Name:

Birth Date:

Matriculation Number:

Exam
Artificial Intelligence 1

July 19, 2021

	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	5.1	5.2	6.1	Sum	grade
total	15	8	5	8	7	8	6	9	9	8	12	95	
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 95 points if you fully solve all problems. You will only need 90 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 (18 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 19, 2021 ....

(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 95 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 90 Punkte bereits als volle Punktzahl, d.h. 5 Punkte sind Bonuspunkte.
8. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (18 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 19, 2021 ....

(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 (Reading and Writing Prolog) 15 pt

Note: The negation-normal-form of a formula F is a formula equivalent to F in which negations only occur immediately in front of propositional variables. For example, the negation-normal form of $\neg(P \wedge Q)$ is $\neg P \vee \neg Q$.

Consider the following partial Prolog program for computing the negation-normal form:

```
contains([H|_],H).
contains([_|T],X) :- contains(T,X).

isForm(pv(A))      :- contains(["p", "q"], A).
isForm(conj(F,G)) :- isForm(F), isForm(G).
isForm(disj(F,G)) :- isForm(F), isForm(G).
isForm(neg(F))     :- isForm(F).
```

$F \wedge G = F \wedge G$

```
nnf(pv(A),H)              :- H = pv(A).

nnf(conj(F,G), H)         :- nnf(F,F2), nnf(G,G2), H = conj(F2,G2).

nnf(disj(F,G), H)         :- nnf(F,F2), nnf(G,G2), H=disj(F2,G2).

nnf(neg(pv(A)),H)         :- H = neg(pv(A)).

nnf(neg(neg(F)), H)       :- nnf(F,H).

nnf(neg(conj(F,G)), H)    :- nnf( disj(neg(F), neg(G)), H).

nnf(neg(disj(F,G)), H)    :- nnf(cnj(neg(F), neg(G)), H).
```

$\neg(F \wedge G) \equiv \neg F \vee \neg G$

4 pt

1. Give the first three results in order that are returned by the query `isForm(F)`?

3 pt

2. Complete the following query such that it can be used as a test case for the program:

   P $\quad \neg P$ $\quad \neg P \vee P$

   `nnf(neg(conj(pv("p"),neg(pv("p")))), disj( neg(pv("p")), pv("p")))`

   $\neg(P \wedge \neg P)$

8 pt

3. Complete the program such that `nnf(F,H)` computes the negation-normal-form of F.

This page was intentionally left blank for extra space

01. $F = pv(\text{"}p\text{"})$

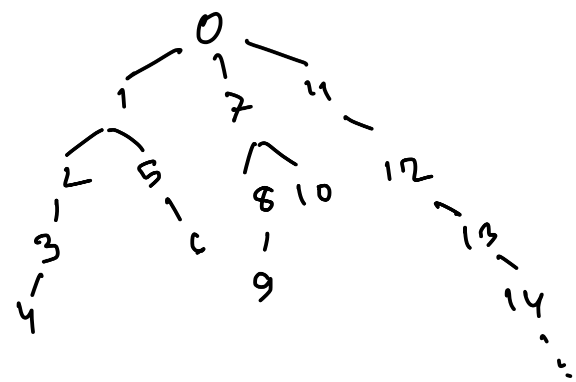
$= pv(\text{"}q\text{"})$

$= conj(pv(\text{"}p\text{"}), pv(\text{"}p\text{"}))$

2 Search

Problem 2.1 (DFS and BFS Concretely) 8 pt

Consider the **infinite** tree whose nodes are the natural numbers with root 0. For every node, the children and their order are as follows:



- children of 0: 1, 7, 11
- children of 1: 2, 5
- children of 2: 3
- children of 3: 4
- children of 5: 6
- children of 7: 8, 10
- children of 8: 9
- children of 11: 12
- children of n for $n \geq 12$: n+1
- other nodes have no children

1. List the nodes in the order of expansion during
   (a) depth-first search 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, — 3 pt
   (b) breadth-first search 0, 1, 7, 11, 2, 5, 8, 10, 12, 3, 6, 9, 13, 4, 14, ... 3 pt
2. Assuming the goal state is 7, how does it matter whether we use depth-first or breadth-first search? 2 pt

Both will find 7

BFS → faster

DFS → slower (traverse all descendants of 1)

This page was intentionally left blank for extra space

Problem 2.2 (Heuristics) 5 pt

Consider A^* search with path cost function g and heuristic function h.

