	100
エラー解析	✓
ソフトタイムアウト(秒)	5
言語制約の使用	✓

```python
import sc3
import sys
import os
import csv

def 勤務カウントOneShift(person,day,vlist):
            v=sc3.GetTaskVar(person,day,0,'日T');#AM
            vlist.append(v)
            v=sc3.GetTaskVar(person,day,1,'日T');#PM
            vlist.append(v)
            v=sc3.GetTaskVar(person,day,0,'有給');#AM
            vlist.append(v)
            v=sc3.GetTaskVar(person,day,1,'有給');#PM
            vlist.append(v)
```

Python Version

Supports Python 3.68 and above.

GUI Notes on Python

■ In the GUI description, we use almost all of the entries as python variables. In some cases, we should modify the variable name to address the parser error in Python. The solver renames them automatically to keep as much of the original character as possible.

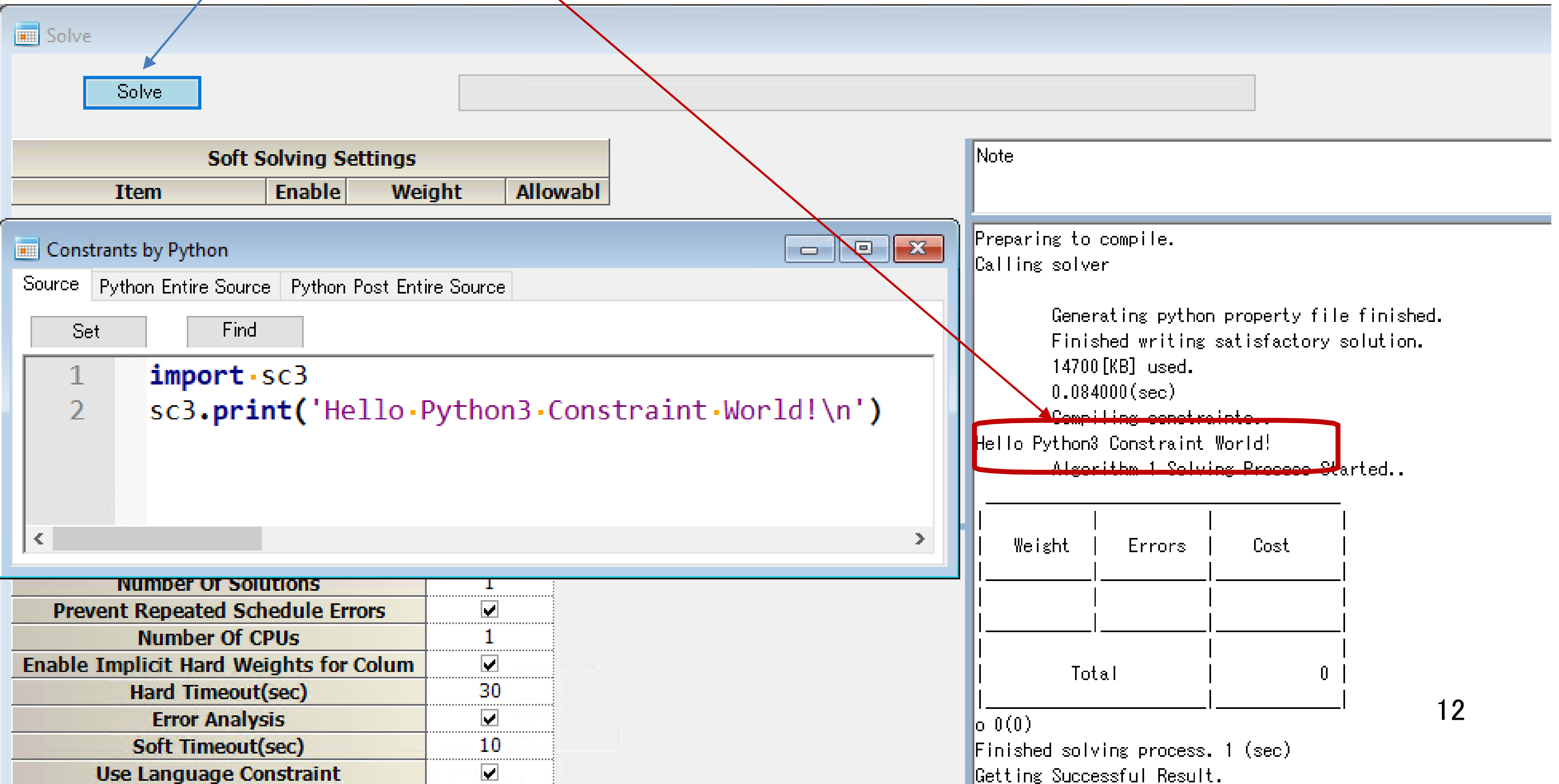
■ However, please keep this in mind for an easier reading of python code. Followings are examples of the error (letters).

'*-()[].'
'1variable' is illegal, variable1 is legal.

■ Of course, you can ignore the rule above if you have no plan to use Python.

Hello World!

- Load , and Solve the project hello_python_world.nurse3.
- You should see a Hello… in the right pane.



```python
import sc3
sc3.print('Hello Python3 Constraint World!\n')
```

Soft Solving Settings			
Item	Enable	Weight	Allowabl

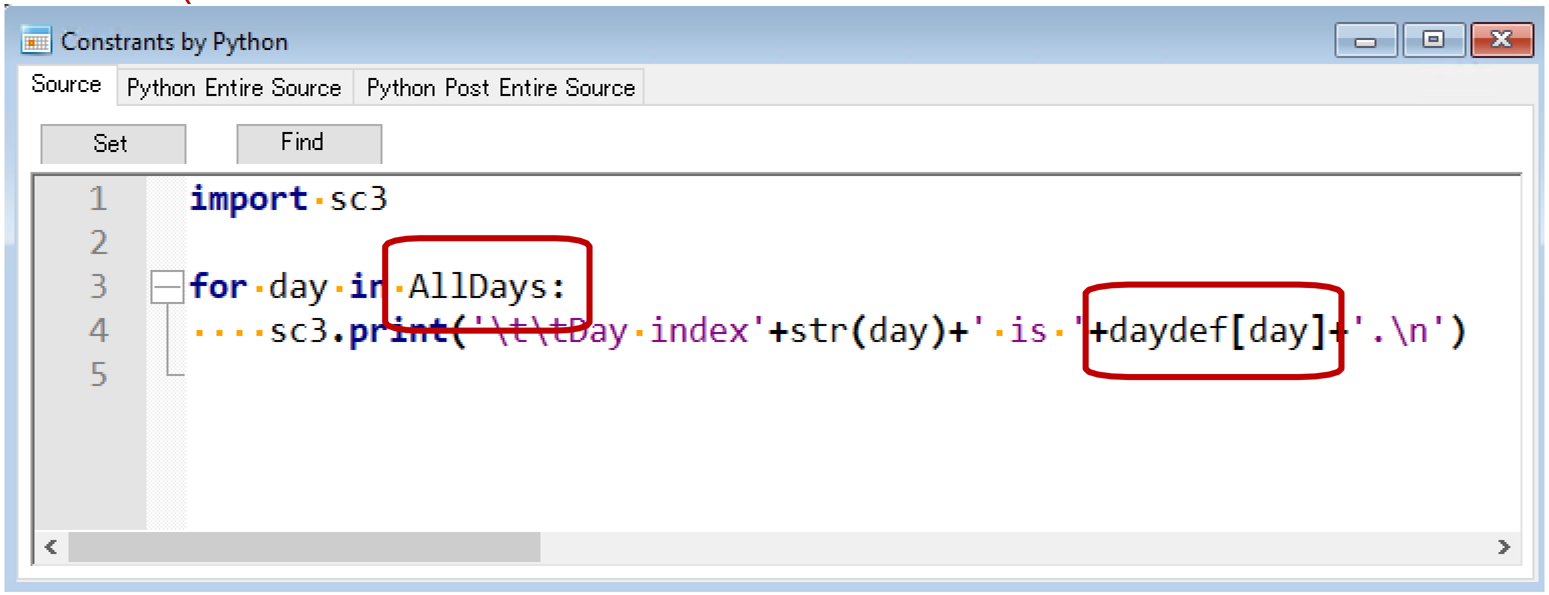
Number Of Solutions	1
Prevent Repeated Schedule Errors	✓
Number Of CPUs	1
Enable Implicit Hard Weights for Colum	✓
Hard Timeout(sec)	30
Error Analysis	✓
Soft Timeout(sec)	10
Use Language Constraint	✓

```
Preparing to compile.
Calling solver

             Generating python property file finished.
             Finished writing satisfactory solution.
             14700[KB] used.
             0.084000(sec)
             Compiling constraints..
Hello Python3 Constraint World!
             Algorithm 1 Solving Process Started..

  _____________________________________________________________
 |                   |                    |                    |
 |     Weight        |      Errors        |       Cost         |
 |___________________|____________________|____________________|
 |                   |                    |                    |
 |___________________|____________________|____________________|
 |                                        |                    |
 |                  Total                 |                  0 |
 |________________________________________|____________________|
o 0(0)
Finished solving process. 1 (sec)
Getting Successful Result.
```

python_tutorial1

■ Load the project Python_tutorial1.nurse3

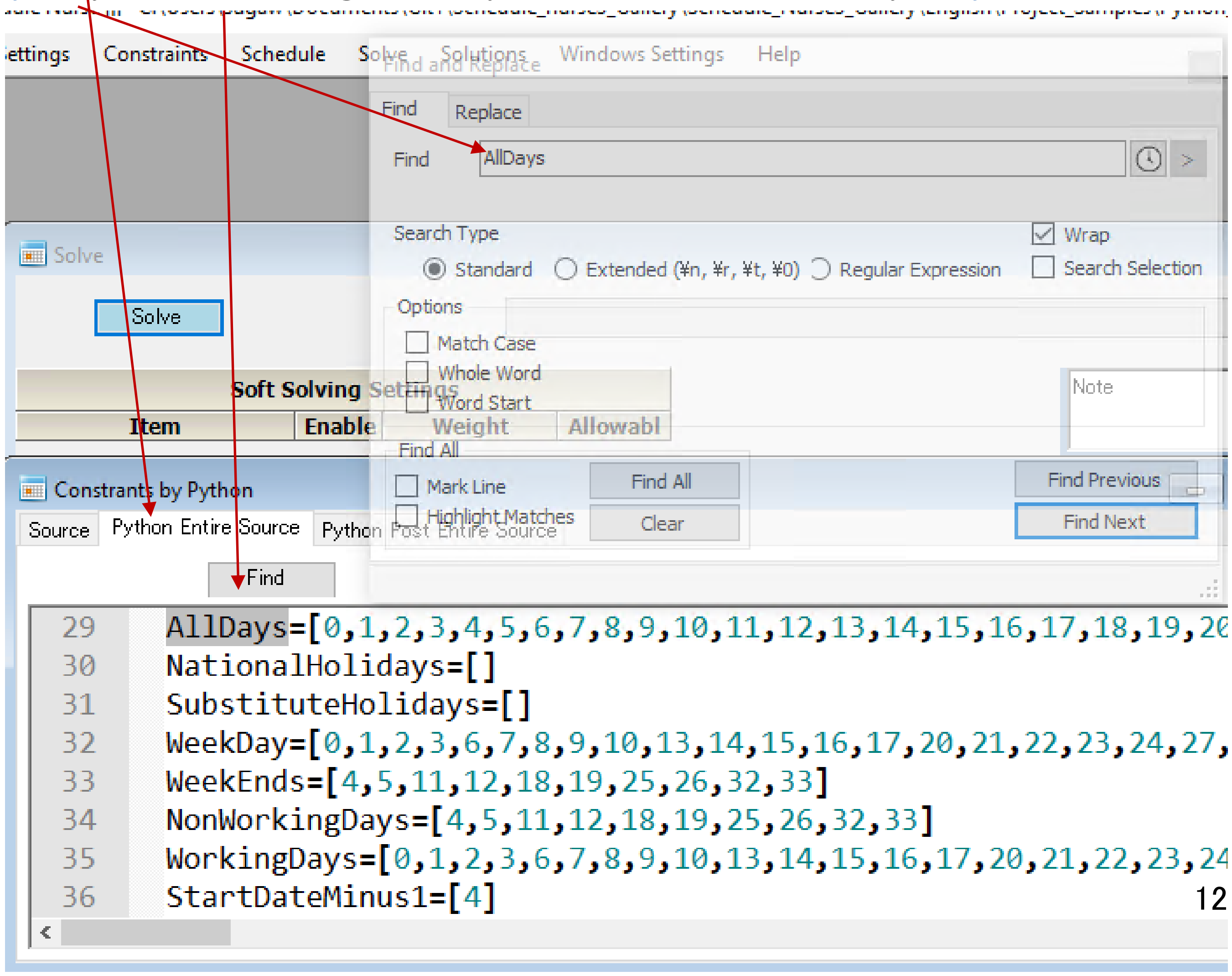
■ Use sc3.print() for displaying value of variables. It can only display string, so you need to convert integer to a string using str().



```
import sc3

for day in AllDays:
    sc3.print('\t\tDay index'+str(day)+' is '+daydef[day]+'.\n')

```

Where is AllDays and daydef?

