Exam Artificial Intelligence, 2008-05-29

You may use a dictionary. You can answer in either English or Swedish
Write short *answers to the open questions: no question requires more than one page of answer.*

1) Suppose we have a graph with nodes S, A, B, and C, where S is the start and G the goal node. The distances between connected nodes are given in this table:
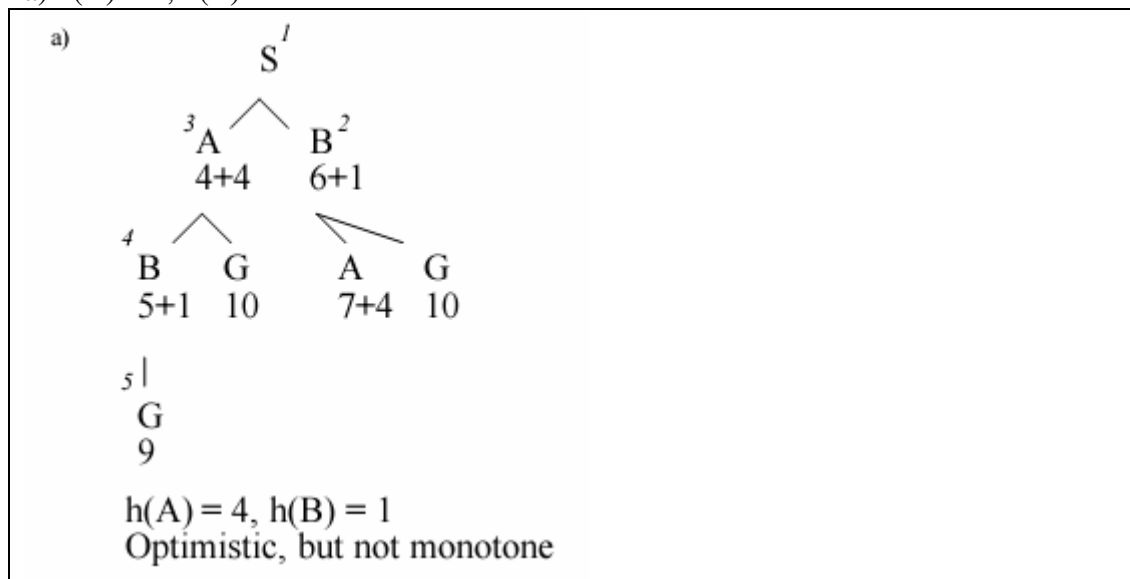
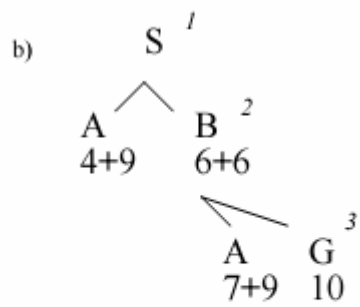|	**S**	**A**	**B**	**G**
S		4	6	
A	4		1	6
B	6	1		4
G		6	4	

Perform a search with method A:
- Draw the tree with the g and h values for all nodes.
- Show in what order the nodes are examined and
- Discuss whether the heuristic function is optimistic and/or monotone and how this affects the search.

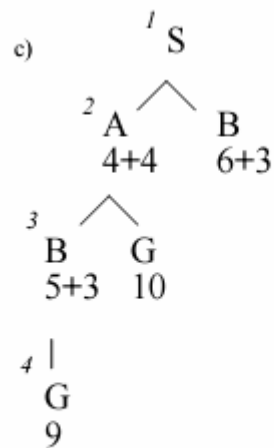Do this for the three cases when the estimated remaining distances are:
a) h(A) = 4, h(B) = 1

b) h(A) = 9, h (B) = 6

b)

S ^{1}

A   B ^{2}
4+9   6+6

A   G ^{3}
7+9   10

h(A) = 9, h(B) = 6
Not optimistic,
and therefore not monotone

---

c) h(A) = 4, h(B) = 3

c)

^{1} S

^{2} A   B
4+4   6+3

^{3}
B   G
5+3   10

$_4$ |
G
9

h(A) = 4, h(B) = 3
Optimistic and monotone

If h(n) is monotone, when a node is found it is via the shortest path, so the node's g-value is known.

