CSE 571 Homework 2

Name: Student ID:

Name: Student ID:

Due: September 22., 2016

For each homework we will state here if you have to work alone or if you can team up with another student.
For this homework you are allowed to work in **teams of two students** for all questions. Each group has to submit one **handwritten** (!) copy and state their names on the front page. Use the same groups as in Blackboard. Failure to do so could result in the allegation of plagiarism! Submission is possible at the end of a lecture or during an office hour till the due date of this homework (which you can find above).
Print this homework and write all your answers in the space below the questions. If you need additional space you might want to use the backside of the pages.
Also,

- unstapled homework will result in a decrease of at least 30% of the achieved points.
- handwritten text which is not readable will be graded with zero points.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8*	Sum
Points	10	7	3	6	4	15	10	15	55 + 15 Bonus
Achieved									

1. Question (10 Points)

For the following tasks we will use the map of Indias souther states given in the figure below. In order to simplify the tasks we will only use the bigger states without city states and islands. Use the following shortcuts for the names of the states: M = Maharashtra, C = Chhattisgarh, O = Odisha, T = Telangana, G = Goa, KA = Karnataka, AP = Andhra Pradesh, KE = Kerala and TN = Tamil Nadu. For the first questions you are going to use Prolog. We recommend you to install the commonly used SWI-Prolog interpreter which you can find under `http://www.swi-prolog.org/`. The interpreter is available for both Windows and Unix. For Ubuntu you can simply type *sudo apt-get install swi-prolog*. Write your rules and facts into a textfile with a name which ends on *.prl* and load it into the interpreter by typing *[nameOfFile.prl]*. The interpreter itself is started in Linux by using the command *swipl*. You can find more information here: `http://www.swi-prolog.org/man/quickstart.html`. For our Prolog program we will handle color names as values of variables and the shortcuts of the state names as variables.

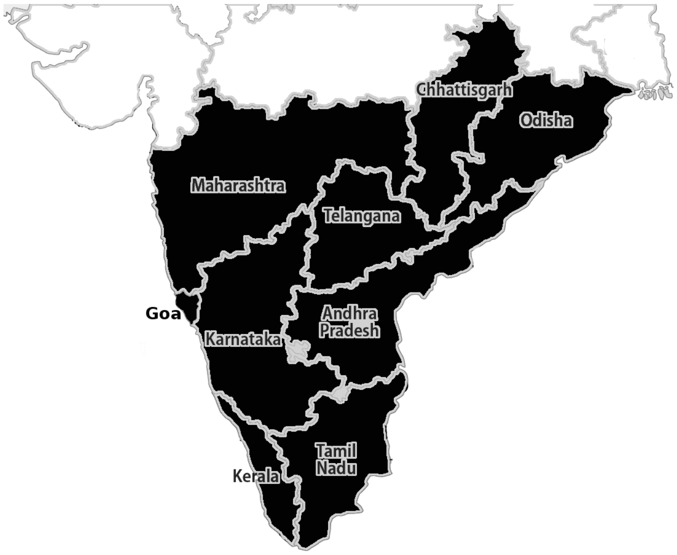


Figure 1: The southern states of India. To simplify the task we use here only a reduced number of states without the northern states and city states.

Hint: Odisha and Telangana have a common border.

1.1) (2 Points) We will write a simple Prolog program to solve the map coloring problem for a map of South India. The colors to use are: red, green, blue and yellow. Subsequently, our four facts are:
color(red).
color(green).
color(blue).
color(yellow).

Write down a rule named *connected(stateA,stateB)* which expresses that two states are succesfully connected if they have both different colors.

1.2) (2 Points) Obviously, the relationship *neighbourship* which we are now going to introduce is commutativ. State (two) rule(s) for *neighbor(stateA,stateB)* by using the previously defined rule such that the relationship of neighborship is commutativ.
(+1 Bonus if you use just one rule)

1.3) (3 Points) Define now our final rule *india(M,C,O,T,G,KA,AP,KE,TN)* which becomes true if we can find a color for each states. Basically, this rule contains the information which states are neighbors.
Hint: The body of your rule should contain 16 literals.

1.4) (1 Point) Execute now the prolog program by starting the interpreter and loading the file as described above. Now execute *india(M,C,O,T,G,KA,AP,KE,TN).* in the interpreter and write down the output of the program:

1.5) (2 Points) Does a solution exist, if we use only three colours?

2. Question (7 Points)

Assume again we have the indian map given from Figure 1. In this task you are going to apply Forward Checking on the CSP problem for the four colors red (r), green (g), blue (b) and yellow (y).
We will do Forward Checking for four assignments, which you can find in the left counn *Assignment*.

2.1) (5 Points) Cross out the colours for each state which are eliminated by Forward Checking in each step.

