Introduction to Artificial Intelligence
Exercise Sheet 4

Laurin van den Bergh, 16-744-401
Yufeng Xiao, 19-763-663
Nora Beringer, 19-734-227

Universität Zürich
Institut für Informatik

Due: March 23, 2022

Exercise 4.1

We can formulate a CSP for the Latin Square Problem as follows:
Set of variables X: X = {A, B, C, 1, 2, 3 }
The set of variables defines here the column and row space.
We have got a 3x3 Latin Square with rows = {A, B, C } and columns = {1, 2, 3 }.
Set of Domain D: D = {1, 2, 3 }
D denotes the variables used to fill the Latin Square.
Set of Constraints C: C = {A1 $\neq$ A2, A1 $\neq$ A3, A1 $\neq$ B1, A1 $\neq$ C1, A2 $\neq$ A3, A2 $\neq$ B2, A2 $\neq$ C2, A3 $\neq$ B3, A3 $\neq$ C3, $\sum$A = 6, $\sum$B = 6, $\sum$C = 6, $\sum$1 = 6, $\sum$2 = 6, $\sum$3 = 6, A $\neq \emptyset$, B $\neq \emptyset$, C $\neq \emptyset$, 1 $\neq \emptyset$, 2 $\neq \emptyset$, 3 $\neq \emptyset$ }
Where $X_{ij} \in X$, hence for example A1 depicts the location of the variable in row A and column 1.
Neighbouring locations and locations in the same row and column space are only allowed to use unused variables from D, so that every column and row depicts each variable exactly once. Therefore no neighbouring location and locations in the same row and column can show the same entry as its neighbour or as any location in the same row and column space.
Furthermore the sum of each row and of each column must be exactly 6 (as $1 + 2 + 3 = 6$), meanwhile each row and column must be completely filled with variables of the Domain D.

Exercise 4.2

a) node-consistent: A single variable (corresponding to a node in the CSP graph) is node-consistent if all the values in the variabe's domain satisfy the variable's unary constraints.

The unary constraint for dom(x) = {2, 4, 6} is $c_1 = \langle (x), x \in \{2x' | x' \in \mathbf{N}\} \rangle$.
This constraint is satisfied as every variable in dom(x) multiplied by 2 is in fact a natural number.

The unary constraint for dom(y) = {1, 4, 9} is $c_2 = \langle (y), y \neq 3 \rangle$.
This constraint is satisfied as every variable in dom(y) is $\neq 3$.

The unary constraint for dom(z) = {0, 1, 2, 3} is $c_3 = \langle (z), z < 4 \rangle$.
This constraint is satisfied as every variable in dom(z) is < 4.

b) arc-consistent: A variable is arc-consistent if every value in its domain satisfies the variable's binary constraints. More formally, X_i is arc-consistent with respect to another variable X_j if for every value in the current Domain D_i there is some value in the Domain D_j that satisfies the binary constraint on the arc (X_i, X_j).

For $c_4 = \langle (x, y), x^2 = 4y \rangle$ this is not the case as only the variable 2 from dom(x) together with variable 1 from dom (y) satisfies this binary constraint ($2^2 = 4$).
There is no other variable in dom(x) which equals 4y when squared.

For $c_5 = \langle (y, z), y = z^2 \rangle$ this is not the case as for the variable $0 \in$ dom(z) the constraint is not fullfilled as there is no variable y' $\in$ dom(y) for which y' $= 0^2$ holds.

For $c_6 = \langle (x, z), x = 2z \rangle$ this is not the case as for the variable $0 \in$ dom(z) the constraint is not fullfilled as there is no variable x' $\in$ dom(x) for which x' = 2*0 holds.

c) path-consistent: Path-consistency tightens the binary constraints by using implicit constraints that are inferred by looking at triples of variables. A two-variable set $\{X_i, X_j\}$ is path-consistent with respect to a third variable X_m if, for every assignment $\{X_i = a, X_j = b\}$ consistent with the constraints (if any) on $\{X_i, X_j\}$, there is an assignment to X_m that satisfies the constraints on $\{X_i, X_m\}$ and $\{X_m, X_j\}$. The name refers to the overall consistency of the path from X_i to X_j with X_m in the middle.

Note for Group: I think there is no path-consistency as there is no arc-consistency. Do you agree or am I missing sth?

d) strongly 3-consistent: A CSP is strongly 3-consistent if it is 3-consistent and is also (3 - 1)-consistent and (3 - 2)-consistent.
2-consistency is the same as arc consistency.
For binary constraint graphs, 3-consistency is the same as path consistency.

Note for Group: If there is no path-consistency there is also no 3-consistency.

Exercise 4.3

a)
b)
c) If we pick the variable which is most likely to fail soon then the failure will be detected immediately which therefore avoids pointless searches through other variables and

on average have fewer sucessful assignments to backtrack over. The least-constraining value prefers the value that rules out the fewest choices for neighboring variables in the constriant graph. Therefore it leaves the maximum flexibility in order to assign following variables.

Exercise 4.4

a)

b)