C1 Artificial Intelligence (25 Points)
Search

(a) (10 points) A heuristic function is said to be consistent if, for every node n and every successor n′ of n, the following holds:

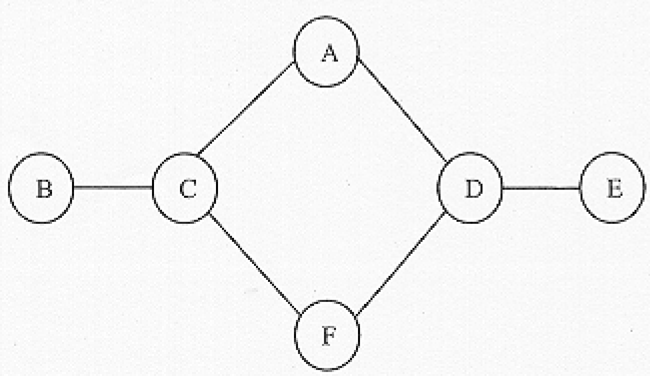
$$h(n) \leq c(n, n') + h(n')$$

where c(n, n′) is the cost of the operator (edge) that takes you from n to n′. Prove that if a heuristic is consistent, then it must be admissible. You may assume that there is a reachable goal node, that there are no negative edge costs, and the heuristic function = 0 at the goal node.

(b) (5 points) Give an example of an admissible heuristic that is not consistent. (You may draw a graph and give a specific example.)

(c) (6 points) Sometimes there is no good evaluation function, h(n), that estimates how close a node is to a goal node for a problem, but there is a good comparison method: a way to estimate whether one node is closer to the goal than another, without assigning numerical values to either. Assuming this comparison function is transitive, it is possible to do a best first search using such a comparison function by simply sorting the nodes according to the comparison function and expanding the node that comes out on bottom (i.e., which is closer to the goal than the others). Is there an analog of A* using a comparison function instead of a state evaluation function h(n)?

(d) (4 points) Consider coloring the graph below using three colors (1, 2, and 3) such that no two nodes that are connected by an edge are colored the same color as a constraint satisfaction problem. Imagine C has already been colored with color 1.



Fill in the following table of domains (copied on to our answer sheet) after running the forward checking algorithm (no search, just forward checking pruning). Circle the row corresponding to the next variable to instantiate if you were using the minimum remaining values (MRV) heuristic (which is also called the "most constrained variable" or "fail first" heuristic). If there are multiple choices, circle all of them.

Variable	Domain
A	
B	
C	1
D	
E	
F	

C2 Artificial Intelligence (25 Points)
Decision Trees

(a) (10 points) The decision tree learning algorithm acts in a greedy fashion in that it chooses which attribute to split on by taking the one with the largest information gain. One could make the algorithm less greedy by giving it a lookahead, that is, change the algorithm to select the best information gain possible at some lookahead, rather than just at the next depth.

Consider implementing this idea in the following manner: when evaluating whether to split on a given attribute, instead of using the standard one step information gain, let the two step information gain for a split a be defined as the largest gain two steps down, in other words, the gain obtained if we split the next level optimally after splitting this node on a. (Note that this criteria is then used to make only a single, one step split. After this single split, the whole procedure is then repeated to select the next attribute to split on in each of the new leaves.)

Write a formula for the two step information gain, assuming that the attributes are binary, and using the following notation (a0 is the first variable to split):

- S^l = the set of examples on the left side of the first split
- S^r = the set of examples on the right side of the first split
- n^l = the number of examples on the left side of the first split
- n^r = the number of examples on the right side of the first split
- A^l = the set of all possible attributes (a^l) which are ways of further splitting on the left side of the first split
- A^r = the set of all possible attributes (a^r) which are ways of further splitting on the right side of the first split
- Gain(S, a) = the standard information gain on a set S splitting on a.

(b) (15 points) In this question we consider decision trees with continuous input attributes A1,…,An and a Boolean output attribute Y. In such trees, the test at each internal node is an inequality of the form Ai > c, where c, the split point, may be any real number (to be chosen by the learning algorithm). The value at each leaf is true or false. In a test-once tree, each attribute may be tested at most once on any path in the tree; in a test-many tree, each attribute may be tested more than once on a path.

Suppose we are given the following four examples:

A1	A2	Y
3	3	False
6	13	True
15	14	True
14	22	False

i. (6 points) Draw a test once decision tree that classifies the examples correctly where the chosen root provides the highest information gain possible.

ii. (9 points) Suppose we sort the examples by their value for attribute Ai. (See, e.g., the values for A2 in the table above.) Explain why the best (i.e., highest information gain) split point for Ai never falls directly between two examples with the same output classification. (E.g., for A2 in the table, it could not be between 13 and 14.)

