

# Grundlagen der Künstlichen Intelligenz

Albert-Ludwigs Universität Freiburg  
Institut für Informatik

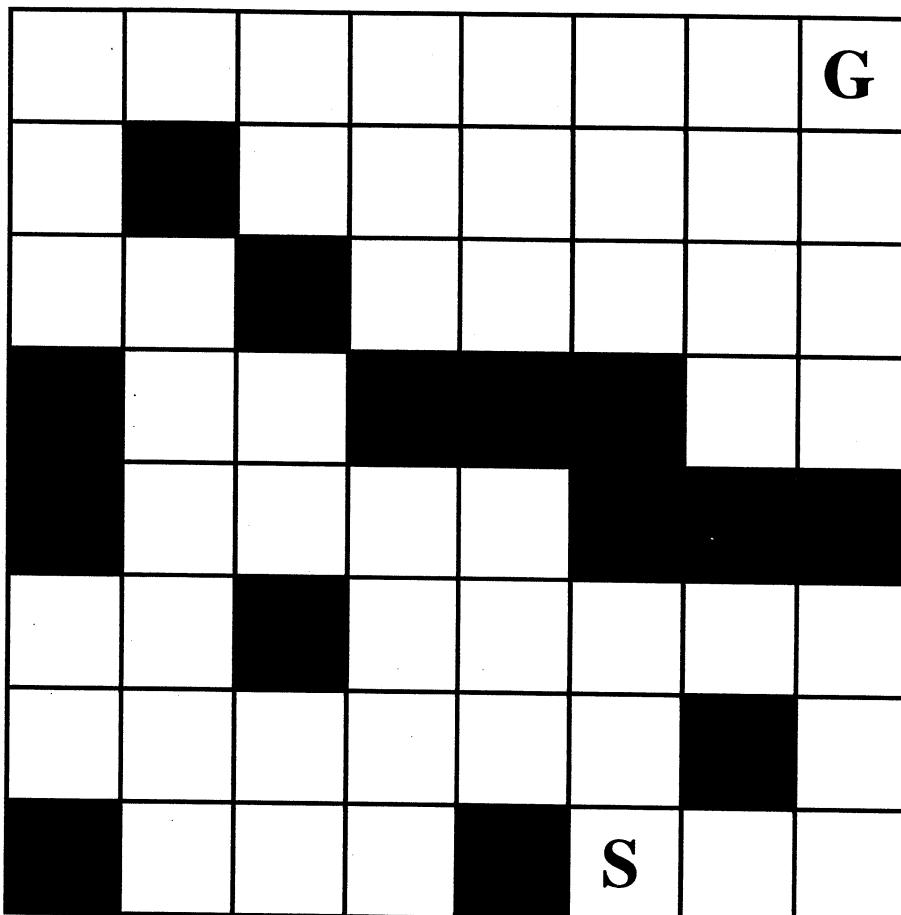
Die Klausur besteht aus **12 Frage**. Insgesamt sind **93 Punkte** zu erreichen.

Die Klausur dauert **90 Minuten**. Es ist nicht erlaubt Bücher, Skripte oder ähnliche Unterlagen zu benutzen. Sollte der Platz für eine Antwort nicht ausreichen, so schreiben Sie bitte auf der Rückseite der jeweiligen Seite weiter. Bitte vermerken Sie auf jeder einzelnen Seite an dem dafür vorgesehenen Platz Ihre **Matrikelnummer**.

Nachname, Vorname: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

1. (10 Punkte) Das Problem der Wegplanung in *Grid-Worlds*, wie sie weiter unten eine sehen, besteht darin, einen Pfad von einem Feld *S* zu einem Feld *G* zu finden. Ein Agent kann jeweils *ein* Feld in horizontaler oder vertikaler Richtung gehen. Schwarze Felder können nicht betreten werden.



Nehmen Sie an, dass der Agent *Bidirectional* sucht, um einen Pfad von *S* nach *G* zu finden. Genauer gesagt, benutzt er eine *Depth-First Search* von *S* ausgehend und eine *Breath-First Search* von *G* ausgehend. In beiden Fällen ist die Reihenfolge der Operatoren *rauf*, *rechts*, *runter* und *links*. Zyklen werden vermieden. Der Agent beginnt bei *S* und führt abwechselnd eine *Dept-First search* Operation und eine *Breath-First Search* Operation aus. Nummerieren Sie in der gegebenen *Grid-World* für beide Suchtechniken die Felder in der Reihenfolge, in der sie expandiert werden. Hören sie auf, sobald eine Lösung gefunden worden ist, und markieren Sie die Lösung.

2. Das folgende Puzzle soll mit Hilfe des  $A^*$  Algorithmus gelöst werden. Gegenstand des Puzzles sind Zahlen zwischen 100 und 999. Anfangs sind zwei Zahlen  $S$  und  $G$  gegeben, sowie eine Menge  $Bad$  von Zahlen. Ein Spielzug besteht darin, eine Zahl in eine andere Zahl zu verwandeln, indem man 1 zu einer Ziffer der Zahl addiert oder 1 von einer Ziffer subtrahiert. Ein gültiger Zug wäre also beispielsweise von 678 nach 679 oder von 234 nach 134. Jeder Zug hat die Kosten 1. Zusätzlich unterliegen die Züge den folgenden Einschränkungen:

- Es ist nicht erlaubt, zu der Ziffer 9 zu addieren oder von der Ziffer 0 zu subtrahieren.
- Es ist nicht erlaubt, einen Zug auszuführen, der die derzeitige Zahl in eine Zahl aus der Menge  $Bad$  überführt.
- Ein Spieler darf dieselbe Ziffer nicht in zwei aufeinanderfolgenden Zügen ändern.

Lösen Sie das Puzzle, indem Sie von  $S$  nach  $G$  mit der kleinstmöglichen Anzahl an Zügen gelangen.

- (1 Punkte) Geben Sie eine *State Description* (Zustandsbeschreibung) an, mit der der  $A^*$  Algorithmus angewendet werden kann.
- (3 Punkte) Definieren Sie eine *admissible Heuristik*, die für dieses Problem in einer  $A^*$  Suche verwendet werden kann. Erläutern Sie, warum Ihre Heuristik admissible ist.
- (6 Punkte) Verwenden Sie die Heuristik aus (b), um die ersten drei (3) *Knoten-Expansionen* der  $A^*$  Suche für  $S = 567$ ,  $G = 777$  und  $Bad = \{666, 667\}$  durchzuführen. Kennzeichnen Sie in dem Baum auch alle legalen Nachfolger von jedem Knoten, den Sie expandieren.

3. (5 Punkte) Finden Sie mittels der *Davis-Putnam* Prozedur ein Modell der folgenden aussagenlogischen Formel, oder beweisen Sie, dass die Formel unerfüllbar ist.

$$(A \vee B) \wedge (\neg B \vee \neg C) \wedge (C \vee D) \wedge (A \vee D) \wedge (\neg A \vee C) \wedge (B \vee \neg D) \wedge (\neg A \vee \neg C)$$

4. Betrachten Sie die folgenden beiden Sätze in Prädikatenlogik erster Stufe.

1.  $\phi_1 \equiv \forall x(\text{boy}(x) \Rightarrow \exists y(\text{girl}(y) \wedge \text{likes}(x, y)))$
2.  $\phi_2 \equiv \exists y(\text{girl}(y) \wedge \forall x(\text{boy}(x) \Rightarrow \text{likes}(x, y)))$

(a) (3 Punkte) Welche Folgerung ist korrekt? Tragen Sie "ja" oder "nein" in die entsprechenden Kästchen ein. (-2 für eine falsche Antwort)

1.   $\phi_1 \models \phi_2$

2.   $\phi_2 \models \phi_1$

(b) (10 Punkte) Beweisen Sie Ihre Behauptung mittels *Resolution Refutation*.

5. (5 Punkte) Geben Sie ein Prolog-Programm an, dass `append(X, Y, Z)` so definiert ist, dass `Z` die Konkatenation der Listen `X` und `Y` ist. Nehmen Sie an, dass alle drei Listen `Z`, `Y` und `X` als *Difference Lists* repräsentiert sind.

6. Betrachten Sie das folgende Prolog-Programm:

```
part(a). part(b). part(c). part(d).
red(a). black(b). red(c). black(c).
color(P,red) :- red(P).
color(P,black) :- black(P).
color(P,unknown).
```

- (a) (4 Punkte) Zeichnen Sie den SLD-Baum für die Anfrage `?- color(a, C).`

- (b) (3 Punkte) In dem Programm wird die Klausel `color(P,red) :- red(P)` durch  
`color(P,red) :- red(P), !.` ersetzt. Markieren Sie im SLD-Baum, welche Zweige dadurch abgeschnitten werden.
- (c) (3 Punkte) Ändert dieser Cut die Semantik des Programmes? Falls ja, erklären Sie kurz, warum. (**-3 Punkte für eine falsche Antwort**)

7. (5 Punkte) Zeichen sie den für *phase transitions* typischen Graphen, und erklären Sie anhand des Graphens in höchstens 5 Sätzen, was eine *phase transition* ist.

