Last Name: First Name:

Matriculation Number:

Exam
Artificial Intelligence 1

April 4, 2024

Please ignore the QR codes; do not write on them, they are for grading support

	To be used for grading, do not write here													
prob.	1.1	2.1	2.2	2.3	3.1	4.1	4.2	5.1	5.2	5.3	6.1	7.1	7.2	Sum	grade
total	10	8	6	7	6	5	6	6	8	7	7	6	8	90	
reached															

Organizational Information

Please read the following directions carefully and acknowledge them with your signature.

1. Please place your student ID card and a photo ID on the table for checking.
2. You can reach 90 points if you fully solve all problems. You will only need 85 points for a perfect score, i.e. 5 points are bonus points.
3. No resources or tools are allowed except for a pen.
4. You have 90 min (sharp) for the exam.
5. Write the solutions directly on the sheets, no other paper will be graded.
6. If you have to abort the exam for health reasons, your inability to sit the exam must be certified by an examination at the University Hospital. Please notify the exam proctors and have them give you the respective form.
7. Please make sure that your copy of the exam is complete (14 pages excluding cover sheet and organizational information pages) and has a clear print. **Do not forget to add your personal information on the cover sheet and to sign this declaration.**

Declaration: With my signature I certify having received the full exam document and having read the organizational information above.

Erlangen, April 4, 2024 ....
(signature)

Organisatorisches

Bitte lesen die folgenden Anweisungen genau und bestätigen Sie diese mit Ihrer Unterschrift.

1. Bitte legen Sie Ihren Studierendenausweis und einen Lichtbildausweis zur Personenkontrolle bereit!
2. Sie können 90 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 85 Punkte bereits als volle Punktzahl, d.h. 5 Punkte sind Bonuspunkte.
3. Es sind keine Hilfsmittel erlaubt außer einem Stift.
4. Die Bearbeitungszeit beträgt genau 90 min.
5. Schreiben Sie die Lösungen direkt auf die ausgeteilten Aufgabenblätter. Andere Blätter werden nicht bewertet.
6. Wenn Sie die Prüfung aus gesundheitlichen Gründen abbrechen müssen, so muss Ihre Prüfungsunfähigkeit durch eine Untersuchung in der Universitätsklinik nachgewiesen werden. Melden Sie sich in jedem Fall bei der Aufsicht und lassen Sie sich das entsprechende Formular aushändigen.

7. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (14 Seiten exklusive Deckblatt und Hinweise) und einwandfreies Druckbild! **Vergessen Sie nicht, auf dem Deckblatt die Angaben zur Person einzutragen und diese Erklärung zu unterschreiben!**

Erklärung: Durch meine Unterschrift bestätige ich den Empfang der vollständigen Klausurunterlagen und die Kenntnisnahme der obigen Informationen.

Erlangen, April 4, 2024 ....
(Unterschrift)

Please consider the following guidelines to avoid losing points:

- If you continue an answer on another page, clearly give the problem number on the new page and a page reference on the old page.
- If you do not want something to be graded, clearly cross it out. Adding a wrong statement to a correct solution may lead to deductions.
- The instructions “Give X”, “List X” or similar mean that only X is needed. If you additionally justify your answer, we will try to give you partial credit for a wrong answer (but there is no guarantee that we will).
- The instruction “Assume X” means that X is information that you may use in your answer.
- The instruction “Model X as a Y” means that you have to describe X formally and exactly as an instance of Y using the definition of Y from the lecture.
- If you are uncertain how long or complex an answer should be, use the number of points as an indication: 1 point roughly corresponds to 1 minute.
- In all calculation questions, you have to simplify as much as reasonably possible without a calculator. For example, $\log 2$ or 3^7 should not be calculated, but $0.4 \cdot 0.3 \cdot 0.5 = 0.06$ should be.