- Click on Python Entire Source, which is output from Solution Solver.
- You can find AllDays by Find Dialog. Also you can find 1D array daydef.



```
AllDays=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
NationalHolidays=[]
SubstituteHolidays=[]
WeekDay=[0,1,2,3,6,7,8,9,10,13,14,15,16,17,20,21,22,23,24,27,
WeekEnds=[4,5,11,12,18,19,25,26,32,33]
NonWorkingDays=[4,5,11,12,18,19,25,26,32,33]
WorkingDays=[0,1,2,3,6,7,8,9,10,13,14,15,16,17,20,21,22,23,24
StartDateMinus1=[4]
```

Log of python_tutorial1

■ See the generated log. You can see the python interpreter displays the sc3.print data, and then the nominal solving process will start.

```
Generating python property file finished.
Compiling constraints..
          Day index0 is 2020-10-27.
          Day index1 is 2020-10-28.
          Day index2 is 2020-10-29.
          …
Algorithm 1 Solving Process Started..
```

GetShiftVar　python_tutorial2

■Each shift is OneHot encoded, which results that A_Member_in_All*AllDays*Shifts per day binary variables exist in the search space. You can call any shift variable per the following format.

```
v=sc3.GetShiftVar(person,day,shift);
```

where person/day are zero-based index and shift is zero-based index or string.

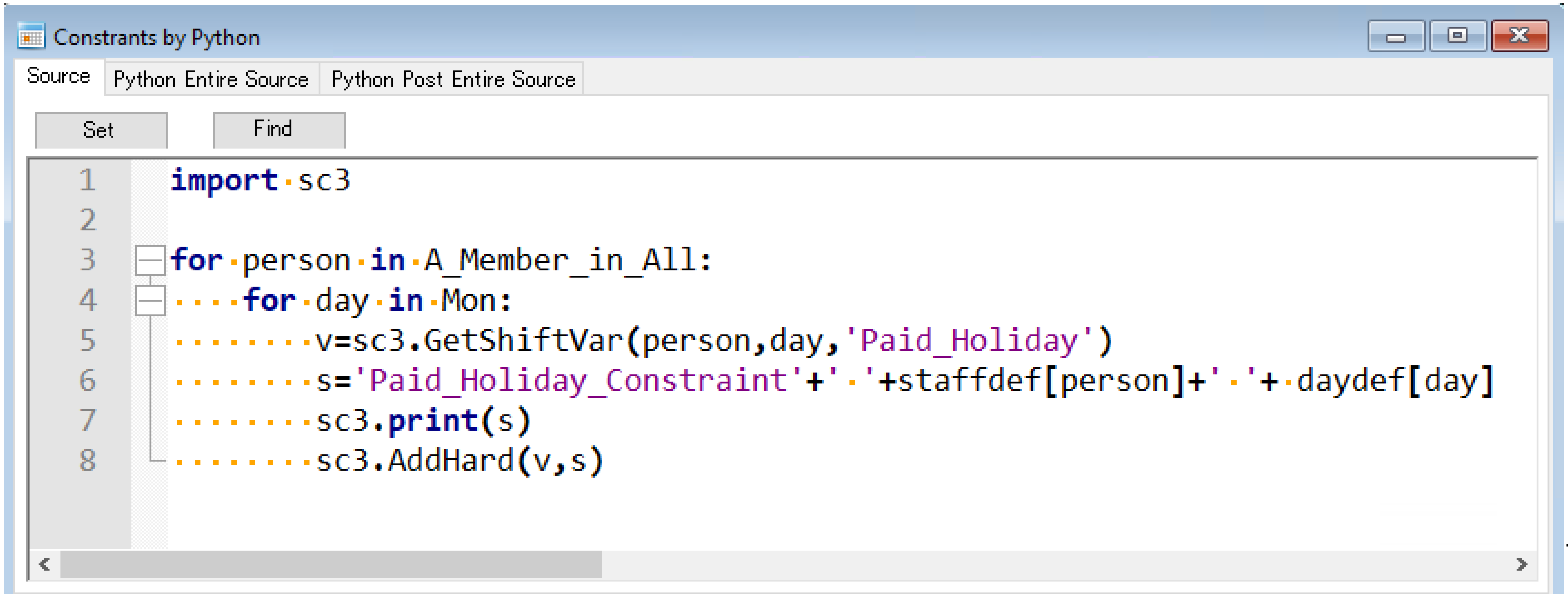
■The most important thing to remember is the activated shift is at least one and at most one per day per staff. For example, if you assert any Shift variable, the other variables on the same person and the day automatically become de-asserted.

AddHard　python_tutorial2

■ The following format is a function to add a hard constraint.

sc3.AddHard(v,string)

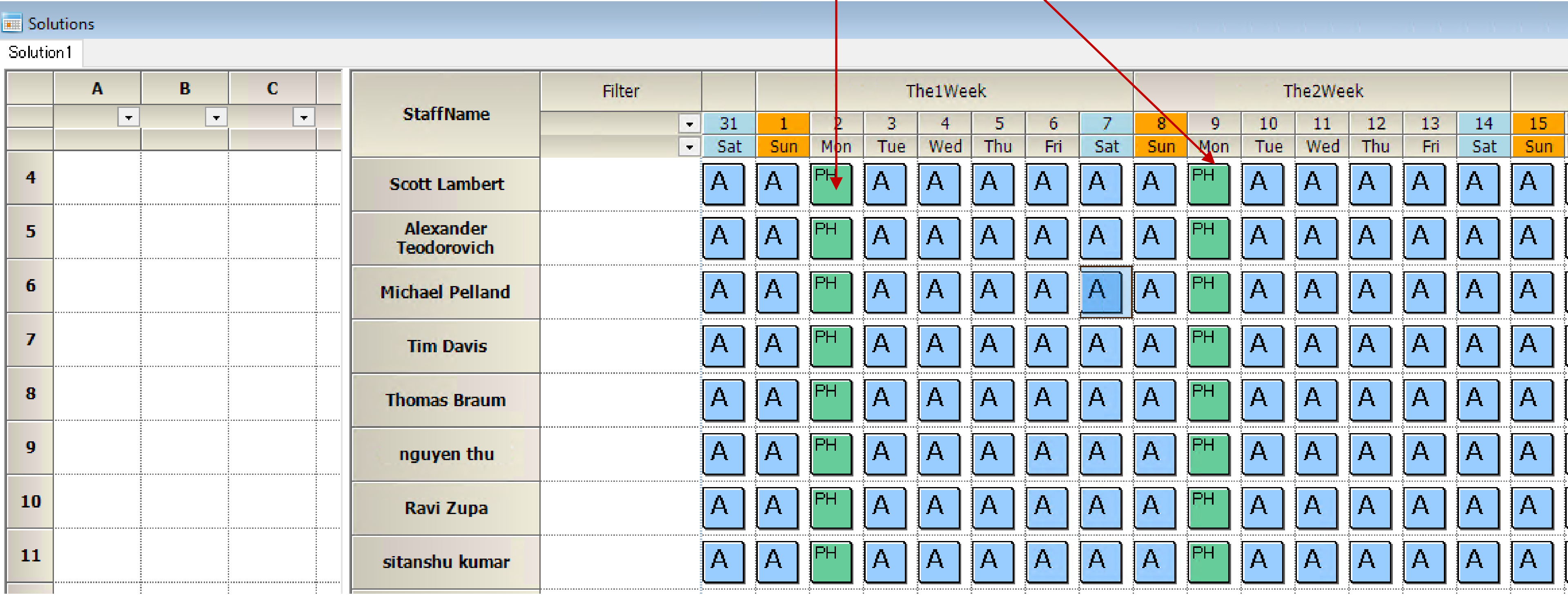
■ v is the shift variable obtained by GetShiftVar or the output variable of And/Or/Not/SeqExpr/SeqComp.



```
import sc3

for person in A_Member_in_All:
    for day in Mon:
        v=sc3.GetShiftVar(person,day,'Paid_Holiday')
        s='Paid_Holiday_Constraint'+' '+staffdef[person]+' '+ daydef[day]
        sc3.print(s)
        sc3.AddHard(v,s)
```

Python_tutorial2 result

- You can see all staff could be given a Paid_Holiday on Mondays.



Solutions

Solution1

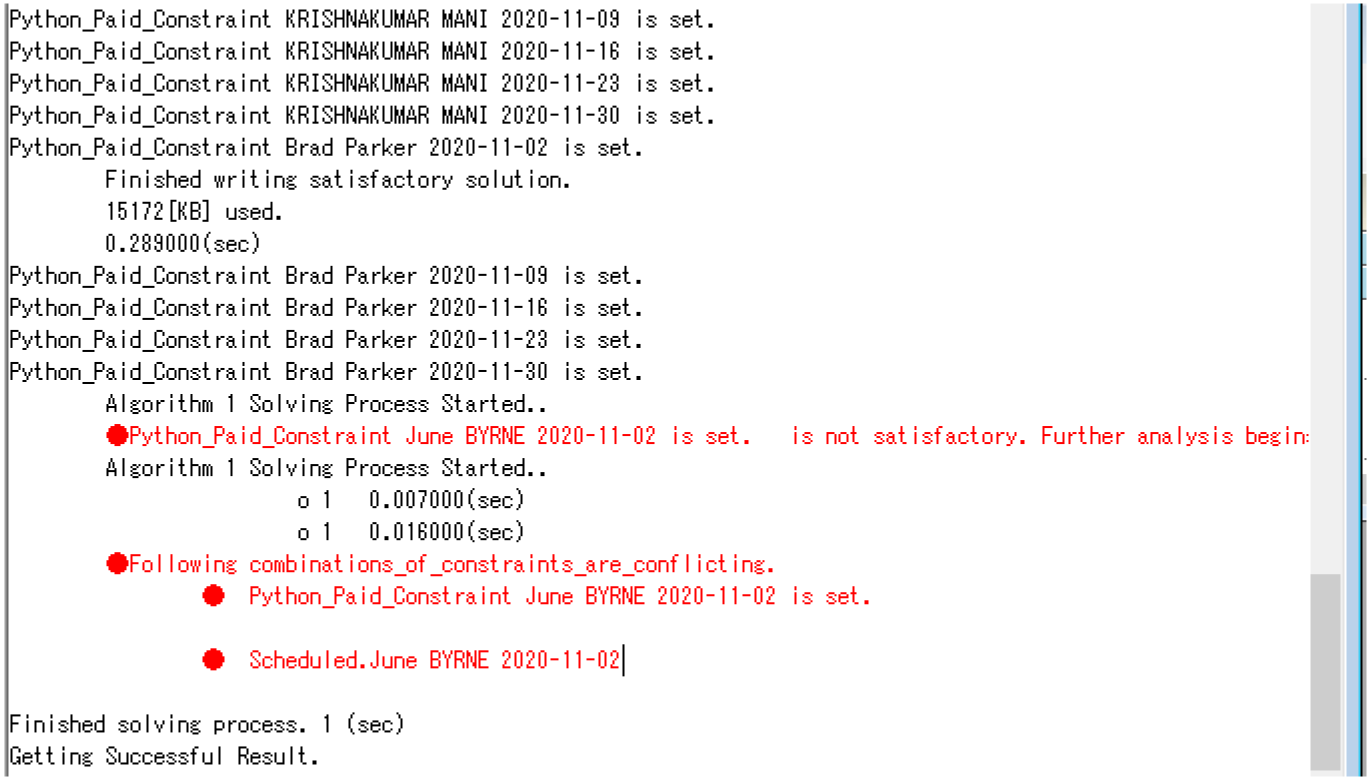
	A	B	C
4			
5			
6			
7			
8			
9			
10			
11			

StaffName	Filter		The1Week							The2Week							
		31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Scott Lambert		A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A
Alexander Teodorovich		A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A
Michael Pelland		A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A
Tim Davis		A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A
Thomas Braum		A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A
nguyen thu		A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A
Ravi Zupa		A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A
sitanshu kumar		A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A

python_tutorial2_error

■ We made every Monday a Paid_Holiday Shift using Python. In contrast, We put the Day_Shift on Monday on the Schedule. Since Python constraints and the Schedule are contradictory and both are hard-constraints, we have no answer.

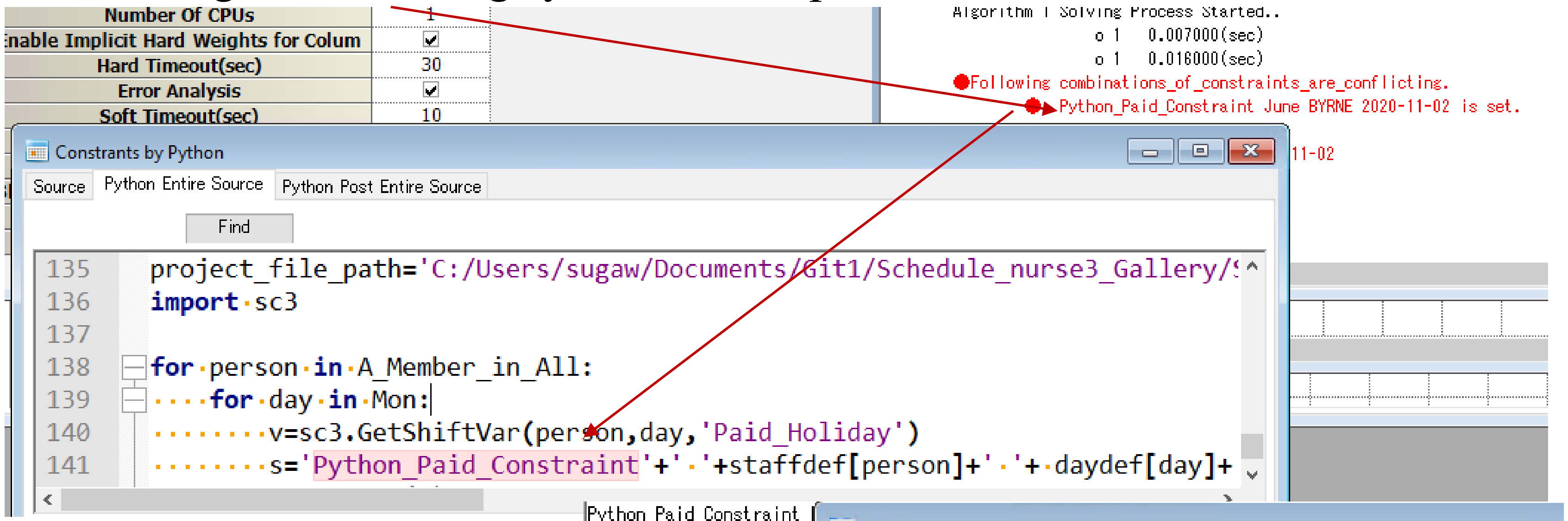
■ Look at the right pane; you'll see Red Marked portions.



```
Python_Paid_Constraint KRISHNAKUMAR MANI 2020-11-09 is set.
Python_Paid_Constraint KRISHNAKUMAR MANI 2020-11-16 is set.
Python_Paid_Constraint KRISHNAKUMAR MANI 2020-11-23 is set.
Python_Paid_Constraint KRISHNAKUMAR MANI 2020-11-30 is set.
Python_Paid_Constraint Brad Parker 2020-11-02 is set.
        Finished writing satisfactory solution.
        15172[KB] used.
        0.289000(sec)
Python_Paid_Constraint Brad Parker 2020-11-09 is set.
Python_Paid_Constraint Brad Parker 2020-11-16 is set.
Python_Paid_Constraint Brad Parker 2020-11-23 is set.
Python_Paid_Constraint Brad Parker 2020-11-30 is set.
        Algorithm 1 Solving Process Started..
        ●Python_Paid_Constraint June BYRNE 2020-11-02 is set.   is not satisfactory. Further analysis begin
        Algorithm 1 Solving Process Started..
                        o 1   0.007000(sec)
                        o 1   0.016000(sec)
        ●Following combinations_of_constraints_are_conflicting.
                ●  Python_Paid_Constraint June BYRNE 2020-11-02 is set.

                ●  Scheduled.June BYRNE 2020-11-02

Finished solving process. 1 (sec)
Getting Successful Result.
```

python_tutorial2_error error position

- Dbl-Clicking on here brings you the error position as below.