1. What is the effect of using the heuristic $h(n) = 0$? 3 pt

2. What is the effect of using the path cost as the heuristic, i.e., if we put $h(n) = g(n)$? 2 pt

01. It is admissible, but useless one as it will behave like uninformed search

02. if, $h(n) = g(n)$

A^* cost min $f^n = h(n) + g(n) = 2g(n)$

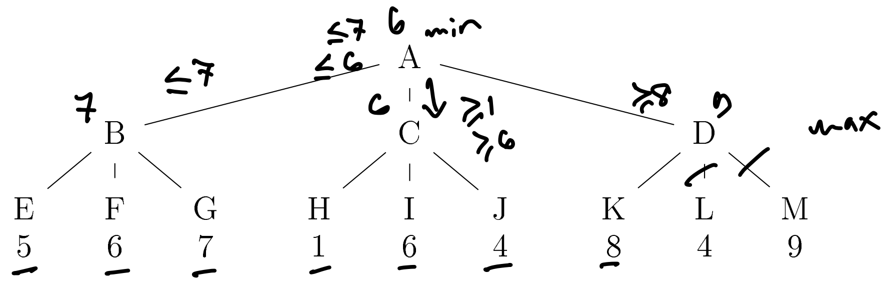
so, it will behave like uninformed search,

as $2g(n)$ leads to the same expansion decision as $g(n)$

Also, it is not admissible $\rightarrow g(n) > h^*(n)$

Problem 2.3 (Adversarial Search) 8 pt

Consider the following minimax game tree for the **minimizing player**'s turn. The values at the leaves are the static evaluation function values of those states.



1. Label each non-leaf node with its minimax value. A = 6, B = 7, C = 6, D = 9 4 pt

2. Which move would be selected by the player? Move C 1 pt

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. L & M 3 pt

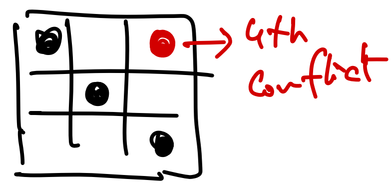
3 Constraint Satisfaction/Propagation

Problem 3.1 (4 Rooks on a Small Board) 7 pt

Consider the following problem: We want to place 4 rooks on a 3×3 chess-board such that no two rooks threaten each other. (Rooks move like queens except not diagonally.)

Model the problem as a constraint satisfaction problem (V, D, C).
Use your model to argue briefly but rigorously why this problem is unsatisfiable.

Note: Make sure you give a formally exact definition, i.e., explicitly define the sets V and all sets D_v. You can describe each constraint as a set of tuples or as a formula.



$V = x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$

$D_v = \{1, 2, 3\}, \; v \in V$

C: $v \neq w$, for all $(v,w) \in V_x^V$, $V_x = \{x_1, x_2, x_3, x_4\}$

$v \neq w$, for all $(v,w) \in V_y^V$, $V_y = \{y_1, y_2, y_3, y_4\}$

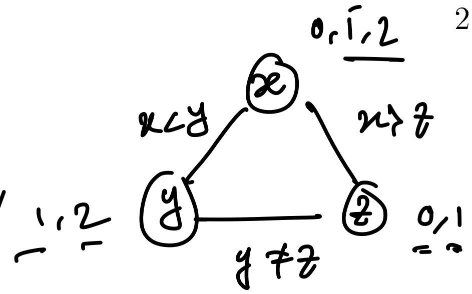
For satisfying this problem requires $\{x_1, x_2, x_3, x_4\}$ & $\{y_1, y_2, y_3, y_4\}$ $\Rightarrow$ all different values, but domain has 3 value.

Problem 3.2 (CSP Formalization) 8 pt

Consider the following binary CSP:

- $V = \{x, y, z\}$
- $D_x = \{0, 1, 2\}$, $D_y = \{1, 2\}$, $D_z = \{0, 1\}$
- Constraints: $x < y$, $y \neq z$, $x > z$

1. Give all pairs (v, w) of variables such that v is arc-consistent relative to w. 3 pt
2. Give all solutions. 2 pt
3. What is special about the constraint $y \neq z$? 1 pt
4. Assume we assign $y = 1$ and apply forward-checking. Give the resulting domains D_x, D_y, D_z. 2 pt



01. (x, y) ✗ (y, z) ✓ (x, z) ✗

(y, x) ✓ (z, y) ✓ (z, x) ✓

02. $(1, 2, 0)$,

03. $y \neq z$ can be dropped, because it is satisfied automatically if other two satisfied

04. $D_x = \{0\}$

$D_y = \{1\}$

$D_z = \{0\}$

4 Logic

Problem 4.1 (Satisfiability and Validity) 6 pt

Consider propositional logic with propositional variables $\{P, Q, R\}$. For each of the following statements, give a counter-example that refutes it:

1. The formula $((P \wedge Q) \vee (Q \wedge R)) \Rightarrow (\neg P \vee \neg R)$ is satisfied by all assignments. 2 pt
2. If a formula F cannot be proved in the natural deduction calculus, then $\neg F$ is valid. 2 pt
3. If, for two formulas F, G, all assignments satisfy $F \Rightarrow G$ and no assignment satisfies F, then no assignment satisfies G. 2 pt

