Last Name: First Name

Matriculation Number: Birth Date:

Seat:

Exam
Artificial Intelligence 1

Feb 14, 2022

	To be used for grading, do not write here													
prob.	1.1	2.1	2.2	3.1	4.1	4.2	5.1	5.2	5.3	6.1	7.1	7.2	Sum	grade
total	10	8	7	6	6	7	6	8	8	9	12	8	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 point distributions for the problems are provisional.
3. You can reach 95 points if you fully solve all problems. You will only need 85 points for a perfect score, i.e. 10 points are bonus points.
4. No resources or tools are allowed except for a pen.
5. You have 90 min (sharp) for the test
6. Write the solutions directly on the sheets, no other paper will be graded.
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 (22 pages excluding 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.**

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

Erlangen, Feb 14, 2022 ....

(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. Sie können 95 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 85 Punkte bereits als volle Punktzahl, d.h. 10 Punkte sind Bonuspunkte.
4. Es sind keine Hilfsmittel erlaubt außer eines Stifts.
5. Die Bearbeitungszeit beträgt genau 90 min.
6. Schreiben Sie die Lösungen direkt auf die ausgeteilten Aufgabenblätter. Andere Blätter werden nicht bewertet.
7. 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.
8. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (22 Seiten exklusive 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 14, 2022 ....

(Unterschrift)

Please consider the following guidelines to avoid losing points:

- If you continue an answer on another page, clearly give the problem number on the new page and a page reference on the old page.
- If you do not want something to be graded, clearly cross it out.
- The instructions "Give X", "List X" or similar mean that only X is needed. If you additionally justify your answer, we will try to give you partial credit for a wrong answer (but there is no guarantee that we will).
- The instruction "Assume X" means that X is information that you may use in your answer.
- The instruction "Model X as a Y" means that you have to describe X formally and exactly as an instance of Y using the definition of Y from the lecture.
- If you are uncertain how long or complex an answer should be, use the number of points as an indication: 1 point roughly corresponds to 1 minute.

1 Prolog

Problem 1.1 (Prolog) 10 pt

Consider the following Prolog code:

```
isNat(zero).
isNat(succ(N)) :− isNat(N).                write

isTree(tree(N,Ts)) :− isNat(N), isTrees(Ts).

isTrees([]).
isTrees([H|T]) :− isTree(H), isTrees(T).

evaluateNat(zero,0).
evaluateNat(succ(N),X) :− evaluateNat(N,Y), X is Y+1.

% If T is a tree, treeSum(T,S) holds if
% S is the sum of the labels of the nodes (evaluated into Prolog integers) .
treeSum( tree(N,Ts), S ) :− evaluateNat(N,E),
                            treeListSum(Ts, F),
                            S is E+F

treeListSum([],S) :− S is 0.

treeListSum([H|T],S) :− treeSum(H,E), treeListSum(T,F), S is E+F.
```

1. What do the following queries return: 3 pt

   (a) isTree(tree(succ(zero),[tree(zero,[]), tree(succ(zero),[])])) yes

   (b) isTree(X) X = tree(zero, []) , X = tree(zero, [tree(z, [])])

   If a query returns multiple results, only give the first two.

2. Complete the implementation of the predicate treeSum. 5 pt
   Hint: Use treeListSum as an auxiliary predicate.

3. Assume we change the isTree predicate to 2 pt
   isTree(tree(N,Ts)) :− write(N), isNat(N), isTrees(Ts).

   Why and how does Prolog's search behavior matter now?

