



Artificial Intelligence

Exam activity 2019-2020

Name: Trif Gheorghe Andrei

Group: 30235

Email: trifandrei@yahoo.com

Assoc. Prof. dr. eng. Adrian Groza

Adrian.Groza@cs.utcluj.ro



Contents

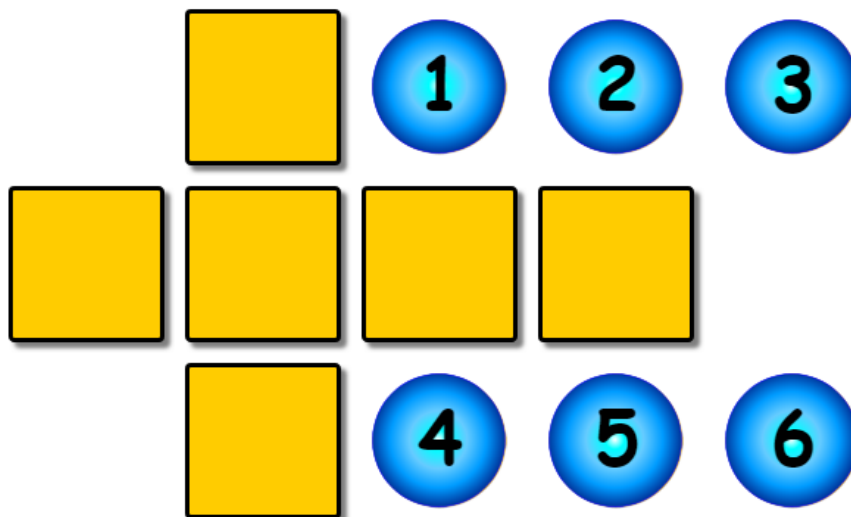
1	Puzzle 1	3
1.1	Problem definition	3
1.2	Implementation	5
1.3	Solutions	7
2	Puzzle 2	10
2.1	Problem definition	10
2.2	Implemetation	11
2.3	Solutions	12
3	Puzzle 3	13
3.1	Problem definition	13
3.2	Implementation	14
3.3	Solutions	15
4	Puzzle 4	16
4.1	Problem definition	16
4.2	Implementation	17
4.3	Solutions	18
5	Puzzle 5	19
5.1	Problem definition	19
5.2	Implementation	20
5.3	Solutions	21

Chapter 1

Puzzle 1

1.1 Problem definition

The first puzzle I chose was Place Numbers. This is a puzzle with many levels. In this puzzle a player must arrange the numbers on the yellow tiles so that the sum of the numbers in the vertical line is the same as the sum of the numbers in the horizontal line at level 1.



In the implementation section is the code for all 8 levels. All goals are exemplified below:

Level 1 the sum of the numbers in the vertical line is the same as the sum of the numbers in the horizontal.

Level 2 the sum of the three numbers in the vertical line is two more than the sum of the four numbers in the horizontal line.

Level 3 the sum of the number in the horizontal line is one more than the sum of the number in the vertical line.

Level 4 arrange the numbers on the yellow tiles so that the product on vertical is equal with product on horizontal.

Level 5 arrange the numbers on the yellow tiles so that the horizontal line is ten bigger than the vertical line.

Level 6 arrange the numbers on the yellow tiles so that the sum on horizontal line is 3 times bigger than the vertical line.

Level 7 arrange the numbers on the yellow tiles so that the sum on horizontal line is half pf the sum than the vertical line.

Level 8 Arrange the numbers on the yellow tiles so that the vertical total multiplied by the center number equals the horizontal total.

