special ~~specialist~~ general ~~general~~

1. **Introduction**

(a) What is the difference between weak and strong AI? 1/ (2)

imprecise vague

The weak AI is used for resolving the task in predefined conditions. The strong AI research the current conditions and world knowledge to act in a current situation.

(b) Describe two possible risks of using strong AI. 1,5/ (2)

smaller ~~risks~~ more severe risks lacking

- Using all possible actions to make the best result (loud voice during the chessplay) (v)
- Do not achieve the goal because of maximising the metrics (staying in one place because of metric of security of passangers) (v)

2. **Agents and Environment**

(a) **Properties of task environments** 2,5 (3)

In the following example, we consider an autonomous vehicle that independently and without the support of a human driver transports the passengers through traffic from their starting point to their destination.

Identify six environmental properties from the lecture that describe the scenario.

Partly observable ✓
Stohastic ✓
Sequential ✓
Dynamic ✓
Multiagent ✓

(b) **Stochastic environment** 2/ (2)

Is every stochastic environment also unknown? Justify your answer.

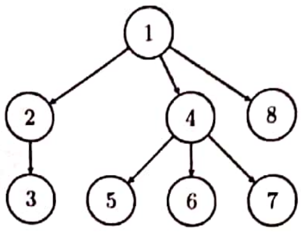
Stochastic enviroment vs Deterministic. ✓
Deterministic environment is when the next state is dependent only on your choice and current position. The stohastic environment is opposite. Unknown enviroment is when you do not know the laws of the current world and need to discover it in the process of the word. So if you do not expect a situation which happens it means that the environment Stohastic.

3. Search

(a) Search methods in trees (1)

The figure shows a tree that was searched using a search method from the lecture. The numbers in the nodes determine the order in which the nodes were "expanded" by the algorithm.
For the tree, **state** which algorithm was used.

DFS ✓



(b) Iterative depth-first search 2 (2)

Name an advantage that iterative depth-first search respectively has over breadth-first search and normal depth-first search.

Breadth-first search disover node layer by layer. Imagine b - branching factor, d - depth of the result and m - max tree depth. The BFS has $O(b^{1+d})$ time and memory complexity. The DFS discover nodes with random choosing the child till the tree leaves. It has $O(b^m)$ time complexity but $O(bm)$ memory complitity. The DFS can goes so far from the destination which in another subtree that is why the iterative DFS is used. It uses the same memory complexity as DFS($O(bm)$), but faster than DFS and does not go deep into the wrong subtree. The complexity is $O(b^d)$. ✓

(c) **A*Star** 4,5 (6)

Apply the A*-Algorithm (A-Star) to the following graph. The **Heuristic (H)** is given in each node. The search starts in node **W** and finds the shortest path to node **C**.

For each step of the A*-algorithm, state the nodes which still have to be searched as well as their scores and mark the node which will be searched in the next step. Finally, state the correct order of the nodes from the shortest path as calculated by the algorithm. You can use the table below.

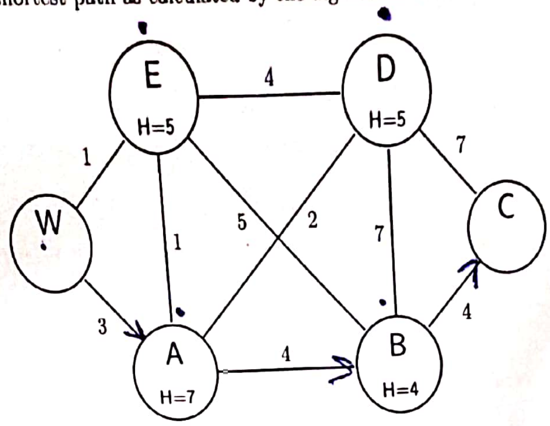


Table 1: Table for the A*-Algorithm

Step	Successors	Costs
1	W	W(0)
2	E, A	E(6), A(10)
3	A, B, D	A(9), D(10), B(10)
4	B, D	B(10), D(10)
5	D, C	D(10), C(11)
6	C	C(11) C(10)

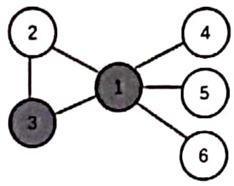
W - E - A - B - C

4. **Complex Search**

(a) Genetic Algorithms for Vertex Cover

Vertex Cover describes the problem to find a smallest possible subset set $S \subset V$ of the nodes of a graph $G = (V, E)$, such that for each edge of the graph at least one of the end nodes of the edge are contained in S. The goal now is to develop a genetic algorithm for the Vertex Cover problem.

