Last Name: First Name:

Matriculation Number:

Retake Exam
Artificial Intelligence 1

September 23, 2025

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	5.1	5.2	5.3	6.1	7.1	7.2	Sum	grade
total	10	8	6	7	8	9	8	8	6	8	8	4	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, September 23, 2025 .....
(signature)

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.
- You can always ask for the translation or explanation of a non-technical word.
- 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 may or may not give you partial credit.
- The instruction "Assume X" means that X is information that your answer may use.
- 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.

This page was intentionally left blank for extra space

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:

```
isTerm(pterm(F,ARGS)) :- string(F), isTermList(ARGS).
isTerm(pvar(X)) :- string(X).

isTermList([]).
isTermList([H|T]) :- isTerm(H), isTermList(T).

isLiteral(plit(P,ARGS)) :- string(P), isTermList(ARGS).

isLiteralList([]).
isLiteralList([H|T]) :- isLiteral(H), isLiteralList(T).

isClause(pclause(H,B)) :- isLiteral(H), isLiteralList(B).
```

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

1. Explain intuitively what the predicate `isClause(pclause(H,B))` computes? 2 Points

2. 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

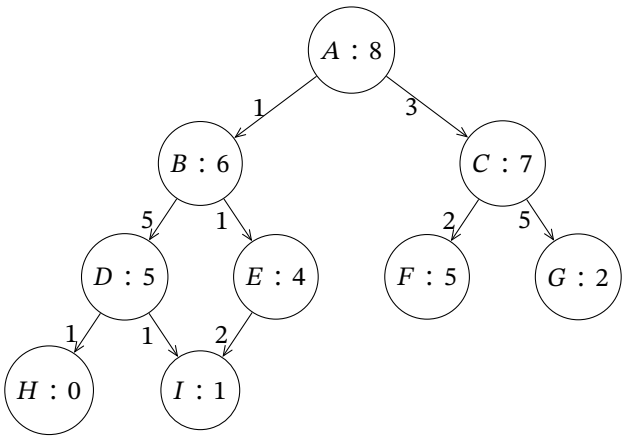
3. Assume that the Prolog term C contains no free variables. How is the result of the query `isClause(`C`)` affected by exchanging the lines 4 and 5? 2 Points

4. Extend the program above with a unary Prolog predicate `isProgram` that succeeds if its argu- 3 Points

ment is of the form `pprog(`P`)` where P is a list of clauses.

2 Search

Problem 2.1 (Search Algorithms)
Consider the graph below. Every edge is labeled with its cost.
Each node is labeled with $n : h(n)$ where n is its name and $h(n)$ its value for heuristic h.



Assume you have already expanded the root node A. In each problem below, list the **next** 4 nodes that will be expanded using the respective search algorithm and (if needed) heuristic h.
If there is a tie, break it using alphabetical order of the nodes.

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 (Remembering expanded states)
When searching in a graph, it can happen that the same node can be reached along two different paths. When inserting a node n into the fringe, the search algorithms presented in the course do not check if n has already been expanded before.

1. In about 1 sentence each, explain the advantage and disadvantage of additionally performing such a check. 3 Points

2. Now assume we have implemented such a check. We consider only *breadth-first* and *uniform-cost* search. 3 Points
   Explain in about 2 sentences whether it is correct to simply skip inserting n into the fringe if n has been expanded before.

Problem 2.3 (Search Problem)
Consider the deterministic search problem $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{I}, \mathcal{G} \rangle$ where

- $\mathcal{S} = \mathbb{Z}$
- $\mathcal{A} = \{-6, -4, 0, 5\}$
- $\mathcal{T}(a, s) = \{a + s\}$
- $\mathcal{I} = \{0\}$
- $\mathcal{G} = \{9\}$

1. Give the state resulting from applying the action sequence $-4, 5, 5$ to the initial state.

1 Points

2. Give a solution to the problem.

3 Points

3. Explain in about 2 sentences whether DFS a good choice for this problem.

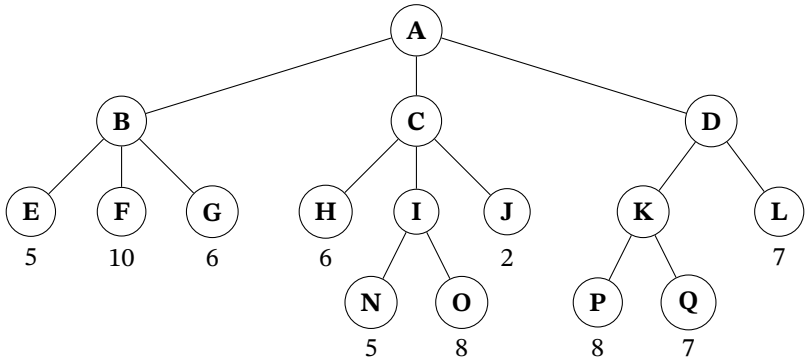
2 Points

4. If we change the set $\mathcal{A}$ to $\{-6, -4, 0, 2\}$, the problem changes substantially. In what way?

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.