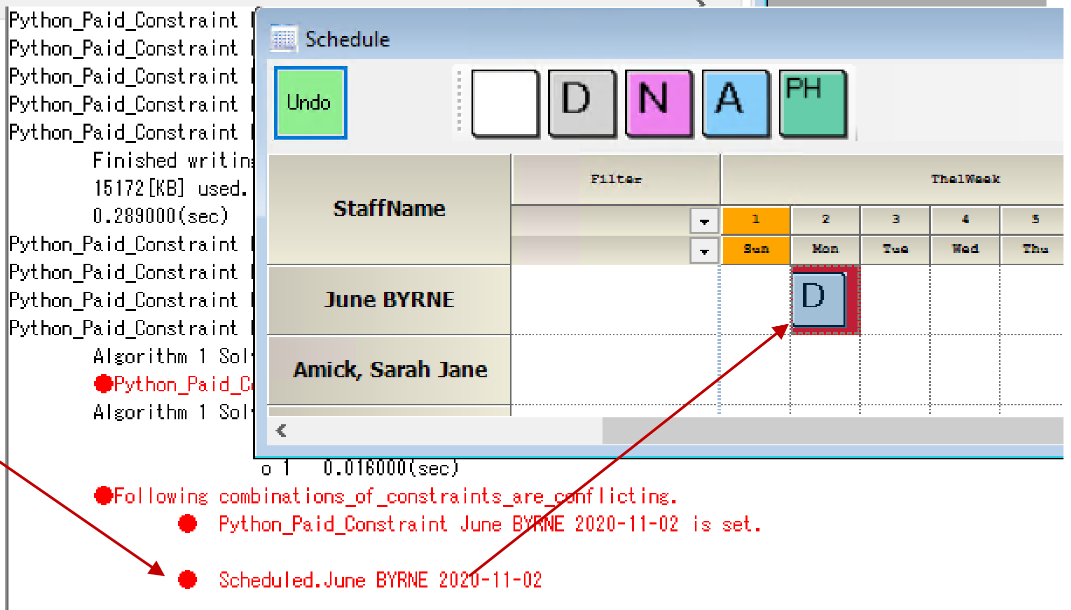


Number Of CPUs	1
nable Implicit Hard Weights for Colum	✓
Hard Timeout(sec)	30
Error Analysis	✓
Soft Timeout(sec)	10

```
Algorithm 1 Solving Process Started..
        o 1    0.007000(sec)
        o 1    0.016000(sec)
●Following combinations_of_constraints_are_conflicting.
    ●Python_Paid_Constraint June BYRNE 2020-11-02 is set.
11-02
```

```python
project_file_path='C:/Users/sugaw/Documents/Git1/Schedule_nurse3_Gallery/S
import sc3

for person in A_Member_in_All:
    for day in Mon:
        v=sc3.GetShiftVar(person,day,'Paid_Holiday')
        s='Python_Paid_Constraint'+' '+staffdef[person]+' '+daydef[day]+
```

- Dbl-Clicking on here brings you the error position as right.



```
Python_Paid_Constraint
Python_Paid_Constraint
Python_Paid_Constraint
Python_Paid_Constraint
Python_Paid_Constraint
        Finished writin
        15172[KB] used.
        0.289000(sec)
Python_Paid_Constraint
Python_Paid_Constraint
Python_Paid_Constraint
Python_Paid_Constraint
        Algorithm 1 Sol
        ●Python_Paid_C
        Algorithm 1 Sol
                o 1    0.016000(sec)
        ●Following combinations_of_constraints_are_conflicting.
            ●  Python_Paid_Constraint June BYRNE 2020-11-02 is set.

            ●  Scheduled.June BYRNE 2020-11-02
```

StaffName	Filter	TheWeek				
		1	2	3	4	5
		Sun	Mon	Tue	Wed	Thu
June BYRNE			D			
Amick, Sarah Jane						

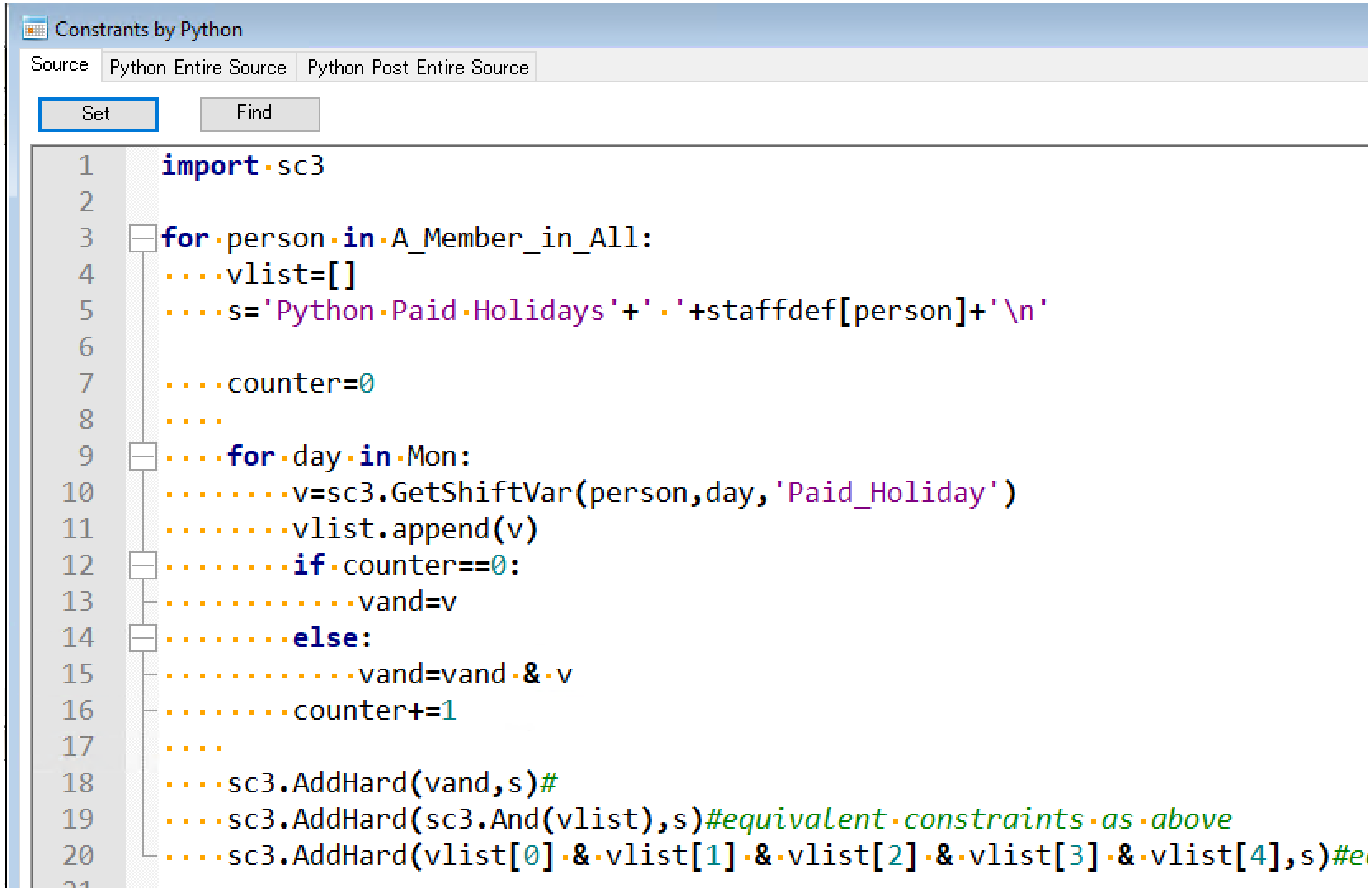
python_tutorial2_error AddHard note

■Please note the added string information at AddHard is a clue for what is conflicting. We strongly recommend you set it up because you'll have a lot of trouble without it.

■The recommended format will be as follows.
1D-Array　Keyword＋SPACE+staff/day_words
2D-Array　Keyword+staff/day_words+SPACE+staff/day_worsds

And　python_tutorial3

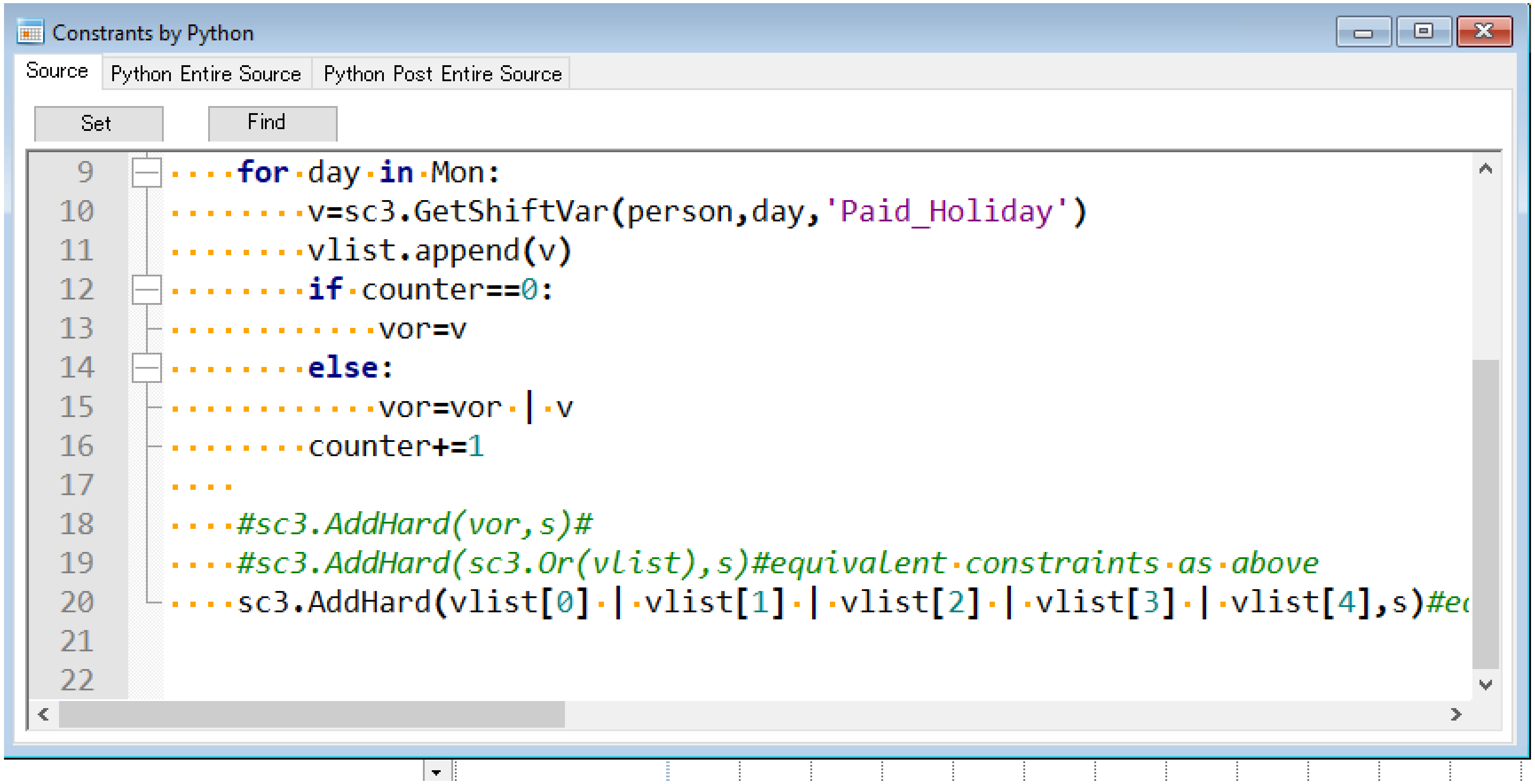
■ The last three lines are all Valid and equivalent constraints.



```python
import sc3

for person in A_Member_in_All:
    vlist=[]
    s='Python Paid Holidays'+' '+staffdef[person]+'\n'

    counter=0
    
    for day in Mon:
        v=sc3.GetShiftVar(person,day,'Paid_Holiday')
        vlist.append(v)
        if counter==0:
            vand=v
        else:
            vand=vand & v
        counter+=1
    
    sc3.AddHard(vand,s)#
    sc3.AddHard(sc3.And(vlist),s)#equivalent constraints as above
    sc3.AddHard(vlist[0] & vlist[1] & vlist[2] & vlist[3] & vlist[4],s)#e
```

