Last Name: First Name:

Matriculation Number:

Retake Exam
Artificial Intelligence 1

October 9, 2023

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	3.1	4.1	4.2	5.1	5.2	5.3	6.1	7.1	7.2	Sum	grade
total	8	10	9	6	7	7	6	6	7	8	10	6	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, October 9, 2023 ....
(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, October 9, 2023 ....

(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.

1 Prolog

Problem 1.1 (Prolog in Prolog)

Consider the following Prolog program that represents Prolog in Prolog, i.e. Prolog terms, literals, and clauses are represented as Prolog terms:

```
1  isTerm(pterm(F,ARGS)) :- string(F), isTermList(ARGS).
2  isTerm(pvar(X)) :- string(X).
3
4  isTermList([]).
5  isTermList([H|T]) :- isTerm(H), isTermList(T).
6
7  isLiteral(plit(P,ARGS)) :- string(P), isTermList(ARGS).
8
9  isLiteralList([]).
10 isLiteralList([H|T]) :- isLiteral(H), isLiteralList(T).
11
12 isClause(pclause(H,B)) :- isLiteral(H), isLiteralList(B).
```

Here `string` is a built-in predicate that succeeds if its argument is a string.

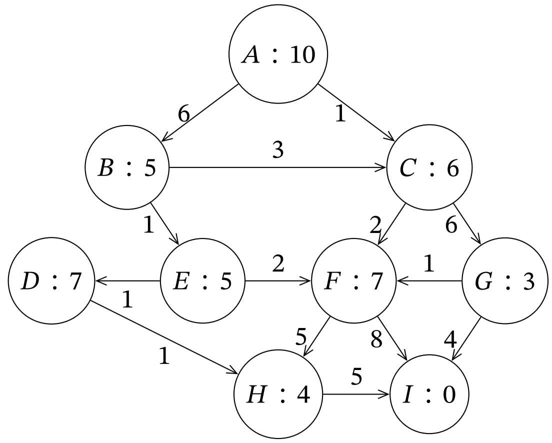
1. Write the Prolog clause `isNat(succ(N)) :- isNat(N)` as a Prolog term relative to the above program (i.e., such that `isClause` succeeds for it). 3 Points

2. Assume that the Prolog term C contains no free variables. How is the result of the query `isClause(`C`)` affected by exchanging the lines 4 and 5? 2 Points

3. Extend the program above with a unary Prolog predicate `isProgram` that succeeds if its argument is of the form `pprog(`P`)` where P is a list of clauses. 3 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. Every edge is labeled with its actual cost.

1. Assume that I is the goal node. Argue whether or not the heuristic is admissible. 2 Points

Now 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.

2. depth-first search 1 Points

3. breadth-first search 1 Points

4. uniform-cost search 2 Points

5. greedy-search 2 Points

6. A^*-search 2 Points

Problem 2.2 (Search Problems)
Consider the search problem $\langle S, A, T, I, G\rangle$ where

- ☐ $S = \mathbb{Z} \times \mathbb{Z}$
- ☐ $A = \{R, S, M\}$
- ☐ T is given by
  - ☐ $T(R, (x, y)) = \{(x, 0), (0, y)\}$
  - ☐ $T(S, (x, y)) = \{(y, x)\}$
  - ☐ $T(M, (x, y)) = \{(x + 1, y)\}$
- ☐ $I = \{(0, 0)\}$
- ☐ $G = \{(3, 3)\}$