8. (8 Punkte) Es sei folgendes *Constraint Satisfaction Problem* (CSP) gegeben. Über der Menge  $\{A, E, G, K, M, O, R, S, T\}$  von 9 Bool'schen Variablen, d.h. alle Variablen haben die Domäne  $\{\text{true}, \text{false}\}$ , sind die folgenden (*Primitive*) *Constraints* gegeben:

$$\begin{aligned}M &\Leftrightarrow A \\G \wedge O &\Leftrightarrow R \\E \wedge R &\Leftrightarrow K \\T \wedge E &\Leftrightarrow S\end{aligned}$$

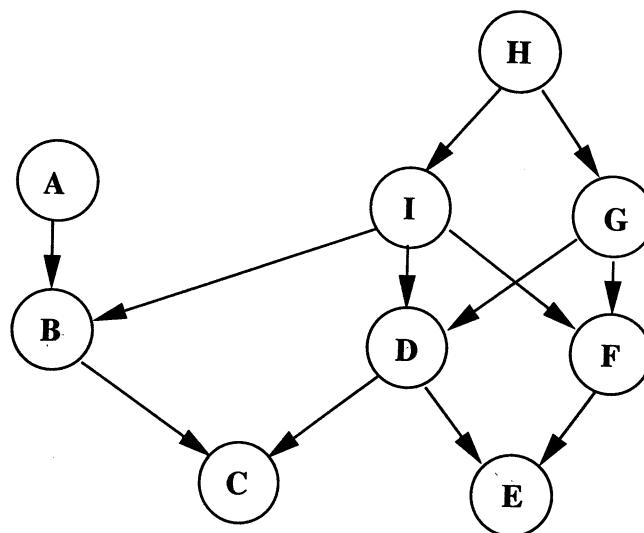
Nehmen Sie an, dass ein Algorithmus zum Lösen von CSPs schon die folgenden Variablenbelegungen getroffen hat:

$$A = \text{false}, G = \text{true}, K = \text{true}, M = \text{true}, O = \text{true}.$$

Geben Sie für jedes (*Primitive*) *Constraint* an, ob es *arc consistent* (AC) und/or *node consistent* (NC) ist gegeben die selektierten Variablenbelegungen. Dazu tragen Sie "ja" oder "nein" in die entsprechenden Kästchen ein (**-1 Punkt für jeden falschen Eintrag**):

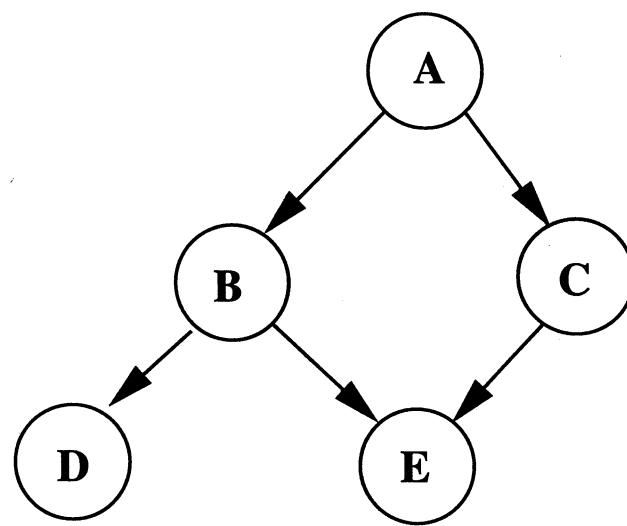
- |       |                          |    |                          |                                |
|-------|--------------------------|----|--------------------------|--------------------------------|
| 1. NC | <input type="checkbox"/> | AC | <input type="checkbox"/> | $M \Leftrightarrow A$          |
| 2. NC | <input type="checkbox"/> | AC | <input type="checkbox"/> | $G \wedge O \Leftrightarrow R$ |
| 3. NC | <input type="checkbox"/> | AC | <input type="checkbox"/> | $E \wedge R \Leftrightarrow K$ |
| 4. NC | <input type="checkbox"/> | AC | <input type="checkbox"/> | $T \wedge E \Leftrightarrow S$ |

9. Betrachten Sie folgendes *Bayesian Network*:



- (a) (3 Punkte) D-separiert die Menge  $\{B, E, H\}$  den Knoten  $A$  von  $G$ ? Geben Sie als Antwort **ja** oder **nein** an. Im Falle von **nein** geben sie einen Pfad an, der nicht blockiert ist. (-2 Punkte für eine falsche Antwort.)
- (b) (3 Punkte) D-separiert die Menge  $\{I\}$  den Knoten  $B$  von  $E$ ? Geben Sie als Antwort **ja** oder **nein** an. Im Falle von **nein** geben sie einen Pfad an, der nicht blockiert ist. (-2 Punkte für eine falsche Antwort.)

10. (6 Punkte) Gegeben sei das folgende *Bayesian Network*:



Zeigen Sie, wie *Variable Elimination* die Wahrscheinlichkeit von  $E = e$  berechnet bei Anwendung der *Elimination Order*  $A, B, C, D$ .

11. (10 Punkte) In dieser Aufgabe soll ein Konzept erlernt werden. Die Instanzen sind durch die Bool'schen Variablen  $A$  und  $B$  beschrieben. Es seien die folgenden zwei Beispiele gegeben:

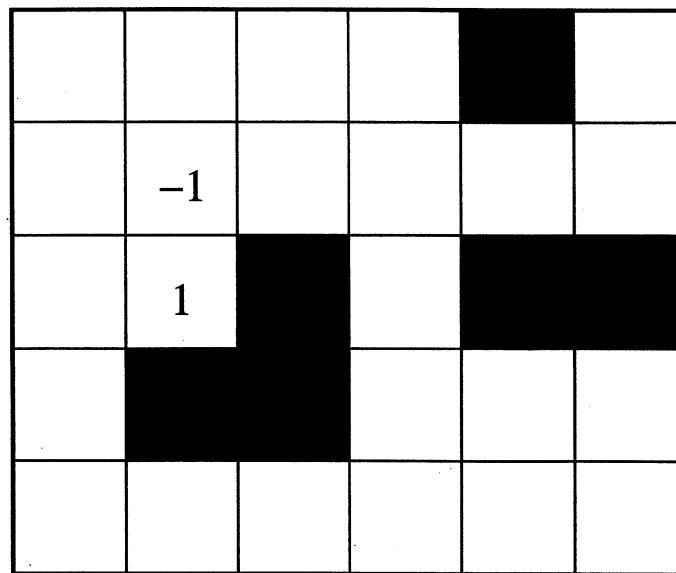
A	B	Class
true	true	positive
false	false	negative

Nehmen Sie an, dass der Hypothesenraum  $H$  aus allen möglichen *Disjunktionen* über den Variablen  $A$  und  $B$  besteht, d.h.

$$H =_{def} \{ \begin{aligned} & true, \\ & A, \neg A, B, \neg B, \\ & A \vee B, A \vee \neg B, \neg A \vee B, \neg A \vee \neg B, \\ & false \end{aligned} \}$$

wobei  $A \vee B$  für ( $A = true \vee B = true$ ) steht. Die zwei weiter oben angegebenen Beispiele induzieren einen *version space* in  $H$ . Geben sie die *G-set* und die *S-set*, die den *version space* beschreiben.

12. (5 Punkte) Betrachten Sie die deterministische Version der folgenden *Grid-World*:



Wie üblich kann der Roboter sich immer nur ein Feld nach *oben*, *links*, *rechts* oder *unten* bewegen. Geben Sie im Diagramm die *optimal policy* an, wobei sich der Nutzen (*utility*) einer Aktionssequenz als

$$\text{Endwert} - \frac{1}{100} \cdot \text{Anzahl der Aktionen.}$$

berechnet.

## Klausur Künstliche Intelligenz Sommersemester 2005

Bearbeitung: 12.09.2005, 14.00-16.00 Uhr

Name, Vorname: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Hauptfach, Nebenfach: \_\_\_\_\_

- Es sind **keine** Hilfsmittel zugelassen (Bücher, Skripte, Taschenrechner etc.).  
*This is a closed-book exam. Usage of books, lecture notes, pocket calculators etc. is not allowed.*
- Abschreiben oder der Versuch werden mit Ausschluss von der Klausur geahndet.  
*Cheating and attempted cheating will lead to disqualification from the exam.*
- Schreiben Sie Namen und Matrikelnummer auf jedes Blatt.  
*Write your name and matriculation number on each sheet.*

Aufgabe	Punkte	Erreichbar
1.1		8
1.2		6
2		6
3.1		5
3.2		5
3.3		2
3.4		3
4.1		6
4.2		2
4.3		4
5		5
6.1		6
6.2		9
6.3		6
7.1		4
7.2		5
7.3		3
8.1		4
8.2		3
9		8
Gesamt		100

Name:

Matr.-Nr.: \_\_\_\_\_

**Aufgabe 1 (8+6 Punkte)**

Betrachten Sie das Spielfeld aus dem bekannten Spiel *Minesweeper*. Ziffern zwischen 0-8 geben die *genaue* Anzahl der Minen in direkt benachbarten Felder an. Ausserhalb des gegebenen Spielfeldes und auf mit Ziffern gekennzeichneten Feldern liegen keine Minen. Auf jedem weiteren Feld liegt maximal eine Mine. Gesucht ist eine mit den Angaben konsistente Belegung des Spielfeldes mit Minen.

*Consider the board of the well known game Minesweeper. Numbers from 0-8 give the exact amount of mines in the neighbouring fields. Beyond the given board are no further mines. Each field of the board can contain maximally one mine. We are looking for a allocation of the board with mines consistent with the given information.*

1. Nehmen Sie an, dass die einzelnen Variablen mit ganzen Zahlen belegt werden. Formulieren sie das Problem als CSP indem sie jedes Constraint als einzelne Formel aufschreiben. Geben Sie die Definitionsbereiche der Variablen an.  
*Assume that variables are integer-valued. Formulate the given problem as CSP. Write down a formula for every constraint. Give the domains.*
2. Berechnen sie eine Lösung unter Verwendung von *Forward checking*.  
*Calculate an assignment using forward checking.*

A	B	C	D
0	E	1	F
G	H	K	L
M	2	2	N

Name:

Matr.-Nr.:

---

**Aufgabe 2 (6 Punkte)**

Bei einem Gewitter gibt es häufiger Blitze. Wenn ein solcher Blitz in einem Wald einschlägt, kann das zu einem Waldbrand führen (es sei denn, es regnet zu stark). Oft entsteht ein Waldbrand aber auch nur durch die Unachtsamkeit der Touristen, die an einem Lagerfeuer feiern. Natürlich wird kaum ein Tourist feiern, wenn es regnet.