Or python_tutorial4

- The last three lines are all Valid and equivalent constraints.



```
    for day in Mon:
        v=sc3.GetShiftVar(person,day,'Paid_Holiday')
        vlist.append(v)
        if counter==0:
            vor=v
        else:
            vor=vor | v
        counter+=1
    
    #sc3.AddHard(vor,s)#
    #sc3.AddHard(sc3.Or(vlist),s)#equivalent constraints as above
    sc3.AddHard(vlist[0] | vlist[1] | vlist[2] | vlist[3] | vlist[4],s)#ec
```

And Or Not : python_tutorial5

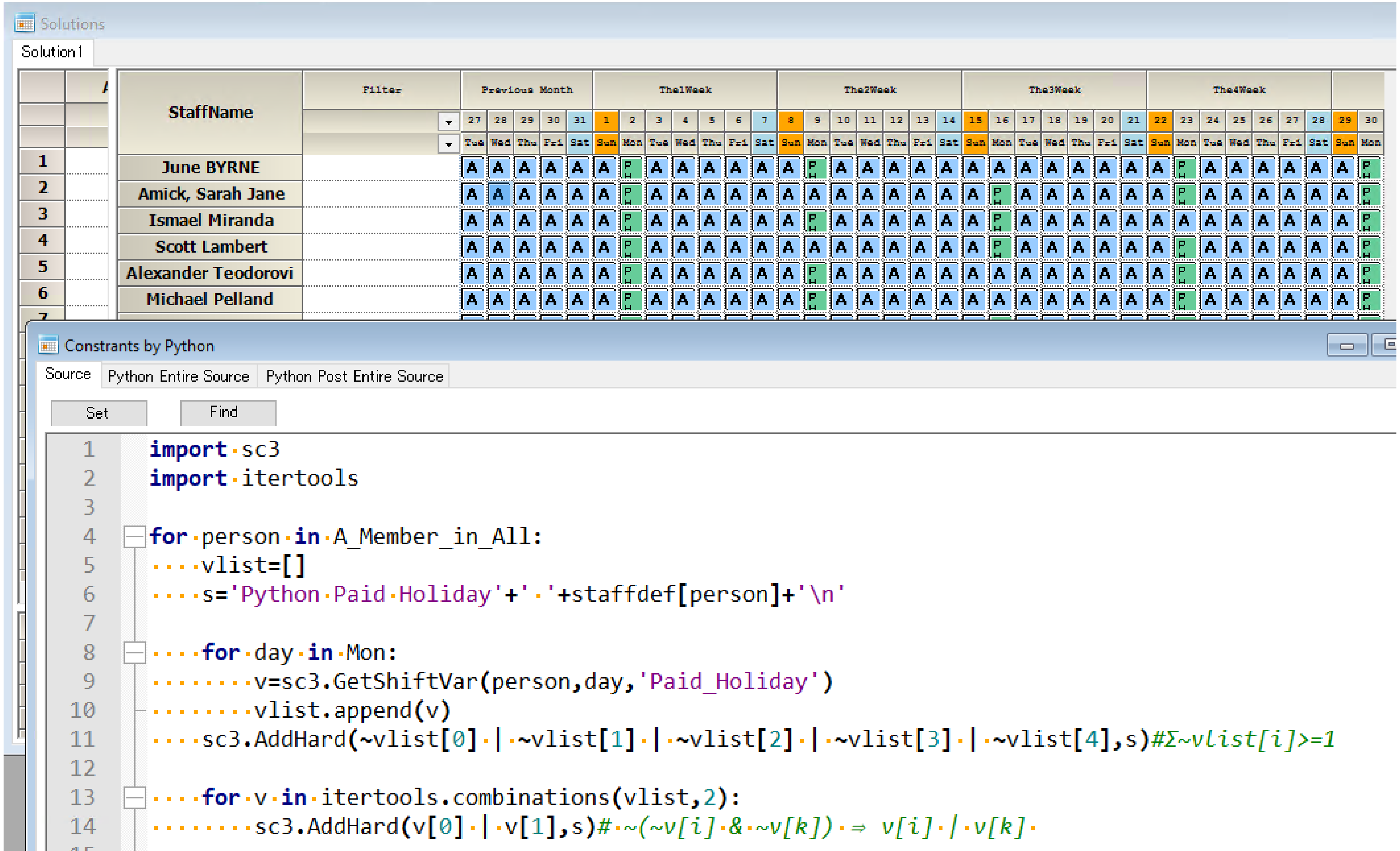
■Consider the problem we have four Paid_Holidays out of five Mondays using only logical operators.
■Since the problem requires Σ~Paid_Holiday equals to 1. We have the following two constraints.

1) Σ~Paid_Holiday >=1;
2) 2) Of all the combinations that take two from five, ~Paid_Holiday and ~Paid_Holiday are prohibited.　i.e. Σ~Paid_Holiday<=1;

■Enumerating all the combinations of 5 to 2 that we can take is a pain, but we can leave that to Python Itertools, which will enumerate all the combinations for us.

python_tutorial5 result

■We were able to achieve the specification that only one of the five Mondays is not a Paid_Holiday.


Solutions

Solution1

	StaffName	Filter	Previous Month					The1Week							The2Week							The3Week							The4Week								
			27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
			Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	June BYRNE		A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	A	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH
2	Amick, Sarah Jane		A	A	A	A	A	A	PH	A	A	A	A	A	A	A	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH
3	Ismael Miranda		A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	A	A	A	A	A	A	A	PH
4	Scott Lambert		A	A	A	A	A	A	PH	A	A	A	A	A	A	A	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH
5	Alexander Teodorovi		A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	A	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH
6	Michael Pelland		A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	A	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH

Constrants by Python

```
import sc3
import itertools

for person in A_Member_in_All:
    vlist=[]
    s='Python Paid Holiday'+' '+staffdef[person]+'\n'

    for day in Mon:
        v=sc3.GetShiftVar(person,day,'Paid_Holiday')
        vlist.append(v)
    sc3.AddHard(~vlist[0] | ~vlist[1] | ~vlist[2] | ~vlist[3] | ~vlist[4],s)#Σ~vlist[i]>=1

    for v in itertools.combinations(vlist,2):
        sc3.AddHard(v[0] | v[1],s)# ~(~v[i] & ~v[k]) ⇒ v[i] | v[k]
```

python_tutorial6 Cardinal Constraints

■The constraints on inequalities are called cardinal constraints. In python_tutorial6, Xi = Paid_Holiday

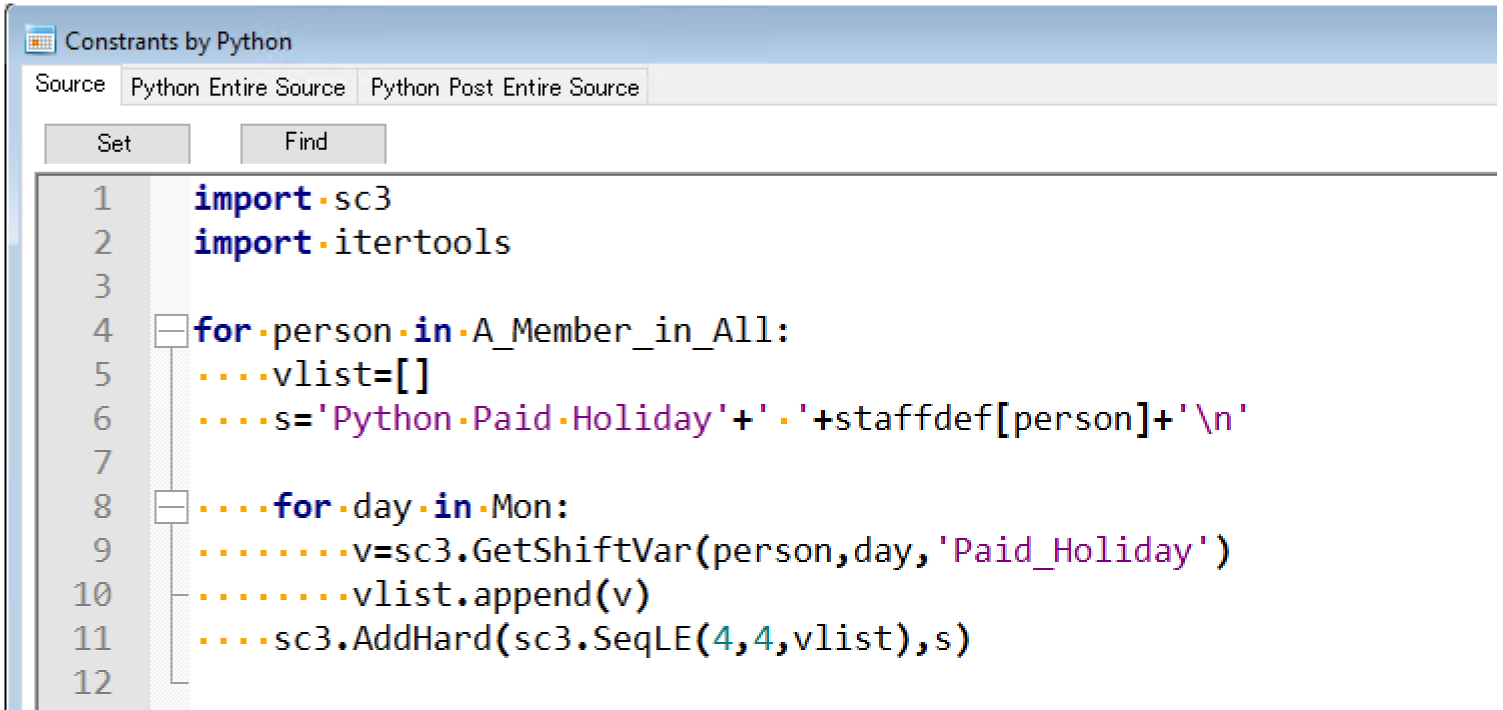
Σ~Xi<=4 AND Σ~Xi>=4

using only the logical expressions And,Or,Not.

■However, the cardinal constraint has its own dedicated function.

sc3.SeqLE(min,max,List)

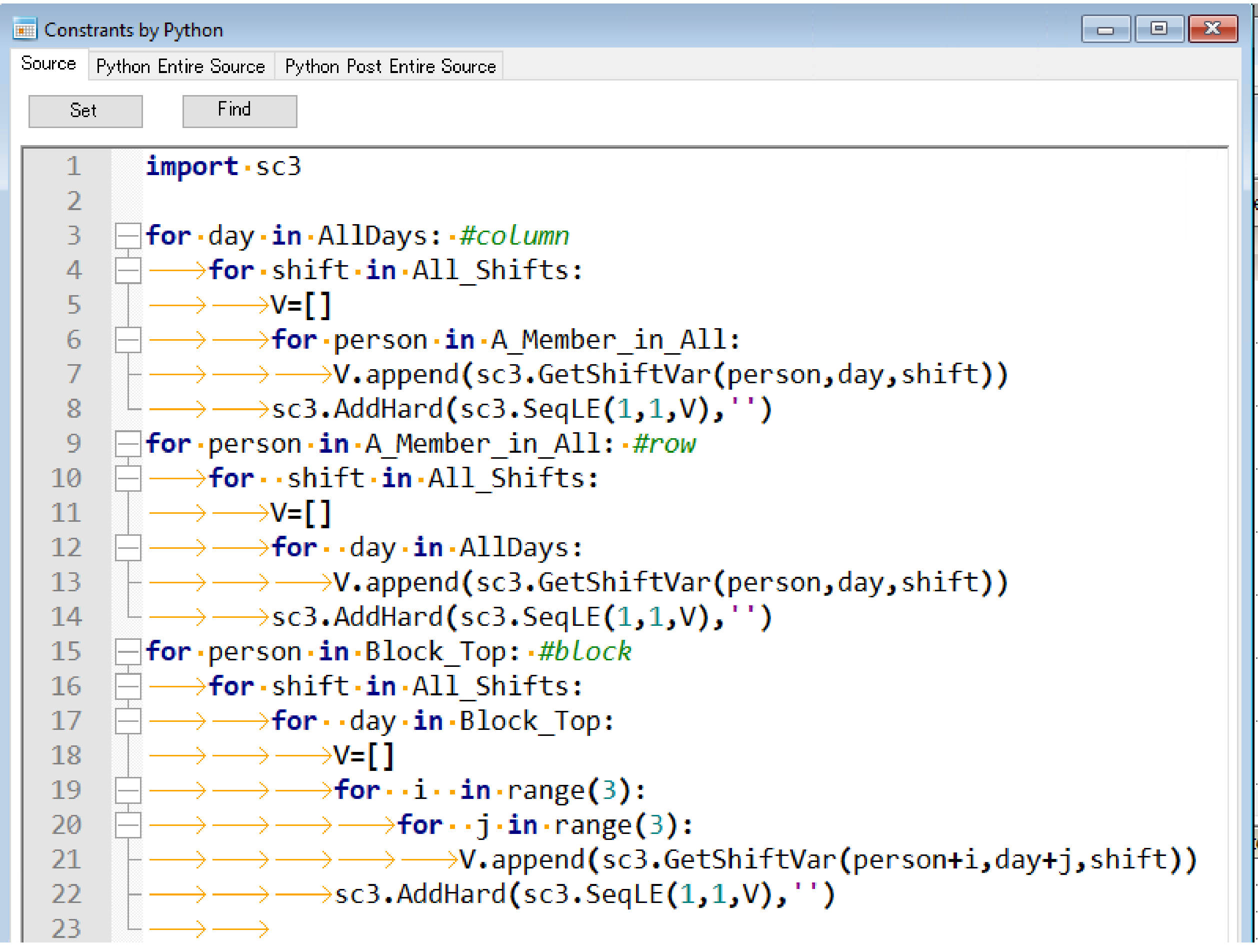
The List must be List of the shift/task variable, or the output of And/Or/Not, etc.