The following figure shows an example for an optimal solution for Vertex Cover. $S = \{1, 3\}$ covers all edges of the graph with only two selected nodes.



i. Describe (using words) a possible fitness function (performance measure) for the given problem, to find a smallest possible subset of nodes such that all edges are covered. 2 (2)

The fitness function will be the number of edges connected with the node.

ii. State the four phases of a genetic algorithm. For each phase, describe how the phase could be adopted for the given problem. 3 (4)

~~Selection (randomly select the edges~~
~~Recombine~~
~~Fix~~
~~Random Mutation~~

Selection (randomly select the nodes by the fitness function)

Recombine (take the row of the adjacency matrix as DNA code)

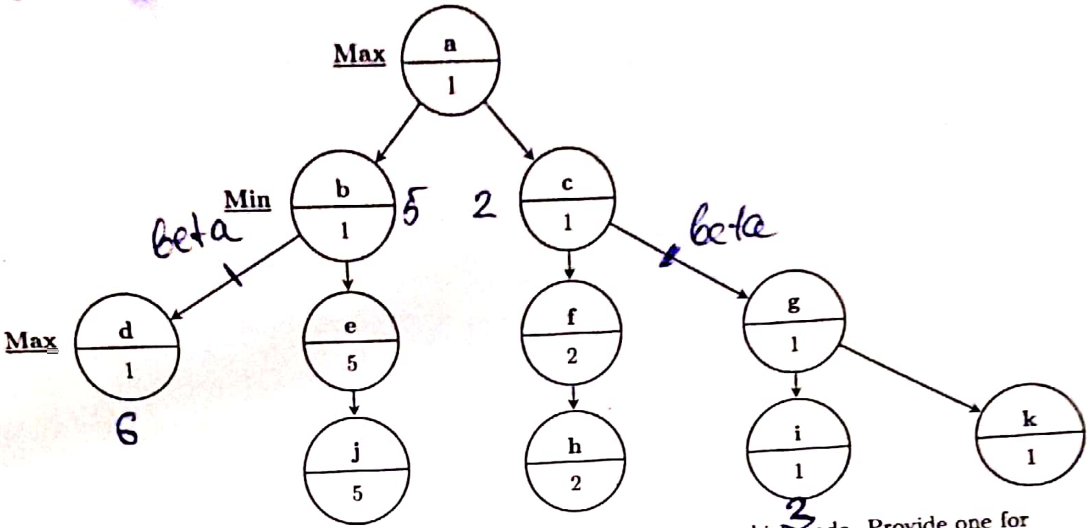
Fix the genes

Random mutation

CamScanner

6/(6)

5. **Antagonistic Search and Games - Alpha/Beta Pruning**



Specify which nodes you need to set to what value in order to eliminate which node. Provide one for each Alpha and Beta Cut. Use the table below. Also indicate the type of the cut.

Node	New Value	eliminated	Type of Cut
i	3	g	beta
d	6	d	~~beta~~ alpha

CamScanner

6. **Constraint Satisfaction Problems**

2/ (2)

(a) What is the difference between edge constistency and path consistency for CSPs?

Edge consistency works only for one vs one nodes ✓

Path consistency have more than two nodes

and works for ~~each~~ all transitive pairs: ✓

(a, b, c): consistency (a, b), consistency (b, c)

3,5/ (4)

(b) **Cryptoarithmetic Puzzle**

Given the following cryptoarithmetic puzzle:

$$\begin{array}{cccc} & O & N & E \\ + & O & N & E \\ \hline & T & W & O \end{array}$$

Specify the problem completely as a CSP.

$o, n, e, t, w \in \{0, \dots, 9\}$ ✓

$x_1, x_2 \in \{0, 1\}$

$E + E = O + \cancel{10x_1} + 10X_1$

$X_1 + n + n = W + 10 \cdot X_2$ ✓

$X_2 + O + O = T$ ✓

$Alldif(O, n, e, T, W)$. (✓)

$\forall\ O, T \neq 0$

CamScanner

A	B	A ↔ B	A → B	B → A	(B → A) ∧ (A → B)
T	F	T	T	T	F
T	F	F	F	T	F
F	T	F	T	F	
F	F	T	T	T	T

7. **Logical Agents** (4)

(a) **Biconditional Elimination**

Proof the following equivalency

$$(A \Leftrightarrow B) \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$$

4

$A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$

We can use such a formula for ~~opening~~ simplifying the formula or use the table of values to show that it is equal.

A	B	A ↔ B
T	T	T
T	F	F
F	T	F
F	F	T

As we se that the values in the table are evual, so the formulas are also equal.

(b) **Resolution** (6)

Show using resolution in the propositional logic that the following formula F is universally true.