*In a thunderstorm there is lightning. If lightning strikes in the woods it can cause a forest fire (unless it rains too hard). But a forest fire is often just caused by the carelessness of tourists partying at a camp fire. Obviously, hardly a tourist will party if it rains.*

1. Modellieren Sie die Abhängigkeiten zwischen den Wahrscheinlichkeitsvariablen "Blitz", "Lagerfeuer", "Gewitter" und "Waldbrand" in einem Bayesschen Netz, d.h. wie würde der entsprechende gerichtete Graph aussehen?

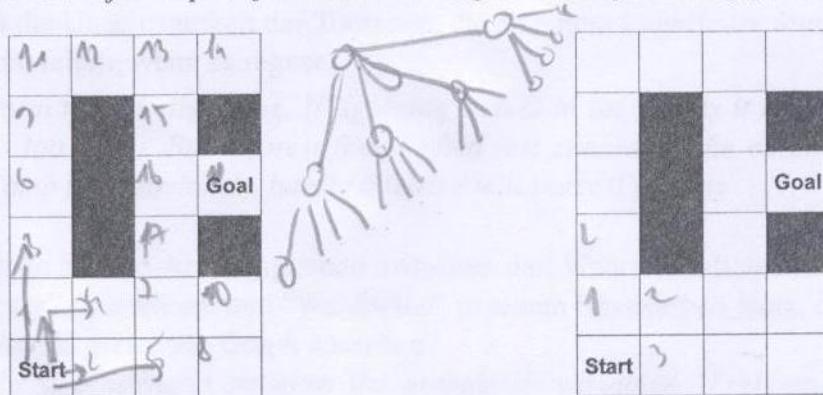
*Model the dependencies between the probability variables "Lightning", "Camp fire", "Thunderstorm", and "Forest fire" in a Bayesian network, i.e. how would the directed graph look?*

Name:

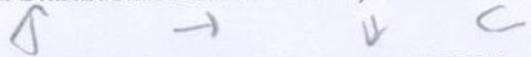
Matr.-Nr.:

### Aufgabe 3 (5+5+2+3 Punkte)

Angenommen, ein Agent soll einen Pfad vom Start zum Ziel in der folgenden Gitterwelt finden:  
Assume an agent has to find a path from start to the goal in the following grid:



Nehmen Sie an, der Agent hat die vier möglichen Aktionen  $N$  (gehe nach Norden),  $E$  (gehe nach Osten),  $S$  (gehe nach Süden),  $W$  (gehe nach Westen) - in dieser Reihenfolge - im allgemeinen und eine Teilmenge davon an den Rändern (z.B. im Startzustand nur  $N$  und  $E$ ).



Suppose the agent has the four possible actions  $N$  (go north),  $E$  (go east),  $S$  (go south),  $W$  (go west) (in this order!) in general and only a subset at the borders (so, e.g. for the start cell we only have  $N$  and  $E$ ).

1. In welcher Reihenfolge würden die Zellen betrachtet werden, wenn wir Breitensuche anwenden? Markieren Sie (im linken Feld) jede Zelle mit der entsprechenden Zahl, angefangen mit 0 und einem Pfeil von der Expansierten Zelle in die neue Zelle.  
*In which order would the cells be visited if we apply breadth-first search? Mark (in the left field) each cell with the corresponding number starting with 0. Add an arrow from the expanded to the new cell.*
2. Was passiert bei Tiefensuche? Markieren Sie entsprechend die Felder im rechten Feld.  
*What happens if we apply depth-first search? Mark the cells accordingly in the right field.*
3. Wie lang ist der Weg vom Start für den jeweils ersten gefunden Weg zum Ziel bei der Tiefensuche und bei der Breitensuche?  
*How long is the path from the start to the goal for the paths found first by depth-first-search and breadth-first-search?*
4. Ist einer der beiden Wege optimal? Begründen Sie ihre Antwort ausführlich (maximal fünf Sätze).  
*Is any of the paths optimal? Justify your answer in detail.*

**Hinweis:** Nehmen Sie an, dass bereits besuchte Zellen markiert werden, sodass keine Zelle mehrfach besucht wird.

**Note:** Assume that cells already visited are marked, so that no cell is visited more than once.

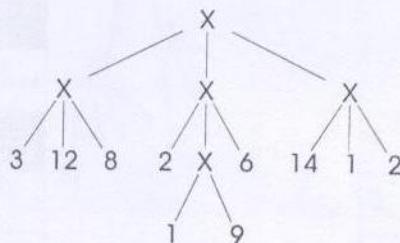
Name:

Matr.-Nr.:

**Aufgabe 4 (6+2+4 Punkte)**

Betrachten Sie den gegebenen Spielbaum. Nehmen Sie an, dass der erste Spieler der MAX-Spieler ist und die Blätter des Baums seinen Nutzwert (Utility) ausdrücken.

*Consider the game search tree in the figure below. Assume the first player is the MAX-player and the values at the leaves of the tree reflect its utility.*



1. Wenden sie den  $\alpha$ - $\beta$ -pruning Algorithmus (von links nach rechts) auf dem gegebenen Spielbaum an. Streichen Sie nichtbesuchte Äste aus dem Originalbaum und tragen sie die  $\alpha$  und  $\beta$  Werte der an den entsprechendne Knoten ein.

*Use the  $\alpha$ - $\beta$ -pruning algorithm (from left to right) on the given tree. Strike out unvisited branches of the given tree and enhance the nodes with their corresponding  $\alpha$  and  $\beta$  values.*

2. Welchen Zug würde MAX idealerweise wählen?

*Which move should MAX choose?*

3. Ordnen Sie den Baum so um das der  $\alpha$ - $\beta$ -pruning Algorithmus jeden der Äste besuchen muss.

*Reorder the tree such that the  $\alpha$ - $\beta$ -pruning algorithm has to visit every branch.*

Name:

Matr.-Nr.:

---

**Aufgabe 5 (5 Punkte)**

Betrachten Sie die folgenden bedingten Wahrscheinlichkeitsverteilungen:

Consider the following conditional probability distributions:

		$\mathbf{P}(A)$	
		$A = \text{true}$	$A = \text{false}$
		0.1	0.9
$\mathbf{P}(B A)$		$\mathbf{P}(C A)$	
		$B = \text{true}$	$B = \text{false}$
$A = \text{true}$	0.8	0.2	$C = \text{true}$
$A = \text{false}$	0.1	0.9	$C = \text{false}$
		0.3	0.7
		0.8	0.2

1. Berechnen Sie unter Zuhilfenahme der gegebenen Tabellen  $P(B = \text{true}|C = \text{true})$ .  
Calculate  $P(B = \text{true}|C = \text{true})$  using the given conditional probability distributions.

Name: \_\_\_\_\_

Matr.-Nr.: \_\_\_\_\_

**Aufgabe 6 (6+9+6 Punkte)**

Jeder der alle Tiere liebt wird von jemandem geliebt. Jeder der ein Tier tötet wird von niemandem geliebt. Jack liebt alle Tiere. Entweder Jack oder Neugier tötete die Katze mit dem Namen Tuna.

*Everyone who loves all animals is loved by someone. Anyone who kills an animal is loved by no one. Jack loves all animals. Either Jack or Curiosity killed the cat, who is named Tuna.*

1. Transformieren Sie die Aussagen in Prädikatenlogik erster Stufe.

*Transform the sentences into first-order predicate logic (PL1).*

2. Überführen Sie obige Ausdrücke in die Klauselform.

*Transform the result from (1) into the clausal form.*

3. Beweisen Sie mit Hilfe der Resolution, dass *die Katze* an *Neugier* starb.

*Prove by resolution that Curiosity killed the cat.*

Name:

Matr.-Nr.:

**Aufgabe 7 (4+5+3 Punkte)**

Die Abbildung zeigt ein bekanntes Problem aus der *Blocksworld*. In (a) ist der Ausgangszustand dargestellt und in (b) der Zielzustand. Wir verwenden die in der Vorlesung erwähnten Prädikate  $\text{Clear}(x)$ ,  $\text{On}(x, y)$  und  $\text{OnTable}(x)$ , um den Zustand zu beschreiben.

The figure below shows a well-known problem from the blocksworld. (a) shows the starting state and (b) the goal state. We use the predicates introduced in the lecture, namely  $\text{Clear}(x)$ ,  $\text{On}(x, y)$ , and  $\text{OnTable}(x)$ , to describe states.

1. Beschreiben Sie den Zielzustand und den Ausgangszustand mit den Prädikaten  $\text{Clear}(x)$ ,  $\text{On}(x, y)$  und  $\text{OnTable}(x)$ .

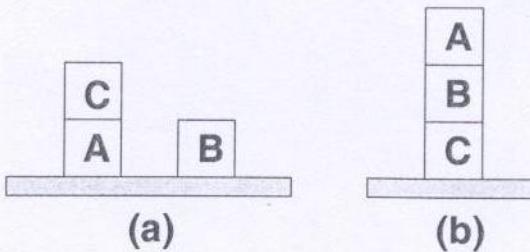
Describe the goal state and the starting state with the predicates  $\text{Clear}(x)$ ,  $\text{On}(x, y)$ , and  $\text{OnTable}(x)$

2. Definieren Sie in STRIPS zwei Operatoren  $\text{Stack}(x, y)$  und  $\text{Unstack}(x, y)$ . Mit  $\text{Stack}(x, y)$  soll ein Block  $x$  vom Tisch auf einen Block  $y$  gelegt werden, mit  $\text{Unstack}(x, y)$  soll ein Block  $x$ , der auf einem Block  $y$  liegt, auf den Tisch gelegt werden.