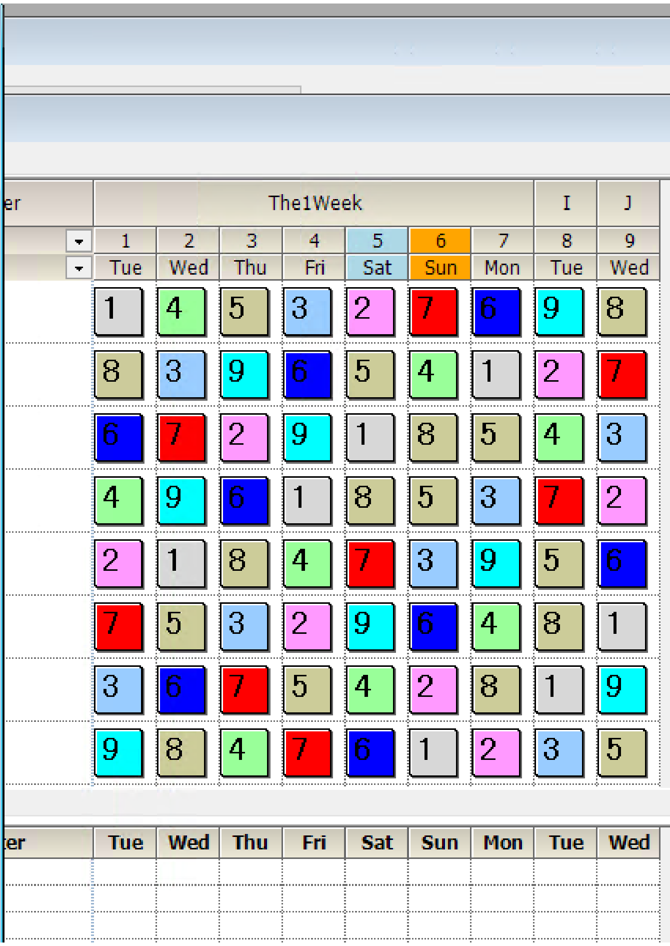
```
import sc3
import itertools

for person in A_Member_in_All:
    vlist=[]
    s='Python Paid Holiday'+' '+staffdef[person]+'\n'

    for day in Mon:
        v=sc3.GetShiftVar(person,day,'Paid_Holiday')
        vlist.append(v)
    sc3.AddHard(sc3.SeqLE(4,4,vlist),s)

```

sudoku

- We used cardinality constraints for the sudoku problem.
- The solver tries to get two solutions in this project, but there is only one solution, so we get a message that the solution2.txt does not exist.



```python
import sc3

for day in AllDays: #column
    for shift in All_Shifts:
        V=[]
        for person in A_Member_in_All:
            V.append(sc3.GetShiftVar(person,day,shift))
        sc3.AddHard(sc3.SeqLE(1,1,V),'')
for person in A_Member_in_All: #row
    for  shift in All_Shifts:
        V=[]
        for  day in AllDays:
            V.append(sc3.GetShiftVar(person,day,shift))
        sc3.AddHard(sc3.SeqLE(1,1,V),'')
for person in Block_Top: #block
    for shift in All_Shifts:
        for  day in Block_Top:
            V=[]
            for  i  in range(3):
                for  j in range(3):
                    V.append(sc3.GetShiftVar(person+i,day+j,shift))
            sc3.AddHard(sc3.SeqLE(1,1,V),'')
        
```



er	The1Week							I	J
	1	2	3	4	5	6	7	8	9
	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
	1	4	5	3	2	7	6	9	8
	8	3	9	6	5	4	1	2	7
	6	7	2	9	1	8	5	4	3
	4	9	6	1	8	5	3	7	2
	2	1	8	4	7	3	9	5	6
	7	5	3	2	9	6	4	8	1
	3	6	7	5	4	2	8	1	9
	9	8	4	7	6	1	2	3	5

ter	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed

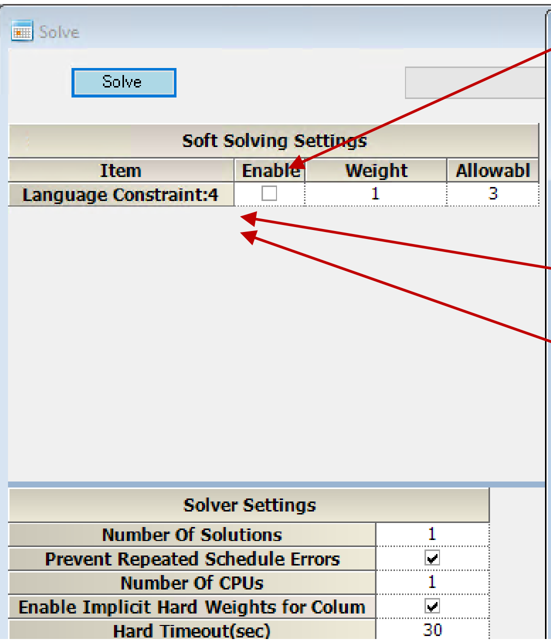
AddSoft python_tutorial7

■ The following is the format of adding a soft constraint.

sc3.AddSoft(variable ,string ,soft_level)

,where soft_level must be a constant(1-7).

■ Since the first time you add a soft level, a checkbox in solving parameters is empty; you need to select the checkbox so that the constraint becomes effective.

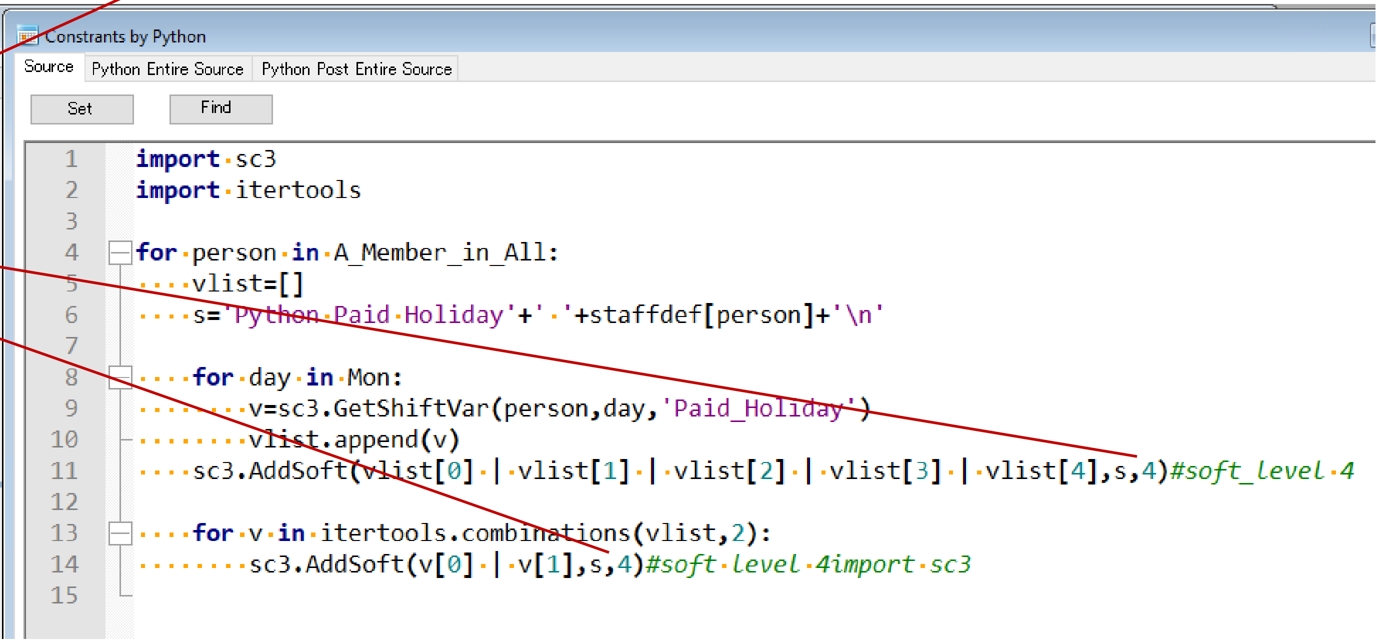


Soft Solving Settings

Item	Enable	Weight	Allowabl
Language Constraint:4	☐	1	3

Solver Settings

Number Of Solutions	1
Prevent Repeated Schedule Errors	✓
Number Of CPUs	1
Enable Implicit Hard Weights for Colum	✓
Hard Timeout(sec)	30



```
import sc3
import itertools

for person in A_Member_in_All:
    vlist=[]
    s='Python Paid Holiday'+' '+staffdef[person]+'\n'

    for day in Mon:
        v=sc3.GetShiftVar(person,day,'Paid_Holiday')
        vlist.append(v)
    sc3.AddSoft(vlist[0] | vlist[1] | vlist[2] | vlist[3] | vlist[4],s,4)#soft_level 4

    for v in itertools.combinations(vlist,2):
        sc3.AddSoft(v[0] | v[1],s,4)#soft level 4import sc3
```

SeqError python_tutorial8

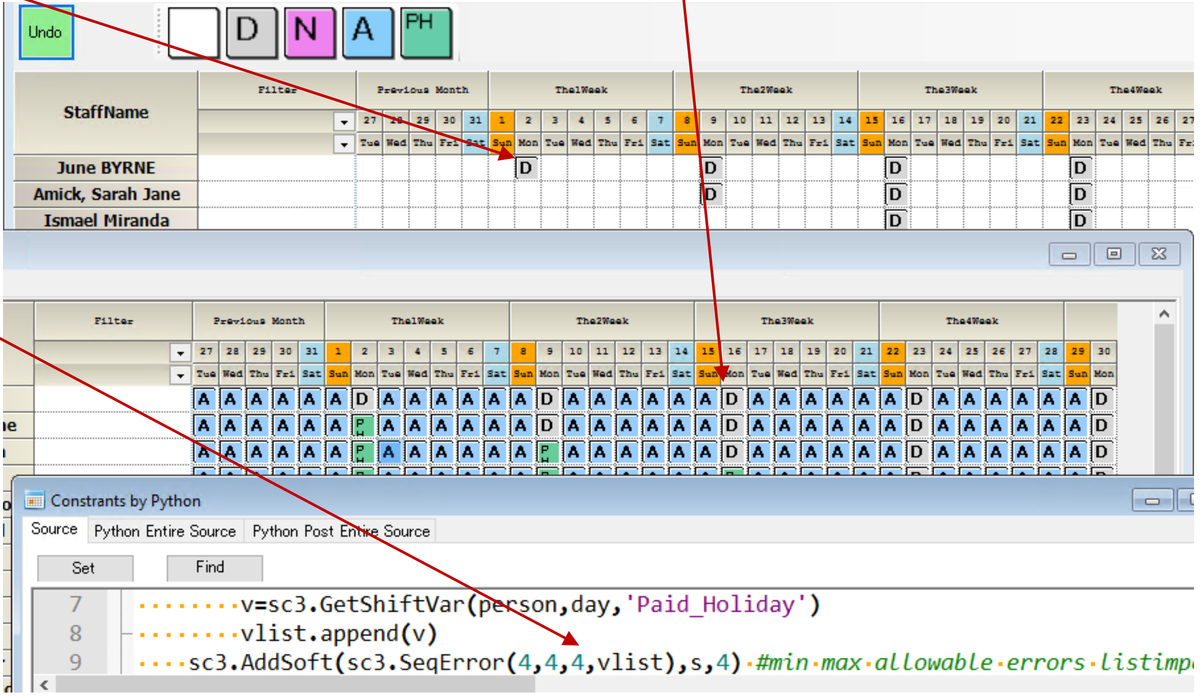
■ The following is the format when we use cardinality constraint as a soft constraint.

sc3.SeqError(min,max,allowable_errors,list)

■ Please note we use sc3.SeqLE for hard constraints instead of the above.

■ Let's look at the result of the project. We should have set it to 4 Paid_Holiday, but the reason it's not is that the we entered DayShifts as hard scheduled entries. A Hard constraint is always the winner for any soft rules.

■ The allowable_errors is the parameter of how many of these errors are allowed. Be careful to be set the number because over the limit should cause a hard error.



