Name:

Birth Date:

Matriculation Number:

Exam
Artificial Intelligence 1

July 29., 2019

	To be used for grading, do not write here														
prob.	1.1	2.1	2.2	2.3	3.1	3.2	4.1	4.2	4.3	5.1	5.2	5.3	6.1	Sum	grade
total	4	3	3	3	3	10	5	5	5	6	8	10	10	75	
reached															

Exam Grade: Bonus Points: Final Grade:

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. The grading information on the cover sheet holds with the proviso of further checking.
3. no resources or tools are allowed except for a pen.
4. You have 90 min(sharp) for the test
5. You can reach 75 points if you fully solve all problems. You will only need 70 points for a perfect score, i.e. 5 points are bonus points.
6. Write the solutions directly on the sheets.
7. 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.
8. Please make sure that your copy of the exam is complete (21 pages including 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 (next page).**

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

Erlangen, July 29., 2019 . . . .
(signature)

Organisatorisches

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

1. Bitte legen Sie Ihren Studentenausweis und einen Lichtbildausweis zur Personenkontrolle bereit!
2. Die angegebene Punkteverteilung gilt unter Vorbehalt.
3. Es sind keine Hilfsmittel erlaubt außer eines Stifts.
4. Die Lösung einer Aufgabe muss auf den vorgesehenen freien Raum auf dem Aufgabenblatt geschrieben werden; die Rückseite des Blatts kann mitverwendet werden. Wenn der Platz nicht ausreicht, können bei der Aufsicht zusätzliche Blätter angefordert werden.
5. 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.
6. Die Bearbeitungszeit beträgt 90 min.
7. Sie können 75 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 70 Punkte bereits als volle Punktzahl, d.h. 5 Punkte sind Bonuspunkte.
8. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (21 Seiten inklusive 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, July 29., 2019 ....

(Unterschrift)

Please consider the following rules; otherwise you may lose points:

- If you continue an answer on another page, please indicate the problem number on the new page and give a page reference on the old page.
- Always justify your statements (we would like to give poins for incorrect answers). Unless you are explicitly allowed to, do not just answer “yes”, “no”, or “42”.
- If you write program code, give comments!

1 Prolog

Problem 1.1 4 pt

1. Program a Prolog predicate uadd for addition and umult for multiplication in unary representation.

Hint: The number 3 in unary representation is the ProLog term s(s(s(o))), i.e. application of the arbitrary function s to an arbitrary value o iterated three times.

Hint: Note that ProLog does not allow you to program (binary) functions, so you must come up with a three-place predicate. You should use add(X,Y,Z) to mean $X+Y=Z$ and program the recursive equations $X+0=X$ (base case) and $X+s(Y)=s(X+Y)$.

---

2. Write a Prolog predicate ufib that computes the n^{th} Fibonacci Number (0, 1, 1, 2, 3, 5, 8, 13,…add the last two to get the next), using the addition predicate above.

This page was intentionally left blank for extra space

2 Search

Problem 2.1 Does a finite state space always lead to a finite search tree? How about a finite space state that is a tree? Justify your answers. 3 pt

Problem 2.2 (A^* **Theory**)
What is the condition on the heuristic function that makes A^* optimal? Does a heuristic with this condition always exist? 3 pt

Problem 2.3 (A^* vs. BFS)

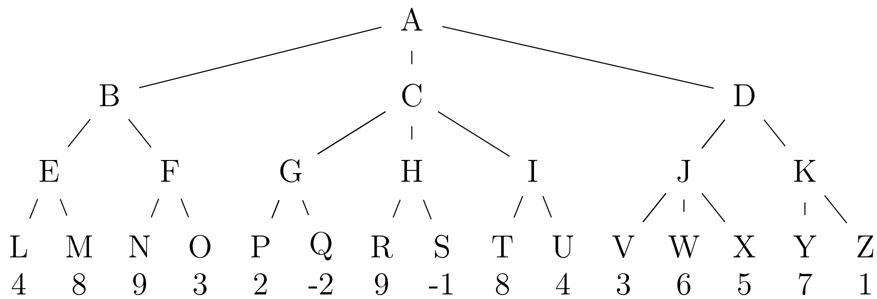
Does A^* search always expand fewer nodes than BFS? Justify you answer. 3 pt

This page was intentionally left blank for extra space

3 Adversarial Search

Problem 3.1 (Minimax Restrictions)
Name at least five criteria that a game has to satisfy in order for the minimax algorithm to be applicable. 3 pt

Problem 3.2 (Minimax Algorithm and Alphabeta Pruning)
Consider the following (complete!) search tree: 10 pt



1. What is the minimax value at node B?
2. Which subtrees in the tree can be pruned during alpha-beta search? What is the criterion for pruning a subtree?

This page was intentionally left blank for extra space

4 Constraint Satisfaction Problems & Inference

Problem 4.1 (Constraint Networks)
A constraint network is a triple $\langle V, D, C\rangle$. Explain the roles of V, D, and C. If you use the word "constraint" you have to define and briefly explain it. 5 pt

same

Problem 4.2 (Arc consistency)

Define the concept of *arc consistency*. 5 pt

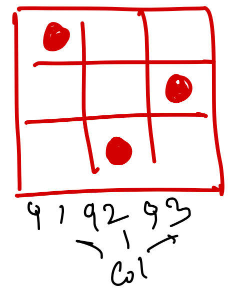
Same

Problem 4.3 (50 Queens)

!? Formalize the *50 Queens Problem* as a constraint network. Hint: You do not have to write down all the constraints explicitly, but it has to be clear what the exact constraints are. 5 pt