Define two operators  $\text{Stack}(x, y)$  and  $\text{Unstack}(x, y)$  using the STRIPS language.  $\text{Stack}(x, y)$  is used to move a block  $x$  from the table on top of a block  $y$ ,  $\text{Unstack}(x, y)$  is used to move a block  $x$  from a block  $y$  to the table.

3. Definieren Sie einen Operator  $\text{Move}(x, y, z)$ , mit dem ein Block  $x$ , auf dem sich keiner weiterer Block befindet, von einem Block  $y$  auf einen freien Block  $z$  gestellt wird. Verwenden Sie dazu die Operatoren  $\text{Stack}$  und  $\text{Unstack}$  aus der vorhergehenden Aufgabe.

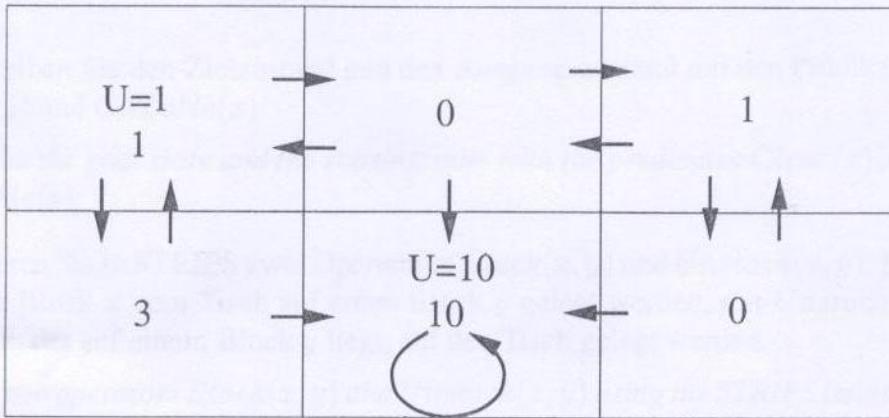
Define an operator  $\text{Move}(x, y, z)$  which is used to move a block  $x$ , which has no other block on top of it, from a block  $y$  on top of a block  $z$ . Use  $\text{Stack}(x, y)$  and  $\text{Unstack}(x, y)$  from the previous question.



**Aufgabe 8 (4+3 Punkte)**

Betrachten Sie die Gitterwelt in der untenstehenden Abbildung. Jeder Zustand ist mit der direkten Belohnung beschriftet, die man für das Erreichen des Zustandes erhält. Die Pfeile stellen mögliche Aktionen dar.

*Consider the grid world shown below. Each state is numbered with the immediate reward gained for reaching this state. The arrows correspond to actions than can be taken.*



1. Schreiben Sie die deterministische Version der Bellman-Gleichung für den oberen mittleren Zustand auf.

*Write down the deterministic version of the Bellman equation for the upper middle state.*

2. Betrachten Sie die Nutzenwerte (U) für den unteren mittleren und den oberen linken Zustand. Nehmen Sie einen Discountfaktor von 0.8 und ein deterministisches Modell an und aktualisieren Sie den Nutzenwert für den unteren linken Zustand. Schreiben Sie alle Berechnungen auf, die dafür durchgeführt werden müssen. Was ist die beste Aktion im unteren linken Zustand, ausgehend vom aktualisierten Nutzenwert?

*Consider the utilities (U) shown for lower middle and upper left state, respectively. Assuming a discount factor of 0.8 and a deterministic model, update the utility of the lower left state. Write down all calculations that have to be performed. Based on the updated utility, what would be the best action to take in the lower left state?*

Name:

Matr.-Nr.: \_\_\_\_\_

**Aufgabe 9 (8 Punkte)**

Entscheidungsbäume können die “alle-ausser-einem” Funktion darstellen, welche genau dann *true* ist, wenn alle bis auf eine der booleschen Variablen  $X_i, (i = 1, \dots, n)$  *true* sind.

*Decision trees can represent the “all-but-one” function, which is true if all but one of the boolean variables  $X_i, (i = 1, \dots, n)$  are true.*

1. Können Entscheidungsbäume die “alle-ausser-einem” Funktion mit einer *polynomiellen* Anzahl an Knoten in  $n$  darstellen? Antworten Sie mit **Ja** oder **Nein**. Begründen Sie kurz Ihre Antwort.

*Can Decision Trees represent the “all-but-one” function using a number of nodes which is polynomial in  $n$ ? Answer with yes or no. Explain your answer.*

**Foundations of Artificial Intelligence**  
**Summer Semester 2006**

Time: September 27th 2006, 10:00-12:00

Full name: \_\_\_\_\_

Matriculation no.: \_\_\_\_\_

Degree course: \_\_\_\_\_

- This is a closed-book exam. Using books, lecture notes, calculators etc. is **forbidden**.
- Cheating and attempted cheating will lead to disqualification from the exam.
- Write your answers directly underneath the exercises. Continue on the following blank page if you need additional space.
- Write your name and matriculation number on each sheet.

exercise	score	max. score
1	9	
2	11	
3.1	2	
3.2	3	
3.3	3	
3.4	4	
4	11	
5.1	6	
5.2	5	
6.1	6	
6.2	5	
7.1	4	
7.2	3	
7.3	3	
7.4	3	
8.1	6	
8.2	5	
9.1	4	
9.2	3	
9.3	4	
bonus	+14	
total		100

Name:

Matr. no.:

### **Exercise 1 (9 marks)**

PREDICATE LOGIC

Prove the *contradiction theorem*:

$\Theta \cup \{\varphi\}$  is unsatisfiable if and only if  $\Theta \models \neg\varphi$

Alternative formulation:

$\Theta \cup \{\varphi\} \models \perp$  if and only if  $\Theta \models \neg\varphi$

You may assume that the *deduction theorem* and *contraposition theorem* are true.

Name: \_\_\_\_\_

Matr. no.: \_\_\_\_\_

**Exercise 2 (11 marks)**

REASONING IN PREDICATE LOGIC

Consider the knowledge base

$$\Theta = \{\forall x(\neg(\exists z(A(z) \wedge B(x))) \vee C(x)), \\ \forall y(\neg B(y) \Rightarrow C(y))\}.$$

Prove that  $\Theta \models (\forall x A(x)) \Rightarrow (\exists y C(y))$ .

Name: \_\_\_\_\_

Matr. no.: \_\_\_\_\_

**Exercise 3 (2+3+3+4 marks)**

**BAYESIAN NETWORKS**

Consider the Bayesian network which is given by the following conditional probability tables:

$P(G)$	$P(S)$	$P(M G, S)$	$G$	$S$
0.90	0.20	0.90	true	true
		0.10	true	false
$P(T M)$	$M$	0.30	false	true
0.37	true	0.60	false	false
0.61	false			

1. Draw the graph representation of the Bayesian network.
2. Indicate the *Markov blanket* for nodes  $M$  and  $S$ , respectively.
3. The information represented by the given Bayesian network could also be represented by a joint probability table. Briefly (at most 3 lines!) explain the main advantages and disadvantages of each approach.
4. Compute  $P(M)$ .

Name: \_\_\_\_\_

Matr. no.: \_\_\_\_\_

**Exercise 4 (11 marks)**

**PARSING**

Consider the following grammar:

1	$S$	$\rightarrow$	$NP\ VP$
2	$NP$	$\rightarrow$	$D\ N$
3	$VP$	$\rightarrow$	$V$
4	$VP$	$\rightarrow$	$V\ NP$
5	$D$	$\rightarrow$	the
6	$N$	$\rightarrow$	professor
7	$N$	$\rightarrow$	student
8	$V$	$\rightarrow$	examines

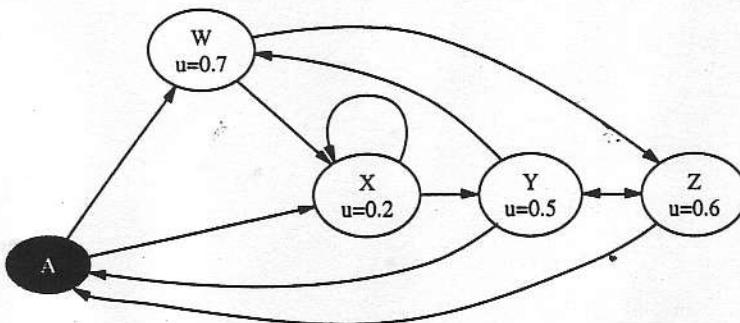
Parse the following sentence using a *bottom up parser*:

the professor examines the student

Give the trace of the algorithm using the notation introduced in the lecture.

**Exercise 5 (6+5 marks)****DECISION THEORY**

Consider the graph depicted below. An agent moves along the graph by following the arcs. Every movement action succeeds with a (small) probability of 0.2; the remaining probability is distributed among the other successors of the current vertex (the agent *must* move by following an arc). The values depicted for vertices  $W$ ,  $X$ ,  $Y$  and  $Z$  denote the utility of the respective state.



1. Calculate the utility of state  $A$ . The immediate reward for this state is 0.2, and the discount factor is  $\gamma = 0.25$ . Write down all steps of your calculation.
2. Which is the best action for the agent in state  $Y$ ? Write down all steps of your calculation and justify your answer.

Name:

Matr. no.:

---

**Exercise 6 (6+5 marks)**

**BOARD GAMES**

The *Nim* game is played as follows: Two players take turns removing one, two, or three coins from a stack of coins. The player who removes the last coin loses the game.