1 Prolog

Problem 1.1 (Analyzing a Prolog Program)
Consider the following Prolog program:

```
1 foo([X|_],X).
2 foo([_|L],X) :- foo(L,X).
3
4 good([], _, _).
5 good([X|Xs], Ys, XYs) :- goodX(X, Ys, XYs), good(Xs, Ys, XYs).
6
7 goodX(X, Ys, [pair(X,Y)|_]) :- foo(Ys, Y).
8 goodX(X, Ys, [_|XYs]) :- goodX(X, Ys, XYs).
```

1. Give a value for `xys` such that the query `good([1,2], [1,2], xys)` returns `true`.

   3 Points

2. Which definition from the course does the program `good(Xs,Ys,XYs)` implement?

   3 Points

For the remaining questions, assume we swap the lines 1 and 2.
Explain (in about 2 sentences each) how this change effects program's of the form `foo(s,t)` regarding

3. …correctness?

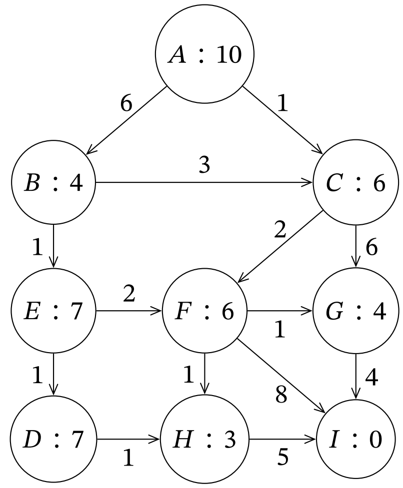
   2 Points

4. …efficiency?

   2 Points

2 Search

Problem 2.1 (Search Algorithms)
Consider the following *directed graph*:



Every *node* is *labeled* with $n : h(n)$ where n is the identifier of the *node* and $h(n)$ is the *heuristic* for estimating the *cost* from n to a *goal node*.
Each node's children are ordered **alphabetically**.
Every *edge* is *labeled* with its actual *cost*.

Assume you have already *expanded* the *node* A. List the **next** 4 ***nodes*** **(i.e., excluding** A**)** that will be *expanded* using the respective *algorithm*.
If there is a tie, break it using **alphabetical order**.

1. *Depth-first search* 1 Points

2. *Breadth-first search* 1 Points

3. *uniform-cost search* 2 Points

4. *greedy search* 2 Points

5. A^**-search* 2 Points

Problem 2.2 (Search in an Infinite Tree)

Consider an infinite tree defined as follows:

- The root has 1 child.
- Every other node has one more child than its parent.

Explain (in about 2 sentences each) what pitfalls we have to watch for (e.g., correctness, complexity, completeness, …) when searching in this tree using …

1. …depth-first search.

3 Points

2. …breadth-first search.

3 Points

Problem 2.3 (Search Problems)
Consider the family of search problems P_n given for $n = 1, 2, \ldots$ by $\langle S, A, T, I, G \rangle$ where

- $S = \{0, 1, \ldots, n\}$
- $A = \{f, b, s\}$
- T is given by
  - $T(f, x) = \{x \oplus 2\}$
  - $T(b, x) = \{x \ominus 2\}$
  - $T(s, x) = \{x^2\} \cap S$

  where $\oplus$ and $\ominus$ are addition/subtraction modulo n.
- $I = \{0\}$
- $G = \{n - 1\}$