If h(n) is optimistic the shortest path will be found.

2) In a Bayesian network we have the following nodes:

  x1: smoking
  x2: visit_to_Africa
  x3: lung_cancer
  x4: bronchitis
  x5: tuberculosis
  x6: coughing
  x7: positive_chest_x_ray

x1 can cause x3 and x4
x3 can cause x6
x4 can cause x6 and x7
x2 can cause x5
x5 can cause x7

a) Draw a graph which shows the dependencies.
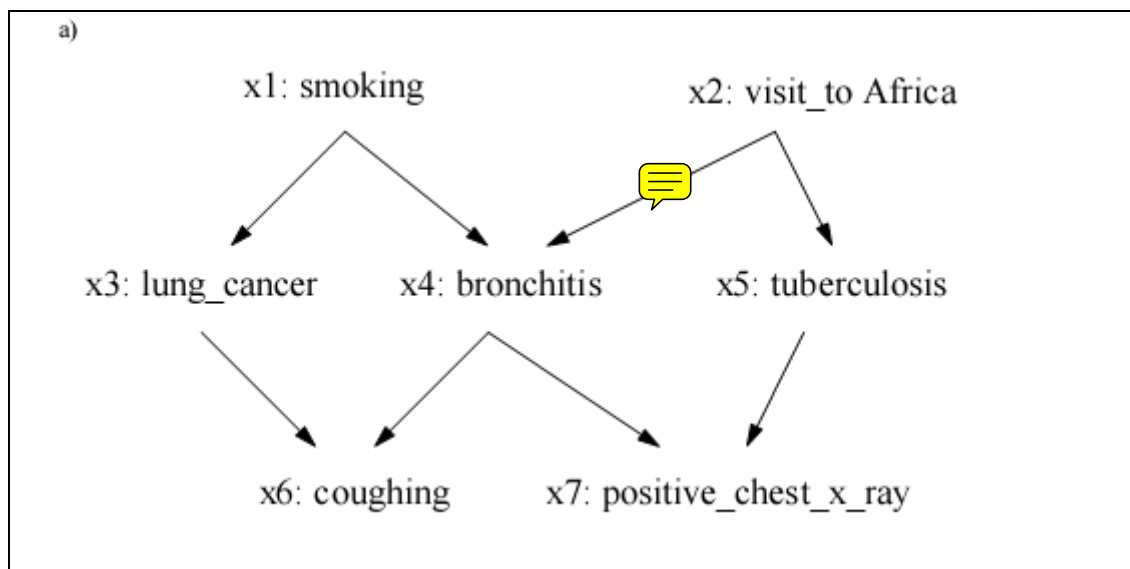
Suppose we learn that x6 is true.
b) How does this affect the probabilities of the other nodes?

You can reason in a number of ways, but I prefer this:

b) x6 is true:
x3, x4 and x1 become more likely by abduction, and x7 due to x4
x5 and x2 are hard to judge, can go up or down

Then we learn that x7 is true.
c) How does this affect the probabilities of the other nodes?

c) x7 is also true:
x4 increases further, while x3 decreases

Then we learn that x2 is true.
d) How does this affect the probabilities of the other nodes?

d) x2 is also true
x5 becomes very likely,
and x4 decreases to the same level as x3