1. Draw the game tree for an initial state with 4 coins, and analyze it with the *Minimax* algorithm. Show an optimal strategy for both players by marking an optimal move in each of the inner nodes of the game tree.
2. Which of the players can force a win in the case of 5 or 6 coins in the initial state, respectively? Use the result from the first part of the exercise. Do not draw a complete game tree.

Name:

Matr. no.:

---

**Exercise 7 (4+3+3+3 marks)**

**CONSTRAINT SATISFACTION PROBLEMS**

Consider the following instance of the  $2 \times 2$  *Sudoku puzzle*:

1			
	2		
			1
4			3

The objective is to fill those cells which are initially empty with the numbers 1, 2, 3 and 4 in such a way that each of these numbers appears exactly once in each row, in each column, and in each of the four  $2 \times 2$  subgrids. *Well-formed* Sudoku puzzles do not require “guesses”.

1. Specify the general  $2 \times 2$  Sudoku puzzle as a CSP.
2. Describe (briefly!) a backtracking algorithm for solving the problem. Name two heuristics that can potentially improve the performance of the algorithm.
3. Describe (briefly!) a hill climbing algorithm for solving the problem.
4. Which of the two algorithms from parts 2. and 3. would you prefer to use in an efficient solver for well-formed Sudoku puzzles? Justify your answer briefly.

Name: \_\_\_\_\_

Matr. no.: \_\_\_\_\_

**Exercise 8 (6+5 marks)**

**PLANNING**

The (simplified) *Towers of Hanoi* problem consists of three pegs and three discs of different size. Initially, all discs are stacked on peg 1, ordered by size (with the largest one at the bottom). The objective is to move all discs to peg 3, ordered in the same way as they are initially. Only one disc may be moved at a time.



The following rules apply:

- Only the topmost disc of each peg may be moved.
  - Discs may only be moved on top of larger discs or onto empty pegs.
1. Express the problem as a planning problem in the STRIPS formalism (initial state, actions, goals). Use the three predicates *clear*, *smaller* and *on*, three objects for the three discs and three objects for the three pegs. A single action *move* is sufficient.
  2. State a solution for the problem.

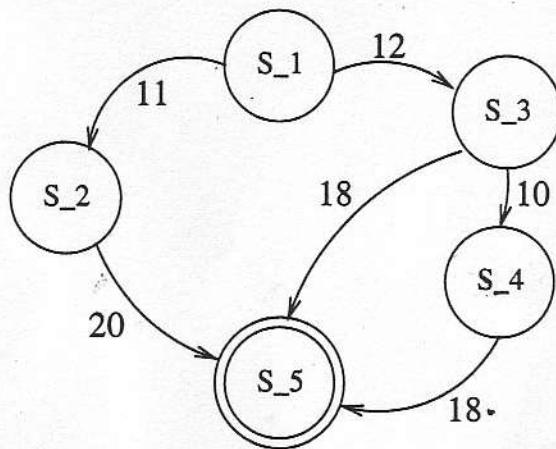
Name: \_\_\_\_\_

Matr. no.: \_\_\_\_\_

**Exercise 9 (4+3+4 marks)**

**INFORMED SEARCH**

Consider the following state space with initial state  $s_1$  and goal state  $s_5$ :



Moreover, consider the following heuristic function:

$$h(s_1) = 17, \quad h(s_2) = 17, \quad h(s_3) = 24, \quad h(s_4) = 17, \quad h(s_5) = 0$$

1. Why might the  $A^*$  algorithm, using this heuristic function, compute a suboptimal solution?
2. Modify the heuristic function in such a way that  $A^*$  is guaranteed to compute an optimal solution.
3. Apply the  $A^*$  algorithm to the problem with the modified heuristic function. Record all search nodes which are generated during the search along with their respective path costs and heuristic estimates. Also state in which order the nodes are expanded.

# Prüfungsprotokoll Staatsexamen 06

Informatik (II, III, KI) Burgard

Funktionentheorie (I & Weierstraßscher Produktsatz) - Wolke

Zahlentheorie (I & Algebraische und transzendenten Zahlen) – Wolke

## Informatik

Was macht man in Info II? ... Bachmann-Landau-Symbolik (aufgeschrieben, erklärt warum Konstante nicht so wichtig ist)

Quicksort (Funktionsweise und Laufzeiten)

Suchbaum (Gestalt, einfügen, entfernen, Laufzeiten)

besser? AVL-Bäume (Idee, einfügen (max. 1 Rotation), entfernen, worst case: logn Rotationen)

Zum letzteren hat er nach einem Beweis gefragt (??). Ich schätze er wollte auf die Gestaltanalyse raus, da er nachher noch den Vorfaktor von 1,4 erwähnt hat.

Wozu braucht man Sprachen? (Wusste ich nicht. Problem - Programm schreiben - basiert auf Sprachen)

Chomsky Hierarchie (Bsp für nicht reguläre und nicht kontextfreie Sprache, Begründung, Pumping Lemma, Grammatiken, Wie sehen die für die Sprachen aus? welche Sprachen sind entscheidbar und warum? (L1 ist entscheidbar, da das Wort zwar immer länger werden kann, es aber nur ein endl. Alphabet und endl. viele Variablen gibt- oder so ähnlich)

NFAs (Unterschied zu DFA), Äquivalenz zu DFA (Beweisidee: Zustandsmengen des NFA werden als Zustände des DFA aufgefasst)

Untentscheidbarkeit? (Wußte nicht, was er wollte, hab mal von P und NP angefangen zu erzählen, dann hat er gemeint, okay, dann machen wir halt das.)

P=NP (mündl. Definition), NP Teilmenge von P?

Aussagenlogik: Unterschied != und !=

Resolution (Korrektheit, Vollständigkeit, widerlegungsvollständig → Formel aus KB logisch folgerbar, indem man zeigt, dass KB und nicht Formel unerfüllbar ist)

Suchen (Kurz Breiten-, Tiefen-, tiefenlimitierte Suche erläutert)

A\*: (kurz erklärt, Heuristik, Zulässigkeit)

Zwei zulässige Heuristiken mit:  $h(x) > h'(x)$ . Welche ist besser?

## Funktionentheorie

n-te Wurzel (wie sieht sie aus? wie viele gibt's?...)

Warum ist eine einmal differenzierbare Funktion im Komplexen unendlich oft diffbar?

Warum kann man jede holomorphe Funktion als Potenzreihe schreiben? (Wie kommt man drauf?, Spezielle Cauchy-Formel)

Was ist das Residuum? ...

## Zahlentheorie

Eigenschaften von  $(\mathbb{Z}_m, +, \cdot)$ ?

Warum existiert immer ein Inverses bei  $\mathbb{Z}_m^*$ ? („Zwei-zeiler“ Beweis)

Wieviele Restklassen mod m gibt es?  $\phi(m)$

Quadratisches Reziprozitätsgesetz (hingeschrieben, Beweisidee, dabei Satz von Gauss, bald abgebrochen)

Reziproke Primzahlsumme (hab mich am Anfang verschrieben und  $O(1/x \ln x)$  statt  $O(1/\ln x)$  hingeschrieben. hab es direkt verbessert, aber er wollte dann wissen, warum es erstes nicht sein kann, wovon ich keinen blassen Schimmer hatte. Wär dann besser als die Funktion selbst oder so), Beweisidee (wusste ich nicht, aber er wollte glaub ich nur das Stichwort Partielle Summation).

Geben sie eine transzendente Zahl an, bei der man die Transzendenz schnell einsieht (er wollte auf die Liouville- Zahlen raus, also Standardbeispiel aus dem Skript, Satz von Liouville angegeben, wie werden b/k gewählt- wusste ich nicht, steht aber im Skript).

Zu Herr Burgard: Super Prüfer. Prüfung war sehr angenehm, er hat hauptsächlich die wichtigen Themen aus Info II und III gefragt (wofür ich ihm sehr dankbar war) und wollte die Ideen dazu haben. Teilweise auch Beweisideen, aber keine ausgeführten komplizierten Beweise. Wichtiger sind die Zusammenhänge und die Laufzeiten von Programmen mit kurzer Erklärung. Dem zweiten Prüfling erging es ähnlich, auch er fand Herr Burgard sehr gut (bei ihm noch gefragt zu KI: Handeln unter Unsicherheit, bayessche Netze, value iteration).

Zu den Themengebieten: Aus Info II hab ich die Kapitel Fibonacci-Heaps und Bruder-Bäume/ B-Bäume beim wiederholen weggelassen, da Herr Burgard gemeint hat, dass sei ja „nicht so wichtig“. Aus KI durfte ich die Predikatenlogik weglassen, und mich nur auf die Aussagenlogik beschränken.

War sehr froh, dass ich Informatik als ein Teilgebiet gewählt habe!

Zu FT: Hab mich in Funktionentheorie auf die wichtigen Sätze und Beweise (ab Kapitel 4) konzentriert und hätte die auch super gekonnt, habe aber die Grundlagen vernachlässigt. Ein schwerer Fehler. Ich wusste auf sämtliche Fragen von Herrn Wolke keine Antworten, er versuchte mich noch drauf zu bringen, aber das half auch nix mehr. Und leider kamen wir dann auch nicht mehr zu den hinteren Kapiteln, also hab ich in Funktionentheorie total versagt. Tipp: wiederholt auch die wichtigsten Aussagen am Anfang, da er z.B das mit der Potenzreihe am gleichen Tag noch mindestens zwei weitere gefragt hat, die auch nicht damit gerechnet hatten.

Zu ZT: lief solala. hat klassische Fragen gefragt, nur leider ab und zu auch bei (meiner Meinung nach) Unwichtigem nachgehackt. Beweise hab ich kaum geführt, allenfalls Beweisideen.