1. For $n = 19$, give the result of applying the action sequence f, f, s in state 0.

   2 Points

2. Under what circumstances does this problem have a solution?

   2 Points

3. For $n = 15$, explain what is special about applying the action sequence f, f, s to state 0.

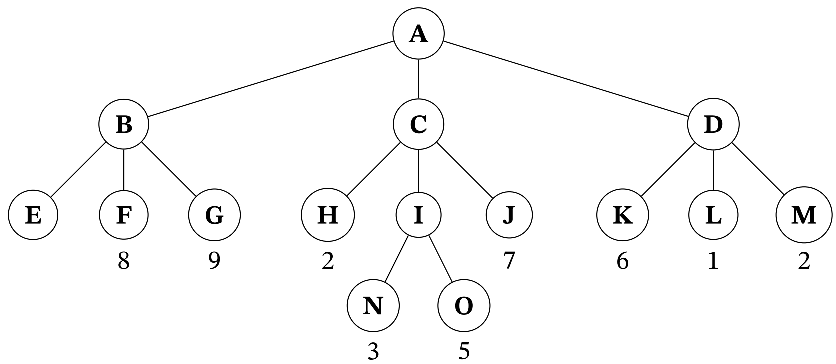
   2 Points

4. Explain (in 1 sentence) why this problem represents a fully observable environment.

   1 Points

3 Adversarial Search

Problem 3.1 (Minimax)
Consider the following *minimax* game tree for the **maximizing** player's turn. The *values* at the *leaves* are the static *evaluation function values* of those *states*; some of those *values* are currently missing.



1. Label the *nodes* I and C with their *minimax* value. 2 Points

2. If possible, label the *node* **E** with an *evaluation function value* that results in the player definitely choosing move **C** (no matter how ties are broken).
   Otherwise, argue why that is impossible. 2 Points

3. Now assume **E** is *labeled* with 5, and we use *αβ-pruning*. We expand *child nodes* in alphabetical order. Which nodes would be *pruned*? 2 Points

4 Constraint Satisfaction

Problem 4.1 (Assignments and Solutions)
Consider the CSP given by

- Variables: $\{a, b, c, d\}$
- Domains: $D_a = D_b = \{0, 1, 2, 3\}$ and $D_c = D_d = \{0, 1, 2, 3, 4, 5\}$
- Constraints:
  - ☐ $a < b$
  - ☐ $b < d - c$
  - ☐ $d - a > 3$
  - ☐ $d = 2c$

1. Check the constraints that make this CSP non-binary. 1 Points

2. Give the unique *solution* to this *CSP*. 2 Points

3. Check the true statements: 2 Points
   - ☐ The *assignment* $a = 0$, $b = 1$ is *consistent*.
   - ☐ The *assignment* $a = 0$, $b = 1$, $c = 2$, $d = 3$ is *consistent* and *total*.
   - ☐ The *assignment* $b = 0$, $c = 2$, $d = 4$ is *consistent* and *total*.

Problem 4.2 (Relating CSP and SAT)
Like in the homework problem, we want to relate CSP and SAT.
Assume a binary CSP with variables $X_1, \ldots, X_n$, a finite domain D_i for every X_i, and a constraint $C_{ij} \subseteq D_i \times D_j$ for every $1 \leq i < j \leq n$. A solution of this CSP instance is an assignment α mapping each X_i to a value in D_i.

1. Give an instance of SAT, i.e., a set P of propositional variables and a propositional formula F over P, that is equivalent to this CSP problem. 4 Points

2. To show the equivalence, for every assignment α for the CSP instance, give an assignment φ for the SAT instance defined in the previous subproblem such that F is satisfied under φ iff all constraints are satisfied under α. 2 Points

5 Logic

Problem 5.1 (Propositional Logic)
Consider the formula $A = ((p \vee q) \Rightarrow (p \wedge q)) \wedge ((p \vee \neg q) \Rightarrow (p \vee \neg r))$ using propositional variables p, q, r.

1. Give all satisfying assignments for A. 2 Points

2. Give a formula in DNF that is equivalent to A. 2 Points

3. Turn A into a theorem by replacing exactly one occurrence of a connective with a different connective. 2 Points

Problem 5.2 (Modeling in First-Order Logic)
Consider the following situation:

- Some individuals are persons, some are animals.
- Persons and animals may like other persons or animals.
- Alice is a person, and she likes the animal Bubbles.
- For every person or animal, we can obtain its mother.

1. Model this situation in first-order logic by giving a signature, i.e., a list of function/predicate symbols with arity. 4 Points

2. State formulas over your signature that capture the following properties: 2 Points
   1. Every individual is a person or an animal but not both.