$F = \neg B \vee (A \wedge D) \vee (\neg A \wedge E \wedge B) \vee (\neg A \wedge C) \vee (A \wedge \neg D) \vee (\neg E \wedge \neg C)$

5/6

$\neg B = \text{True}: F = \text{True}$

$\neg B = \text{False}: (A \wedge D) \vee (\bar{A} \wedge E) \vee (\bar{A} \wedge C) \vee (A \wedge \bar{D}) \vee (\bar{E} \wedge \bar{C})$

$A = \text{True}: D \vee \bar{D} \vee (\bar{E} \wedge \bar{C}) = \text{True}$

$A = \text{False}: E \vee C \vee (\bar{E} \wedge \bar{C})$

$E = \text{True}: \text{True}$

$E = \text{False}: C \vee \bar{C}: \text{True}$

So in all variants we have True resuelt for this formula.

Yes, but that is not resolution!

8. **First order logic**

(a) Fomulate the following sentence as a formula in the first-order logic. Use only the given predicates. (2)

Sentence	Predicates	Meaning
There is sombody, who doesn't like cake	likesCake(x)	x likes cake

1

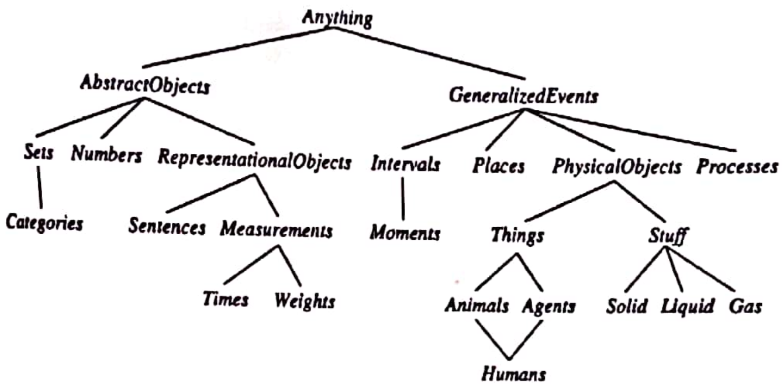
$\forall x \neg \text{likesCacke}(x)$

$\exists$

9. **Knowledge representation**

(a) **Ontologies**

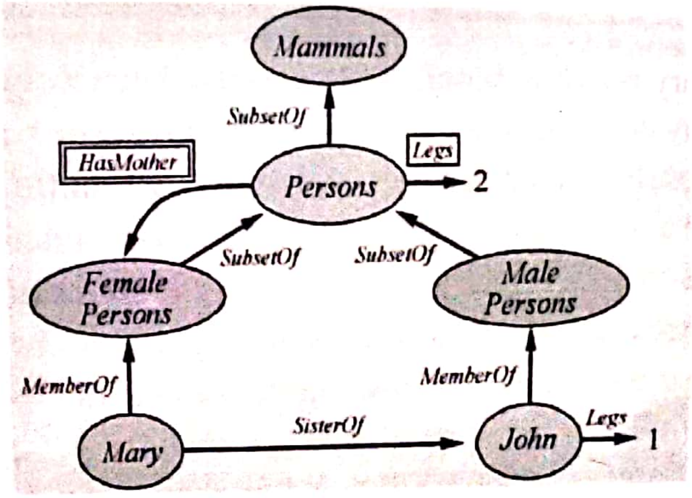
Specify for the object "Football" the most specific fitting category for the two given, distinct contexts from the following upper ontology :



i. Context 1: You want to buy a football in a sporting goods store (1)

ii. Context 2: "We're going to play in the afternoon today. Bring your football with you." (1)

(b) **Semantic Networks**



i. State, how Default-Reasoning works in semantic networks (In your answer, you can refer to the example semantic net in the figure above. (2)

ii. State, how Default Reasoning is represented in the Default Logic. (2)

CamScanner

016

10. **Planning**

Planning problems in which an initial state is to be transferred to a target state with a sequence of actions usually have a very large search space. Planning algorithms based on backward chaining seem to have an advantage over forward chaining because they limit the search space more. Nevertheless, algorithms with forward chaining are more efficient because one can automatically generate strong heuristics for pruning the search space.

(a) Specify two different types of such heuristics and explain them using the example of the following action (the semantics of the action do not matter here; it's just about the structure) **(3)**

```
(Action (a, b, c),
    PRECOND: On (a, b) ∧ Tile (a) ∧ Blank (c) ∧ Adjacent (b, c)
    Effect: On (a, c) ∧ Blank (b) ∧ ¬On(a, b) ∧ ¬Blank (c) )
```

(b) Another idea to improve efficiency is hierarchical planning. State, what kind of knowledge is required for that and why the search space is drastically restricted by this. **(3)**