Ergebnis: 3,0 (da es in Info sehr gut, ZT mittelmäßig und FT schlecht lief).

PS : Toi, toi, toi. Wünsch euch viel Glück. Nicht verzweifeln! Durch kommt man schon irgendwie. Was will man mehr?

# Prüfungsprotokoll

## Diplom HF

Jörg Müller  
SS 2007

(Advanced AI Techniques, Introduction to Mobile  
Robotics)

### 1 Advanced AI Techniques

#### 1.1 Reinforcement Learning (RL), Burgard

##### **Worin besteht RL?**

Bestimmung von optimalen Aktionen, z.B. in Markov Decision Processes  
(kurz erklärt)

##### **Unterschied zwischen RL, Supervised Learning (SL)**

Trainingsinput ist bei RL nur Reward, bei SL die optimale Aktion

##### **Dynamische Programmierung (DP): Wie wird die optimale Aktion bestimmt?**

Mit Policy-Iteration oder Value-Iteration, (zu) kurz erklärt, hat nochmal nachgefragt, Bellman Equation hingeschrieben.

##### **Monte Carlo (MC): Was ist der Unterschied zu DP?**

Es werden nicht alle möglichen Nachfolgezustände betrachtet, sondern anhand einer gesamten Sequenz gelernt.

##### **Vorteile von MC?**

Kann bei Interaktion mit Welt lernen; kein Modell notwendig; geringerer Einfluss einer Markov-Verletzung.

##### **Temporal Difference (TD)?**

Zu MC abgegrenzt: Zusätzlich zu Sampling Bootstrapping, Backup-Schritt aufgeschrieben.

##### **Was ist das exploration exploitation Dilemma?**

Man will möglichst oft die beste Aktion ausführen, um den Reward zu maximieren (exploit), muss aber auch explorieren, um den gesamten Zustands-Aktions-Raum abzudecken.

#### 1.2 Game Theory, Nebel

##### **Unterschied der Spieltheorie zu RL?**

Mehrere Agenten, ...

**Strategische Spiele: Definition**

$G = \langle N, (A_i), (U_i) \rangle$ , erklärt ...

**Was ist ein Nash Equilibrium (NE)?**

Gleichgewichtszustand (Aktionsprofil), bei dem es sich für keinen Spieler lohnt, davon abzuweichen, d.h. die Auszahlung bei Abweichung nur geringer werden kann.

**Gibt es immer ein NE?**

Nicht für pure Strategien. Nach Satz von Nash gibt es aber immer ein NE in gemischten Strategien.

**Was sind denn gemischte Strategien**

$N = \langle N, (\Delta(A_i)), (U_i) \rangle$ , erklärt ...

**Wie berechnet man NE in gemischten Strategien?**

Mit Support-Lemma (Support und Support-Lemma aufgeschrieben/erklärt): Nutzen der puren Strategien aus Support sind gleich. Lösung des Gleichungssystems für zwei Spieler und zwei Aktionen einfach.

**Kompetitive Spiele: Zentrales Konzept?**

Maximinizer definiert, Maximinizer Satz aufgeschrieben.

**Berechnung des NE?**

Aktionen  $A_1 = \{a_{11}, \dots, a_{1n}\}$ ,  $A_2 = \{a_{21}, \dots, a_{2m}\}$ .

Lineare Gleichungen:

- $\sum_i \alpha(a_{1i}) = 1$
- $\alpha(a_{1i}) > 0 \quad \forall i \in \{1, \dots, n\}$
- $\sum_i \alpha(a_{1i}) u_1(a_{1i}, a_{2j}) \geq u \quad \forall j \in \{1, \dots, m\}$

Lösung durch Maximierung von  $u$  mit linearer Programmierung.

**Definition Bayessche Spiele**

...

## 2 Introduction to Mobile Robotics, Burgard

**Wo wir gerade bei Bayes sind: Was sind Bayessche Filter?**

Recursive Bayes Filter Gleichung hingeschrieben.

**Wozu wird der Bayes Filter verwendet?**

Z.B. Roboter Lokalisierung, allgemein Abschätzung eines Zustands mit Observationen und Zustandsübergängen.

**Was für Implementierungen des Bayes Filters gibt es?**

Kalman Filter, Diskrete Filter, Particle Filter. Jeden in einem Satz beschrieben.

**Wie funktioniert ein Particle Filter?**

Repräsentation der Wahrscheinlichkeitsverteilung durch Particles. Algorithmus beschrieben:

- Resampling nur sehr kurz erklärt. Hat nachgefragt → Stochastic Universal Resampling genau erklärt
- Motion Modell: Für jeden Particle Sampling normalverteilter Fehler in Odometrie-Messung  $\langle \delta_{\text{rot}1}, \delta_{\text{trans}}, \delta_{\text{rot}2} \rangle$  und Integration der Odometrie mit diesem Fehler.
- Sensor Modell: Gewichtung der Particles =  $\frac{\text{target distribution}}{\text{proposal distribution}} = p(z_t | x_t)$ .

Dabei jeweils auf die entsprechenden Teile der Recursive Bayes Filter Gleichung hingewiesen.

### Wie bekommt man das Sensormodell?

Beam-basierte Sensormodell mit Annahmen beschrieben; Modellierung des Sensors und der Fehlerquellen, Parametrisierung der Fehlerquellen, Lernen der Fehlerparameter.

### Wie funktioniert ein Kalman Filter?

Modell ist:

$$\begin{aligned} x_t &= A_t x_{t-1} + B_t u_t + \varepsilon_t \\ z_t &= C_t x_t + \delta_t \end{aligned}$$

Komponenten der Gleichungen beschrieben. Mehr wollte er nicht hören.

### Nachteile des Kalman Filter?

nur für lineare Systeme mit normalverteiltem Rauschen.

### Was macht man im nichtlinearen Fall?

EKF: linearisierung durch Taylorentwicklung 1. Ordnung.

$$\begin{aligned} x_t &= g(x_{t-1}, u_t) \\ z_t &= h(x_t) \end{aligned}$$

Jakobi-Matrix definiert, Taylorentwicklung hingeschrieben.

$$\text{Übergang z.B. von } A_t \text{ zu } G_t = \left. \frac{\partial g}{\partial x_{t-1}} \right|_{\mu_{t-1}}.$$

### Malt eine eindimensionale Normalverteilung und einen Pfeil für die Odometrie-Messung auf. Wie sieht die Abschätzung des Kalman-Filters aus?

Erwartungswert wird um  $u$  verschoben. Varianz wird größer, also Kurve flacher.

### Malt jetzt noch eine Observation dazu. Wie geht die ein?

Observation hat geringere Varianz, also wird der neue Erwartungswert zwischen dem bisherigen und dem der Observation, aber näher an dem der Observation liegen. Die Varianz ist kleiner, als die der bisherigen Abschätzung und die der Observation.

### Warum wird die Varianz kleiner?

Regel für Multiplikation zweier Normalverteilungen aufgeschrieben:  $\Sigma' = \frac{1}{\Sigma_1^{-1} + \Sigma_2^{-1}}$ .

### Was macht man bei SLAM?

(Wollte erst Rao-Blackwellization erklären, aber er war noch beim Kalman Filter.)

Zustand enthält beim EKF-SLAM Roboterposition und  $(x, y)$ -Positionen

der Landmarks.

Kovarianzmatrix mit Kreuz-Korrelationen.

### **Wieso sieht die Kovarianzmatrix wie ein Schachbrett aus?**

$x$ -Positionen der Landmarks korrelieren untereinander, genauso  $y$ -Positionen.

Im Grenzfall sind die Landmarks voll korreliert. Da die Anordnung gewöhnlich  $l_{1x}, l_{1y}, l_{2x}, l_{2y}, \dots$  ist, entsteht ein Schachbrett.

### **Bemerkung**

Es war insgesamt eine entspannte Prüfungsatmosphäre

Prof. Burgard hat manchmal nachgefragt, weil ich wohl nicht genau das gesagt habe, was er hören wollte oder für wichtig hielt, aber sonst haben beide ruhig und präzise Fragen gestellt.

Viel Glück bei Euren Prüfungen

**Klausur**  
**Grundlagen der Künstlichen Intelligenz**  
**Sommersemester 2007**

Bearbeitung: 01.10.2007, 10:15-11:45 Uhr

Name, Vorname:

Matrikelnummer:

Studiengang:

- Informatik (Diplom)
- Informatik (Bachelor)
- Informatik (Master)
- ACS (Master)
- Sonstiges (bitte angeben): \_\_\_\_\_

- Schreiben Sie Ihren **vollen Namen**, Ihre **Matrikelnummer** und Ihren **Studiengang** auf dieses Deckblatt sowie Ihren **vollen Namen** auf alle weiteren Blätter.
- Überprüfen Sie die Klausur auf Vollständigkeit. Die Klausur besteht aus **8 Aufgaben**.
- Schreiben Sie Ihre Antworten auf den freien Platz unter der Aufgabenstellung. Nutzen Sie die folgende leere Seite, falls Sie weiteren Platz benötigen.
- Bücher, Notizen und ähnliches dürfen nicht verwendet werden. Dasselbe gilt für elektronische Geräte (Taschenrechner, Mobiltelefon, PDA, ...).
- Die Bearbeitungszeit beträgt **90 Minuten**.