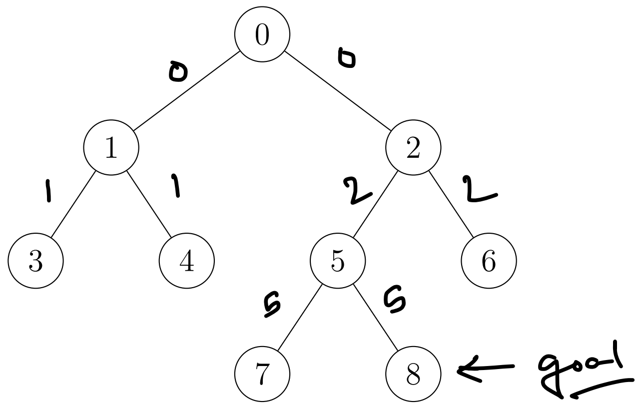
   isTrees $\xrightarrow[\text{dFS}]{\text{call}}$ isTree → write(N) print nodes in dfs order

This page was intentionally left blank for extra space

2 Search

Problem 2.1 (Search Algorithms) 8 pt

Consider the following tree with root 0:



Every edge from parent node i to child node j has path cost i.
As a heuristic for estimating the distance from node n to a goal node, we use $h(n) = 8 - n$.

Assume you have already expanded the root node. List the **next** 4 nodes that will be expanded using

1. depth-first search 1 pt

   0 1 3 4 2

2. breadth-first search 1 pt

   0 1 2 3 4

3. uniform-cost search 2 pt

   f : ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5 6
   c : 0 0 1 1 2 2

   v: 0 1 2 3 4

4. greedy search 2 pt

   f : 1 ~~2~~ ~~5~~ ~~6~~ 7 ~~8~~
   c=h(n) : 7 6 3 2 1 0
   v: 0 2 6 5 8

5. A^*-search 2 pt

   f : ~~1~~ ~~2~~ ~~5~~ ~~6~~ 7 8 3 4
   g(n)+h(n) : 0+7 0+6 2+3 2+2 7+1 7+0 1+5 1+4

   v : 0 2 6 5 1

If there is a tie, first expand the node with the smaller number.

Problem 2.2 (Search Problems) 7 pt

Consider the search problem (S, A, T, I, G) where

- $S = \mathbb{Z}$
- $A = \{-6, -4, 0, 5\}$
- $T(a, s) = \{a + s\}$
- $I = \{0\}$
- $G = \{7\}$

1. Give the state resulting from applying the action sequence $-6, -4, 5$ to the initial state. 1 pt

$0 \xrightarrow{-6} 0-6 = -6 \xrightarrow{-4} -6-4 = -10 \xrightarrow{5} -10+5 = -5$

2. Give a solution to the problem. 3 pt

$\underbrace{5, 5, 5,}_{15} \underbrace{-4, -4}_{-8} \Rightarrow 7$

3. Is DFS a good choice for this problem and why (not)? 2 pt

$0 \xrightarrow{-6} -6 \xrightarrow{-6} -12 \xrightarrow{-6} -18$

No, it will not find a sol^n.