1.2 Implementation

```
set ( arithmetic ).
assign ( domain_size , 7 ). % number [ 1 .. 6 ]
```

```
list ( distinct ).
[ g1 , g2 , g3 , g4 , g5 , g6 ].
end_of_list .
```

```
formulas ( assumptions ).
```

```
%The vertical line is formed by the grids g1,g3,g6.
%The horizontal line is formed by the grids g2,g3,g4,g5.
%The grids can't take 0 as value
```

```
( g1=1 | g1=2 | g1=3 | g1=4 | g1=5 | g1=6 ).
( g2=1 | g2=2 | g2=3 | g2=4 | g2=5 | g2=6 ).
( g3=1 | g3=2 | g3=3 | g3=4 | g3=5 | g3=6 ).
( g4=1 | g4=2 | g4=3 | g4=4 | g4=5 | g4=6 ).
( g5=1 | g5=2 | g5=3 | g5=4 | g5=5 | g5=6 ).
( g6=1 | g6=2 | g6=3 | g6=4 | g6=5 | g6=6 ).
```

```
%Level 1
%sum on the vline=sum on the hline.
%g1+g3+g6=g2+g3+g4+g5.
```

```
%Level 2
%the sum on the horizontal line is =2*sum of the vertical.
%2*(g1+g3+g6)=g2+g3+g4+g5.
```

```
%Level 3
%The sum of the horizontal line is 1 more then the sum of the vertical.
%g2+g3+g4+g5=g1+g3+g6+1.
```

```
%Level 4
%product in the horizontal line is equal with the product in vertical.
%g1*g3*g6=g2*g3*g4*g5.
```

```
%Level 5
```

```
%The sum of the horizontal line is ten bigger then the vertical  
% $g_2+g_3+g_4+g_5=g_1+g_3+g_6+10$ .
```

```
%Level 6
```

```
%The sum of the numbers in horizontal line is 3 times the sum o  
% $g_2+g_3+g_4+g_5=(g_1+g_3+g_6)*3$ .
```

```
%Level 7
```

```
%The sum of the number in the vertical line is half of the sum o  
% $(g_2+g_3+g_4+g_5) \bmod 2 = 0$ .  
% $(g_2+g_3+g_4+g_5)/2=g_1+g_3+g_6$ .
```

```
%Level 8
```

```
%Arrange the numbers on the yellow tiles so that the vertical to  
% $(g_2+g_3+g_4+g_5)=(g_1+g_3+g_6)*g_3$ .
```

```
end_of_list.
```

NOTE!: To execute a level in mace4 just uncomment the level and comment level 8. Use command `mace4 -m -1 -f input.in` to find all models and `mace4 -c -f input.in` to generate just one model.

1.3 Solutions

Level 1: In this level mace4 generated 36 models. Here are some solutions for this level.

Models	G1	G2	G3	G4	G5	G6
1	2	1	5	3	4	6
2	4	2	3	1	6	5
3	5	1	3	6	2	4
4	6	1	5	4	3	2
5	6	5	1	3	2	4

Level 2: Here are generated 24 models.

Models	G1	G2	G3	G4	G5	G6
1	2	4	6	3	5	1
2	4	5	3	2	6	1
3	1	6	3	5	2	4
4	1	6	3	2	5	4
5	2	5	6	3	4	1

Level 3: Here are generated 72 models.

Models	G1	G2	G3	G4	G5	G6
1	5	4	6	1	3	2
2	6	1	2	4	5	3
3	6	1	4	5	3	2
4	4	1	6	2	5	3
5	3	6	4	1	2	5

Level 4: Here are generated 24 models.

Models	G1	G2	G3	G4	G5	G6
1	6	4	5	1	3	2
2	4	6	5	2	1	3
3	4	2	5	1	6	3
4	3	6	5	2	1	4
5	3	2	5	6	1	4

Level 5: Here are generated 24 models.

Models	G1	G2	G3	G4	G5	G6
1	3	4	1	5	6	2
2	2	6	1	5	4	3
3	2	5	1	6	4	3
4	2	4	5	6	3	1
5	3	4	1	6	5	2

Level 6: Here are generated 12 models.

Models	G1	G2	G3	G4	G5	G6
1	2	5	3	6	4	1
2	2	4	3	6	5	1
3	1	6	3	5	4	2
4	1	5	3	6	4	2
5	1	4	3	6	5	2

Level 7: Here are generated 24 models.

Models	G1	G2	G3	G4	G5	G6
1	4	5	3	6	2	1
2	4	2	3	6	5	1
3	2	5	6	4	3	1
4	2	4	6	3	5	1
5	1	4	3	5	2	4

Level 8: Here are generated 24 models.