Assignments	M	C	O	G	KA	T	AP	KE	TN
Initialization	g r b y	g r b y	g r b y	g r b y	g r b y	g r b y	g r b y	g r b y	g r b y
T = g	g r b y	g r b y	g r b y	g r b y	g r b y	**green**	g r b y	g r b y	g r b y
KA = b	g r b y	g r b y	g r b y	g r b y	**blue**	**green**	g r b y	g r b y	g r b y
M = r	**red**	g r b y	g r b y	g r b y	**blue**	**green**	g r b y	g r b y	g r b y
O = y	**red**	g r b y	**yellow**	g r b y	**blue**	**green**	g r b y	g r b y	g r b y

2.2) (1 Point) What would happen if in the next step red would be assigned to TN?

2.3) (1 Point) What color is going to be assigned to Chhattisgarh?

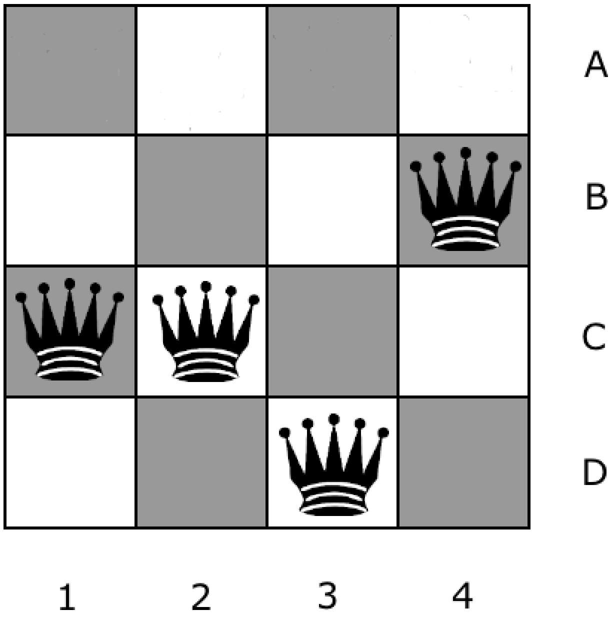
3. Question (3 Points)

Draw the minimal map which requires at least four colors such that each area has a different color than it neighbours.

4. Question (6 Points)

The min-conflicts algorithm attempts to solve CSPs iteratively. It starts by assigning some value to each of the variables, ignoring the constraints when doing so. Then, while at least one constraint is violated, it repeats the following: (1) randomly choose a variable that is currenly violating a constraint, (2) assign to it the value in its domain such that after the assignment the total number of constraints violated is minimized (among all possible selections of values in its domain).

In this question, you are asked to execute the min-conflicts algorithm on a simple problem: the 4-queens problem in the figure shown below. Each queen is dedicated to its own column (i.e. we have variables Q_1 , Q_2, Q_3, and Q_4 and the domain for each one of them is A, B, C, D). In the configuration shown below, we have $Q_1 = C$, $Q_2 = C$, $Q_3 = D$, $Q_4 = B$. Two queens are in conflict if they share the same row, diagonal, or column (though in this setting, they can never share the same column).



You will execute min-conflicts for this problem three times, starting with the state shown in the figure above. When selecting a variable to reassign, min-conflicts chooses a conflicted variable at random. **For this problem, assume that your random number generator always chooses the leftmost conflicted queen. When moving a queen, move it to the square in its column that leads to the fewest conflicts with other queens. If there are ties, choose the topmost square among them.**

4.1) Starting with the queens in the configuration shown in the above figure, which queen will be moved, and where will it be moved to? Cross the right fields.

	1	2	3	4
Queen				

	A	B	C	D
Position				

4.2) Continuing off of Part 1, which queen will be moved, and where will it be moved to?

	1	2	3	4
Queen				

	A	B	C	D
Position				

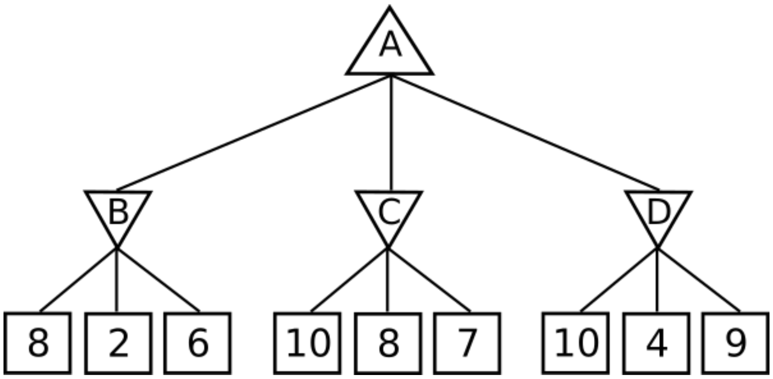
4.3) Continuing off of Part 2, which queen will be moved, and where will it be moved to?