3) In the book the following context-free rules are used to describe noun phrases:

noun_phrase ↔ noun
noun_phrase ↔ article noun
noun ↔ man
noun ↔ dog
article ↔ a
article ↔ the

a) Extend the grammar to include preposition phrases. A preposition phrase is a preposition followed by a noun phrase. Prepositions are *in* and *on*.
A noun phrase can now be a noun, a noun followed by a preposition phrase, an article followed by a noun, or an article followed by a noun followed by a preposition phrase.
The complete grammar (including the two nouns from b) is now:
noun_phrase ↔ noun
noun_phrase ↔ noun preposition_phrase
noun_phrase ↔ article noun
noun_phrase ↔ article noun preposition_phrase
preposition_phrase↔ preposition noun_phrase
noun↔ man
noun↔ dog
noun↔ house
noun↔ prairie
preposition ↔ in
preposition ↔ on
article↔ a
article↔ the

b) Show also the parse tree for the phrase:

the dog in the house on the prairie

(You have to include two more nouns too.)
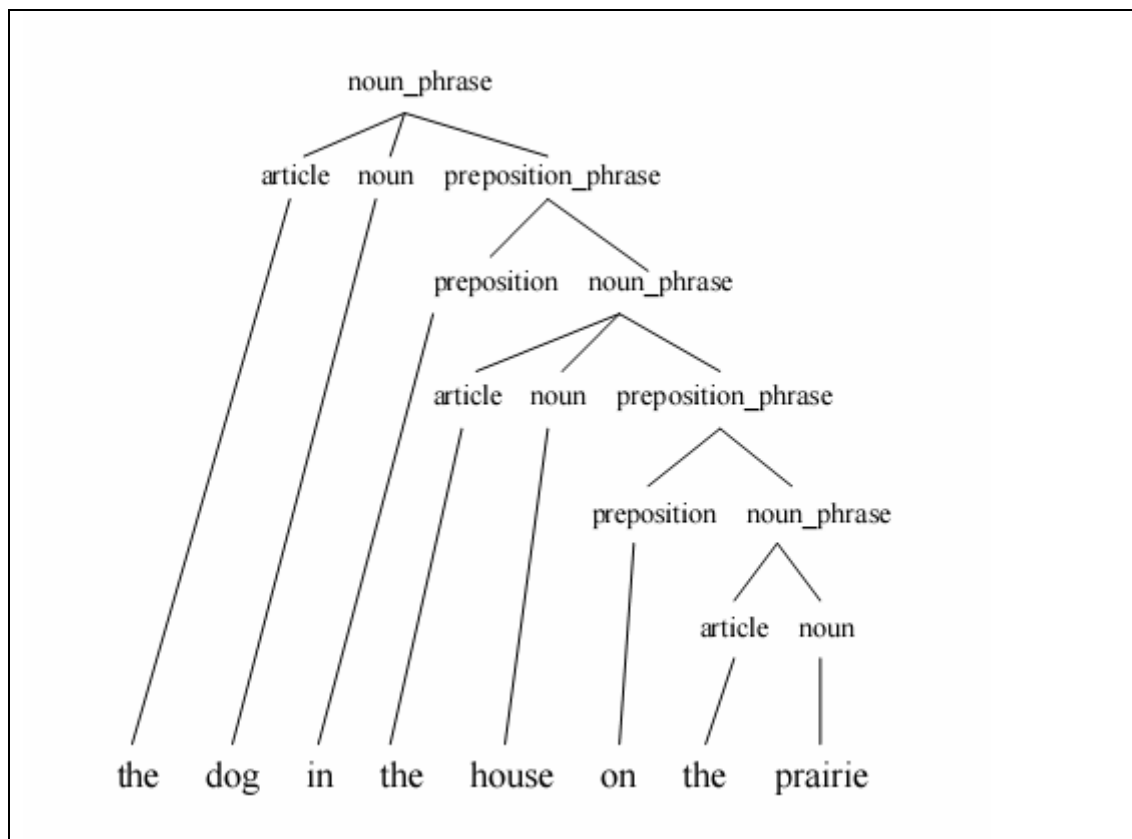
4) We want to learn if a creature lives or dies, based on its genome. For simplicity we consider creatures with genomes of only 6 letters (each letter can be A, C, G or T). We find the following examples.