1. Which move will the player choose?

2 Points

2. What change to the label of only a **single** node can you make such that the player chooses a different move?

2 Points

For the remaining questions: Use $\alpha\beta$-pruning and expand child nodes in alphabetical order.
Ignore any change you may have applied in the previous problem.

3. Which nodes would be pruned?

2 Points

4. What change to the labels of the nodes N and O can you make such that J is pruned but not I?

2 Points

4 Constraint Satisfaction

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

- Variables: $\{a, b, c, d\}$
- Domains: $D_a = D_b = D_c = D_d = \{0, 1, 2, 3\}$
- Constraints:
  - ☐ $a + b = 3$
  - ☐ $2 \leq c \cdot d < 4$
  - ☐ $a + c > 4$
  - ☐ $2a + b + c > 7$

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

2. Give all solutions. 3 Points

3. By testing arc-consistency only for the third constraint, which domain values can be eliminated immediately? 2 Points

4. After assigning $a = 2$, what is the domain of c after applying forward-checking using the third constraint? 1 Points

5. Give an equivalent binary CSP. 2 Points

5 Logic

Problem 5.1 (Propositional Logic)
Consider the formula A of propositional logic given by

$$(p \wedge q) \Rightarrow \big((p \Rightarrow \neg q) \Rightarrow q\big)$$

1. Give the propositional variables that occur in A.

2 Points

2. Give all assignments that satisfy A.

2 Points

3. Give a DNF for A.

2 Points

4. What is the **smallest** number n for which we can give n formulas such that every other formula in p and q is equivalent to one of them?

2 Points

Problem 5.2 (First-Order Logic)
Consider the formula A of first-order logic with equality given by

$$(\forall x.\exists y.p(x,y)) \Rightarrow \forall x.p(x,f(x))$$

over the smallest signature that makes it well-formed.

1. Give that signature. 2 Points

2. Give a model that satisfies A. 2 Points

3. Now assume $\forall x.\exists y.p(x,y)$ is the only axiom. Give a second axiom such that the models (U,I) that satisfy both axioms are exactly the ones where $I(p)$ (which is a binary relation on U) also represents a unary function from U to U. 2 Points

4. In about 1 sentence, explain why the dual of A (i.e., with $\Leftarrow$ instead of $\Rightarrow$) is a theorem. 2 Points

Problem 5.3 (Proving in Natural Deduction)

1. Prove the formula

$$\big(\forall x.p(x, f(x))\big) \Rightarrow \forall y.\exists u.p(y, u)$$

using a natural deduction–style proof.

6 Points

6 Knowledge Representation

Problem 6.1 (ALC Description Logic)
Consider the following ALC setting:

- concepts: `food`, `drink`, `person`
- relations: `likes`, `goesWith`

We abbreviate every concept/relation by its first letter.

1. Give an ALC ABox with 1 piece of food and 2 persons that like it.

2 Points

2. Give an ALC TBox that formalizes the property that any person who likes some food also likes some drink.

1 Points

3. Give an ALC TBox that formalizes the property that persons cannot be liked.

1 Points

4. Give an ALC formalization for the concept of persons who only like food that a drink goes with.

2 Points

5. Give the translation to first-order logic of the ALC statement $(\forall l.f) \sqsubseteq (\exists l.d)$.

2 Points

7 Planning

Problem 7.1 (STRIPS Planning)
Consider a map of Australia. Starting from Sydney (S), we want to visit Darwin (D), Brisbane (B), and Perth (P), passing Adelaide (A).
Using $C = \{A, B, D, P, S\}$, we use the following STRIPS task:

- The facts are $\texttt{at}(x)$, $\texttt{visited}(x)$, and $\texttt{road}(x, y)$ for $x, y \in C$.
- The actions are $\texttt{drive}(x, y)$ for $x, y \in C$ with precondition/add-list/delete-list given by, respectively, $\{\texttt{at}(x), \texttt{road}(x, y)\}$, $\{\texttt{at}(y), \texttt{visited}(y)\}$, and $\{\texttt{at}(x)\}$.
- The initial state is $I = \{\texttt{at}(S), \texttt{visited}(S)\} \cup \mathit{Roads}$ where $\mathit{Roads} = \{\texttt{road}(A, P), \texttt{road}(A, D), \texttt{road}(D, A), \texttt{road}(A, S), \texttt{road}(S, A), \texttt{road}(S, B), \texttt{road}(B, S)\}$.
- The goal is $G = \{\texttt{visited}(B), \texttt{visited}(D), \texttt{visited}(P)\}$.

You may abbreviate all fact/action names by their first letter.

1. Give an optimal plan for this task, or argue why no plan exists.

2 Points

2. Give an optimal plan for the delete-relaxed task, or argue why no plan exists.

2 Points

3. Let $I' = I \cup \{\texttt{visited}(D)\}$. Give $h^*(I')$.

2 Points

4. Let $h(s) = |G \setminus s|$. Explain whether h is an admissible heuristic.

2 Points

Problem 7.2 (Partial Order Planning)

1. In about 2 sentences, explain the key properties and significance of the Sussmann anomaly.

2 Points

2. In 1-2 sentences, explain the general idea of a causal link.

2 Points