4. If we change the set A to $\{-6, -4, 0, 2\}$, the problem changes substantially. In what way? 1 pt

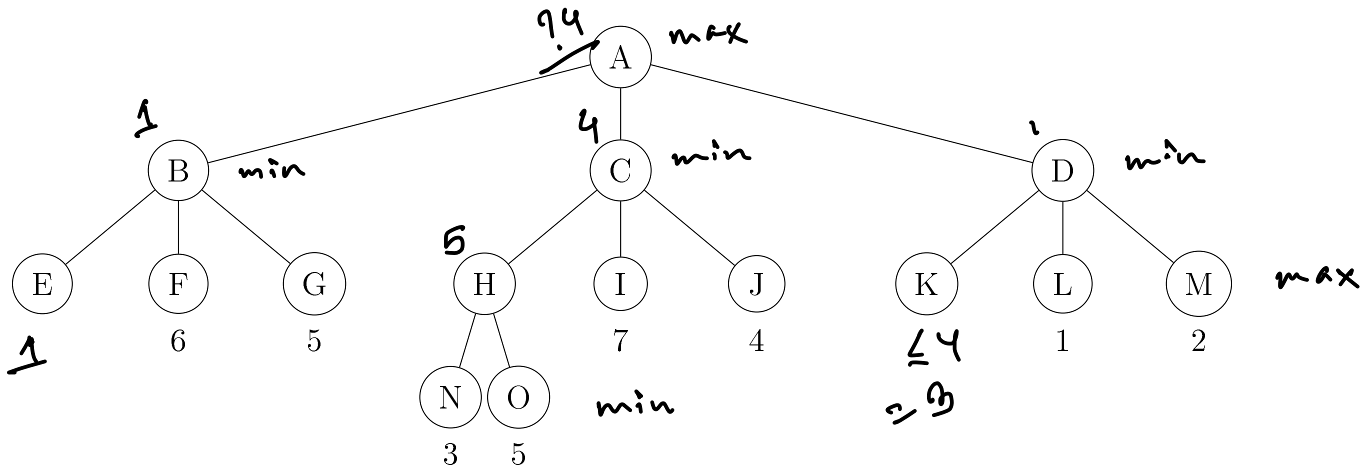
even $\rightarrow \times$ 7 odd

There is no sol^n.

3 Adversarial Search

Problem 3.1 (Minimax) 6 pt

Consider the following minimax game tree (**without** alpha-beta pruning) for the **maximizing player**'s turn. The values at the leafs are the static evaluation function values of those states; some of those values are currently missing.



1. Label the nodes H and C with their minimax values. H=5, C=4 2 pt
2. Label the node E with an evaluation function value that results in the player choosing move B. 2 pt
3. Now assume E is labeled 1.
   Label the node K with an evaluation function value that results in α-β-search pruning (i.e., not visiting) L and M.
   Assume children of a node are visited in alphabetical order. 2 pt

02. anything ≥ 4

03. anything ≤ 4

4 Constraint Satisfaction/Propagation

Problem 4.1 (Modeling) 6 pt

You are designing an exam consisting of 4 questions, and you need to assign a positive integer point value to each question.
The total points of the exam should be 20.
Questions 1 and 2 are hard and should contribute at most 10 points together.
Questions 2 and 3 cover essential topics and should contribute at least 12 points together.
Question 4 is long and should contribute more points than **any** of the others.

Model this problem as a (not necessarily binary) constraint satisfaction problem (V, D, C). Briefly explain the meaning of the variables.

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.

Var. $V = q_1, q_2, q_3, q_4$

Domain $D_{q_i} = \{1, 2, 3, \ldots\}$ for all $i = 1, 2, 3, 4$

or, $D_{q_i} = \{1, 2, 3, \ldots, n\}$, $n \geq 17$

Constraints C: $q_1 + q_2 + q_3 + q_4 = 20$

$q_1 + q_2 \leq 10$

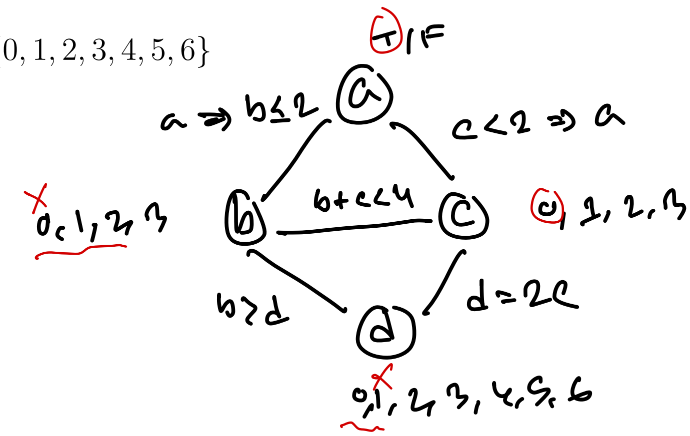
$q_2 + q_3 \geq 12$

$q_4 > q_i$, $i = 1, 2, 3$

Problem 4.2 (Solving) 7 pt

Consider the following binary CSP:

- $V = \{a, b, c, d\}$
- $D_a = \texttt{bool}$, $D_b = D_c = \{0, 1, 2, 3\}$, $D_d = \{0, 1, 2, 3, 4, 5, 6\}$
- Constraints:
  - if a, then $b \leq 2$
  - if $c < 2$, then a
  - $b + c < 4$
  - $b > d$
  - $d = 2c$