C3 Artificial Intelligence (25 points)

Belief Networks

Let H_x be a random variable denoting the handedness of an individual x, with possible values l or r. A common hypothesis is that left- or right-handedness is inherited by a simple mechanism; i.e., perhaps there is a gene G_x, also with values l or r, and perhaps actual handedness turns out the same (with some probability s) as the gene an individual possesses. Furthermore, perhaps the gene itself is equally likely to be inherited from either of an individual's parents, with a small non-zero probability m of a random mutation flipping the handedness.

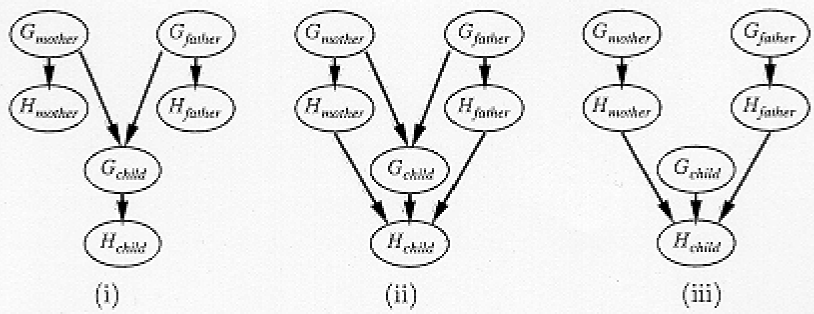


Fig. 1: Three possible structures for a Bayesian network describing genetic inheritance

(a) (3 pts) Which of the three network structures in Fig. 1 claim that $P(G_{father}, G_{mother}, G_{child}) = P(G_{father})P(G_{mother})P(G_{child})$? There may be more than one.

(b) (3 pts) Which of the three networks make independence claims that are consistent with the hypothesis? There may be more than one.

(c) (3 pts) Which of the three networks is the ***best*** description of the hypothesis? (i) (ii) (iii)

(d) (5 pts) Draw the CPT (Conditional Probability Table) for the G_{child} node in networks (i) or (ii) and fill in all of the values.

(e) (6 pts) Suppose that $P(G_{father} = l) = P(G_{mother} = l) = x$. In network (i) or (ii), derive an expression for $P(G_{child} = l)$ in terms of m and x only, by conditioning on its parent nodes.

(f) (5 pts) Under conditions of genetic equilibrium, we expect the distribution of genes to be the same across generations. Use this and your answer to (e) to calculate the value of x, and, given what you know about handedness in humans (i.e. that very few humans are left handed), explain why the hypothesis described at the beginning of this question must be wrong.

C4 Artificial Intelligence (25 points)

Planning

A landlord has decided she needs to do some maintenance on her apartment building. In order to satisfy the tenants, she needs to have the building painted and the windows cleaned. A scaffold is needed for painting the building and cleaning the windows. (If you don't know what a scaffold is, think of it as a ladder.) Setting this up formally as a planning problem we get the following:

Initial state: NoScaffolding(B1)

Goal Condition: Painted(B1), WindowsClean(B1), NoScaffolding(B1)

Operator	Preconditions	Add effects	Delete effects
Paint(b)	*Scaffold(b)*	*Painted(b)*	
Clean(b)	*Scaffold(b)*	*WindowsClean(b)*	
BuildScaffold(b)	*NoScaffolding(b)*	*Scaffold(b)*	*NoScaffolding(b)*
RemoveScaffold(b)	*Scaffold(b)*	*NoScaffolding(b)*	*Scaffold(b)*

In the above STRIPS specification, the schema variable b represents buildings, and we have just one building, namely B1. In the initial state NoScaffolding(B1) is true and all other propositions are false. We want to end up in a goal state where the building is painted, the windows are cleaned, and the scaffold is removed.

(a) *Planning Graphs [14 points]*

The figure below shows a planning graph for the problem. The small square boxes denote "Maintain" actions, and at each action level, we have added all actions whose preconditions are present in the previous level. The precondition and add effect links are represented with solid lines, and the delete effect links are represented by dashed lines. Your task will be to add appropriate mutex (mutual exclusion) links. For brevity, we have abbreviated the names of the actions and propositions and have omitted the parameter B1.

