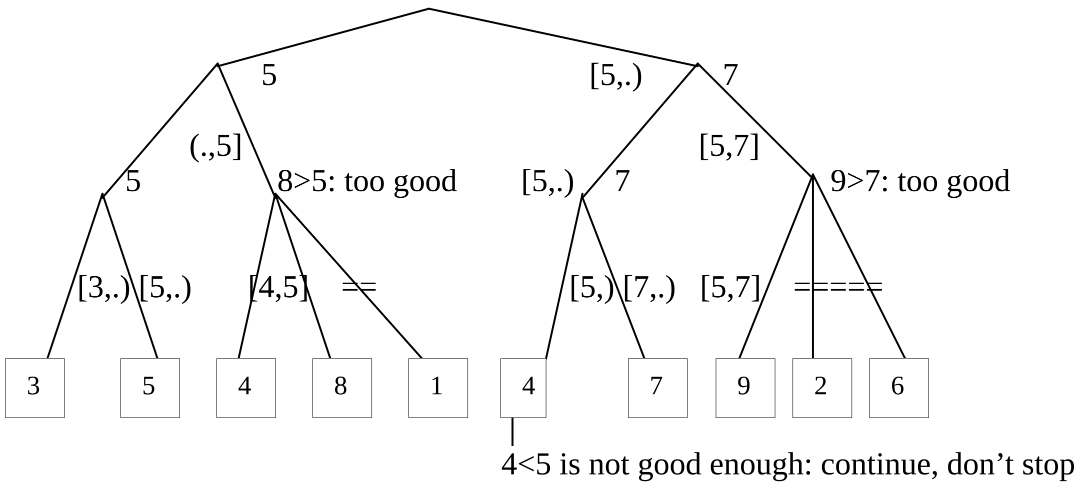
Exam - Artificial Intelligence 1DL340   2010-10-19

You may use a dictionary. You can answer in either English or Swedish.

Grading:
Required for 3: 13.5
Required for 4: 18
Required for 5: 22.5

1) α–β pruning (5p)

a) Perform min-max search with α–β pruning in the tree below. The top node is a max node. Show how the values of α and β evolve, and where pruning takes place. (3p)

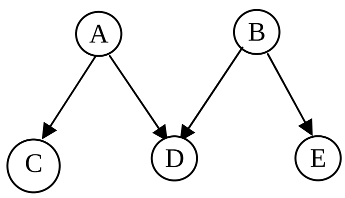


b) In general (i.e. not related to the example above), the order in which moves are investigated influences how much pruning the α–β algorithm can accomplish. What order of moves is expected to make α–β pruning most effective? (2p)

We want to investigate the best moves *first. We want as much pruning as possible, and that is achieved by having α as high as possible and β as low as possible. The best (highest value) moves for MAX give the highest α and the best (lowest value) moves for MIN give the lowest β. Of course, we don't exactly* know *the best moves yet (that is what the algorithm is for), but we may have a reasonable guess (heuristic). For instance in Connect-4: the middle columns are often better moves.*
Incorrect/incomplete answers:
- “depth-first”. Yes, the algorithm is always depth-first, the freedom lies in the order in which we evaluate the moves (children) of a node.
- “the order that give the most pruning”. True, but not specific enough.

---

2) Uncertainty, Bayesian networks



a) Suppose we have the Bayesian network above. We learn that D is true. How does that affect the probability of A, B, C, and E? If we then learn that A is false, how are the probabilities affected? **(2)**

If D is true the other nodes become more likely. If then A is false B and E are even more likely, but C less likely.

b) When are two events independent of each other? **(2)**

The answer I expected was p. 181:
A and B are independent if and only if $p(A \cap B) = p(A)p(B)$.
But I also accept the answer from p. 366-367and a correct reasoning based on the network above.

---

3) Natural language

Describe the following concepts with a few sentences:

a) ATN *see p. 639*
b) Case frame *see p. 645*
c) Context-free grammar *see p. 637* **(6)**

---

4) Search

a) Describe Hill climbing. What is its most severe problem? s*ee p. 127* **(3)**
b) When is one heuristic function more informed than another? s*ee p. 148* **(2)**

---

5) Decision trees
a) Explain briefly the concept *decision tree*.
The ID3 algorithm "learns" a decision tree from examples. These examples would be consistent with several different decision trees. Describe
b) *which* decision tree ID3 learns, and
c) *why* this tree is preferable over others, and
d) outline *how* ID3 constructs this tree from the examples. **(5)**

a) A decision tree contains the knowledge necessary to put examples into categories. Each internal node contains a question, and for each possible answer there is a child. The leaves are the categories.
So when given an example, we start from the root, ask the question, follow the answer to the child, ask the next question, and so on, until we reach a leave. [1.5 points]
b) the simplest one [1 point]
c) Occam's Razor – this is actually a heuristic: that the simplest answer is usually more likely to be correct than a complex one. Or in modern day English: "If it looks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck." [1 point]
So it's not about efficiency *(asking* fewer *questions), but about* validity *(asking* relevant *questions).*
d) It is a top-down recursive algorithm: if you decide what the root question is, then each of the answers will lead you to a similar, simpler (one less question, fewer cases) problem. You stop if you know the category (or if you have no more questions that you can ask – in this way you can deal with inconsistent knowledge).

