Artificial Intelligence DD2380, HT 2008
Homework 2

Due: See BILDA (around Sept 17)
Where: BILDA.
Scope: This homework covers chapters 5-10.

Instructions

In the spirit of the goal oriented grading system this homework starts with some general questions (part A) for receiving a passing grade. You are to solve Part A of the homework individually and hand it in (upload to BILDA) individually and the answers must be in your own words. Upload the answers as a single **pdf**-file.

The second part of the homework (Part B) requires you to make an implementation to solve a problem. You are encouraged to solve this part of the homework in pairs. You have to use a programming language that is freely available and commonly known (Matlab, Python, C/C++, Java) and you have to provide the source code as well for testing. The results of Part B should be uploaded to BILDA separately from Part A. Each upload should be **a SINGLE zip-file**. See BILDA for more instructions.

The third part cannot be started until after you have handed in Part A and B.

We expect you to follow the code of honor (http://www.csc.kth.se/utbildning/hederskodex/).

Grading

The max number of points on parts A-C is 76p. The grade is given by

A	$\geq$65
B	$\geq$55
C	$\geq$45
D	$\geq$30
E	$\geq$15

Language

The homework should be written in English.

Part A

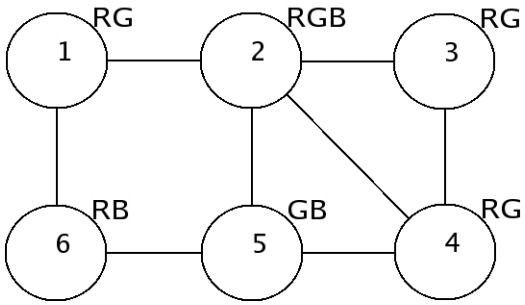
Exercise 1, 16p

Each of the below questions is worth max 2p

1. What do we need to know about the utility function in order to apply alpha-beta pruning in a game that includes chance?
2. When arc-consistency checking finishes?
3. Imagine you have implemented minimax algorithm with breadth-first instead of depth-first. Would it be possible to do alpha-beta pruning? Why or why not?
4. In a CSP, how simple backjumping method improves forward checking?
5. How does propositional logic scale when the amount of objects and relations in the environment increases substantially?
6. What is the main problem of minimax algorithm in card games?
7. Can an inference algorithm derive a sentence that is not entailed?
8. How does Generalized Modus Ponens extends the standard Modus Ponens?

Exercise 2, 10p

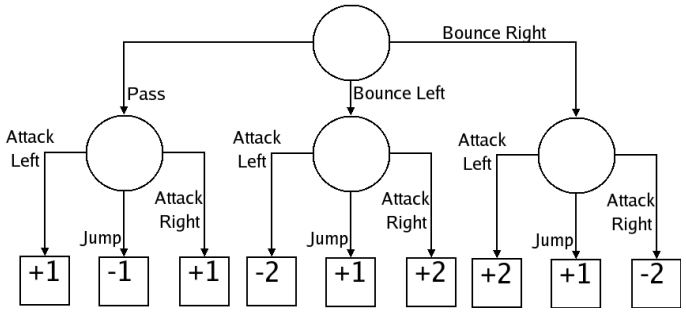
Consider the following constraint graph for a graph coloring problem (the constraints indicate that connected nodes cannot have the same color). The domains are shown next to each variable node.



1. Show the sequence of variable assignments during a pure backtracking search. Assume that the variables are examined in numerical order and the values are assigned in the order shown next to each node (show also the invalid assignments made). Show assignments by writing the variable number and the value, e.g. 1R. Don't write more than 10 assignments, even if it would take more to find a consistent answer. (3p)
2. Show the sequence of variable assignments during backtracking with forward checking, assume that the variables are examined in numerical order and the values are assigned in the order shown next to each node. Show assignments by writing the variable number and the value, e.g. 1R. Don't write more than 10 assignments, even if it would take more to find a consistent answer. (3p)
3. Is there a better rule for choosing which variable to assign next? In affirmative case, show the sequence of variable assignments for backtracking with forward checking and that ordering rule. (4p)

Exercise 3, 10p

You have the ball in a basketball game, and in front of you there is a fast defender. Your possible movements are shown in the tree below, followed in the tree by the possible movements by the defender. In the end of the tree you can see the value for you of that particular set of action-reaction.



1. What first move should you do? Motivate!(2p)
2. Using alpha-beta pruning, consider the nodes from left to right, which nodes are cut off? Circle the nodes that are not examined. Motivate! (4p)
3. Do you think this is a good method for evaluating such action? If not, what is the difference between this problem and a tic-tac-toe game, which is suitable for this method? (4p)

Exercise 4, 10p

Consider the following sentences:

- All limes are green
- Everything that is not green is not a lime
- This tomato is not green

1. Write each of these sentences in first-order logic, using predicates *Lime*, *Green* and the constant symbol *Tomato* (6p)
2. Can the sight of a red tomato provide evidence of all the limes being green? Motivate it in terms of inference rules! (4p)

Part B

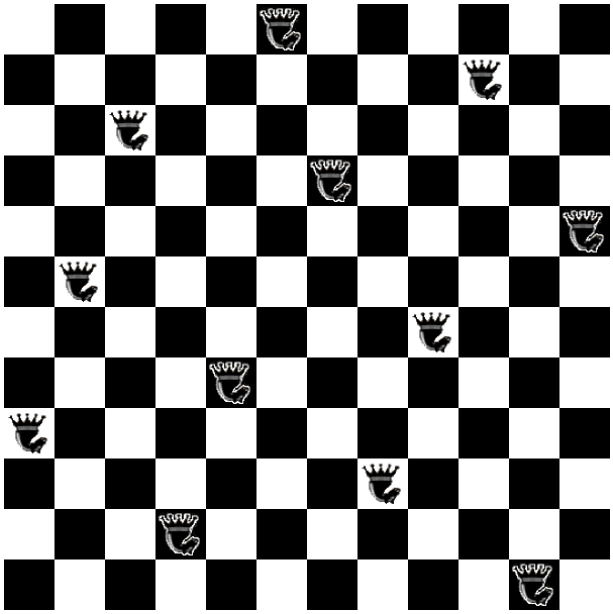
Exercise 5, 20p

In chess, an amazon is a fairy piece that combines the movement of a knight (L shapes) and a queen (rows, columns and diagonals) .The problem of N-amazons consist of placing N amazons in a chess board NxN such that none of them attack each other (same as N-queens problem with different movements).

1. Formulate the problem as a CSP and write a program that will solve it for you. Performance (maximum size of board solved) affects the grade.(10p)

2. Explain the method you chose for solving the problem, specially its limitations.(5p)

3. Analyze the performance of your method by running it with different board sizes (sizes between 10 and 1000, for example), taking note of details of the execution such as execution time, iterations, etc. If possible, propose a way of improving the performance. (5p)



Part C

Exercise 6, 10p

REVIEW