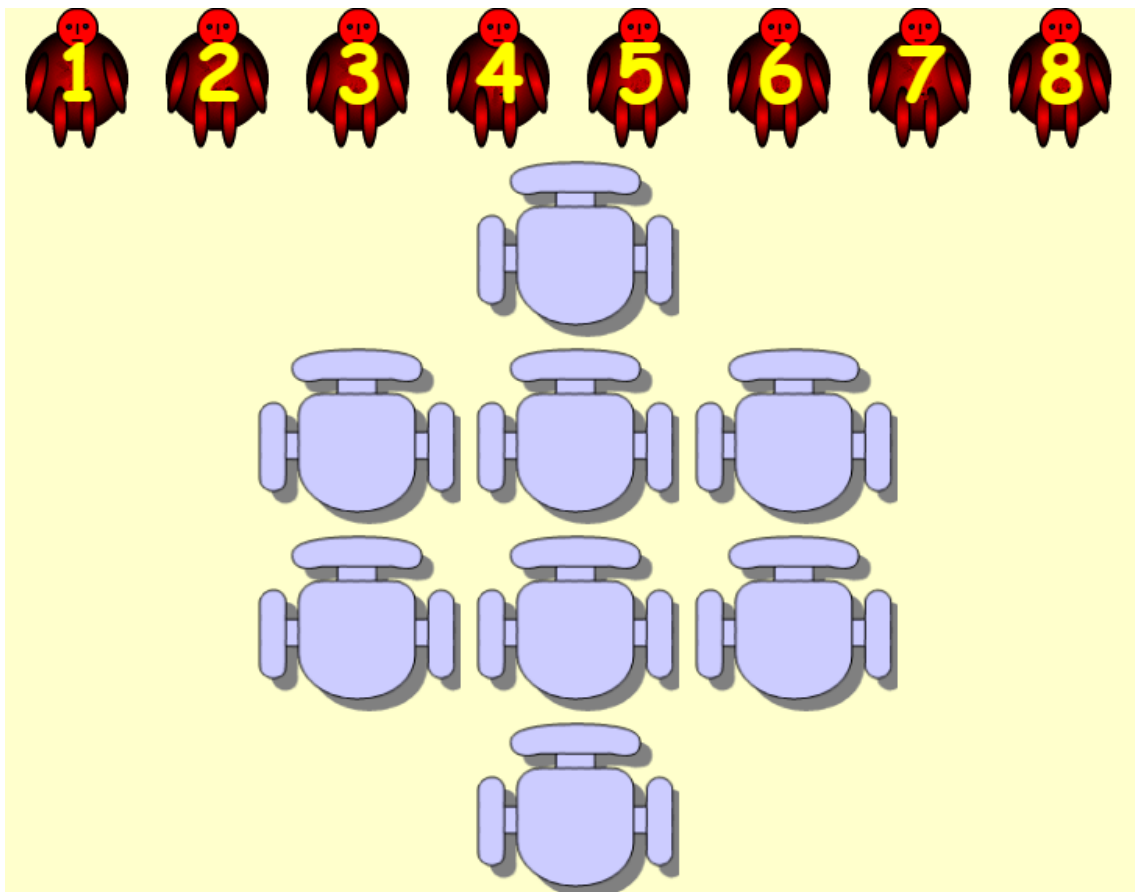
Models	G1	G2	G3	G4	G5	G6
1	6	5	1	3	2	4
2	4	5	1	3	5	6
3	4	2	1	3	5	6
4	2	4	3	6	5	1
5	1	6	3	5	4	2

Chapter 2

Puzzle 2

2.1 Problem definition

In the puzzle no.2 a player must arrange the students numbered 1 to 8 on a chairs so that no two consecutively numbered students sit next to each other either vertically, horizontally or diagonally.



In this puzzle is just one level .But the most difficult thing on this puzzle is the fact that no tow consecutively numbered sit next to each other.

