Name:

Birth Date:

Matriculation Number:

Exam
Künstliche Intelligenz 1

Jul 24., 2017

	To be used for grading, do not write here													
prob.	1.1	1.2	2.1	2.2	3.1	3.2	3.3	4.1	4.2	4.3	4.4	5.1	Sum	grade
total	5	5	5	7	8	5	5	6	5	6	5	18	80	
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 80 points if you fully solve all problems. You will only need 55 points for a perfect score, i.e. 25 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 (16 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, Jul 24., 2017 .....
(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 80 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 55 Punkte bereits als volle Punktzahl, d.h. 25 Punkte sind Bonuspunkte.
8. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (16 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, Jul 24., 2017 .....

(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 Search

Problem 1.1 (A^* Theory)
What is the condition on the heuristic function that makes A^* optimal? Does a heuristic with this condition always exist? 5 pt

Problem 1.2 (A^* vs. BFS)
Does A^* search always expand fewer nodes than BFS? Justify you answer. 5 pt

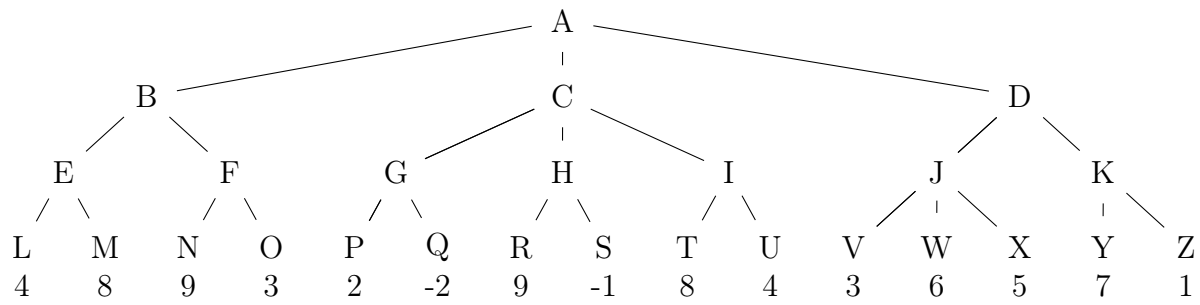
2 Adversarial Search

Problem 2.1 (Minimax Restrictions)

Name at least five criteria that a game has to satisfy in order for the minimax algorithm to be applicable. 5 pt

Problem 2.2 (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. 7 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.

3 Constraint Satisfaction Problems & Inference

Problem 3.1 (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. 8 pt

Problem 3.2 (Arc consistency)
Define the concept of *arc consistency*. 5 pt

Problem 3.3 (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

This page was intentionally left blank for extra space

4 Logic

Problem 4.1 (Inference and Entailment)

6 pt

1. Define and briefly explain the entailment and inference relations.
2. How do we write "**A** entails **B**" and "from **A** we can infer/deduce **B**" in symbolism?
3. What is the difference between the two relations.

Problem 4.2 (First-Order Tableaux)

Prove the following formula using the first-order free variable tableaux calculus. We have $P \in \Sigma_1^p$. 5 pt

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

Problem 4.3 (First-Order Resolution)

Prove the following formula using resolution. 6 pt

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

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

Problem 4.4 (The Zip Function)

The zip function takes two lists with lengths that differ at most by 1, and outputs a list of lists containing one element from the first list and the element with the same index from the other list, possibly followed by a one-element list with the left-over argument. 5 pt

Create a `ProLog` predicate with 3 arguments: the first two would be the two lists you want to zip, and the third one would be the result. For instance:

```
?- zip([1,2,3],[4,5,6],L).
L = [[1, 4], [2, 5], [3, 6]].

?- zip([1,2],[3,4,5],L).
L = [[1, 3], [2, 4], [5]].
```

Feel free to implement any helper functions.

Hint: Remember that you can pattern match a list L as [HEAD|TAIL].

5 Planning

Problem 5.1 Consider the following problem. A fused bulb hangs out of reach from the ceiling. A robot needs to repair the bulb. The room also contains a box. Pushing that box into the correct position, and climbing onto the box, will bring the bulb into reach for the robot. 18 pt

The exercise is to model this problem as a STRIPS planning task. In doing so, assume the following framework. The robot is currently at position "A", the fused bulb is at position "B", and the box is at position "C". The robot and box are at the same height "Low", the fused bulb is at height "High". By climbing onto the box, the robot changes from "Low" to "High"; vice versa when climbing off the box. The actions available for the robot are "*Go*" from one place to another (only possible if the robot is at "Low"), "$Push$" an object from one place to another (only possible if the robot and object are at "Low"), "$ClimbUp$" onto or "$ClimbDown$" from an object, and "*Repair*" to fix an object. The robot needs to be at the same place and height as an object in order to repair it.

Note that the robot can only push an object or climb onto an object if both of them are at the same location. Furthermore, in case of pushing an object the robot changes to the destination location as well.

(a) Write a STRIPS formalization of the initial state and goal descriptions.

(b) Write a STRIPS formalization of the five actions: Go, $Push$, $ClimbUp$, $ClimbDown$, and $Repair$. In doing so, please do make use of "object variables", i.e., write the actions up in a parameterized way. State, for each parameter, by which objects it can be instantiated.

In both (a) and (b), make use of the following predicates: (do not use any other predicates)

- $At(x, y)$: To indicate that object $x \in \{Box, Bulb, Robot\}$ is at position $y \in \{A, B, C\}$.
- $Height(x, y)$: To indicate that object $x \in \{Box, Bulb, Robot\}$ is at height $y \in \{Low, High\}$.
- $Pushable(x)$: To indicate that object $x \in \{Box, Bulb, Robot\}$ can be pushed.
- $Climbable(x)$: To indicate that the robot can climb on object $x \in \{Box, Bulb, Robot\}$.
- $Repaired(x)$: To indicate that object $x \in \{Box, Bulb, Robot\}$ is repaired.

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