Name:

Birth Date:

Matriculation Number:

Exam
Künstliche Intelligenz 1

July 16., 2018

	To be used for grading, do not write here												
prob.	1.1	1.2	2.1	2.2	3.1	3.2	4.1	4.2	5.1	5.2	6.1	Sum	grade
total	6	12	5	5	3	12	2	10	10	6	10	81	
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 81 points if you fully solve all problems. You will only need 55 points for a perfect score, i.e. 26 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 (20 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 16., 2018 . . . . . .

(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 81 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 55 Punkte bereits als volle Punktzahl, d.h. 26 Punkte sind Bonuspunkte.
8. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (20 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 16., 2018 .....

(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 6 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 nth Fibonacci Number (0, 1, 1, 2, 3, 5, 8, 13,... add the last two to get the next), using the addition predicate above.

Problem 1.2 (DFS in Prolog)

We want to implement DFS in `ProLog` using the following data structures for search trees: 12 pt

```
subtrees([]).
subtrees([(Cost,T)|Rest]) :- number(Cost),istree(T), subtrees(Rest).
istree(tree(_,Children)) :- subtrees(Children).
```

Write a Prolog predicate dfs such that dfs(G,T,X,Y) on a tree T returns the path to the goal G in X and the cost of the path in Y

This page was intentionally left blank for extra space

2 Search

Problem 2.1 (A^* vs. BFS)
Does A^* search always expand fewer nodes than BFS? Justify you answer. 5 pt

Problem 2.2 (A looping greedy search)

Draw a graph and give a heuristic so that a greedy search for a path from a node A to a node B gets stuck in a loop. Draw the development of the search tree, starting from A, until one node is visited for the second time. 5 pt

Indicate, in one or two sentences, how the search algorithm could be modified or changed in order to solve the problem without getting stuck in a loop.

This page was intentionally left blank for extra space

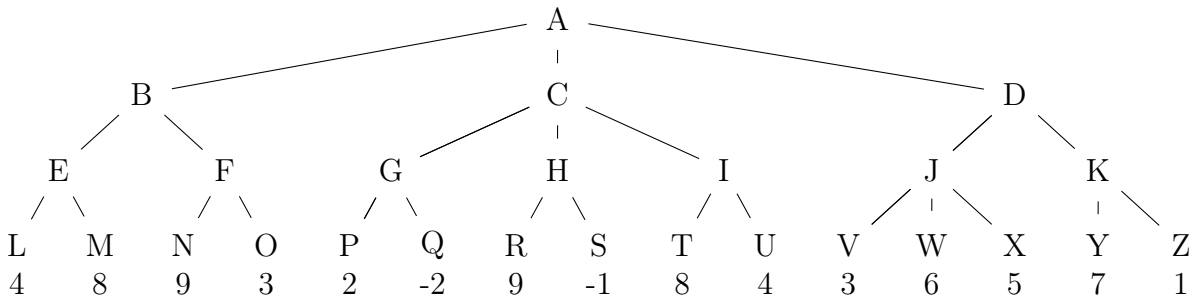
3 Adversarial Search

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 (Game Tree)

Consider the following game tree. Assume it is the maximizing player's turn to move. The values at the leaves are the static evaluation function values of the states at each of those nodes. 12 pt



1. Compute the minimax game value of nodes A, B, C, and D
2. Which move would be selected by Max?
3. List the nodes that the alpha-beta algorithm would prune (i.e., not visit). Assume children of a node are visited left-to-right.

This page was intentionally left blank for extra space

4 Constraint Satisfaction Problems & Inference

Problem 4.1 (Arc consistency)
Define the concept of *arc consistency*. 2 pt

Problem 4.2 (Scheduling CS Classes)

You are in charge of scheduling for computer science classes that meet Mondays, Wednesdays and Fridays. There are 5 classes that meet on these days and 3 professors who will be teaching these classes. You are constrained by the fact that each professor can only teach one class at a time. The classes are: 10 pt

- Class 1 - Intro to Programming: meets from 8:00-9:00am
- Class 2 - Intro to Artificial Intelligence: meets from 8:30-9:30am
- Class 3 - Natural Language Processing: meets from 9:00-10:00am
- Class 4 - Computer Vision: meets from 9:00-10:00am
- Class 5 - Machine Learning: meets from 9:30-10:30am

The professors are:

- Professor A, who is available to teach Classes 3 and 4.
- Professor B, who is available to teach Classes 2, 3, 4, and 5.
- Professor C, who is available to teach Classes 1, 2, 3, 4, 5.

1. Formulate this problem as a CSP problem in which there is one variable per class, stating the domains, and constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit.

2. Give the constraint graph associated with your CSP

3. Show the domains of the variables after running arc-consistency on this initial graph (after having already enforced any unary constraints).

This page was intentionally left blank for extra space

5 Logic

Problem 5.1 (First-Order Resolution)
Prove the following formula using resolution. 10 pt

$$\forall X \forall Y \forall Z \exists U \exists V \exists W [(P(X,Y) \Rightarrow (P(Z,a) \Rightarrow R(a))) \Rightarrow ((P(U,V) \wedge P(W,a)) \Rightarrow R(a))]$$

We assume a signature with $P \in \Sigma_2^p$, $R \in \Sigma_1^p$, and $a \in \Sigma_0^f$.

Problem 5.2 (Natural Deduction)

Prove the following formula using the propositional Natural Deduction calculus. 6 pt

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

This page was intentionally left blank for extra space

6 Planning

Problem 6.1 (STRIPS)

You are given a water spout and two jugs, one holding p and one holding q gallons, where $p < q$ and p and q are relatively prime. 10 pt

Starting with both jugs empty, the goal is to have exactly k gallons in one of the jugs. You can only fill the jugs from the spout fully.

We use the following predicates:

$$P = \{Jug_p(n) \mid 0 \leq n \leq p,\ n \in \mathbb{N}\} \cup \{Jug_q(n) \mid 0 \leq n \leq q,\ n \in \mathbb{N}\}$$

The intial state is $I = \{Jug_p(0), Jug_q(0)\}$ and the goal state $G = Jug_p(k)$.

Give the pre, add and del lists for the following actions:

- $Empty_p/Empty_q$: Empties jug p/q completely
- $FillUp_p/FillUp_q$: Fill up jug p/q fully
- For all x, y with $0 \leq x \leq p,\ 0 \leq y \leq q$:
  $Fillp_{x,y}/Fillq_{x,y}$: pour the contents of jug q/p into jug p/q until the former is empty or the latter is full.

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

This page was intentionally left blank for extra space