2.2 Implemetation

```
set(arithmetic).
assign(domain_size,9). % 9 numbers

list(distinct).
[0,g1,g2,g3,g4,g5,g6,g7,g8]. % The grids
end_of_list.
formulas(assumptions).
% If grid 1 is adjacent to grid 2, then grid 2 is also adjacent

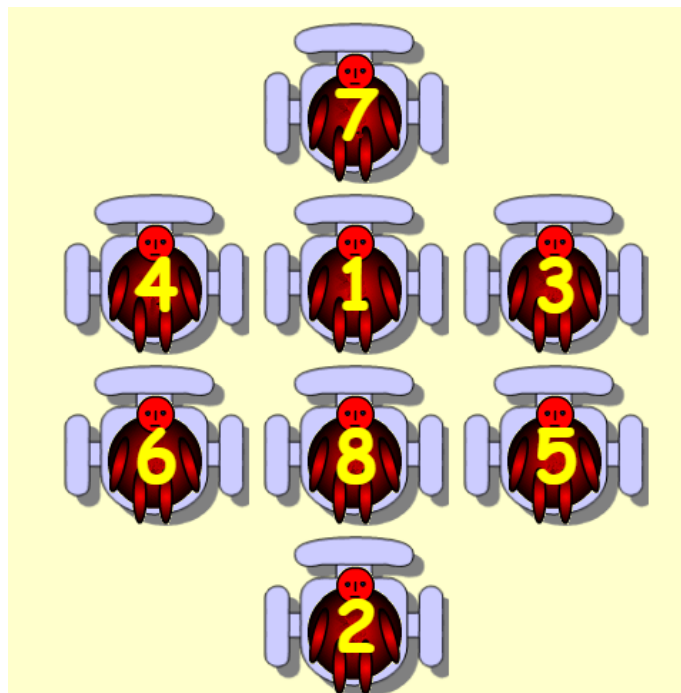
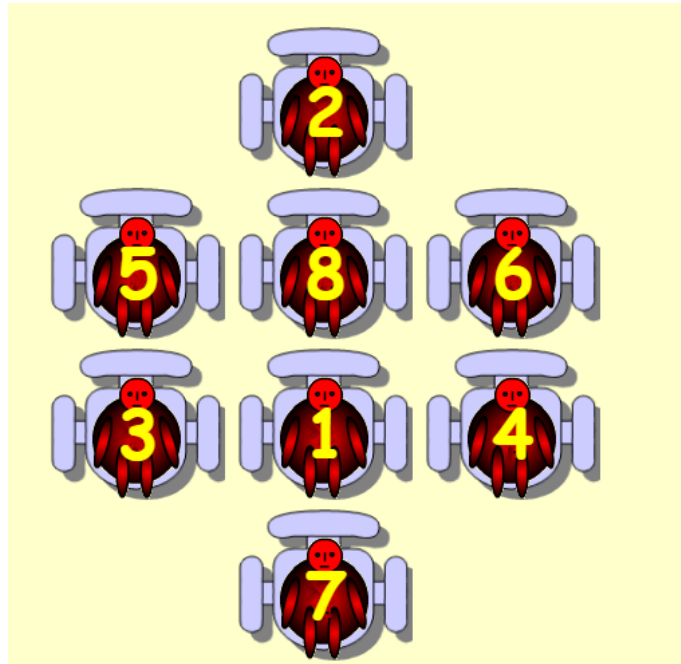
adjacent(x,y) <=> adjacent(y,x).
% We check if the number on two adjacent grids are not consecutive

adjacent(x,y) => abs(x - y) != 1.
% Marking the adjacent grids (diagonally, vertically and horizontally)
% g1
% g2g3g4
% g5g6g7
% g8
adjacent(g1,g2).
adjacent(g1,g3).
adjacent(g1,g4).
adjacent(g2,g3).
adjacent(g2,g5).
adjacent(g2,g6).
adjacent(g3,g4).
adjacent(g3,g5).
adjacent(g3,g6).
adjacent(g3,g7).
adjacent(g4,g6).
adjacent(g4,g7).
adjacent(g5,g6).
adjacent(g5,g8).
adjacent(g6,g7).
adjacent(g6,g8).
adjacent(g7,g8).

end_of_list.
```

2.3 Solutions

NOTE: use command `mace4 -c -f puzzle2.in`. The model generated by mace4 should be 1 of the solutions. Here i put as an example 2 solutions.

