Name:

Birth Date:

Matriculation Number:

# Exam
# Artificial Intelligence 1

Feb 11., 2019

| | To be used for grading, do not write here | |
|---|---|---|
| prob. | Sum | grade |
| total | 0 | |
| 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 0 points if you fully solve all problems. You will only need 80 points for a perfect score, i.e. -80 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 (**??** 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, Feb 11., 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 0 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 80 Punkte bereits als volle Punktzahl, d.h. -80 Punkte sind Bonuspunkte.
8. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (**??** 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, Feb 11., 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 $n^{\text{th}}$ 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)** 12 pt

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

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

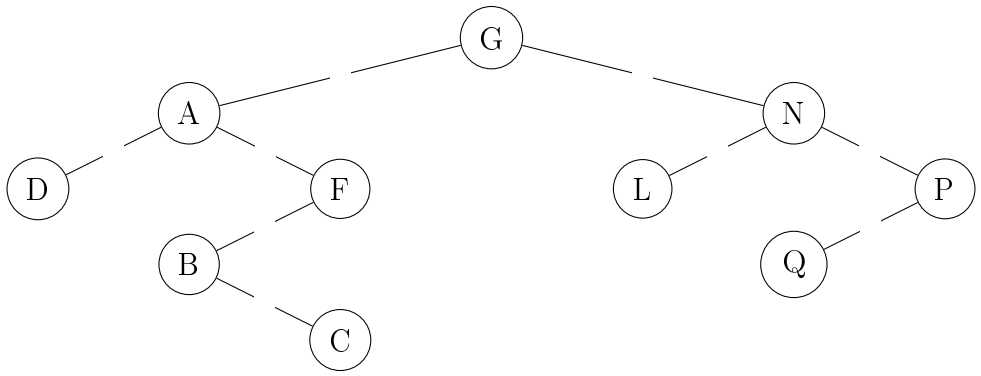
Write a Prolog predicate dfs such that dfs(G,T,X,Y) on a tree T returns the path to the goal G in X and the cost of the path in Y

This page was intentionally left blank for extra space

# 2 Search

**Problem 2.1 (BFS)** 3 pt

Apply BFS and DFS on the following tree exhaustively (the goal is not node G!).




List the nodes in the order they are expanded:

1. BFS
2. DFS

**Problem 2.2 (Admissibility limits)** 5 pt

The condition for a heuristic $h(n)$ to be admissible is that for all nodes $n$ holds that $(0 \leq h(n) \leq h^*(n))$, where $h^*(n)$ is the true cost from $n$ to goal. What happens when for all nodes, $h(n) = 0$ and when $h(n) = h^*(n)$ ?

This page was intentionally left blank for extra space

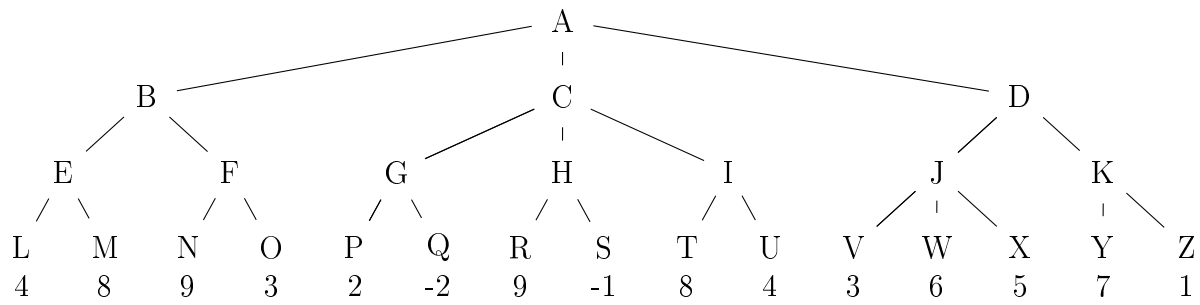
# 3 Adversarial Search

**Problem 3.1 (Minimax Restrictions)** 3 pt

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

**Problem 3.2 (Game Tree)** 7 pt

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.




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 (Constraint Networks)** 6 pt

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.