	Erreichbare Punkte	Punkte
Aufgabe 1	12	
Aufgabe 2	12	
Aufgabe 3	14	
Aufgabe 4	12	
Aufgabe 5	12	
Aufgabe 6	12	
Aufgabe 7	14	
Aufgabe 8	12	
Summe (Klausur)	100	
Bonuspunkte (Übungsaufgaben)	14	
		Summe:
		Note:

Name: \_\_\_\_\_ Matr.-Nr.: \_\_\_\_\_

**Aufgabe 1 (8+4 Punkte)**

**HEURISTISCHE SUCHE**

Eine Heuristik  $h$  heißt **konsistent**, wenn gilt:

$$\begin{array}{ll} h(s) = 0 & \text{für alle Zielzustände } s \\ h(s) \leq c(s, s') + h(s') & \text{für alle Zustände } s \text{ mit Nachfolger } s' \end{array}$$

Dabei bezeichnet  $c(s, s')$  die Kosten beim Übergang von  $s$  zu  $s'$ .

Zeigen Sie:

- (a) Jede konsistente Heuristik ist zulässig (*admissible*).
- (b) Nicht jede zulässige Heuristik ist konsistent.

Name:

Matr.-Nr.:

---

**Aufgabe 2 (12 Punkte)**

**CONSTRAINT-SATISFACTION-PROBLEME**

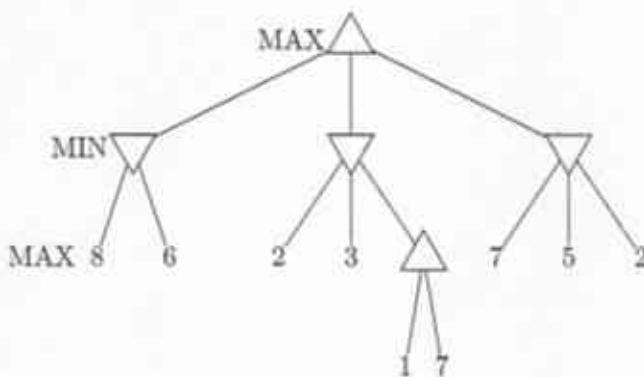
Betrachten Sie das Constraint-Satisfaction-Problem mit den Variablen  $v_1, v_2, v_3$  und  $v_4$ , jeweils mit Wertebereich  $\{1, 2, 3, 4\}$  und den folgenden Constraints:

- $C_1: v_3 < v_1$
- $C_2: v_2 < v_3$
- $C_3: v_4 = v_2 + 2$
- $C_4: v_1 \neq v_4$

Stellen Sie für dieses Problem Kantenkonsistenz (*arc consistency*) her. Geben Sie dabei in jedem Schritt an, welche Werte Sie aus welcher Domäne streichen, und aufgrund welches Constraints dies geschieht. Geben Sie eine Lösung des Problems an.

**Aufgabe 3 (8+2+4 Punkte)**

Betrachten Sie den folgenden Spielbaum für ein 2-Personen-Spiel:



- Wenden Sie den  $\alpha$ - $\beta$ -Pruning-Algorithmus (von links nach rechts) auf den gegebenen Spielbaum an. Streichen Sie nicht-besuchte Teilebäume aus dem Originalbaum und tragen Sie die  $\alpha$ - und  $\beta$ -Werte an den besuchten Knoten ein.
- Welchen Zug würde MAX am Wurzelknoten wählen?
- Ordnen Sie den Baum so um, dass der  $\alpha$ - $\beta$ -Pruning-Algorithmus jeden Knoten besuchen muss.

Name: \_\_\_\_\_ Matr.-Nr.: \_\_\_\_\_

**Aufgabe 4 (4+8 Punkte)**

AUSSAGENLOGIK

Sei  $\Phi = \{\neg C \vee E \vee D, E \vee \neg D, \neg B \vee \neg E, \neg A \vee B, \neg B \vee \neg F \vee C, \neg A \vee F\}$  und  $\phi = \neg A$ .

- (a) Geben Sie eine aussagenlogische Belegung  $v$  mit  $v \models \Phi$  an.
- (b) Zeigen Sie mit Hilfe der Resolutionsmethode, dass  $\Phi \models \phi$ .

Name: \_\_\_\_\_ Matr.-Nr.: \_\_\_\_\_

**Aufgabe 5 (8+4 Punkte)**

**PRÄDIKATENLOGIK**

(a) Formalisieren Sie die folgenden beiden Aussagen in der Prädikatenlogik:

- Jeder Barbier rasiert alle Personen, die sich nicht selbst rasieren.
- Kein Barbier rasiert Personen, die sich selbst rasieren.

(b) Geben Sie die Klauselform für die Formeln an.

Name:

Matr.-Nr.:

**Aufgabe 6 (7+4+1 Punkte)**

**ENTSCHEIDUNGSBÄUME**

Ein Freund hat im letzten Monat acht Reisen durchgeführt. Sie haben die folgenden Reisedaten notiert und wollen daraus lernen, wann er das Auto als Verkehrsmittel wählt und wann den Zug.

Nr.	Entfernung	Gepäck	Anzahl Mitfahrer	Verkehrsmittel
1	0-20 km	viel	2	Auto
2	21-100 km	wenig	0	Zug
3	101-500 km	viel	1	Zug
4	0-20 km	viel	1	Auto
5	21-100 km	wenig	2	Auto
6	101-500 km	viel	2	Zug
7	101-500 km	wenig	0	Zug
8	0-20 km	viel	0	Auto

- Geben Sie einen Entscheidungsbaum an, der für die gegebenen Beispiele anhand der Attribute *Entfernung*, *Gepäck* und *Mitfahrer* klassifiziert, ob das Auto als Verkehrsmittel gewählt wird oder nicht.
- Leiten Sie aus Ihrem Entscheidungsbaum eine logische Formel ab, die genau dann wahr ist, wenn Ihr Freund den Zug wählt.
- Formulieren Sie die Wahl des Verkehrsmittels in einem möglichst kurzen, natürlichsprachlichen Satz.

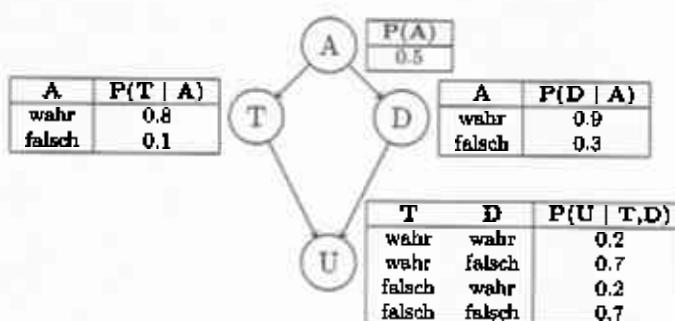
Name:

Matr.-Nr.:

### Aufgabe 7 (12+2 Punkte)

BAYESSCHE NETZE

Betrachten Sie das folgende Bayes'sche Netz:



Hierbei bezeichne

- U das Ereignis, dass ein Autofahrer im laufenden Jahr einen Unfall verursacht
- T gibt an ob er ein teures Fahrzeug besitzt
- D gibt an ob er eher defensiv fährt und
- A ist genau dann wahr, wenn der Fahrer älter als 50 Jahre ist.

Mittels der Formeln für Bayes'sche Netze können Sie berechnen, dass  $P(U) = 0.4$  und  $P(U|T) = 0.283$  ergibt. Es scheint also so zu sein, dass das Fahren eines teuren Autos die Unfallwahrscheinlichkeit reduziert.

- (a) Berechnen Sie  $P(U|A)$  und  $P(U|A, T)$  oder alternativ  $P(U|\neg A)$  und  $P(U|\neg A, T)$ .
- (b) Was lässt sich aus diesen Ergebnissen über den Zusammenhang zwischen teuren Autos und dem Unfallrisiko aussagen?

**Anmerkung:** Dieses Szenario ist frei erfunden und beruht nicht auf tatsächlichen statistischen Daten.

**Tipp:** Verwenden Sie Brüche statt der Dezimaldarstellung der Werte um die Rechnungen zu vereinfachen.

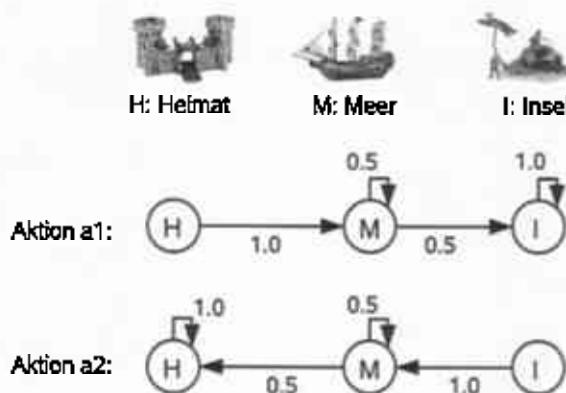
Name:

Matr.-Nr.:

**Aufgabe 8 (3+3+4+2 Punkte)**

VALUE ITERATION

Betrachten Sie das folgende Markov-Entscheidungsproblem eines Piraten, der einen Goldschatz von einer Schatzinsel nach Hause transportieren möchte.



Der Pirat startet im Zustand **H** ohne Gold an Bord. Das offene Meer **M** kann er nur in 50% der Fälle in einem Schritt verlassen. Sobald er die Insel **I** erreicht, nimmt er automatisch das Gold auf. Erreicht er die Heimat **H** mit Gold (und nur dann) erhält er einen Reward von 100 und sein Abenteuer endet. Nehmen Sie  $\gamma = 0.5$  als Discountfaktor an.

- Geben Sie eine geeignete Menge von Zuständen an, durch die dieses Problem formalisiert werden kann.
- Wie viele Dimensionen hat Ihr Zustandsraum?
- Führen Sie die ersten drei Schritte des Value-Iteration Algorithmus durch, um eine erste Abschätzung für den Nutzen der einzelnen Zustände zu bekommen. Als Anfangsverteilung besitze das Ziel "Erreichen der Heimat mit Gold" den Nutzen 100 und alle anderen Zustände den Nutzen 0.
- Geben Sie die optimale Policy für dieses Problem an.