$V = \{Q_1, Q_2, ----, Q_{50}\}$,

↳ Here Q_i means ith Q position, $\boxed{(\text{row}, \text{col}^m) = (Q_i, i)}$



$D_{Q_i} = \{1, —, 50\}$, for all $i \in \{1, —50\}$

C: $Q_i \neq Q_j$ where $i \neq j$ & $i \in \{1, —50\}$
$j \in \{1 --- 50\}$

This page was intentionally left blank for extra space

5 Logic

Problem 5.1 (Natural Deduction)

Prove the validity of the following formula in Natural Deduction: 6 pt

$$((A \vee B) \wedge (A \Rightarrow C) \wedge (B \Rightarrow C)) \Rightarrow C$$

1.	$(A \vee B) \wedge (A \Rightarrow C) \wedge (B \Rightarrow C)$	Assumption 1
2.	$A \vee B$	$\wedge E_l$ on 1
3.	$(A \Rightarrow C) \wedge (B \Rightarrow C)$	$\wedge E_r$ on 1
4.	$A \Rightarrow C$	$\wedge E_l$ on 3
5.	$B \Rightarrow C$	$\wedge E_r$ on 4
6.	A	Assumption 2
7.	C	$\Rightarrow E$ on 4, 6
8.	B	Assumption 2
9.	C	$\Rightarrow E$ on 5, 8
10.	C	$\vee E$ on 2, 7, 9
11.	$((A \vee B) \wedge (A \Rightarrow C) \wedge (B \Rightarrow C)) \Rightarrow C$	$\Rightarrow I$ on 1, 10

Problem 5.2 (First-Order Tableaux)

Prove the following formula using the first-order free variable tableaux calculus. We have $P \in \Sigma_1^p$. 8 pt

$$\exists X. (P(X) \Rightarrow \forall Y.P(Y))$$

$$\exists X. (P(X) \Rightarrow \forall Y. P(Y))^F$$

$$(P(\underline{V_X}) \Rightarrow \forall Y. P(Y))^F \quad ?$$

$$P(V_X)^T$$

$$\forall Y. P(Y)^F$$

$$P(c_Y)^F$$

$$\bot \; [c_Y / V_X]$$

Problem 5.3 (First-Order Resolution)

Prove the following formula using resolution. 10 pt

$P \in \Sigma_1^p, R \in \Sigma_2^p, a, b \in \Sigma_0^f$

$$\exists X.\forall Y.\exists Z.\exists W.\, ((\neg P(Z) \wedge \neg R(b,a)) \vee \neg R(a,b) \vee R(W,a) \vee (P(Y) \wedge R(X,b)))$$

Negation:

$$\forall x.\exists Y.\forall z.\forall w\, \neg\big((\neg P(z) \wedge \neg R(b,a)) \vee \neg R(a,b) \vee R(w,a) \vee (P(Y) \wedge R(x,b))\big)$$

$$\equiv \forall x.\exists Y.\forall z.\forall w\, \big(\neg(\neg P(z) \wedge \neg R(b,a)) \wedge R(a,b) \wedge \neg R(w,a) \wedge \neg(P(Y) \wedge R(x,b))\big)$$

$$\equiv \forall x.\exists Y.\forall z.\forall w\, \big((P(z) \vee R(b,a)) \wedge R(a,b) \wedge \neg R(w,a) \wedge (\neg P(Y) \vee \neg R(x,b))\big)$$

Substituting bound var. $Y = f_Y(x)$

$$(P(z) \vee R(b,a)) \wedge R(a,b) \wedge \neg R(w,a) \wedge (\neg P(f_Y(x)) \vee \neg R(x,b))$$

Resol^n:

$$\{P(z)^T, R(b,a)^T\} + \{R(w,a)^F\}\ [b/w] \Rightarrow \{P(z)^T\}$$

$$\{R(a,b)^T\} + \{P(f_Y(x))^F, R(x,b)^F\}\ [a/x] \Rightarrow \{P(f_Y(a))^F\}$$

$$\{P(z)^T\}[f_Y(a)/z] + \{P(f_Y(a))^F\} \Rightarrow \{\}$$

This page was intentionally left blank for extra space

6 Planning

Problem 6.1 Cheeta is an intelligent and lazy monkey. Your task is to help him grab a banana. 10 pt

Initially, the monkey and a boat are "Low" on the ground at position "A" and the banana hangs "High" at position "B" on a tree. Cheeta can climb up the tree where the banana hangs, meaning that it can climb from height "Low" to "High" at B. Cheeta can grab the banana when they are both at the same position and at the same height.

There is a river between "A" and "B", and Cheeta can only travel between these positions by boat. Cheeta can only enter the boat if they are at the same position and height (the boat is a height "Low").

The following STRIPS model is used. The facts are:

- $At(x, y)$: $x \in \{Boat, Banana, Cheeta\}$ is at position $y \in \{A, B, Boat\}$.
- $Height(x, y)$: $x \in \{Boat, Banana, Cheeta\}$ is at height $y \in \{Low, High\}$.
- $Climbable(x)$: Cheeta can climb at position $x \in \{A, B\}$.
- $Grabbed()$: Cheeta has grabbed the banana.

The goal for Cheeta is to grab the banana using the following available actions:

- $Drive(x, y)$ to get from x to y
- $GetIn(x)$ to get in the boat (at location x)
- $GetOut(x)$ to get out of the boat (at location x)
- $Climb(x, y, z)$ to climb at position x from height y to height z
- $Grab(x, y)$ to grab the banana (at position x and height y)

(a) Properly define the actions $Drive(x, y)$ and $GetIn(x)$

(b) Give the initial state and a solution to this planning problem

This page was intentionally left blank for extra space

This page was intentionally left blank for extra space

This page was intentionally left blank for extra space