StaffName	Filter	Previous Month					The1Week							The2Week							The3Week							The4Week					
		27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
		Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fr
June BYRNE								D							D							D							D				
Amick, Sarah Jane															D							D							D				
Ismael Miranda																						D							D				

	Filter	Previous Month					The1Week							The2Week							The3Week							The4Week								
		27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
		A	A	A	A	A	A	D	A	A	A	A	A	A	D	A	A	A	A	A	A	D	A	A	A	A	A	A	D	A	A	A	A	A	A	D
ne		A	A	A	A	A	A	PH	A	A	A	A	A	A	D	A	A	A	A	A	A	D	A	A	A	A	A	A	D	A	A	A	A	A	A	D
		A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	D	A	A	A	A	A	A	D	A	A	A	A	A	A	D

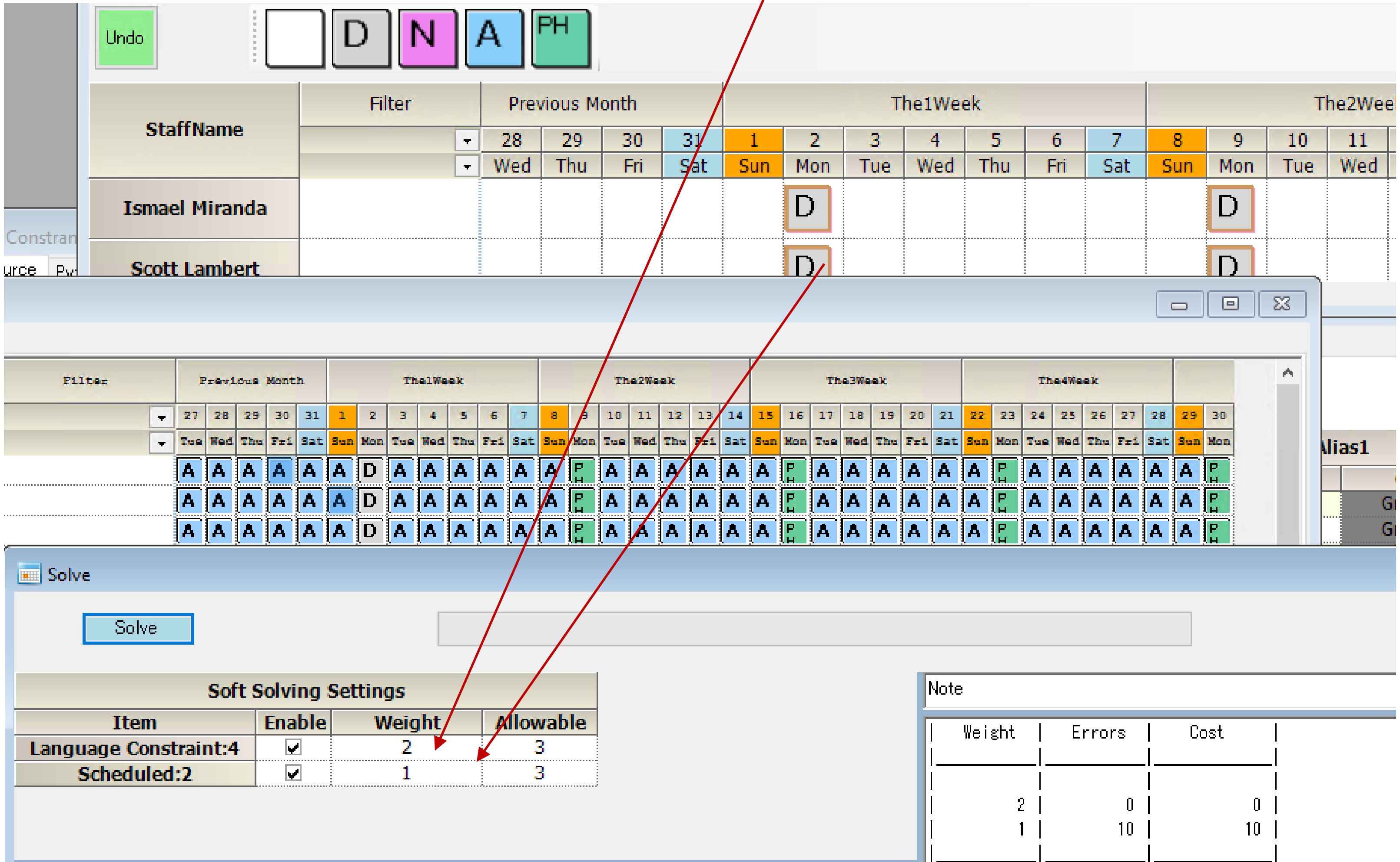
Constrants by Python

Source | Python Entire Source | Python Post Entire Source

```
        v=sc3.GetShiftVar(person,day,'Paid_Holiday')
        vlist.append(v)
    sc3.AddSoft(sc3.SeqError(4,4,4,vlist),s,4) #min max allowable errors listimp
```

SeqError python_tutorial9

■ When we make the weights of the schedule constraints larger than the cardinality constraints, there are no errors in the cardinality constraints. Instead, we should have a changed schedule due to weaker weight. In this way, we can change the priority without modifying the Python source code.



StaffName	Filter	28 Wed	29 Thu	30 Fri	31 Sat	1 Sun	2 Mon	3 Tue	4 Wed	5 Thu	6 Fri	7 Sat	8 Sun	9 Mon	10 Tue	11 Wed
		Previous Month				The1Week							The2Wee			
Ismael Miranda							D							D		
Scott Lambert							D							D		

Filter	27 Tue	28 Wed	29 Thu	30 Fri	31 Sat	1 Sun	2 Mon	3 Tue	4 Wed	5 Thu	6 Fri	7 Sat	8 Sun	9 Mon	10 Tue	11 Wed	12 Thu	13 Fri	14 Sat	15 Sun	16 Mon	17 Tue	18 Wed	19 Thu	20 Fri	21 Sat	22 Sun	23 Mon	24 Tue	25 Wed	26 Thu	27 Fri	28 Sat	29 Sun	30 Mon
	Previous Month					The1Week							The2Week							The3Week							The4Week								
	A	A	A	A	A	A	D	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH
	A	A	A	A	A	A	D	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH
	A	A	A	A	A	A	D	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH	A	A	A	A	A	A	PH

Soft Solving Settings

Item	Enable	Weight	Allowable
Language Constraint:4	✓	2	3
Scheduled:2	✓	1	3

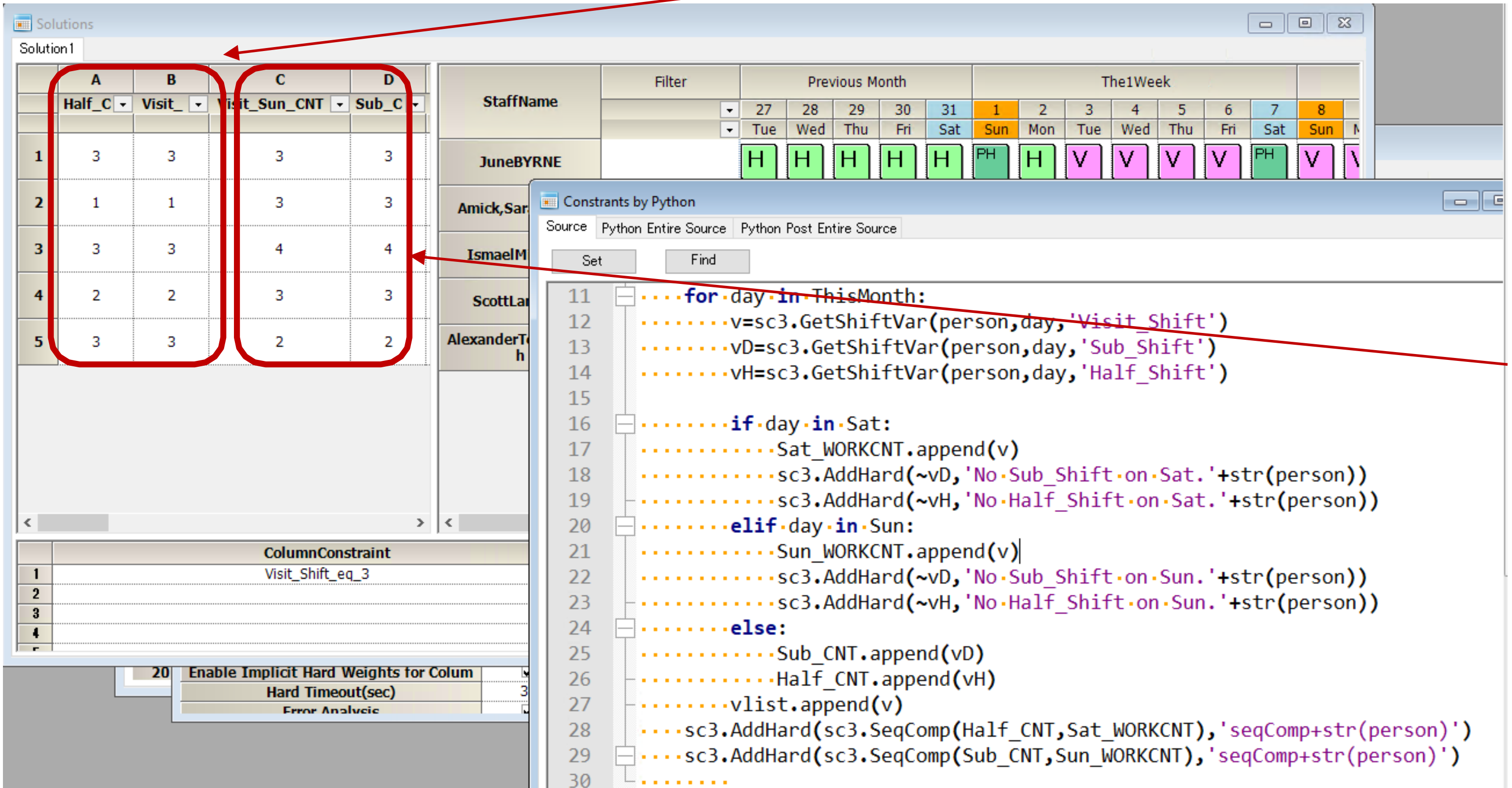
Note

```
| Weight   | Errors  | Cost    |
|__________|_________|_________|
|          |         |         |
|        2 |       0 |       0 |
|        1 |      10 |      10 |
```

SeqComp　python_tutorial10

■**sc3.SeqComp(X,Y)** is a function that returns True if ΣX(i)==ΣY(i).
When combined with AddHard, you can constraint ΣX(i) equals to ΣY(i).

The number of Saturday Visit-Shift equals to the number of Half-Shift for the month.



Solutions

Solution1

	A	B	C	D
	Half_C	Visit_	Visit_Sun_CNT	Sub_C
1	3	3	3	3
2	1	1	3	3
3	3	3	4	4
4	2	2	3	3
5	3	3	2	2

StaffName	Filter	Previous Month					The1Week							
		27	28	29	30	31	1	2	3	4	5	6	7	8
		Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
JuneBYRNE		H	H	H	H	H	PH	H	V	V	V	V	PH	V
Amick,Sar														
IsmaelM														
ScottLar														
AlexanderT h														

	ColumnConstraint
1	Visit_Shift_eq_3
2	
3	
4	

20	Enable Implicit Hard Weights for Colum
	Hard Timeout(sec)
	Error Analysis

Constraints by Python

Source　Python Entire Source　Python Post Entire Source

Set　Find

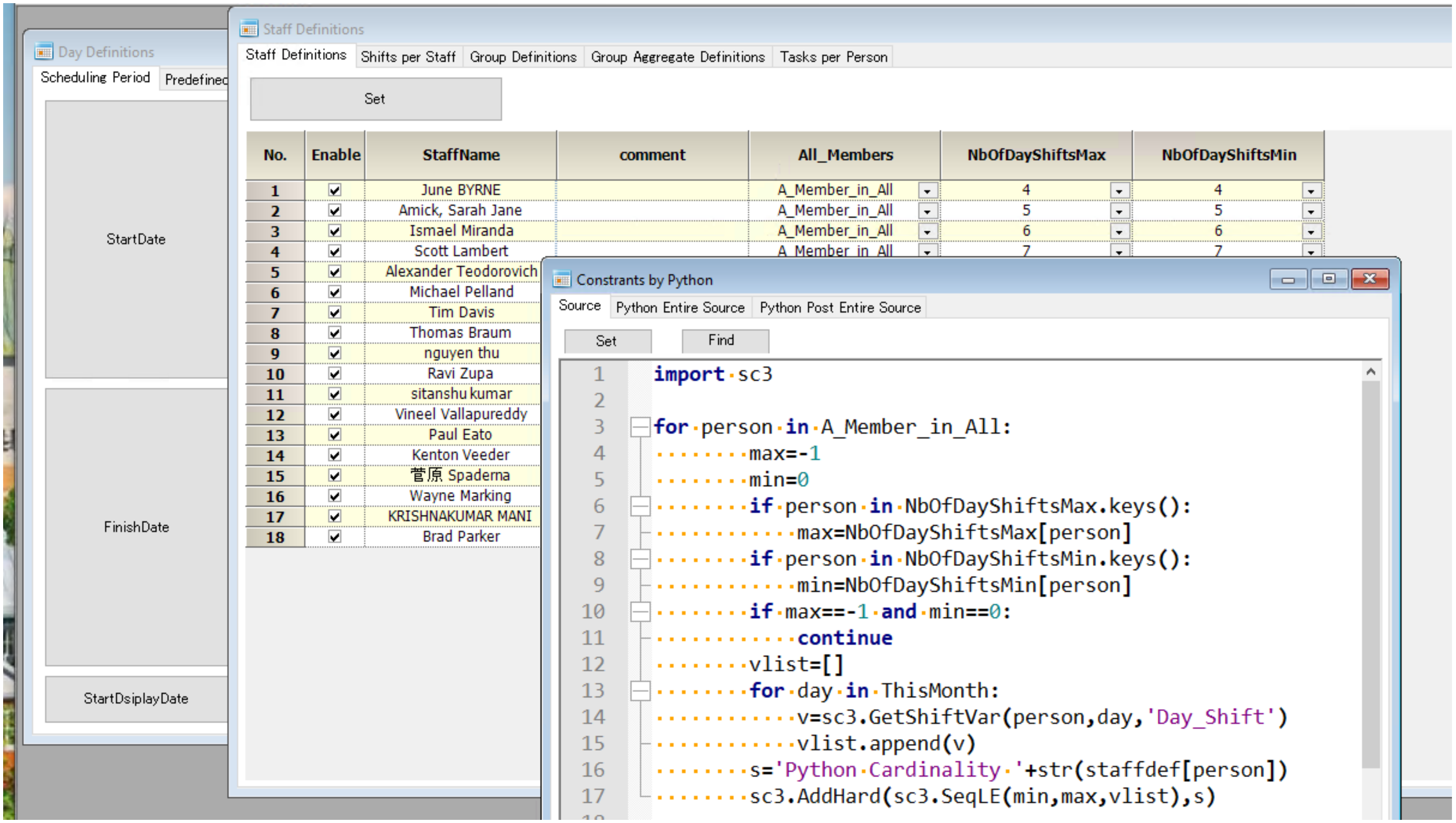
```
    for day in ThisMonth:
        v=sc3.GetShiftVar(person,day,'Visit_Shift')
        vD=sc3.GetShiftVar(person,day,'Sub_Shift')
        vH=sc3.GetShiftVar(person,day,'Half_Shift')

        if day in Sat:
            Sat_WORKCNT.append(v)
            sc3.AddHard(~vD,'No Sub_Shift on Sat.'+str(person))
            sc3.AddHard(~vH,'No Half_Shift on Sat.'+str(person))
        elif day in Sun:
            Sun_WORKCNT.append(v)
            sc3.AddHard(~vD,'No Sub_Shift on Sun.'+str(person))
            sc3.AddHard(~vH,'No Half_Shift on Sun.'+str(person))
        else:
            Sub_CNT.append(vD)
            Half_CNT.append(vH)
        vlist.append(v)
    sc3.AddHard(sc3.SeqComp(Half_CNT,Sat_WORKCNT),'seqComp+str(person)')
    sc3.AddHard(sc3.SeqComp(Sub_CNT,Sun_WORKCNT),'seqComp+str(person)')
```

The number of Sunday Visit-Shift is equal to the number of Sub-Shift for the month.

python_tutorial13　cardinality constraint per staff

■You can retrieve any staff property that consists of only numbers as a Python Dictionary.

