Last Name: First Name:

Matriculation Number:

Seat:

Exam
Artificial Intelligence 1

Feb 13, 2023

	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	Sum	grade
total	8	8	4	7	6	6	8	6	8	6	8	10	85	
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 85 points if you fully solve all problems. You will only need 80 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 (18 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, Feb 13, 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 85 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 80 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 (18 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, Feb 13, 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.

This page was intentionally left blank for extra space

1 Prolog

Problem 1.1 (First-Order Terms in Prolog) 8 pt

Consider the following Prolog program:

```
isSignature([]).
isSignature([funSym(Name,Arity)|Tail]) :-
  string(Name), integer(Arity), Arity >= 0, isSignature(Tail).

isTerm(_, Vars, var(Name)) :-
  contains(Vars, Name).
isTerm(Sig, Vars, applyFunSym(Name,Args)) :-
  contains(Sig,funSym(Name,L)),
  len(Args,L), isTermList(Sig,Vars,Args).

isTermList(_, _, []).
isTermList(Sig, Vars, [T|Ts]) :-
  isTerm(Sig, Vars, T), isTermList(Sig, Vars, Ts).

% computes the length of a list
len([],0).
len([_|T],M) :- len(T,L), M is L+1.

% checks if a list contains a certain value
contains([H|_],H).
contains([_|T],X) :- contains(T,X).
```

Here `string` and `integer` are built-in predicates that succeed if their argument is a string or an integer, respectively.

1. What does the predicate `isTerm(S,V,T)` compute? 3 pt

2. What do the following queries return: 3 pt

   (a) `isSignature([funSym("G",1)])`

   (b) `isTerm([funSym("f",1)], ["X"], X)`

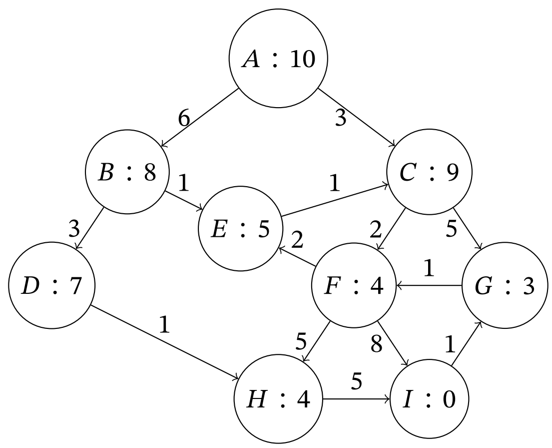
   If a query returns multiple results, only give the first two.

3. Now assume we change line 9 by switching the order of the two predicates, i.e., `isTermList(Sig,Vars,Args), len(Args,L)`.
   Under what circumstances and how would Prolog's search behavior change the run time of `isTerm`? 2 pt

2 Search

Problem 2.1 (Search Algorithms) 8 pt

Consider the following 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.

Assume you have already expanded the node A. List the next 4 nodes (i.e., excluding A) that will be expanded using

1. depth-first search 1 pt
2. breadth-first search 1 pt
3. uniform-cost search 2 pt
4. greedy search 2 pt
5. A^*-search 2 pt

If there is a tie, break it using alphabetical order.

Problem 2.2 (Complexity) 4 pt

Consider trees with maximal branching factor b, maximal depth m, and minimal depth d of a solution.

Give the worst-case complexity classes in b, d, and m (using O-notation) of

1. time complexity of breadth-first search 1 pt

2. space complexity of breadth-first search 1 pt

3. time complexity of depth-first search 1 pt

4. space complexity of depth-first search 1 pt

Problem 2.3 (Search Problems) 7 pt

Consider the search problem $\langle S, A, T, I, G\rangle$ where

- ☐ $S = \mathbb{Z} \times \mathbb{Z}$
- ☐ $A = \{X, Y, M\}$
- ☐ T is given by
  - ☐ $T(X, (x, y)) = \{(-x, y)\}$
  - ☐ $T(Y, (x, y)) = \{(x, -y)\}$
  - ☐ $T(M, (x, y)) = \{(x + 1, y), (x, y + 1)\}$
- ☐ $I = \{(0, 0)\}$
- ☐ $G = \{(x, y) \in S \mid x^2 + y = 8\}$