1. Give all pairs (v, w) of variables such that v is **not** arc-consistent relative to w. 3 pt

(a,b) ✓ (a,c) ✓ (b,c) ✓ (b,d) ✗ (c,d) ✓

(b,a) ✓ (c,a) ✓ (c,b) ✓ (d,b) ✗ (d,c) ✗

2. Give a solution. 1 pt

$a = T$, $b \in \{1,2\}$, $c = 0$, $d = 0$

3. Give an inconsistent total assignment to the variables. 1 pt

$a = F$, $b = 0$, $c = 0$, $d = 0$

4. Assume we assign a to be true, and apply forward-checking. Give the resulting domains D_b and D_c. 2 pt

$D_b = \{0, 1, 2\}$

$D_c = \{0, 1, 2, 3\}$

5 Logic

Problem 5.1 (Propositional Logic) 6 pt

We use the propositional variables P, Q, R. Consider the formula A given by

$$(P \vee Q) \Rightarrow \neg(Q \wedge (R \Rightarrow Q))$$

1. Using the assignment $\varphi(P) = T$, $\varphi(Q) = F$, and $\varphi(R) = T$, give the value $I_\varphi(A)$. 2 pt

$(T \vee F) \Rightarrow \neg(F \wedge (T \to F))$

T $\quad \neg F \quad F$

$T \quad T \to T$

$I_\varphi(A) = T$

2. Argue whether A is valid (i.e., give a proof or a counter-example). 4 pt

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

$(P \vee Q)^T$

$\neg(Q \wedge (R \to Q))^F$

$(Q \wedge (R \to Q))^T$

Q^T

$(R \to Q)^T$

$P^T \mid Q^T \quad \| \quad R^F \mid Q^T$

open

A in invalid

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

$(P \vee Q)^T \to P^T / Q^T$

$\neg(Q \wedge (R \to Q))^F \to (Q \wedge (R \to Q))^T$

$Q^T / (R \to Q)^T$

T

$Q = T$

Counter Example

Problem 5.2 (Definitions) 8 pt

Consider the following signature of first-order logic:

- unary function symbol f
- unary predicate symbol P

Give counter-examples for the following statements:

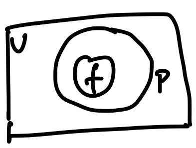
1. If a formula A is not a theorem, then $\neg A$ is a theorem. 2 pt

?? non-theorem non-contradiction, e.g. $A = \forall x\, P(x)$

2. Every model that satisfies $\forall x.P(f(x))$ also satisfies $\forall x.P(x)$. 3 pt

$n \to$ anything $\mathbb{N}$

$I(f) \subseteq I(P) \subset U$



e.g. $U = \mathbb{N}$, $I(P) = \{0\}$, $I(f)(n) = 0 \Downarrow f(n) = 0$

$\forall x\, P(f(x))$	$\forall x\, P(x)$
$P(f(0)) = P(0) \hookrightarrow T$	$P(0) \longrightarrow T$
$P(f(5)) = P(0) \hookrightarrow T$	$P(5) \to F$

3. In the model given by universe $\mathbb{N}$, $I(P) = \{n \in \mathbb{N} | n > 5\}$, and $I(f)(n) = n + 1$, we have $I_\varphi\Big(\big(P(f(x)) \wedge \neg P(y)\big) \Rightarrow P(f(y))\Big) = T$ for all assignments φ. 3 pt

$\downarrow$

$f(n) = n+1$

$\big(P(f(x)) \wedge \neg P(y)\big) \Rightarrow P(f(y))^F$

$\big(P(f(x)) \wedge \neg P(y)\big)^T$ and $P(f(y))^F$

$P(f(x))^T$

if $f(x) > 5$

$\Rightarrow x+1 > 5$

$\Rightarrow x > 4$ / $x \geq 5$

$\varphi(x) \geq 5$

if $f(y) \leq 5 \Rightarrow y+1 \leq 5 \Rightarrow y \leq 4$

$\varphi(y) \leq 4$

any assignment with $\varphi(x) \geq 5$ & $\varphi(y) \leq 4$

Problem 5.3 (Proving in Natural Deduction) 8 pt

??