■For example ,we set the number of DayShifts per each staff based on the dictionary on Python as figure below.

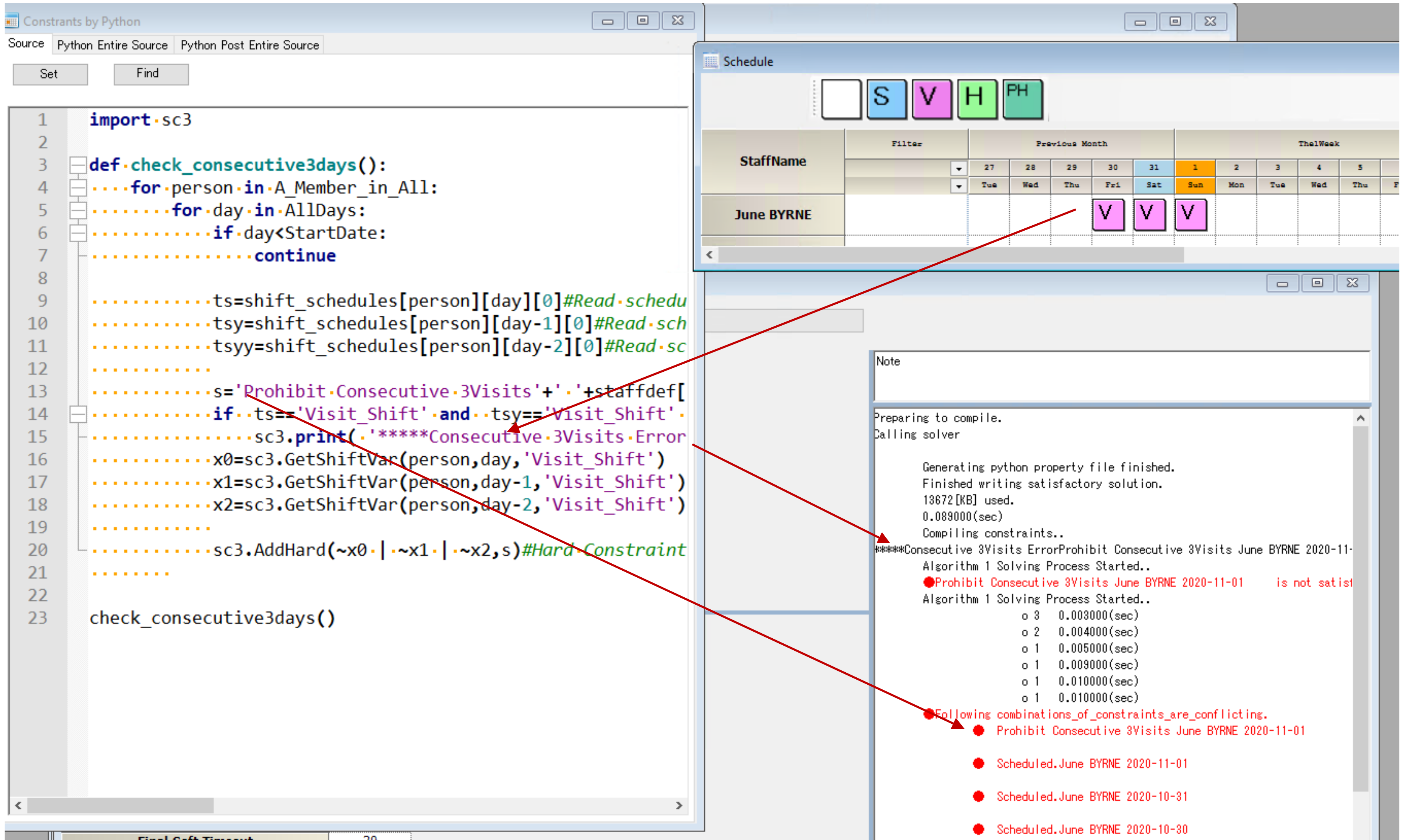


No.	Enable	StaffName	comment	All_Members	NbOfDayShiftsMax	NbOfDayShiftsMin
1	✓	June BYRNE		A_Member_in_All	4	4
2	✓	Amick, Sarah Jane		A_Member_in_All	5	5
3	✓	Ismael Miranda		A_Member_in_All	6	6
4	✓	Scott Lambert		A_Member_in_All	7	7
5	✓	Alexander Teodorovich				
6	✓	Michael Pelland				
7	✓	Tim Davis				
8	✓	Thomas Braum				
9	✓	nguyen thu				
10	✓	Ravi Zupa				
11	✓	sitanshu kumar				
12	✓	Vineel Vallapureddy				
13	✓	Paul Eato				
14	✓	Kenton Veeder				
15	✓	菅原 Spaderna				
16	✓	Wayne Marking				
17	✓	KRISHNAKUMAR MANI				
18	✓	Brad Parker				

```python
import sc3

for person in A_Member_in_All:
        max=-1
        min=0
        if person in NbOfDayShiftsMax.keys():
            max=NbOfDayShiftsMax[person]
        if person in NbOfDayShiftsMin.keys():
            min=NbOfDayShiftsMin[person]
        if max==-1 and min==0:
            continue
        vlist=[]
        for day in ThisMonth:
            v=sc3.GetShiftVar(person,day,'Day_Shift')
            vlist.append(v)
        s='Python Cardinality '+str(staffdef[person])
        sc3.AddHard(sc3.SeqLE(min,max,vlist),s)
```

python_tutorial11_error

■ This is an example of reading scheduled shifts and showing an error before the solver issues the hard error.



```
import sc3

def check_consecutive3days():
    for person in A_Member_in_All:
        for day in AllDays:
            if day<StartDate:
                continue

            ts=shift_schedules[person][day][0]#Read schedu
            tsy=shift_schedules[person][day-1][0]#Read sch
            tsyy=shift_schedules[person][day-2][0]#Read sc
            
            s='Prohibit Consecutive 3Visits'+' '+staffdef[
            if  ts=='Visit_Shift' and  tsy=='Visit_Shift' 
                sc3.print( '*****Consecutive 3Visits Error
            x0=sc3.GetShiftVar(person,day,'Visit_Shift')
            x1=sc3.GetShiftVar(person,day-1,'Visit_Shift')
            x2=sc3.GetShiftVar(person,day-2,'Visit_Shift')
            
            sc3.AddHard(~x0 | ~x1 | ~x2,s)#Hard Constraint
        

check_consecutive3days()
```

StaffName	Filter	Previous Month					TheIWeek				
		27	28	29	30	31	1	2	3	4	5
		Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu
June BYRNE					V	V	V				

Note

```
Preparing to compile.
Calling solver

        Generating python property file finished.
        Finished writing satisfactory solution.
        13672[KB] used.
        0.089000(sec)
        Compiling constraints..
*****Consecutive 3Visits ErrorProhibit Consecutive 3Visits June BYRNE 2020-11-
        Algorithm 1 Solving Process Started..
        ●Prohibit Consecutive 3Visits June BYRNE 2020-11-01      is not satisf
        Algorithm 1 Solving Process Started..
                        o 3   0.003000(sec)
                        o 2   0.004000(sec)
                        o 1   0.005000(sec)
                        o 1   0.009000(sec)
                        o 1   0.010000(sec)
                        o 1   0.010000(sec)
        ●Following combinations_of_constraints_are_conflicting.
            ●  Prohibit Consecutive 3Visits June BYRNE 2020-11-01

            ●  Scheduled.June BYRNE 2020-11-01

            ●  Scheduled.June BYRNE 2020-10-31

            ●  Scheduled.June BYRNE 2020-10-30
```

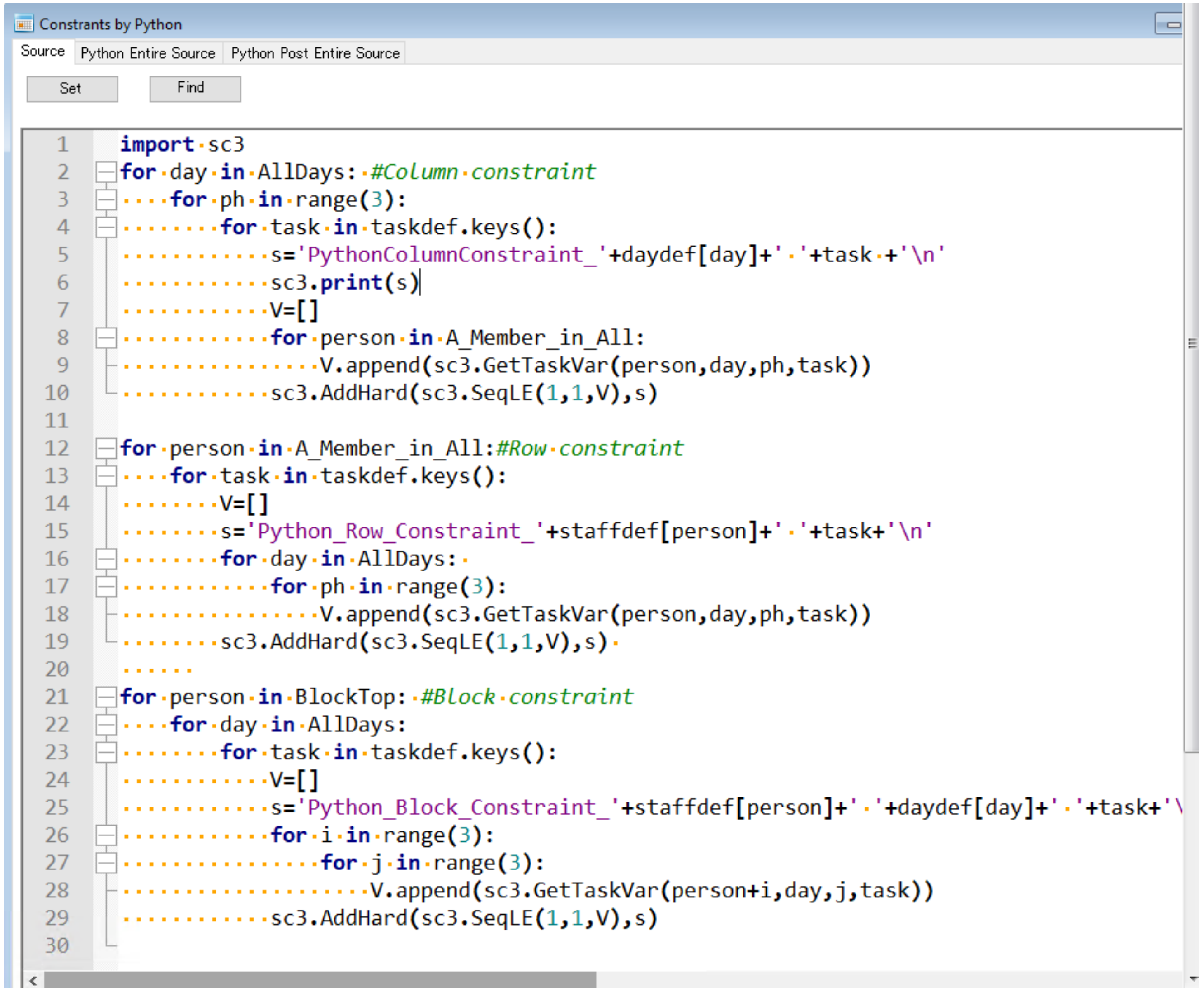
GetTaskVar sudoku_task

■Use GetTaskVar instead of GetShiftVar in phase mode.

■The example is sudoku_task

■The format is as follows.