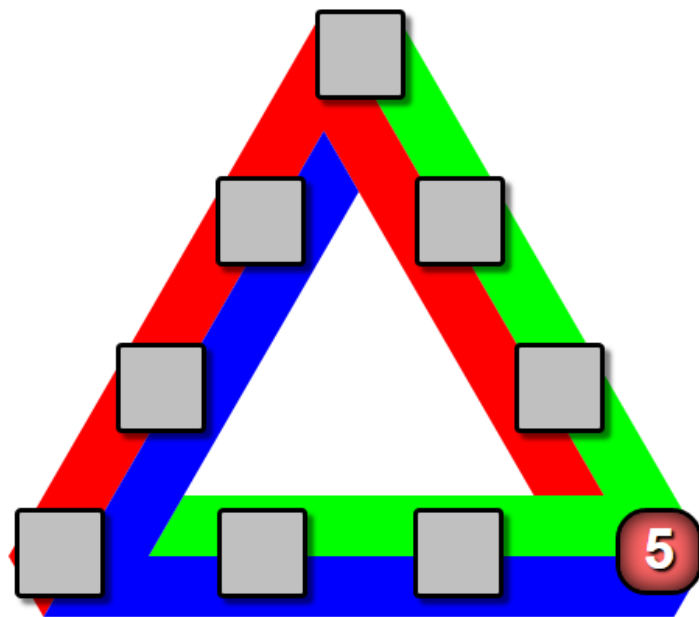


Chapter 3

Puzzle 3

3.1 Problem definition

In this puzzle is a triangle like in the image below and we must arrange the number from 1 to 9 so that the sum of each line to be equal with 20. In this puzzle i will add a general level to complicate more this puzzle.



In the general level the box with 5 in it will be empty and the condition for each line will be $L1=L2=L3$.

3.2 Implementation

```
set(arithmetic).
assign(domain_size,10). % 9 numbers

%For level 5
%list(distinct).
%[0,g1,g2,g3,g4,g5,g6,5,g8,g9]. % The grids for each line

%For general level.
list(distinct).
[0,g1,g2,g3,g4,g5,g6,g7,g8,g9].

end_of_list.

formulas(assumptions).

%Level 5

%g1+g2+g3+g4=20.
%g4+g5+g6+5=20.
%5+g8+g9+g1=20.

%General Level

g1+g2+g3+g4=g4+g5+g6+g7.
g7+g8+g9+g1=g4+g5+g6+g7.

end_of_list.
```

NOTE!: use commands `mace4 -m -l -f puzzle3.in` and `mace4 -c -f puzzle3.in` to run the file .

3.3 Solutions

In the general level are 864 models. Here are some of them.

Model	G1	G2	G3	G4	G5	G6	G7	G8	G9
1	9	8	1	3	7	5	6	4	2
2	8	3	4	5	6	7	2	1	9
3	7	9	2	3	4	6	8	1	5
4	7	4	2	8	1	9	3	5	6
5	5	8	4	3	1	9	7	6	2

At level 5 mace4 found 96 models. But in this level g7 have a fixed value .

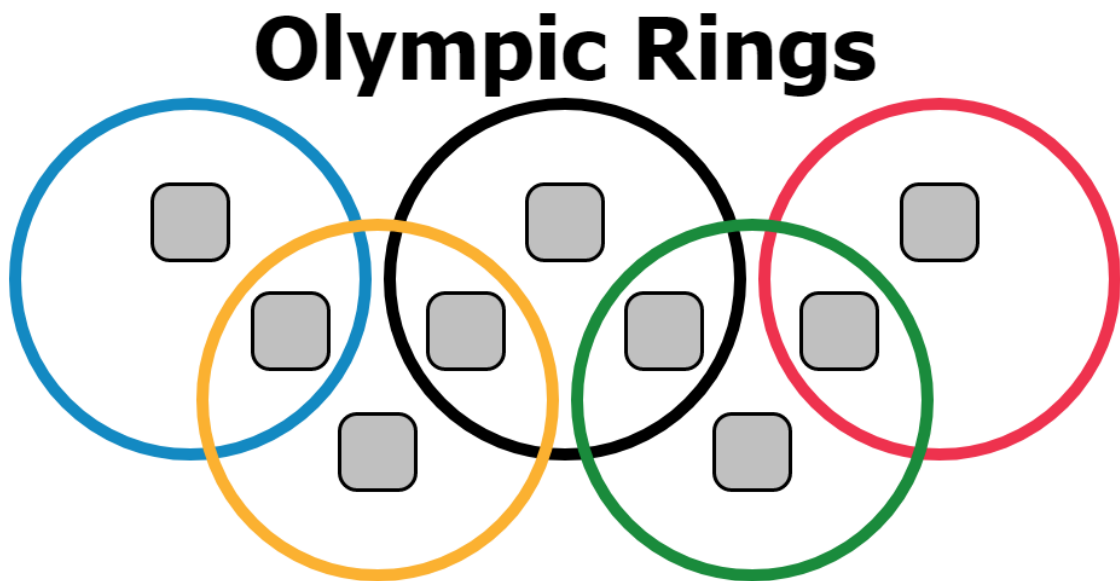
Model	G1	G2	G3	G4	G5	G6	G7	G8	G9
1	9	7	3	1	8	6	5	2	4
2	6	7	3	4	9	2	5	1	8
3	6	3	7	4	9	2	5	8	1
4	4	9	1	6	2	7	5	8	3
5	3	9	1	7	2	6	5	8	4

