Department of Computer Science
Technical University of Cluj-Napoca

Artificial Intelligence

Laboratory activity

Name:Lazar Catalin and Vaida Diana
Group:30433
Email:lazar.mihai@protonmail.com
Email:dianalaura5445@gmail.com

Teaching Assistant: Adrian Groza
Adrian.Groza@cs.utcluj.ro

Contents

Table 1: Lab scheduling

Activity	**Deadline**
Searching agents, Linux, Latex, Python, Pacman	W_1
Uninformed search	W_2
Informed Search	W_3
Adversarial search	W_4
Propositional logic	W_5
First order logic	W_6
Inference in first order logic	W_7
Knowledge representation in first order logic	W_8
Classical planning	W_9
Contingent, conformant and probabilistic planning	W_{10}
Multi-agent planing	W_{11}
Modelling planning domains	W_{12}
Planning with event calculus	W_{14}

Lab organisation.

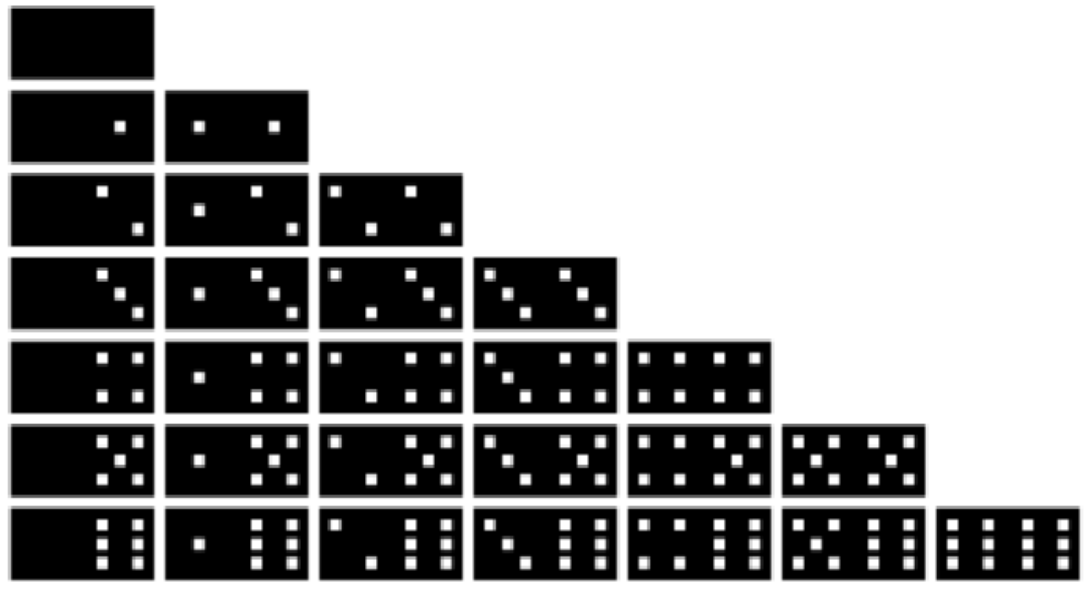
1. Laboratory work is 25% from the final grade.
2. There are three deliverables in total: 1. Search, 2. Logic, 3. Planning.
3. Before each deadline, you have to send your work (latex documentation/code) at moodle.cs.utcluj.ro
4. We use Linux and Latex
5. Plagiarism: Don't be a cheater! Cheating affects your colleagues, scholarships and a lot more.