1. Check the box of the part of the definition that makes this problem fully observable. 1 Points

2. Give the possible states resulting from applying the action sequence M, R, M to the initial state. 2 Points

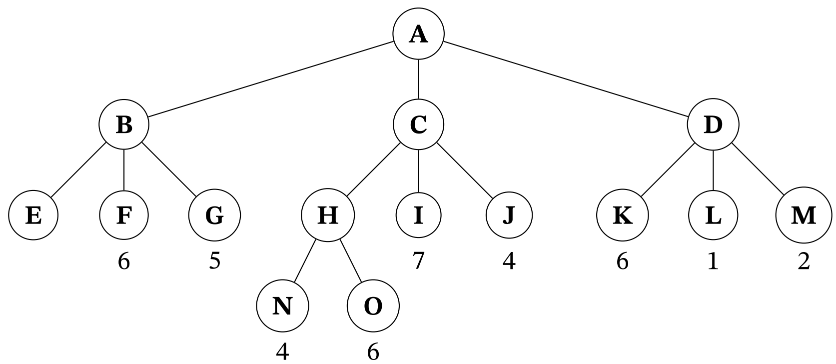
3. Which states are reachable from the initial state? 2 Points

4. Give a solution of minimal length to this problem. 2 Points

5. Assume we use $h((x, y)) = 1/(1 + x + y)$ as a heuristic. Which actions will a greedy search algorithm choose for the first two steps? 2 Points

3 Adversarial Search

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



1. Label the nodes H and C with their minimax values. 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 $\alpha\beta$-pruning. We expand child nodes in alphabetical order. Which nodes would be pruned? 2 Points

4 Constraint Satisfaction/Propagation

Problem 4.1 (Modeling) 7 Points

You want to schedule a tournament in which teams A, B, C, D play each other once. The six games must take place over the next 3 days. But team A must not play twice on the same day. Team B is only available for the next 2 days. And team C wants to play against D a day before playing against anybody else.

Model this problem as a constraint satisfaction problem $\langle V, D, C\rangle$. Explain how the solutions correspond to the possible match schedules.

Problem 4.2 (Solving)
Consider the following binary CSP:

- $V = \{a, b, c, d, e\}$
- $D_a = D_b = D_c = \{0, 1, 2, 3\}$, $D_d = D_e = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- Constraints:
  - $e^2 - d^2 < 18$
  - $d = 2c$
  - $e - a > 5$
  - $a < b$ and $b < c$ and $d < e$

1. Check the boxes for (v, w) if v is arc-consistent relative to w. 2 Points

   ☐ (a, b) ☐ (b, a) ☐ (a, e) ☐ (e, a) ☐ (c, d) ☐ (d, c)

2. Give the **three** solutions. 3 Points

3. Now assume we replace the last constraint with $b < \min\{c, d\}$ (where min is the minimum operator). Transform the resulting problem into an equivalent binary one. 2 Points

5 Logic

Problem 5.1 (Propositional Logic)
We use the propositional variables X, Y, and Z. Consider the formula A given by

$$(X \wedge (Y \Rightarrow Z)) \Rightarrow \neg(X \wedge Y)$$

1. Give a satisfying assignment σ and a falsifying assignment φ for A.

3 Points

2. Which (if any) of the formulas A and $\neg A$ is a theorem?

1 Points

3. Give the shortest formula in CNF that is equivalent to $A \Rightarrow A$.

2 Points

Problem 5.2 (Predicate Logic)
Consider the following signature of predicate logic:

- binary function symbol f
- unary predicate symbol p

1. Give a model for that signature. 2 Points

2. Consider the formula $A = p(x) \wedge p(y)$ and assume a model with universe $\mathbb{N}$ and $\mathcal{I}(p) = \{0\}$. Give an assignment α such that $\mathcal{I}_\alpha(A)$ holds. 2 Points

3. Prove or refute the following statement: A model that satisfies $\forall x.p(x)$ satisfies all formulas. 2 Points

Problem 5.3 (Proving in Tableau Calculus) 7 Points

We use the propositional variables P, Q, and R and the following abbreviations

$$A = Q \wedge (P \Rightarrow Q)$$

$$B = P \Rightarrow A$$

$$C = P \Rightarrow R$$

Using the tableau calculus, find a falsifying assignment for the formula $C \Rightarrow B$.

6 Knowledge Representation

Problem 6.1 (Description Logic)
Consider the following $\mathcal{ALC}$-setting:

- concepts: `pizza`, `icecream`, `food`, `topping`
- relations: `canHaveTopping`
- individuals: `margarita`, `vanilla`, `ham`, `syrup`

You may abbreviate every concept/relation/individual by its first letter.