Complete the following sequent-style natural deduction proof by filling in all boxes:

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\Gamma \vdash \forall x.P(x) \Rightarrow \neg Q(x)$}
\RightLabel{$\forall E$}
\UnaryInfC{$\Gamma \vdash \boxed{P(c) \rightarrow \neg Q(c)}$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\Gamma \vdash \exists y.P(y)$}
\RightLabel{$\exists E$}
\UnaryInfC{$\Gamma \vdash \boxed{P(c)}$}
\RightLabel{$\Rightarrow E$}
\BinaryInfC{$\Gamma \vdash \boxed{\neg Q(c)}$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\Gamma \vdash \forall z.Q(z)$}
\RightLabel{$\forall E$}
\UnaryInfC{$\Gamma \vdash \boxed{Q(c)}$}
\RightLabel{$falseI$}
\BinaryInfC{$\Gamma \vdash \boxed{false}$}
\RightLabel{$\neg I$}
\UnaryInfC{$\boxed{\forall x.P(x) \rightarrow \neg Q(x), \exists y.P(y) \vdash \neg \forall z.Q(z)}$}
\RightLabel{$\Rightarrow I$}
\UnaryInfC{$\boxed{\forall x.P(x) \rightarrow \neg Q(x) \vdash ((\exists y.P(y)) \rightarrow \neg \forall z.Q(z))}$}
\RightLabel{$\Rightarrow I$}
\UnaryInfC{$\vdash (\forall x.P(x) \Rightarrow \neg Q(x)) \Rightarrow ((\exists y.P(y)) \Rightarrow \neg \forall z.Q(z))$}
\end{prooftree}
$$

where we abbreviate

$$\Gamma = \boxed{\forall x.P(x) \rightarrow \neg Q(x), \exists y.P(y), \forall z.Q(z)}$$

iif, $H \vdash_{ND_0} A \rightarrow B$, $H, A \vdash_{ND_0} B$

There is a copy of the proof on the next page. You can use it if you mess up and want to start over.

This is the same proof as on the previous page. You can use it if you mess up and want to start over.

$$\dfrac{\overline{\Gamma \vdash \forall x.P(x) \Rightarrow \neg Q(x)}\,Ax}{\Gamma \vdash \boxed{P(c) \Rightarrow \neg Q(c)}}\,\forall E \qquad \dfrac{\overline{\Gamma \vdash \exists y.P(y)}\,Ax}{\Gamma \vdash \boxed{P(c)}}\,\exists E$$

$$\dfrac{\Gamma \vdash \boxed{P(c) \Rightarrow \neg Q(c)} \qquad \Gamma \vdash \boxed{P(c)}}{\Gamma \vdash \boxed{\neg Q(c)}} \Rightarrow E \qquad \dfrac{\overline{\Gamma \vdash \forall z.Q(z)}\,Ax}{\Gamma \vdash \boxed{Q(c)}}\,\forall E$$

$$\dfrac{\Gamma \vdash \boxed{\neg Q(c)} \qquad \Gamma \vdash \boxed{Q(c)}}{\Gamma \vdash \boxed{\text{false}}}\,falseI$$

$$\dfrac{\Gamma \vdash \boxed{\text{false}}}{\boxed{\forall x.P(x) \Rightarrow \neg Q(x),\ \exists y.P(y) \vdash \neg \forall z.Q(z)}}\,\neg I$$

$$\dfrac{\boxed{\forall x.P(x) \Rightarrow \neg Q(x),\ \exists y.P(y) \vdash \neg \forall z.Q(z)}}{\boxed{\forall x.P(x) \Rightarrow \neg Q(x) \vdash (\exists y.P(y) \Rightarrow \neg \forall z.Q(z))}} \Rightarrow I$$