Chapter 1

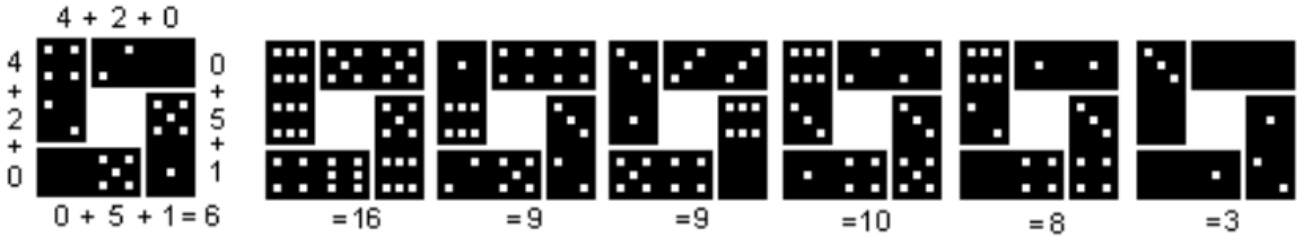
A2: Logics

Dominoes is a family of tile-based games played with gaming pieces. Each domino is a rectangular tile, usually with a line dividing its face into two square ends. Each end is marked with a number of spots (also called pips or dots) or is blank. The backs of the tiles in a set are indistinguishable, either blank or having some common design. The gaming pieces make up a domino set, sometimes called a deck or pack. The traditional European domino set consists of 28 tiles, also known as pieces, bones, rocks, stones, men, cards or just dominoes, featuring all combinations of spot counts between zero and six.

The problems we did are the seven square problem and the quadrilles(French mathematician Edouard Lucas (1842-1891)).
At first, we went a different approach but we encountered all kinds of problems.
With the big help of Mr. Professor Adrian Groza, we represented the dominos. Every domino is formed from it's left and right part, being 56 in total, 28 dominos, and each part has to be smaller or equal to 6. Every combination of parts is unique.



Seven Squares
You can lay seven square frames with all the 28 dominos, so that the sums of the numbers of points are the same at all four sides (book 1).