The root question is the one that maximizes the expected information gain*. See section 10.3.2 for further explanation (not required for this question).* [1.5 points]

---

6) Candidate elimination
We have a rather complex coffee machine.
First you choose coffee (K), espresso (E) or chocolate (C).
You select additions: none (N), sugar only (S), cream only (C), or both (B).
You can get a small (S) cup, or medium (M) or large (L).

However, the machine cannot produce all 36 possible combinations.
Clue 1. The machine can produce a small espresso with sugar only.
Clue 2. The machine can produce a large chocolate with cream only.

a) Explain why, given clues 1 and 2, candidate elimination cannot learn which drinks the machine can produce.

From now on, the goal is to learn what drinks the machine *cannot* produce. In this light, clue 1 and 2 are *negative* examples. A positive example is:
Clue 3. The machine cannot produce a small espresso with cream only.

b) Use clue 3, clue 1 and clue2, in that order, in the candidate elimination algorithm, and explain (in understandable words) what your conclusion is so far.

Instead of having one parameter "additions" with 4 values, we could have two parameters: sugar or not (S/NS) and cream or not (C/NC).

c) Explain the effect of this different representation on what can be learned. If you find it hard to explain in general, continue the example with

Clue 4. The machine cannot produce a small chocolate with both sugar and cream. **(5)**

I use the notation (type, addition, cupsize)
a) [1 point]
Clue 1: +(E, S, S)   *Clue 2:* +(C, C, L)
Running the CE algorithm, only (*,*,*) *generalizes both clues, so the algorithm has finished learning and the learned result is that the machine can produce all combinations. However, it is given that it cannot.*
Comment: further examples would not help. Indeed, in order to learn what the machine cannot produce, we would need negative examples. But from these two clues alone, we can conclude that any negative example will make the algorithm fail.

b) [2 points]

Clue 3: +(E,C,S)	(E,C,S)	(*,*,*)
Clue 1: -(E,S,S)	(E,C,S)	(*,C,*)
Clue 2: -(C,C,L)	(E,C,S)	(E,C,*), (*,C,S)

Important: your answer should show that learning is not finished: we have no definite answer yet to what the machine can or cannot produce.
We are sure that the machine cannot produce a small espresso with cream only.
Maybe the "small" doesn't matter, and the machine cannot produce espresso with cream only. (Idea: if you put cream in espresso, it's cappuccino!)
Maybe the "espresso" doesn't matter, and the machine cannot produce a small drink with cream only. (Idea: a small cup is too small for cream!)

c) [2 points]
Now we can distinguish more categories, so we can learn more.
*The number of base combinations doesn't change (3*2*2*3=3*4*3=36) but the number of categories changes to (3+1)*(2+1)*(2+1)*(3+1) = 4*9*4, compared to the 4*5*4 we had before (each parameter gets a "+1" for the *-value).*

*For instance, if a small cup is too small for cream (C), you expect that it is also too small for cream and sugar (B). But previously, we could not represent "B or C": we could only generalize B and C to *, which includes drinks without cream. So we could not represent (and therefore not learn) the concept "drinks with cream".*
In the new situation B=(S,C) and C=(NS,C), so "B or C" can now be represented as (,C). The category "drinks with cream" is thus (*,*,C,*).*

In the old situation, clue 4 leads to an inconsistency:
(E,C,S) and (C,B,S) generalize to (,*,S), which is more general than (E,C,*) or (*,C,S).*

In the new situation, we get:

+(E,NS,C,S)	(E,NS,C,S)	(*,*,*)
-(E,S,NC,S)	(E,NS,C,S)	(*,NS,*,*), (*,*,C,*)
-(C,NS,C,L)	(E,NS,C,S)	(E,NS,*,*), (E,*,C,*) (*,NS,*,S), (*,*,C,S)
+(C,S,C,S)	(*,*,C,S)	(*,*,C,S)

So indeed, we have now learned that the machine cannot produce a drink in a small cup that contains cream (any drink, with or without sugar).