$$\dfrac{\boxed{\forall x.P(x) \Rightarrow \neg Q(x) \vdash (\exists y.P(y) \Rightarrow \neg \forall z.Q(z))}}{\vdash \underbrace{(\forall x.P(x) \Rightarrow \neg Q(x))}\Rightarrow \underbrace{\big((\exists y.P(y)) \Rightarrow \neg\forall z.Q(z)\big)}} \Rightarrow I$$

where we abbreviate

$$\Gamma = \boxed{}$$

$\Rightarrow$

$\vdash$

iif, $H \vdash_{ND_0} A \to B$, | $H, A \vdash_{ND_0} B$

6 Knowledge Representation

Problem 6.1 (Specifying Properties in ALC) 9 pt

Consider the following ALC setting:

- concepts: `human`, `child`, `grownup`, `animal`
- relations: `isChildOf`, `owns`

We abbreviate every concept/relation by its first letter.

1. Give an ALC ABox that is not consistent with the axiom $h \sqcap \exists o.a = \bot$. 2 pt

   ?

   $x : h,\ y : a,\ x\ o\ y$ $\quad o(x,y)$

2. Give an ALC TBox that formalizes the following properties 3 pt

   - Children and grownups are humans, and humans are children or grownups. No one is both a child and a grownup.

     $h = c \sqcup g$

     $c \sqcap g = \bot$

   - Humans cannot be owned.

     $\exists o.h = \bot$

     ?!

     $\exists y\ o(x,y) \wedge h(y)$

     there is no y $\quad$ y is human

3. Give an ALC formalization for the concept of children who have a parent who owns an animal. 2 pt

   $c \sqcap \exists i.\exists o.a$

4. Give the translation to first-order logic of the ALC statement $(\forall i.h) \sqsubseteq (\exists i.g)$. 2 pt

   PL¹

   $\forall x.(\forall y. i(x,y) \Rightarrow h(y)) \Rightarrow (\exists y. i(x,y) \wedge g(y))$

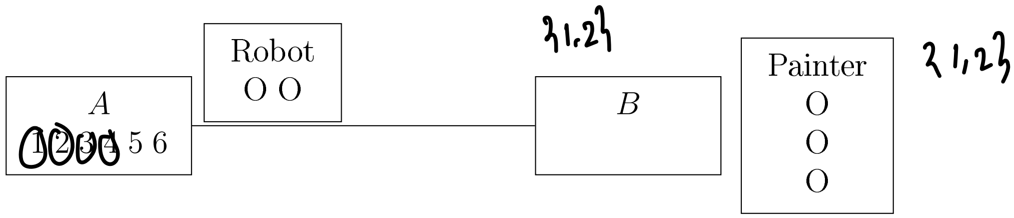
This page was intentionally left blank for extra space

$$\left.\begin{array}{r}1+1+1=3\\ =3\\ =3\end{array}\right\} 9$$

7 Planning

Problem 7.1 (STRIPS) 12 pt

Consider a set of objects $Obj = \{1, 2, 3, 4, 5, 6\}$ that can be at location A or B. Currently all objects are at location A and **unpainted**. Eventually all objects are needed in location A and **painted**. At location B, a painting station is available that can paint up to 3 objects at a time. A robot is available (currently at location A) that can move up to 2 objects at a time from one location to another.



We formalize this problem as a STRIPS task (P, A, I, G) where the set P of facts contains

- $\texttt{at}(l, o)$ for $l \in \{A, B\}$ and $o \in Obj \cup \{Robot\}$
- $\texttt{painted}(o)$ for $o \in Obj$

and the set A of actions contains

- $\texttt{move}(l, m, O)$ for $l, m \in \{A, B\}$, $O \subseteq Obj$, $|O| \leq 2$ given by
  - precondition: $\texttt{at}(l, o)$ for all $o \in O \cup \{Robot\}$
  - add list: $\texttt{at}(m, o)$ for all $o \in O \cup \{Robot\}$
  - delete list: same as precondition
- $\texttt{paint}(O)$ for $O \subseteq Obj$, $|O| \leq 3$ given by
  - precondition: $\texttt{at}(B, o)$ for all $o \in O$
  - add list: $\texttt{painted}(o)$ for all $o \in O$
  - delete list: nothing