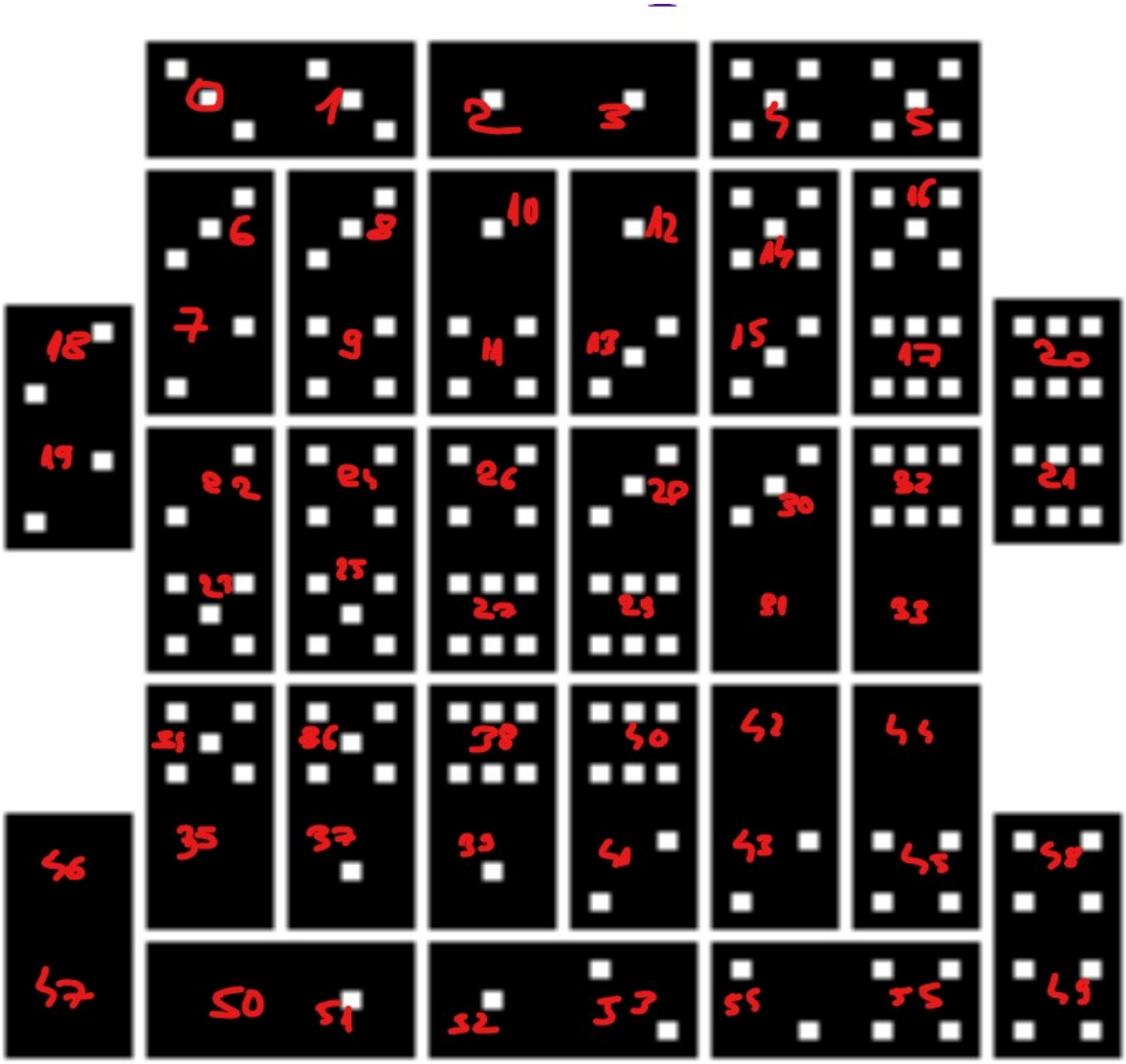


We did a combination from the two figures, our dominos are placed like in the first figure, but we made the algorithm in the second figure. Thus for the first square, for example,the sum of the points from the first domino and the 'first' part of the second domino from the right is equal to the second domino and the part from 'down' of the third domino. This repeats for the other pieces in the square and the rest of the squares.
Like this - f(0) + f(1) + f(2) = f(2) + f(3) + f(4).