1. Check the box of the row that makes this problem non-deterministic. 1 pt

2. Give the possible states resulting from applying the action sequence M, X, Y to the initial state. 2 pt

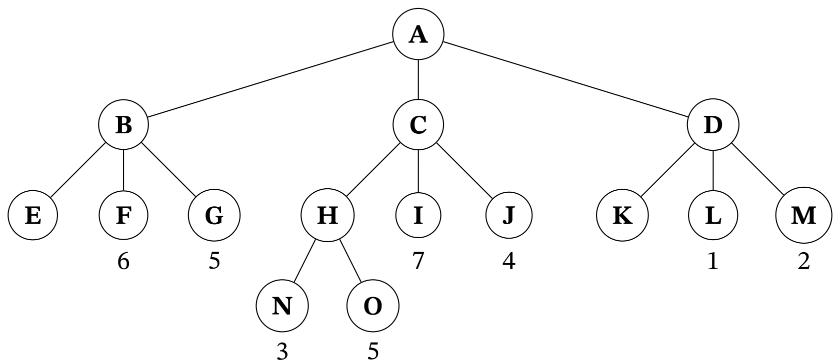
3. Give a solution of minimal length to this problem. 3 pt

4. Assume we have added an action D to A, and we want the successor state after applying D to be the one where both coordinates are swapped. Give the necessary extension of the transition model. 1 pt

3 Adversarial Search

Problem 3.1 (Minimax) 6 pt

Consider the following minimax game tree for the **minimizing** 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 pt

2. If possible, label the node **K** with an evaluation function value that results in the player definitely choosing move **D** (i.e., no matter how **E** is labeled or how ties are broken). 2 pt
   Otherwise, argue why that is impossible.

3. If possible, label the node **E** with an evaluation function value that results in the player definitely choosing move **D** (i.e., no matter how **K** is labeled or how ties are broken). 2 pt
   Otherwise, argue why that is impossible.

4 Constraint Satisfaction/Propagation

Problem 4.1 (Modeling) 6 pt

You want to place 4 items into 3 boxes. The sizes of the items are S_1, S_2, S_3, S_4. The capacities of the boxes are C_1, C_2, C_3. The total of the sizes of the items in each box must not exceed its capacity.

Model this problem as a (not necessarily binary) constraint satisfaction problem $\langle V, D, C \rangle$. Briefly explain how the solutions correspond to the placement of the items.

Problem 4.2 (Solving) 8 pt

Consider the following binary CSP:

- $V = \{a, b, c, d, e, f\}$
- $D_a = D_b = D_c = \{0, 1, 2, 3\}$, $D_d = D_e = D_f = \{0, 1, 2, 3, 4, 5, 6\}$
- Constraints:
  - $a < 2$ or $d > 5$
  - $b \cdot e > 10$
  - $e < f$
  - $d = 2c$
  - $f^2 - b^2 > 17$
  - $a > c$

1. Check the boxes for (v, w) where v is **not** arc-consistent relative to w. 2 pt

   ☐ (b, c) ☐ (c, b) ☐ (b, f) ☐ (f, b) ☐ (c, d) ☐ (d, c)

2. Give an **inconsistent** total assignment to the variables. 1 pt

3. Give **the two** solutions. 3 pt

4. Now assume we add the constraint $f = a + e$.

   (a) What distinguishes this constraint crucially from the other constraints? 1 pt

   (b) How does this difference affect constraint propagation? 1 pt

5 Logic

Problem 5.1 (Propositional Logic) 6 pt

We use the propositional variables P and Q. Consider the formula A given by

$$(P \vee (P \Rightarrow Q)) \Rightarrow \neg(P \wedge Q)$$

1. Give the interpretation $\mathcal{I}_\varphi(A)$ for every possible assignment φ, e.g., via a truth table. 2 pt

2. Give some formula in CNF that is equivalent to $\neg A$. 2 pt

3. Assume we have constructed a saturated tableau with root A^F that has a single open branch. What useful information about A can we read off that open branch? 2 pt

Problem 5.2 (Predicate Logic) 8 pt

Consider the following signature of predicate logic:

