EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

MATH. - NATURWISS. FAKULTÄT
FACHBEREICH INFORMATIK
KOGNITIVE SYSTEME · PROF. A. ZELL

Artificial Intelligence

Assignment 2

Assignment due by: 09.11.2016, Discussion: 15.11.2016

Tutors: Yann Berquin (yann.berquin@uni-tuebingen.de), Isabel Patiño (isabel.patino@uni-tuebingen.de) and Hauke Neitzel (hauke.neitzel@uni-tuebingen.de).

Question 1 Rationality (6 points)

Suggestion: Draw probability trees.

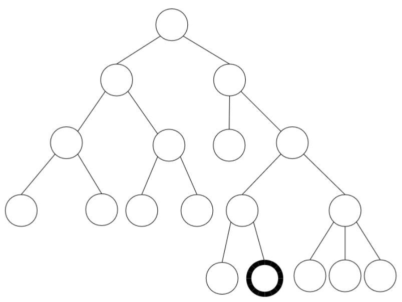
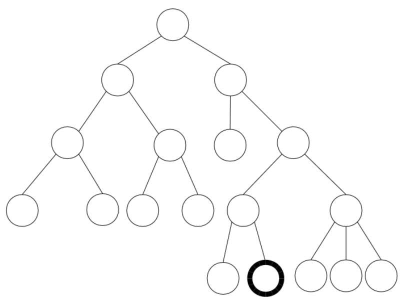
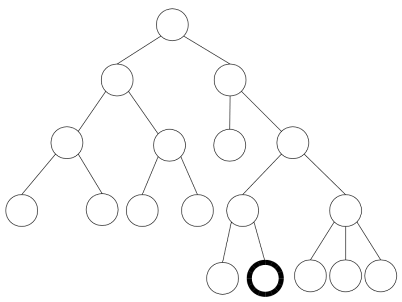
(a) One of your friend, a notable gambler with poor knowledge on probabilities, challenges you to a game with a pair of dice with the following rules: should one of the dice be a six he gives you 12 euros, should both dice be sixes he gives you 32 euros, however, should the dice be anything else you owe him 8 euros. Is it in your interest to play this game only once (i.e. one round)? Is it in your interest to play it twice (i.e. two rounds)? What is your average expected income per round in this game? Please justify each answer.

(b) Suppose you are on an oral exam, and you are given the choice of three topics a, b and c without knowing their respective contents. However, the examiner, who is well known for his twisted exams, tells you that one of the three topics will give you full grade automatically, because it is so easy, while the others are so complex you are doomed to a long and painful failure. You pick a topic without looking at it, say topic a, and the examiner, who knows what the topics are, shows you another topic, say topic b, which has one of the two incredibly difficult topics. He then says to you, "Do you want to pick topic c?" Is it to your advantage to switch your choice? Please justify your answer.

Question 2 Search strategies (3 points)

Enumerate the nodes depending on the search algorithm. The bold node is the goal node, the search should stop once the goal node is reached.

(a) Breadth-first search
(b) Depth-first search
(c) Iterative deepening depth-first search

Question 3 Programming in LISP (3+3+3+2 = 11 points)

When doing these exercises you are **not** allowed to use either looping constructs or built in functions that solve the exercise by themselves.

(a) Implement a function $(\text{listMerge}\ a\ b)$ that adds both lists into a list c. a and b have the same length. The function listMerge should return a sorted list composed of a, b and the result of the addition (c). Duplicated elements should be removed. Example: From the lists $a =$(1 3 8 7 2), $b =$ (7 2 1 4 0), and $c =$(8 5 9 11 2) the function should return (1 2 3 4 5 7 8 9 11).

(b) Implement a function $(\text{searchTree}\ a\ tr)$ that looks for the value a in a binary tree tr. The function should return how deep a is in the binary tree i.e. in which level is a located. If a does not exist in the tree, the solution should be nil. Each node of the binary tree is represented as a list. The list is composed of (i) the node element (scalar value), (ii) a list representing the left child and (ii) a second list representing the right child. The absence of a child is expressed as nil. Example: `(list 2 (list 4 nil nil) (list 7 nil nil))` represents a binary tree with the element 2 as root with children 4 an 7.

(c) Implement a function $(\text{deleteTree}\ a\ tr)$, which deletes a from a binary tree tr. The function should return the new binary tree. If a is not found in the tree, the initial binary tree should be returned. The structure of the tree is the same as in task b.

(d) Implement a function $(\text{functionList}\ a\ f)$, where a is a list and f a function. functionList should return the values of a in which f is not nil.

```
> (filter-list (list 1 2 5 15 24 13 14)
                          (if (evenp x)
                           (+ 1 x)
                           nil))
> (3 25 15)
```