1. Give the initial state I and the goal G. 2 pt

$I = at(A, o)$ for all $o \in obj \cup \{Robot\}$

$G = at(A, o), painted(o)$ for all $o \in obj$

2. After applying move$(A, B, \{1, 2\})$ in I, multiple actions are applicable. Give two of them. 3 pt

applicable actions:

- move$(B, A, \{1,2\})$
- painted$(\{1,2\})$

3. Give the value $h^*(I)$. 2 pt

take 2 obj to B $\xrightarrow{1}$ paint $\xrightarrow{1}$ back to A (1)

3x

$3 \times (1+1+1) = \underline{9}$

4. Give the value $h^+(I)$. 2 pt

relaxed plan

take 2obj to B (3) $\rightarrow$ paint (2)

$3+2=5$

5. Let $U_s(l)$ and $P_s(l)$ be the numbers of unpainted and painted objects at location l in state s. For each of the following heuristics $h(s)$, say if it is admissible. 3 pt

(a) $2 \cdot U_s(A) + U_s(B)$

(b) 0

(c) $U_s(A) + \texttt{roundDown}((U_s(A) + U_s(B))/3) + \texttt{roundDown}((P_s(B) + U_s(B))/2)$

(a) for $s = I$, $U_I(A) = 6$, $U_I(B) = 0$

$2 \cdot 6 + 0 = 12 < h^*_I = 9$, not admissible

(b) $0 \rightarrow$ always $\leq h^*_s$ so, admissible

(c) for $s = I$, $h_s = 6 + (6+0)/3 + (0+0)/2 = 8 \leq h^*_s = 9$

for s $h_s = 0 + (0+0)/3 + (2+0)/2 = 1$

$h^*_s = 1$

$\underline{h_s \leq h^*_s}$

A: 1,2,3,4 painted

B: 5,6 painted

$\underline{\text{admissible}}$

Problem 7.2 (Partial Order Planning) 8 pt

Consider the planning task (P, A, I, G) where

- facts $P = \{p, q, r, s\}$
- actions $A = \{X, Y, Z\}$ where the preconditions (above the box) and effects (below the box) of the actions are given by

$$\begin{array}{ccc} p & q & p \\ \boxed{\text{X}} & \boxed{\text{Y}} & \boxed{\text{Z}} \\ q & \neg p, r & \neg p, s \end{array}$$

- initial state $I = \{p\}$
- goal $G = \{r, s\}$

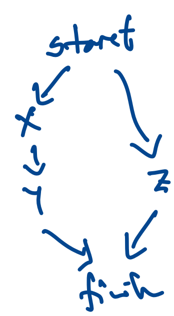
Our goal is to build a partially ordered plan. Recall that the steps consist of the actions plus the start and finish step; and that the effect of an action consists of the added facts and the negations of the deleted facts.

1. Give the start step and finish step. 2 pt

$$\boxed{\text{start}} \quad \overset{r,\,s}{\boxed{\text{finish}}}$$
$$p$$

2. Give all causal links between the steps. 2 pt

start $\xrightarrow{p}$ X, start $\xrightarrow{p}$ Z, X $\xrightarrow{q}$ Y, Y $\xrightarrow{r}$ finish, Z $\xrightarrow{s}$ finish



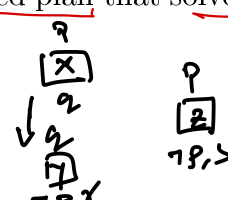
3. Give an example of a step that clobbers a link. 2 pt

Y & Z clobber link $\xrightarrow{p}$

4. Give the temporal ordering that yields a partially ordered plan that solves the task. 2 pt

start $\xrightarrow{p}$ X | start $\xrightarrow{p}$ Z

Z X ≺ Z | Y Z ≺ Y



When reusing this question, note that (while not relevant in this case) in general the same action can occur multiple times as different steps.

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

This page was intentionally left blank for extra space

This page was intentionally left blank for extra space