- unary function symbol f
- binary function symbol g
- binary predicate symbol p

1. Give a formula that uses all three symbols and is a theorem. 2 pt

2. Now consider the formulas A and B given by

$$A = \forall x.\forall y.p(f(x), g(x, y))$$

$$B = \forall x.\forall y.p(x, y)$$

   Give a model $\langle D, \mathcal{I}\rangle$ in which A is true but B is false. 3 pt

3. Prove or refute the following statement: If the formula A has one free variable x, and there is a model in which A is true under all assignments, then A is a theorem. 3 pt

Problem 5.3 (Proving in Tableau Calculus) 6 pt

We use the propositional variables P, Q, and R and define formulas A, B, and C by

$$\begin{aligned} A &= Q \wedge (Q \Rightarrow R) \\ B &= P \Rightarrow A \\ C &= P \Rightarrow R \end{aligned}$$

Prove the formula $B \Rightarrow C$ using the tableaux calculus.

This page was intentionally left blank for extra space

6 Knowledge Representation

Problem 6.1 (Description Logic) 8 pt

Consider the following $\mathcal{ALC}$-setting:

- concepts: `student`, `instructor`, `course`, `person`
- relations: `takes`, `gives`

We abbreviate every concept/relation by its first letter.

1. Give an $\mathcal{ALC}$-ABox using individuals *Alice*, *Bob*, and $Math$ such that *Alice* is a student taking the $Math$-course given by the instructor *Bob*. 2 pt
   You can abbreviate every individual by their first letter.

2. Give an $\mathcal{ALC}$-TBox with two concept axioms expressing the following: 2 pt

   (a) Every person is an instructor or a student.

   (b) Courses are only given by instructors.

3. Consider the model $\langle D, \mathcal{I} \rangle$ given by: 2 pt

   - $D = \{MK, Carol, Donald, AI1, AI2\}$
   - $\mathcal{I}(\mathtt{s}) = \{Carol,\ Donald\}$
   - $\mathcal{I}(\mathtt{i}) = \{MK\}$
   - $\mathcal{I}(\mathtt{c}) = \{AI1,\ AI2\}$
   - $\mathcal{I}(\mathtt{p}) = \{MK,\ Carol,\ Donald\}$
   - $\mathcal{I}(\mathtt{t}) = \{(Carol,\ AI1), (Donald,\ AI2)\}$
   - $\mathcal{I}(\mathrm{g}) = \{(MK, AI1),\ (MK, AI2)\}$

   We define the concept G by

$$G \ = \ (\exists \mathtt{t}.\mathtt{c}) \sqcup (\exists \mathrm{g}.\mathtt{c})$$

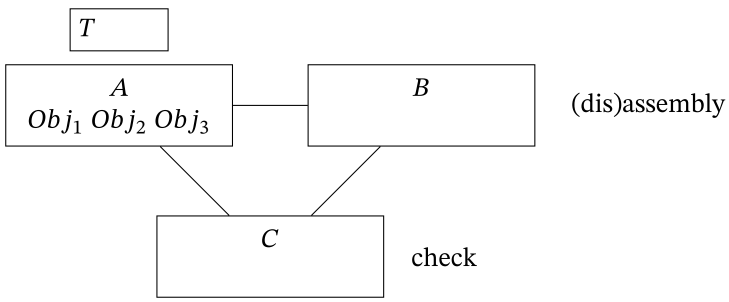
   Give the interpretation $\mathcal{I}(G)$.

4. Give the result of the first-order translation $\overline{G \sqsubseteq \mathtt{p}}^{fo(x)}$ (where G is as in the previous question). 2 pt

7 Planning

Problem 7.1 (STRIPS) 10 pt

Consider a machine that processes objects $Obj = \{Obj_1, Obj_2, Obj_3\}$, 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 transporter T is available (currently at location A) that can move **one** object at a time from one location to any other location.



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

- $\mathtt{at}(l, o)$ for $l \in \{A, B, C\}$ and $o \in Obj \cup \{T\}$
- $\mathtt{isCh}(o)$ and $\mathtt{isAss}(o)$ and $\mathtt{isDis}(o)$ for $o \in Obj$

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

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

1. Complete the definition of the action $\mathtt{check}(o)$. 2 pt

2. Give the initial state I and the goal G. 2 pt

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

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

5. Prove or refute that the following heuristic is admissible: $h(s)$ is 5 times the number of unchecked objects in state s. 2 pt

This page was intentionally left blank for extra space