GetTaskVar(person,day ,phase,task)



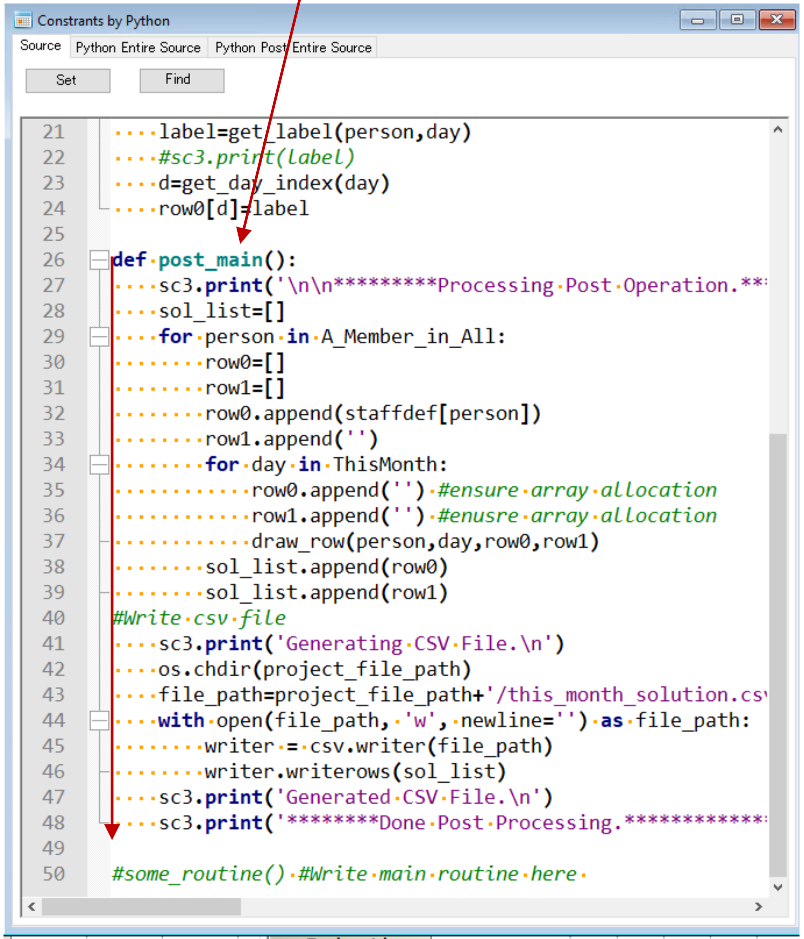
```
import sc3
for day in AllDays: #Column constraint
    for ph in range(3):
        for task in taskdef.keys():
            s='PythonColumnConstraint_'+daydef[day]+' '+task +'\n'
            sc3.print(s)
            V=[]
            for person in A_Member_in_All:
                V.append(sc3.GetTaskVar(person,day,ph,task))
            sc3.AddHard(sc3.SeqLE(1,1,V),s)

for person in A_Member_in_All:#Row constraint
    for task in taskdef.keys():
        V=[]
        s='Python_Row_Constraint_'+staffdef[person]+' '+task+'\n'
        for day in AllDays: 
            for ph in range(3):
                V.append(sc3.GetTaskVar(person,day,ph,task))
        sc3.AddHard(sc3.SeqLE(1,1,V),s) 
      
for person in BlockTop: #Block constraint
    for day in AllDays:
        for task in taskdef.keys():
            V=[]
            s='Python_Block_Constraint_'+staffdef[person]+' '+daydef[day]+' '+task+'\
            for i in range(3):
                for j in range(3):
                    V.append(sc3.GetTaskVar(person+i,day,j,task))
            sc3.AddHard(sc3.SeqLE(1,1,V),s)
```

Post Processing

■After solution generation, for example, you should write the following when you want to output a formatted solution.

■The example is Task_Projects/task_import1.nurse3.

Define def post_main(): in the source with the following order.

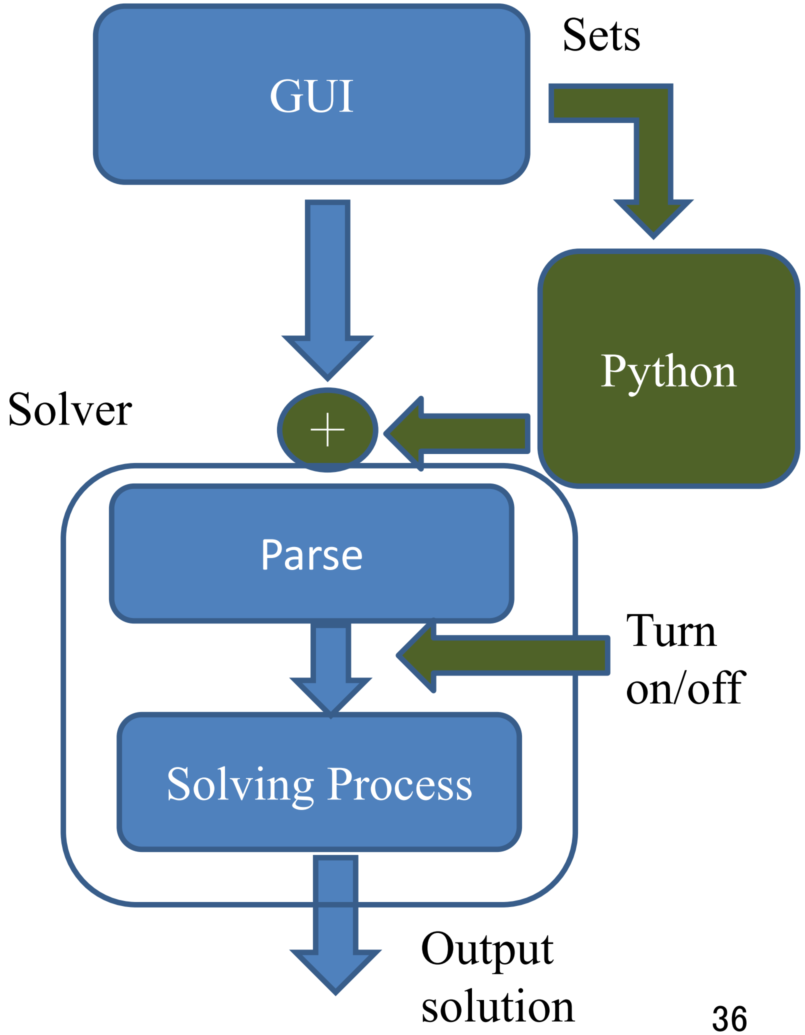


```
    label=get_label(person,day)
    #sc3.print(label)
    d=get_day_index(day)
    row0[d]=label

def post_main():
    sc3.print('\n\n*********Processing Post Operation.**
    sol_list=[]
    for person in A_Member_in_All:
        row0=[]
        row1=[]
        row0.append(staffdef[person])
        row1.append('')
        for day in ThisMonth:
            row0.append('') #ensure array allocation
            row1.append('') #enusre array allocation
            draw_row(person,day,row0,row1)
        sol_list.append(row0)
        sol_list.append(row1)
#Write csv file
    sc3.print('Generating CSV File.\n')
    os.chdir(project_file_path)
    file_path=project_file_path+'/this_month_solution.cs
    with open(file_path, 'w', newline='') as file_path:
        writer = csv.writer(file_path)
        writer.writerows(sol_list)
    sc3.print('Generated CSV File.\n')
    sc3.print('********Done Post Processing.************

#some_routine() #Write main routine here
```

Teodorovich

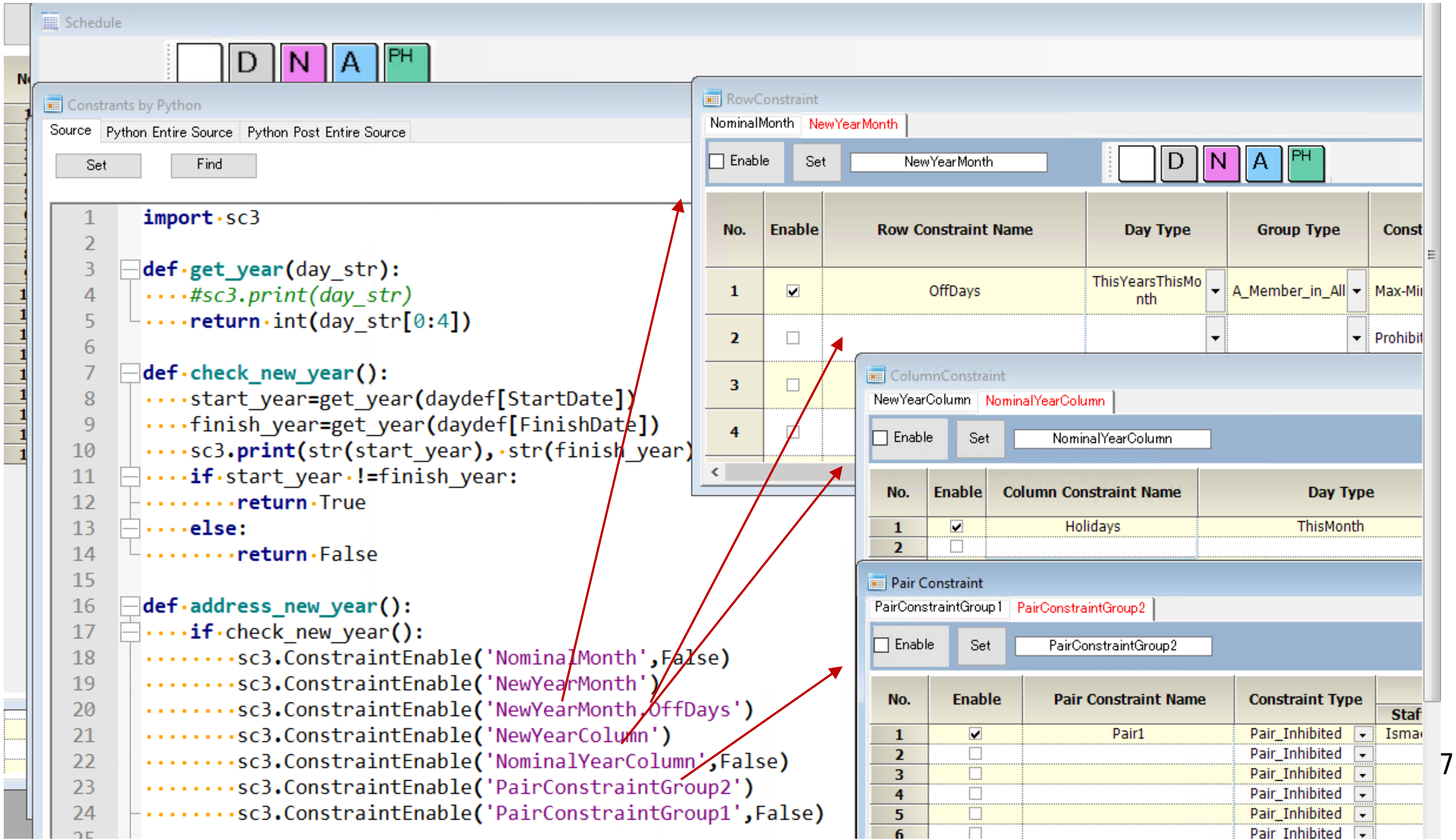
Constraints dynamic on/off

■We can turn on/off a constraint or a constraint group dynamically.
■The timing for turning on/off is right after the parsing in the solver



Constraint group on/off python_tutorial12

- sc3.ConstraintEnable(name) # name is groupname or groupname . Item_name
- sc3.ConstraintEnable(name, enable)



```
import sc3

def get_year(day_str):
    #sc3.print(day_str)
    return int(day_str[0:4])

def check_new_year():
    start_year=get_year(daydef[StartDate])
    finish_year=get_year(daydef[FinishDate])
    sc3.print(str(start_year), str(finish_year)
    if start_year !=finish_year:
        return True
    else:
        return False

def address_new_year():
    if check_new_year():
        sc3.ConstraintEnable('NominalMonth',False)
        sc3.ConstraintEnable('NewYearMonth')
        sc3.ConstraintEnable('NewYearMonth.OffDays')
        sc3.ConstraintEnable('NewYearColumn')
        sc3.ConstraintEnable('NominalYearColumn',False)
        sc3.ConstraintEnable('PairConstraintGroup2')
        sc3.ConstraintEnable('PairConstraintGroup1',False)
```

No.	Enable	Row Constraint Name	Day Type	Group Type	Const
1	✓	OffDays	ThisYearsThisMonth	A_Member_in_All	Max-Mi
2	☐				Prohibi
3	☐				
4	☐				

No.	Enable	Column Constraint Name	Day Type
1	✓	Holidays	ThisMonth
2	☐		

No.	Enable	Pair Constraint Name	Constraint Type	Staf
1	✓	Pair1	Pair_Inhibited	Isma
2	☐		Pair_Inhibited	
3	☐		Pair_Inhibited	
4	☐		Pair_Inhibited	
5	☐		Pair_Inhibited	
6	☐		Pair_Inhibited	

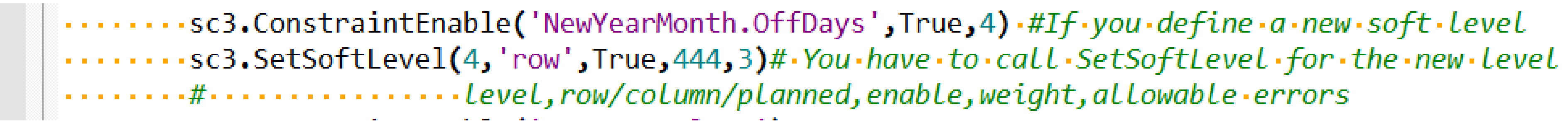
Constraint group on/off with soft level python_tutorial12_soft_le

■**sc3.ConstraintEnable(name, enable,soft_level)**

■**sc3.SetSoftLeveltraintEnable(soft_level,type, enable,weight,allowable_errors)**

type:’row’,’column’,’planned’

■If you define a new level in ConstraintEnable, you should also call SetSoftLevel because the solver doesn't know what weight should use on the constraint.



```
        sc3.ConstraintEnable('NewYearMonth.OffDays',True,4) #If you define a new soft level
        sc3.SetSoftLevel(4,'row',True,444,3)# You have to call SetSoftLevel for the new level
        #                level,row/column/planned,enable,weight,allowable errors
```

Python Constraint Function Summary

	Format	Description	Can be used in	Return
SeqLE	SeqLE(min,max,List)	Hard　Cardinality	AddHard	
SeqError	SeqError(min,max,allowable erros,List)	Soft Cardinality	AddSoft	
SeqComp	SeqComp(ListA,ListB)	ΣListA==ΣListB		1 bit
SeqExpr	SeqExpr(min,max,Type、List)	Type 0:ΣList <=max 1:ΣList>=min 2:Σlist<=max && Σlist>=min		1 bit