**Problem 4.2 (Arc Consistency for Chains)** 10 pt

Consider the general less-than chain below, which we interpret as a CSP: Each of the $N$ variables $X_i$ has the domain $\{1M\}$. The constraints between adjacent variables $X_i$ and $X_{i+1}$ require that $X_i < X_{i+1}$.

$$X_1 < X_2 < X_3 < \cdots < X_N$$

1. For now, assume $N = M = 5$.
   (a) How many solutions does the CSP have?
   (b) What will the domain of $X_1$ be after enforcing the consistency of only the arc $X_1 \to X_2$?
   (c) What will the domain of $X_1$ be after enforcing the consistency of only the arcs $X_2 \to X_3$ and (then) $X_1 \to X_2$?
   (d) What will the domain of $X_1$ be after fully enforcing arc consistency?
2. Now consider the general case for arbitrary $N$ and $M$.
   (a) Imagine you wish to construct a similar family of CSPs which forces one of the two following types of solutions: either all values must be ascending or all values must be descending, from left to right. For example, if $M = N = 3$, there would be exactly two solutions: $\{1, 2, 3\}$ and $\{3, 2, 1\}$. Explain how to formulate this variant. Your answer should include a constraint graph and precise statements of variables and constraints.

This page was intentionally left blank for extra space

# 5 Logic

**Problem 5.1 (First-Order Resolution)** 10 pt

Prove the following formula using resolution.
$P, Q \in \Sigma_1^p; R \in \Sigma_2^p; c, d \in \Sigma_0^f$

$$\exists X.\forall Y.\exists W.\exists Z.\neg\left((R(Z,Y) \vee \neg P(Z)) \wedge (\neg Q(d) \vee P(c)) \wedge (Q(d) \vee \neg P(c))\right.$$

$$\left.\wedge(\neg R(Z,Y) \vee \neg P(W) \vee \neg Q(X)) \wedge P(c)\right)$$

---

**Hint:** Note, that the formula is already (close to) a negated CNF, so if you spend any significant amount of time transforming the formula, you are most likely doing something wrong.

---

**Problem 5.2 (Natural Deduction)** 8 pt

Let $R \in \Sigma_2^p$, $P \in \Sigma_1^p$, $c \in \Sigma_0^f$.

Prove the following formula in Natural Deduction:

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

This page was intentionally left blank for extra space

# 6 Planning

**Problem 6.1 (Planning Bike Repair)** 12 pt

Consider the following problem. A bicycle has a front wheel and a back wheel installed and both wheels have a flat tire. A robot needs to repair the bicycle. The room also contains a tire pump and a box with all the other equipment needed by the robot to repair a bicycle. The robot can repair a wheel with the help of the box and the tire pump when the robot and the three objects are at the same position. The bicycle is repaired when the robot has done a final overall check which requires both tires to be repaired and to be installed on the bicycle again. For this check, the box is also needed at the same position as the bicycle and the robot.

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 bicycle is at position "B", and the "Frontwheel" and the "Backwheel" are installed on the "Bicycle". The "Box" is at position "C" and the "Pump" at position "D". The actions available for the robot are:

- "*Go*" from one position to another. The four possible positions A, B, C, and D are connected in such a way that the robot can reach every other place in one "*Go*".
- "*Push*" an object from one place to another. The bicycle is not pushable and the wheels are only pushable if not installed on the bicycle; obviously the robot cannot push itself; every other object is always pushable. "*Push*" moves both the object and the robot.
- "*Remove*" a wheel from the bike.
- "*RepairWheel*" to fix a wheel with a flat tire.
- "*InstallWheel*" to put a wheel back on the bike.
- "*FinalCheck*" to make sure that not only the two wheels are repaired and installed but also the rest of the bike is in good condition. The box is needed at the same position for this.

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

(b) Write a STRIPS formalization **of the actions** *Remove* **and** *FinalCheck*, **and only these two actions**. In doing so, please make use of "object variables", i.e., write the actions up in a parametrized way. State, for each parameter, by which objects it can be instantiated.

In both (a) and (b), make use of only the following predicates:

- $At(x, y)$: To indicate that object $x \in \{Robot, Bicycle, Frontwheel, Backwheel, Box, Pump\}$ is at position $y \in \{Bicycle, A, B, C, D\}$.

- $Pushable(x)$: To indicate that object $x \in \{Robot, Bicycle, Frontwheel, Backwheel, Box, Pump\}$ can be pushed.
- $Repaired(x)$: To indicate that object $x \in \{Robot, Bicycle, Frontwheel, Backwheel, Box, Pump\}$ is repaired.
- $FlatTire(x)$: To indicate that object $x \in \{Robot, Bicycle, Frontwheel, Backwheel, Box, Pump\}$ has a flat tire.

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