Chapter 4

Puzzle 4

4.1 Problem definition

In this puzzle there are 5 rings. Each ring has some boxes. We must fill the boxes with numbers from 1 to 9. Each box can contain a different number. The goal is the sum of each box in a ring is the same. Here are 2 levels: Level 3 where the sum of each ring is 14. Level 4 here the sum of r_1, r_2, r_3, r_4, r_5 are equal.



4.2 Implementation

```
set(arithmetic).
assign(domain_size,10). % 9 numbers

list(distinct).
[0,g1,g2,g3,g4,g5,g6,g7,g8,g9]. % The grids
end_of_list.

formulas(assumptions).
%Level 3

g1+g2=14.
g2+g3+g4=14.
g4+g5+g6=14.
g6+g7+g8=14.
g8+g9=14.

%Level 4 each ring have the same sum.

%g2+g3+g4=g1+g2.
%g2+g3+g4=g4+g5+g6.
%g4+g5+g6=g6+g7+g8.
%g6+g7+g8=g8+g9.

end_of_list.

NOTE!: use commands mace4 -m -l -f puzzle4.in and mace4 -c -f puzzle4.in
to run the file .
```

4.3 Solutions

In the level 3 mace4 generated 2 models :

Model	G1	G2	G3	G4	G5	G6	G7	G8	G9
1	8	6	1	7	4	3	2	9	5
2	5	9	2	3	4	7	1	6	8

In the level 4 mace4 generated models. Here are some models.

Model	G1	G2	G3	G4	G5	G6	G7	G8	G9
1	9	4	1	8	3	2	5	6	7
2	9	2	5	4	6	1	7	3	8
3	8	3	7	1	6	4	5	2	9
4	7	6	5	2	8	3	1	9	4
5	7	6	5	2	3	8	1	4	9

Chapter 5

Puzzle 5

5.1 Problem definition

In this puzzle a player must create numbers like in the pictures bellow. The sum of those number must be equal with the number on the lemon.



5.2 Implementation

```

set(arithmetic).
assign(domain_size,10). % 9 numbers

formulas(assumptions).

%Each aple can take a value form 1 to 9.
(a1=0 | a1=1 | a1=2 | a1=3 | a1=4 | a1=5 | a1=6 | a1=7 | a1=8 | a1=9)
(a2=0 | a2=1 | a2=2 | a2=3 | a2=4 | a2=5 | a2=6 | a2=7 | a2=8 | a2=9)
(a3=0 | a3=1 | a3=2 | a3=3 | a3=4 | a3=5 | a3=6 | a3=7 | a3=8 | a3=9)
(a4=0 | a4=1 | a4=2 | a4=3 | a4=4 | a4=5 | a4=6 | a4=7 | a4=8 | a4=9)

%      No1      +      No2  +      No3      +      No4
=lemon.
((a1*10) + a2)+((a3*10) +a4)+((a2*10) + a4)+((a1*10) + a3)=283.

%  a1      a2 -no1
%  a3      a4 -no2
%  |      |
%  no4    no3
end_of_list.

```

NOTE!: use commands `mace4 -m -1 -f puzzle5.in` and `mace4 -c -f puzzle5.in` to run the file .

5.3 Solutions

In the puzzle the number of models depends of the target. For each target number this can be different. For target=199 mace4 generated 50 models. Here are some of them.

Model	A1	A2	A3	A4
1	9	0	1	4
2	8	3	0	3
3	7	3	2	2
4	7	0	5	2