   2. Every mother likes their offspring.

3. Give a model over your signature, i.e., a domain D and an interpretation $I(s)$ for every function/predicate symbol, in which all properties from the above subproblem hold. 2 Points

Problem 5.3 (Tableaux Calculus)
Consider the following tableau where $A = (p \wedge q) \vee (\neg p \wedge \neg q)$.

$$
\begin{array}{c}
A^F \\ \hline
(p \wedge q)^F \\
(\neg p \wedge \neg q)^F \\
\begin{array}{c|c}
\begin{array}{c} p^F \\ \begin{array}{c|c} (\neg p)^F & (\neg q)^F \\ p^T & q^T \end{array} \end{array}
&
\begin{array}{c} q^F \\ \begin{array}{c|c} (\neg p)^F & (\neg q)^F \\ p^T & q^T \end{array} \end{array}
\end{array}
\end{array}
$$

1. How can you tell that this tableau is saturated?

2 Points

2. Explain how, using the tableau, we can find all satisfying assignments for A.

2 Points

3. Give the fully saturated tableau for the root A^T.

3 Points

6 Knowledge Representation

Problem 6.1 (ALC)
Consider the following description logic signature

- concept symbols: i (for instructor), s (for student), c (for course), p (for program)
- role symbol m (for is-member-of) used for
  - instructors giving a course
  - students taking a course
  - students being enrolled in a program
  - courses being part of a program

We use an extension of ALC, in which there are dual roles: there is a role m^{-1} that captures the relation has-as-member, e.g., $MK\ m\ AI$ iff $AI\ m^{-1}\ MK$.

1. Give an axiom for the above signature that captures that instructors can only be members of courses. 1 Points

2. Give an axiom for the above signature that captures: courses that are taken by a student, must be given by an instructor. 2 Points

3. Calculate the translation to first-order logic of $s \sqsubseteq (\forall m.\exists m.p)$. 2 Points

4. Given a model $\langle D, [\![-]\!]\rangle$, define an appropriate case of the interpretation mapping for the formula $\forall r^{-1}.C$. 2 Points

7 Planning

Problem 7.1 (Admissible Heuristics in Gripper)
Consider the following situation from the homeworks:

- We have two rooms, A and B, one robot initially located in room A, and n balls that are initially located on the floor in room A.
- The goal is to have all balls on the floor in room B.
- The robot can move between the rooms and has a gripper for picking up and releasing balls. Moving, picking up, and releasing are one action each.

Given state s, we write s_A and s_B for the number of balls on the floor in room A or B, i.e., $n - s_A - s_B$ is the number of balls held by the robot.

For each of the following heuristics, argue (by informal proof or counter-example) whether they are admissible.

1. Assume the gripper can hold up to **1 ball** at a time.
   Potential heuristic: $h(s) = n - s_B$. 2 Points

2. Assume the gripper can hold up to **1 ball** at a time.
   Potential heuristic: $h(s) = 4 \cdot s_A$. 2 Points

3. Assume the gripper can hold up to **2 balls** at a time, which are picked up **in one action** and released in one action. 2 Points
   Potential heuristic: $h(s) = n - s_B$.

Problem 7.2 (Relaxation)
We want to solve a STRIPS planning task.

1. Explain (in about 2 sentences) the purpose of relaxed planning. 2 Points

2. Explain (in about 2 sentences) why it is bad to relax too much or too little. 2 Points

Now consider a concrete task given by a finite set B of size n and

- facts: $inA(b)$, $inB(b)$, $held(b)$ for $b \in B$, $Rfree$, $RinA$, $RinB$
- actions

action	precondition	add list	delete list
$move_A$	$RinB$	$RinA$	$RinB$
$move_B$	$RinA$	$RinB$	$RinA$
$pickup(b)$	$RinA$, $Rfree$, $inA(b)$	$held(b)$	$Rfree$, $inA(b)$
$release(b)$	$RinB$, $held(b)$	$inB(b)$, $Rfree$	$held(b)$

- initial state I: $inA(b)$ for $b \in B$, $Rfree$, $RinA$
- goal state: $inB(b)$ for $b \in B$

3. Let h^+ be the heuristic obtained from the delete-relaxation. Give the value of $h^+(I)$. 2 Points

4. Let h^+ be the heuristic obtained from the only-adds-relaxation. Give the value of $h^+(I)$. 2 Points

This page was intentionally left blank for extra space