01. $((P \wedge Q) \vee (Q \wedge R)) \Rightarrow (\neg P \vee \neg R)^F$

$(P \wedge Q) \vee (Q \wedge R)^T$ | $(\neg P \vee \neg R)^F$

$P \wedge Q^T$ | $Q \wedge R^T$ ... $\neg P^F \Rightarrow P^T$

P^T ... $\neg R^F \Rightarrow R^T$

Q^T

$\boxed{P^T, Q^T, R^T}$

02. F can not be proved, but satisfiable / $\neg F$ is valid (False)

03. $F \to G^T$

F^F

$G <^T_F$

$\Downarrow$

satisfiable

?

F	G	$F \to G$
T	T	T
~~T~~	~~F~~	~~F~~
F	T	T
F	F	T

$\neg$

Let, $F = P \wedge \neg P$ | $F \Rightarrow G \Rightarrow$ for all φ satisfies

$G = P$ | $F \Rightarrow$ none satisfies

$G \Rightarrow$ satisfiable

$P <^T_F$

??

Problem 4.2 (Proving in Natural Deduction) 9 pt

Prove the following formula using natural deduction:

$$(\forall x.P(x)) \Rightarrow (\underbrace{\underbrace{(\forall y.P(y) \Rightarrow Q(y))} \Rightarrow \forall z.Q(z)})$$

$P \Rightarrow ((P \Rightarrow Q) \Rightarrow Z)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall x.P(x), \forall y.P(y) \Rightarrow Q(y) \vdash \forall x\, P(x)}\text{Ax}}{\forall x.P(x), \forall y.P(y) \Rightarrow Q(y) \vdash P(z)}\forall E(z) \qquad \dfrac{\dfrac{}{\forall x.P(x), \forall y.P(y) \Rightarrow Q(y) \vdash \forall y.P(y) \Rightarrow Q(y)}\text{Ax}}{\forall x.P(x), \forall y.P(y) \Rightarrow Q(y) \vdash P(z) \Rightarrow Q(z)}\forall E(z)}{\forall x.P(x), \forall y.P(y) \Rightarrow Q(y) \vdash Q(z)}\Rightarrow E}{\forall x.P(x), \forall y.P(y) \Rightarrow Q(y) \vdash \forall z.Q(z)}\forall I}{\forall x.P(x) \vdash (\forall y.P(y) \Rightarrow Q(y)) \Rightarrow \forall z.Q(z)}\Rightarrow I}{\vdash (\forall x.P(x)) \Rightarrow ((\forall y.P(y) \Rightarrow Q(y)) \Rightarrow \forall z.Q(z))}\Rightarrow I$$

??

01.	$\forall x.P(x)$	Assup. 1
02.	$P(c)$	$\forall E$ on 1
03.	$Q(c)$	Assup 2
04.	$P(c) \Rightarrow Q(c)$	$\Rightarrow I$ on 3
05.	$\forall y.P(y) \Rightarrow Q(y)$	$\forall I$ on 4
06.	$\forall z.Q(z)$	$\forall I$ on 3
07.	$(\forall y.P(y) \Rightarrow Q(y)) \Rightarrow \forall z.Q(z)$	$\Rightarrow I$ on 6
08.	$(\forall x.P(x)) \Rightarrow ((\forall y.P(y) \Rightarrow Q(y)) \Rightarrow \forall z.Q(z))$	$\Rightarrow I$ on 7

This page was intentionally left blank for extra space

5 Knowledge Representation

Problem 5.1 (Specifying Properties in ALC) 9 pt

Consider the following ALC setting:

- concepts: `animal`, `plant`
- relations: `eats`

We abbreviate every concept/relation by its first letter.

Note: For the purposes of this question, carnivores are the animals that eat other animals, and herbivores are the animals that only eat plants.

---

1. Give ALC expressions for 3 pt
   (a) the concept of all herbivores 3 pt
   (b) the statement that carnivores do not eat each other 3 pt
2. Assume a domain $\mathcal{D}$ and interpretations $A \subseteq \mathcal{D}$, $P \subseteq \mathcal{D}$, and $E \subseteq \mathcal{D} \times \mathcal{D}$ of a, p, and e, respectively. Give the semantics of the formula $\forall e.\big((\exists e.a) \sqcap (\exists e.p)\big)$.