Mark mutual exclusion relationships between propositions at levels P0, P1, and P2 only, and between actions at levels A1 and A2. The levels after P3 have been shown just to indicate that the planning graph does not end there. *In addition to marking the mutexes on the figure, also explicitly write down for each of the levels P0, P1, and P2 the pairs of mutually exclusive propositions at that level, and for levels A1 and A2 the pairs of mutually exclusive actions those levels (each action and proposition is numbered to make this easy).* To help you out, we repeat here the (recursive) definitions for mutexes for propositions and actions:

I. No propositions are mutex at proposition level P0

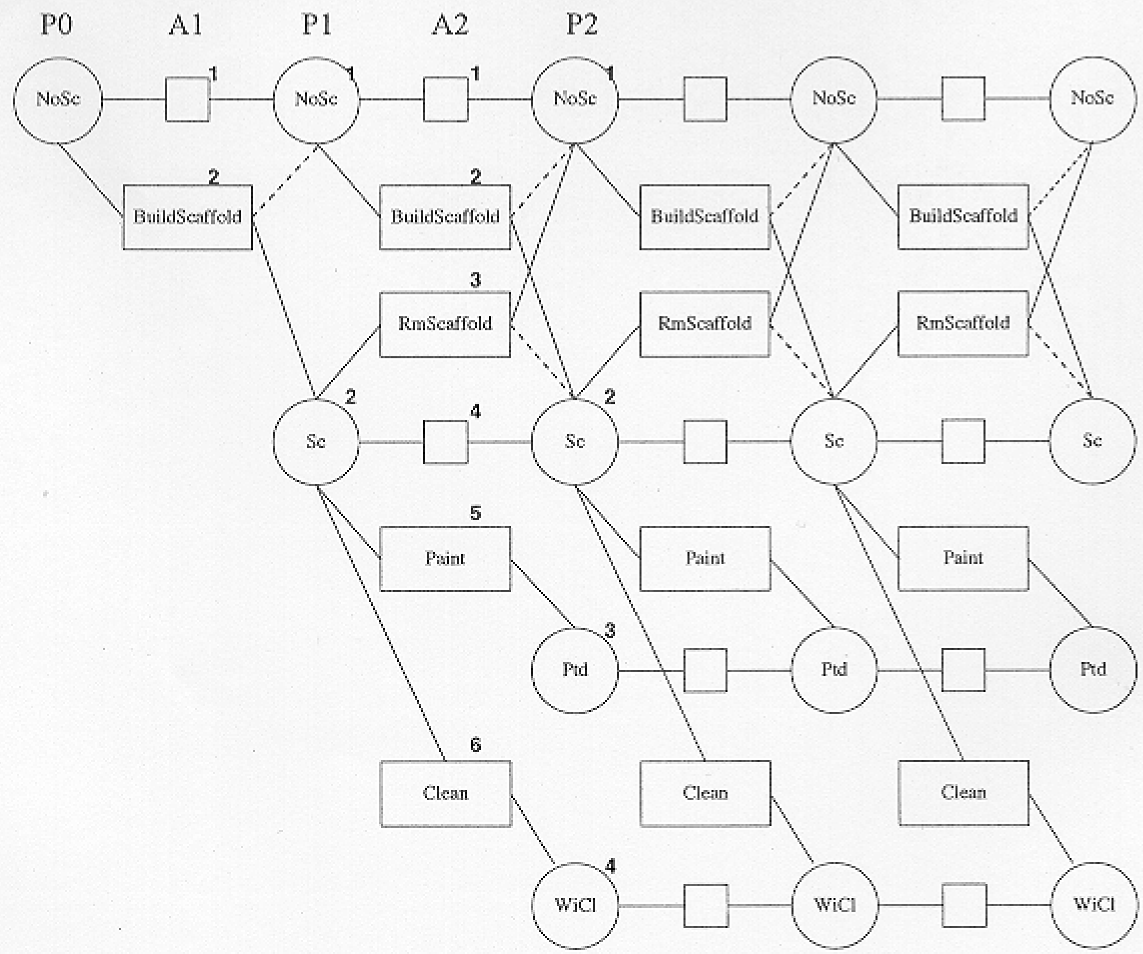
II. Two actions a1 and a2 are mutex at action level Ai if any of the following conditions is true:

   A. Inconsistent effects

   B. Interference

   C. Competing needs

III. Two propositions p1 and p2 are mutex at proposition level Pi if they have inconsistent support.



(b) ***Plan Extraction [3 points]***

Write down the plan solution that is extracted at level P2. If no solution is extracted, explain why.

(b) ***Planning Heuristics [8 points]***

Consider the following heuristic:

$$h(s) = |P_+(s) - P_+(w_0)|$$

Here w_0 is the initial state and $P_+(w_0)$={NoScaffolding}, |S| denotes the number of elements in a set, and "–" is set difference. In words, this heuristic has as its value the number of propositions in the current state that are not true in the initial state.

A. Is h admissible for our particular problem of building a plan for the landlord?

B. Is h admissible for STRIPS planning problems in general?

State your answers to parts A and B, and justify your answer to each part with either a proof of admissibility or with a counterexample to h being admissible.