1. Give the $\mathcal{ALC}$-ABox with assertions that model common sense knowledge (e.g., we do not put vanilla or syrup on pizza even though it is technically possible). 2 Points

2. Give an $\mathcal{ALC}$-TBox with two axioms expressing the following: 2 Points
   - All food is pizza or icecream.
   - Only pizza can have toppings.

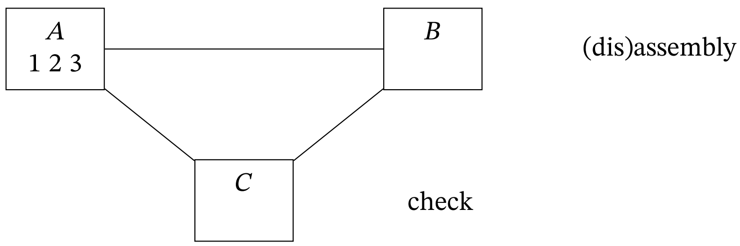
3. Give an $\mathcal{ALC}$-TBox in which the concept `pizza` is inconsistent. 2 Points

4. Give the result of translating the following formula to first-order logic: $(\forall c.i) \sqsubseteq (p \sqcap t)$ 2 Points

This page was intentionally left blank for extra space

7 Planning

Problem 7.1 (STRIPS)
Consider a machine that processes objects $Obj = \{1, 2, 3\}$, which can be at location A, B, or C. Currently all objects are at location A and **unchecked** and **assembled**. Eventually all objects are needed in location A and **checked** and **assembled**.
At location B, objects can be assembled or disassembled. At location C, disassembled objects can be checked.
A transport system is available that can move **exactly two objects at a time** from one location to any other location.



We formalize this problem as a STRIPS task where the **facts** are

- position $l \in \{A, B, C\}$ of object $o \in Obj$: $\mathtt{at}(l, o)$
- state of object $o \in Obj$: $\mathtt{isCh}(o)$ (checked), $\mathtt{isAss}(o)$ (assembled), and $\mathtt{isDis}(o)$ (disassembled)

and the **actions** are given by table below for any $l, m \in \{A, B, C\}$ and $o, p \in Obj$

action	precondition	add list	delete list
$\mathtt{move}(l, m, o, p)$	$\mathtt{at}(l, o)$, $\mathtt{at}(l, p)$	$\mathtt{at}(m, o)$, $\mathtt{at}(m, p)$	$\mathtt{at}(l, o)$, $\mathtt{at}(l, p)$
$\mathtt{assemble}(o)$	$\mathtt{at}(B, o)$	$\mathtt{isAss}(o)$	$\mathtt{isDis}(o)$
$\mathtt{disassemble}(o)$	$\mathtt{at}(B, o)$	$\mathtt{isDis}(o)$	$\mathtt{isAss}(o)$
$\mathtt{check}(o)$	$\mathtt{at}(C, o)$, $\mathtt{isDis}(o)$	$\mathtt{isCh}(o)$	

1. Give the initial state I and the goal G. 2 Points

2. Give the state after applying the two actions $\mathtt{move}(A, B, 1, 2)$ followed by $\mathtt{disassemble}(1)$. 2 Points

3. Give the value $h^*(I)$. 2 Points

4. Give the value $h^+(I)$. 2 Points

5. Consider the heuristics h that computes $h(s)$ as $2a + d + 3$ where a is the number of unchecked assembled and d the number of unchecked disassembled objects in state s. Argue whether h is admissible. 2 Points

Problem 7.2 (Planning Complexity)

1. What is the difference between satisficing and optimal planning? 2 Points

2. Give the named complexity class (e.g., P, NP, etc.) of deciding the existence of a plan for a STRIPS problem. 1 Points

3. Now we consider only STRIPS problems in which all delete lists are empty. Show that the number of facts is an upper bound for the length of an optimal plan. 3 Points