CATCAT – lives
TACCAT – dies
GATCAG – lives
ACTAAC – dies
TTTCCC – dies
GATCTA – lives
AATTGA – dies

a) Use the Candidate Elimination algorithm on this example to learn the category of living creatures. Candidate categories are strings of 6 letters from {A,C,G,T,*}. Use the examples in the above order, and show the sets S and G for each step.

example	S	G
+ CATCAT	CATCAT	******
- TACCAT	CATCAT	C***** **T***
+ GATCAG	*ATCA*	**T***
- ACTAAC	*ATCA*	*AT*** **TC**
-TTTCCC	*ATCA*	*AT*** **TCA*
+ GATCTA	*ATC**	*AT***
- AATTGA	*ATC**	*ATC**

b) Use the ID3 algorithm to learn a decision tree for the above examples. Decisions are of the kind "What is the n:th letter?" (NOTE: normally, you would need to know how to compute information gain values. But this particular case relies only on the principle of ID3.)

ID3 chooses the question with the highest information gain. In this case, the question "what is the 1st letter?" is enough to reach a complete decision:
C or G: lives, T or A: dies.

c) Explain how the characteristics of the two algorithms (CE and ID3) differ, and explain in general terms how they can come to such different conclusions.

When using patterns over {ACGT*}, CE can learn only a very limited set of concepts. For instance, it cannot learn the answer that ID3 finds, because it cannot represent it. CE reaches a final conclusion based on these examples, any more data could only lead to inconsistency, not to a better answer.
ID3 can in principle learn any set of living/dying patterns. It would continue to refine itself, as long as the data itself is not inconsistent (the same creature living and dying). These 7 examples are so few that there is very little confidence that we have learned the correct concept. It might just be "by accident" that the 1st letter can distinguish these cases.
CE treats living and dying asymmetrically - everyone chose to treat "living" as positive examples and "dying" as negative examples, but you could try the reverse.
ID3 treats living and dying symmetrically. Moreover, it would not be constrained to two outcomes, but could handle more than two (healthy - handicapped - dead).

d) Suppose that you had not 7 but a million measurements (which is about 250 per possible genome) and that they are inconsistent. What method would you use in that case to predict if a creature with a known genome lives or dies?

Neither CE nor ID3 can handle inconsistencies, so you need to choose another method!

In this case some statistical methods would be appropriate.
You can use Bayes' rule, but that would effectively mean the following algorithm:
for a given genome XXXXXX
look up all occurrences of XXXXXX in the database
see how many are alive, how many are dead -
from this compute the "live expectancy" percentage.
This is not exactly learning, and it would not take into account "similar" genomes (whether that is good or bad is a question for biologists).

You could try other methods, such as clustering, as well.

5) Describe the architecture of a typical rule based expert system.

See http://www.it.uu.se/edu/course/homepage/ai/vt08/Expert.pdf (first slide) or
Luger page 279 Figure 8.1.
Note: Match is more decision-tree based than rule based. It is certainly not a "typical
rule based expert system". Therefore an answer describing the tables in Match is
incorrect. The architecture described in the figures above is more general, and applies
to most expert systems including Match.

6) Explain in a few sentences the essence of the following concepts:
    a) horizon effect (in minimax search)
    b) the frame problem

If we use a fixed depth in minimax, the search doesn't look further than that, and
must then rely on the static evaluation. As a result, the player may try to postpone a
loss, and prefer a big loss beyond the horizon over a small loss now. A solution is to
make the search depth more flexible, and deepen the search in "volatile" situations.

How do you specify, in some formalism, what does *not* change (after an action, or
receiving new information). Specifying everything explicitly does not scale up, so
there must be a way to put a "frame" around an event and nothing outside the frame
is affected.