	1	2	3	4
Queen				

	A	B	C	D
Position				

5. Question (4 Points)

Consider the zero-sum game tree shown below. The triangles that point up represent maximizing players and triangles which point down represent minimizing players. Assume that each player plays optimal and carry out the minimax algorithm.

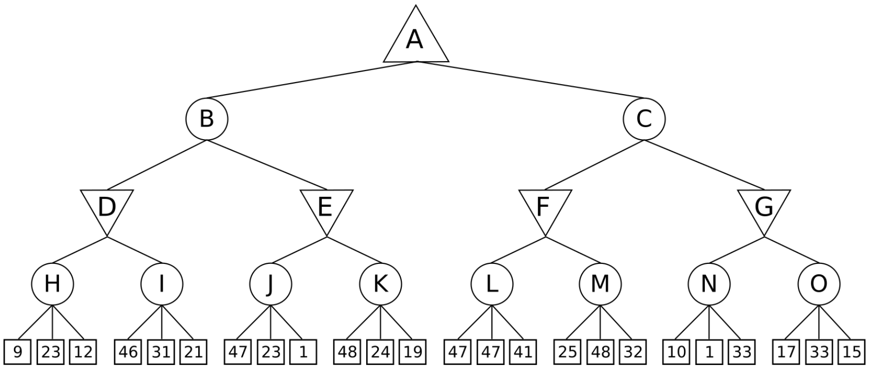


Enter the values of the nodes in the following table:

	A	B	C	D
Value				

6. Question (15 Points)

Consider the zero-sum game tree shown below. The triangles that point up represent maximizing players and triangles which point down represent minimizing players. Nodes which are circles are chance nodes. We assume here that at a chance node each edge has an equal probability to be chosen. Assume that each player plays optimal and carry out the expectimax algorithm.



Enter the values of the nodes in the tables below. You can round the values to the first two decimals places.

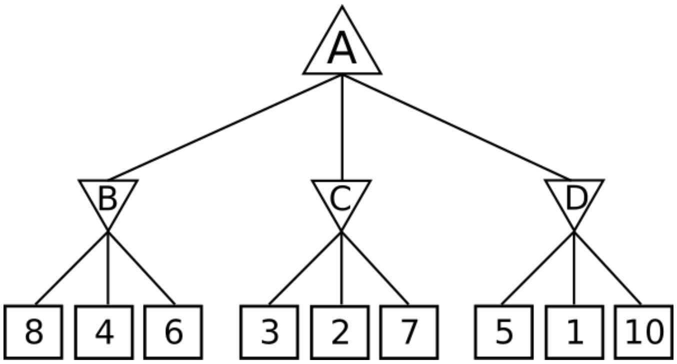
	A	B	C	D	E	F
Values						

	G	H	I	J	K	L
Values						

	M	N	O
Values			

7. Question (10 Points)

Assume the game tree below is given. The triangles that point up represent maximizing players and triangles which point down represent minimizing players. Assuming that both players play optimal, apply the alpha-beta pruning algorithm on the tree.



The search goes from left to right: If you have to visit the next child node, the leftmost unvisited child node is chosen.
Fill in the values of the nodes in the first table and make a cross in the second table for the leaf nodes which do not get expanded/visited.

	A	B	C	D
Value				

	8	4	6	3	2	7	5	1	10
Not visited									

8. Bonus Question (15 Points)

There exists a well-known riddle, which is basically a constraint satisfaction problem. Your task in this homework question, is to solve this problem by writing a Prolog program which answers the question 'Who owns the Fish?' (and others, of course, like who lives in the white house?). Submit your solution, i.e. your Prolog file, on Blackboard to receive the bonus points. There will be a specific 'Homework 2 Bonus Question Prolog'-Submission on the Assignments page.

Let us assume that there are five houses of different colors next to each other on the same road. In each house lives a man of a different nationality. Every man has his favorite drink, his favorite brand of cigarettes, and keeps pets of a particular kind.

- The Englishman lives in the red house.
- The Swede keeps dogs.
- The Dane drinks tea.
- The green house is just to the left of the white one.
- The owner of the green house drinks coffee.
- The Pall Mall smoker keeps birds.
- The owner of the yellow house smokes Dunhills.
- The man in the center house drinks milk.
- The Norwegian lives in the first house.
- The Blend smoker has a neighbor who keeps cats.
- The man who smokes Blue Masters drinks bier.
- The man who keeps horses lives next to the Dunhill smoker.
- The German smokes Prince.
- The Norwegian lives next to the blue house.
- The Blend smoker has a neighbor who drinks water.