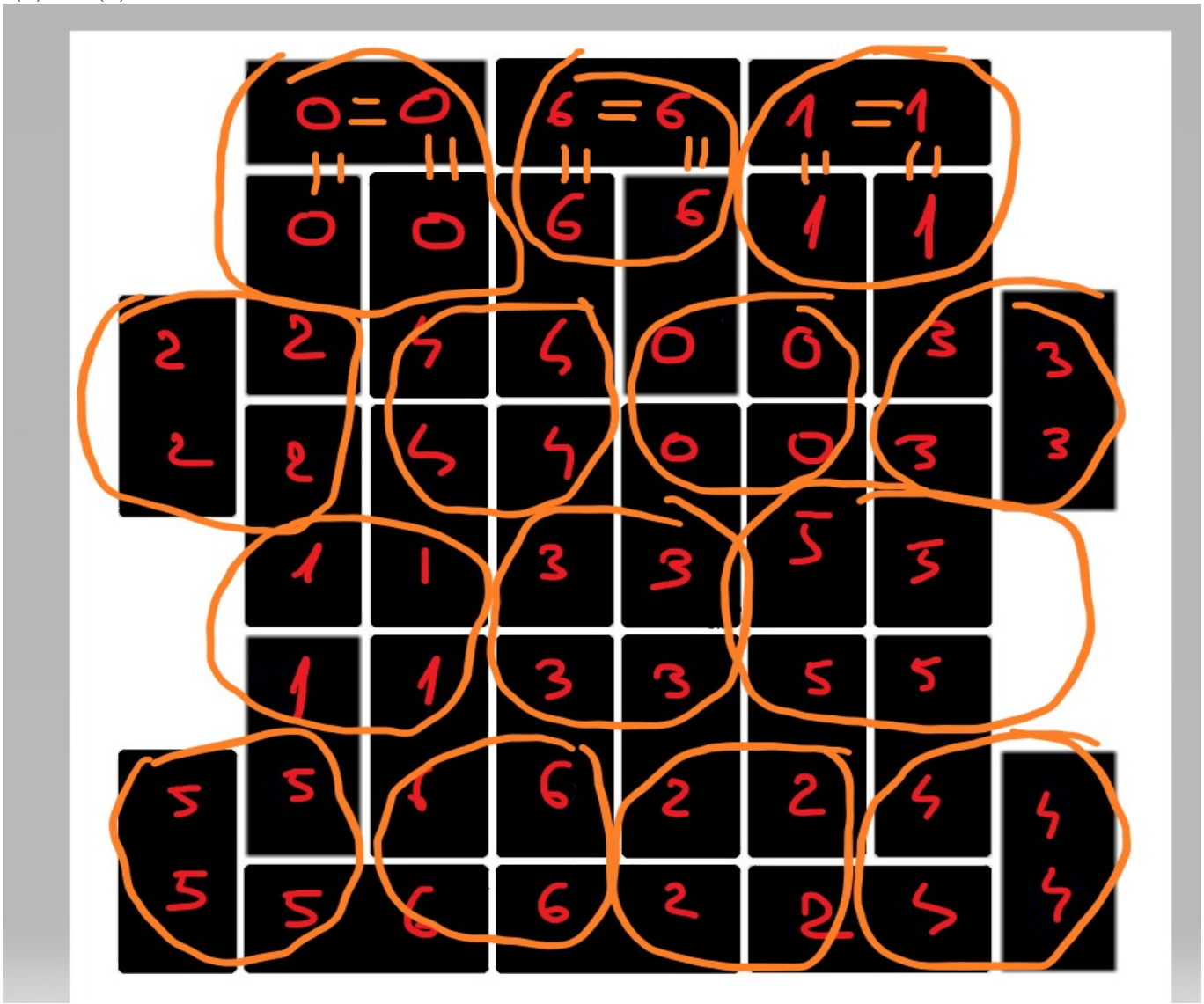
Quadrilles

Quadrilles go back to the French mathematician Edouard Lucas (1842-1891). Those are compact figures with all 28 dominos. 2x2squares with the same numbers of points are included in.

Following the algorithm, this mean the first three top dominos will have pieces that have equal points as well as the other 4 dominos from the 'side'. The rest of the dominos will be placed so they will have equal points to their neighbor piece that isn't a part of their respective domino.



In the figure down, we found one of the solutions for this arrangement of dominos, separating every 'part' into a square.
So for the first part -
f(0) = f(1).
f(0) = f(6).
f(1) = f(8).



Bibliography

http://www.mathematische-basteleien.de/dominos.htm
Mr.Professor Adrian Groza's help

Appendix A

Your original code

Don't be a cheater! Cheating affects your colleagues, scholarships and a lot more. This section should contain only code developed by you, without any line re-used from other sources. This section helps me to correctly evaluate your amount of work and results obtained.

```
formulas(quadrille).

f(0) = f(1).
f(0) = f(6).
f(1) = f(8).

f(2) = f(3).
f(2) = f(10).
f(3) = f(12).

f(4) = f(5).
f(4) = f(14).
f(5) = f(16).

f(18) = f(7).
f(7) = f(22).
f(18) = f(19).

f(9) = f(24).
f(11) = f(26).
f(9) = f(11).

f(13) = f(28).
f(13) = f(15).
f(15) = f(30).

f(17) = f(32).
f(20) = f(21).
f(17) = f(20).

f(23) = f(34).
f(23) = f(25).
f(25) = f(36).
```

```
f(27) = f(38).
f(29) = f(40).
f(27) = f(29).

f(31) = f(42).
f(31) = f(33).
f(33) = f(44).

f(46) = f(35).
f(46) = f(47).
f(35) = f(50).

f(37) = f(51).
f(39) = f(52).
f(37) = f(39).

f(41) = f(53).
f(43) = f(54).
f(41) = f(43).

f(45) = f(55).
f(48) = f(49).
f(45) = f(48).

end_of_list.


formulas(seven_squares).

f(0) + f(1) + f(2) = f(2) + f(3) + f(4).
f(2) + f(3) + f(4) = f(4) + f(5) + f(6).
f(4) + f(5) + f(6) = f(6) + f(7) + f(0).


f(8) + f(9) + f(10) = f(10) + f(11) + f(12).
f(10) + f(11) + f(12) = f(12) + f(13) + f(14).
f(12) + f(13) + f(14) = f(14) + f(15) + f(8).

f(16) + f(17) + f(18) = f(18) + f(19) + f(20).
f(18) + f(19) + f(20) = f(20) + f(21) + f(22).
f(20) + f(21) + f(22) = f(22) + f(23) + f(16).

f(24) + f(25) + f(26) = f(26) + f(27) + f(28).
f(26) + f(27) + f(28) = f(28) + f(29) + f(30).
f(28) + f(29) + f(30) = f(30) + f(31) + f(24).

f(32) + f(33) + f(34) = f(34) + f(35) + f(36).
f(34) + f(35) + f(36) = f(36) + f(37) + f(38).
f(36) + f(37) + f(38) = f(38) + f(39) + f(32).

f(40) + f(41) + f(42) = f(42) + f(43) + f(44).
```

```
f(42) + f(43) + f(44) = f(44) + f(45) + f(46).
f(44) + f(45) + f(46) = f(46) + f(47) + f(40).

f(48) + f(49) + f(50) = f(50) + f(51) + f(52).
f(50) + f(51) + f(52) = f(52) + f(53) + f(54).
f(52) + f(53) + f(54) = f(54) + f(55) + f(48).
end_of_list.
```