01. (a) $a \sqcap \forall e.p$

(b) $c = a \sqcap \exists e.a$

$c \sqcap \exists e.c \equiv \bot$

02. $\forall u.\big(\forall v.\ e(u,v) \Rightarrow \big(\exists x.\ e(v,x) \wedge a(x)\big) \wedge \big(\exists y.\ e(v,y) \wedge p(y)\big)\big)$

semantics $\{u \in D \mid \forall v \in D,$ if $\langle u,v\rangle \in E$ then, for $\exists x,y \in D$, $\langle v,x\rangle \in E, x \in A, \langle v,y\rangle \in E, y \in P\}$

??

Problem 5.2 (Extending ALC) 8 pt

Consider ALC concepts as given by the grammar

$$C ::= a \mid \top \mid \bot \mid \overline{C} \mid C \sqcap C \mid C \sqcup C \mid \exists R.C \mid \forall R.C$$

and with the

- semantics that maps every concept C to $[\![C]\!] \subseteq \mathcal{D}$,
- translation to first-order logic with equality that translates every concept C to a formula $\overline{C}^{fo(x)}$ with free variable x,

both defined by induction on concepts.

We want to extend ALC with the following two concept constructors:

- $C - D$ for the C's that are not D's
- $\exists^! R.C$ for objects that are connected via role R to *at most* one C

Give the cases that we must add to grammar, semantics, and translation to obtain that extension.

(a) Syntax with productions (grammar)

$$C ::= C - C \mid \exists^! R.C$$

(b) Semantics

$$[\![C - D]\!] = [\![C]\!] \setminus [\![D]\!]$$

$$[\![\exists^! R.C]\!] = \{x \in \mathcal{D} \mid \text{there is at most on } y \in [\![C]\!] \text{ with } (x,y) \in [\![R]\!]\}$$

(c) translation

$$\overline{C - D}^{fo(x)} := \overline{C}^{fo(x)} \wedge \neg \overline{D}^{fo(x)}$$

$$\boxed{A - B = A \wedge \neg B}$$

$$\overline{\exists^! R.C}^{fo(x)} := ??$$

$$\boxed{\overline{\exists R.C}^{fo(x)} := \exists y.\, R(x,y) \wedge \overline{C}^{fo(x)}}$$

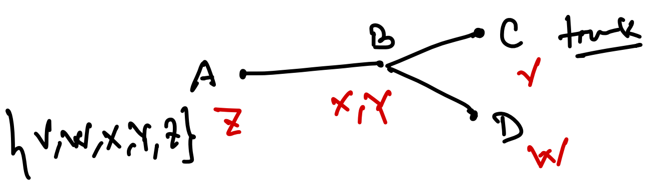
6 Planning

Problem 6.1 (Planning Deliveries in STRIPS) 12 pt

Consider a truck that can carry 4 objects at a time and is supposed to deliver objects $Obj = \{V, W, X, Y, Z\}$ from location A to certain locations $Loc = \{A, B, C, D\}$ along some roads $Roads = \{\{A, B\}, \{B, C\}, \{B, D\}\}$. We use the following STRIPS task:

- facts: $\{\mathtt{at}(l, o) \,|\, l \in Loc, o \in Obj\} \cup \{\mathtt{truck}(l) \,|\, l \in Loc\}$
- actions $\mathtt{move}(l, m, O)$ for $\{l, m\} \in Roads$, $O \subseteq Obj$, $|O| \leq 4$ given by
  - precondition: $\mathtt{at}(l, o)$ for all $o \in O$, $\mathtt{truck}(l)$
  - add list: $\mathtt{at}(m, o)$ for all $o \in O$, $\mathtt{truck}(m)$
  - delete list: same as precondition
- initial state: $\mathtt{truck}(C)$, $\mathtt{at}(A, o)$ for $o \in Obj$
- goal state: $\mathtt{at}(C, V)$, $\mathtt{at}(D, W)$, $\mathtt{at}(B, X)$, $\mathtt{at}(B, Y)$, $\mathtt{at}(A, Z)$

1. Give a sequence of two actions that is applicable in the initial state and give the resulting state. 4 pt
2. Give an optimal plan for the task above. 4 pt
3. Consider the following heuristics: $h(s) = \frac{1}{4} \sum_{o \in Obj} d(s, o)$ where $d(s, o)$ is the number of roads separating the location of o in state s from its location in the goal state. Argue whether this heuristic is admissible or not. 4 pt



01. move { C, B, ∅ }, move { B, Q, ∅ } where Q ∈ {A, C, D}

results: truck(Q), at(A, o) for o ∈ obj

02. move{C,B,∅}, move (B, A, ∅), move (A, B, {V,W,X,Y}), move (B, C, {V,W}), move (C,B,{W}), move (B,D,{W})

?

This page was intentionally left blank for extra space

?? / Q3. $h^*(I) = 6$

$h(s) \Rightarrow$ $v = 2$

$w = 2$

$x = 1$

$y = 1$

$z = 0$

6 / 4

$h(s) \leq h^*(s)$

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