**Tipp:** Verwenden Sie Brüche statt der Dezimaldarstellung der Werte um die Rechnungen zu vereinfachen.

# Prüfungsprotokoll

## Diplomprüfung

### Vertiefung: Künstliche Intelligenz

in den Fächern:

**CSMR** (Computer-Supported Modeling and Reasoning)

**Advanced AI Techniques**

**Spieltheorie** (WS 07/08)

26.April 2008

Prüfer: Prof. Dr. Nebel, Prof. Dr. Burgard

Note: 1,0

Kerstin Haring

#### Bemerkung:

*Die Reihenfolge der Prüfungen war frei wählbar. Ich habe mit Spieltheorie begonnen. Die Fragen sind fett dargestellt, die Antworten normal und meine Kommentare im Nachhinein sind kursiv. Da ich dieses Protokoll im Nachhinein verfasst habe, kann ich mich nicht 100% an die Fragestellungen erinnern, bin mir aber sicher immer die relevanten Themengebiete abzudecken.*

#### CSMR:

Schein eingereicht. Note: 1,3

Wer einen Schein einreicht sollte ihn auf jeden Fall dabei haben!

#### Spieltheorie:

##### **Was ist denn Spieltheorie?**

Spieltheorie ist die Analyse strategischer Entscheidungssituationen, in denen mehrere Spieler miteinander interagieren.

Dabei ist das Resultat eines Spiels von den Entscheidungen der Mitspieler abhängig. Damit stellt sich die Frage nach dem Ergebnis, das sich ergibt, falls alle Spieler "rational" handeln, d.h. ihren (erwarteten) Nutzen maximieren, wobei sie davon ausgehen, dass ihre Mitspieler ebenso rational handeln. Genau wegen dieser Rationalität ist die Spieltheorie für die Informatik interessant.

##### **Kennen Sie denn ein Spiel, bei dem ein Spieler Interesse hat, nicht rational zu spielen?**

Naja, wenn z.B. ein Spieler dem anderen das Schlechteste wünscht und dafür selber Einbußen in seinem Nutzen in Kauf nimmt.

*Haben da noch ein bisschen diskutiert, ich bekam ein Hinweis auf ein weiteres Spiel, bei dem durch Unwissen irrational gespielt wurde, und mir fiel dann noch das 2/3 average Spiel ein, das wir sogar in der VL gespielt haben, das habe ich dann noch erklärt.*

##### **Was ist denn ein strategisches Spiel? Woraus besteht es?**

- $G = \langle N, (A_i), (u_i) \rangle$

- $N$  ist endliche Spielermenge
- nicht-leere Menge  $A_i$  von Aktionen/Strategien für jeden Spieler  $i$
- Auszahlungsfunktion  $u_i: A \rightarrow \mathbb{R}$  für jeden Spieler  $i$
- $A = \prod_{i \in N} A_i$
- $G$  heißt endlich, wenn  $A$  endlich ist.
- Beide Spieler legen vor Spielbeginn ihre Strategie fest, Ergebnis ist Strategiekombination.

### Was sind dominate Strategien? Definition?

- Spiel  $G = \langle N, (A_i), (u_i) \rangle$
- Eine Aktion  $a_j^x \in A_j$  heißt strikt dominiert, falls es
- eine Aktion  $a_j' \in A_j$  gibt, so dass
- $u_j(a_{-j}, a_j') > u_j(a_{-j}, a_j^x)$  für alle Profile  $a \in A$  gilt
- Es ist nicht rational strikt dominante Strategien zu spielen, weil man dann garantiert weniger bekommt als man bekommen könnte. Es ist auch nicht rational, strikt dominante Strategien nicht zu spielen.

### Sind die Ergebnisse eindeutig?

Ja, das Ergebnis bei iterativer Elimination strikt dominanter Strategien ist eindeutig und damit unabhängig von der Reihenfolge der Elimination.

### Was für eine Dominanz gibt es noch? Sind hier die Ergebnisse eindeutig?

*Schwache Dominanz, ebenfalls mit vollständiger Definition erklärt und gesagt, dass die schwache Dominanz nicht eindeutig sein muss, weil das Ergebnis von der Eliminationsreihenfolge abhängt.*

### Was ist ein Nash-Gleichgewicht? Definition?

Ein NG ist eine Strategiekombination, in der kein Spieler durch Abweichung einen Vorteil erlangen kann.

(1) Ein Profil  $a^x \in A$  ist ein NG in einem strategischen Spiel  $G = \langle N, (A_i), (u_i) \rangle$  gdw.

$$u_i(a^x) = u_i(a_{-i}^x, a_i^x) \geq u_i(a_{-i}^x, a_i) \quad f.a. \quad a_i \in A_i \quad f.a. \quad i \in N$$

### Welche Definition hatten wir noch? Wofür braucht man die denn?

(2) Ein Profil  $a^x \in A$  ist ein NG in einem strategischen Spiel  $G = \langle N, (A_i), (u_i) \rangle$  gdw.

$$B_i(a_i) = \{a_i \in A_i \mid u_i(a_{-i}, a_i) \geq u_i(a_{-i}, a_i')\} \quad f.a. \quad a_i \in A_i \quad f.a. \quad i \in N \quad \text{und}$$

$$a_i^x \in B_i(a_{-i}^x) \quad f.a. \quad i \in N$$

*hier hat Prof. Nebel gemeint, ich soll das mal wieder weg machen, das wäre keine Definition.  
Erinnerte mich aber an das Skript und habe dann natürlich erklärt, dass man das auf dem Weg zu der dritten Definition braucht. Das hat er dann auch eingesehen.*

(3) Ein Profil  $a^x \in A$  ist ein NG gdw.

$$B(a^x) = \prod_{i \in N} B_i(a_{-i}^x) \quad \text{und} \quad a^x \in B(a^x) \quad (\text{wird für Beweis des Satzes von Nash verwendet})$$

### Wie lautet der Satz von Nash?

Jedes endliche strategische Spiel hat ein NG in seiner gemischten Erweiterung.

### Was sind gemischte Erweiterungen?

Die gemischte Erweiterung eines Spiels  $G = \langle N, (A_i), (u_i) \rangle$  ist  $G = \langle N, (\Delta A_i), (U_i) \rangle$

- $\Delta A_i$  ist die Menge der Wahrscheinlichkeitsverteilungen über die Menge A
- $U_i: \prod_{j \in N} \Delta(A_j) \rightarrow \mathbb{R}$  jedem Profil  $\alpha$  den erwarteten Nutzen von Spieler i unter der von  $\alpha$  induzierten Wahrscheinlichkeitsverteilung zuordnet.

*Hier irgendwo kamen wir auf das Elfmeterspiel, keine Ahnung wie genau, es kam auf jeden Fall zur Sprache.*

### Wie berechnet man da NGs?

Mit dem Support Lemma.

### Support-Lemma Definition? Was folgt daraus?

*Habe mich tatsächlich an einem lockeren Spruch versucht:*

Also, es sagt, dass wir zuerst den support definieren müssen:

- Sei  $\alpha_i$  eine gemischte Strategie. Der support von  $\alpha_i$  ist die Menge
- $supp(\alpha_i) = \{a_i \in A_i \mid \alpha_i(a_i) > 0\}$
- Sei  $G = \langle N, (A_i), (u_i) \rangle$  strategisches Spiel
- Dann ist  $\alpha^x \in \prod_{i \in N} \Delta(A_i)$  ein NG in gemischten Strategien gdw. alle reinen  $a_i \in supp(\alpha_i)$  für jeden Spieler i eine beste Antwort auf  $\alpha_{-i}^x$  ist.

Wenn die anderen Spieler ihre gemischte Strategie beibehalten, ist es also für jeden einzelnen Spieler egal, ob er seine gemischte Strategie oder eine Einzelaktion daraus spielt.

Aktionen, die in dem support einer gemischten Strategie in einem NG sind, sind immer beste Antworten auf das NG Profil und haben damit auch den gleichen Nutzen!

### Wie wird das dann berechnet?

Benutze stattdessen Instanzen des Linear Complementarity Problem.

*Ich habe gehofft, dass er nicht weiter nachfragt*

### Und wie geht das dann konkret?

*D'oh! Schwachstelle erwischt.*

*Habe was von linearen Ungleichungen erwähnt, bin kurz in die Lösung von NSS mit LP abgedriftet und es dann bemerkt und zugegeben, dass ich hier nicht weiterkomme.*

### Sollen wir lieber an einer anderen Stelle weitermachen?

*Einfachste Frage der Prüfung.*

Ja gerne

### Ist ein NG bei allen Spielen aussagekräftig?

Nein, nicht bei extensiven Spielen:

*(habe hier was von übersetzen in strategische Form und leeren Drohungen erwähnt)*

$$\Gamma = \langle N, H, P, (u_i) \rangle$$

- endliche nicht-leere Menge N von Spielern
- Menge H von Sequenzen (Historien), Menge Z der terminalen Historien
- Spielerfunktion  $P: H \setminus Z \rightarrow N$
- Auszahlungsfunktion  $u_i: Z \rightarrow \mathbb{R}$  f.a.  $i \in N$

Extensive Spiele können unendlich viele Verzweigungen haben und die Pfade können unendlich lang sein.

- Wir haben deshalb auch zwei Formen der Endlichkeit, und zwar heißen Spiele endlich, wenn ihre Menge der Historien  $H$  endlich ist, und sie haben einen endlichen Horizont, wenn die Länge der Historien  $h$  nach oben beschränkt ist.

### Was sind denn leere Drohungen?

Wenn eine Strategie ein Verhalten festlegt, dass im Falle des Erreichens eines bestimmten Knotens nicht rational wäre.

Beispiel 56 Skript:

Was damit gemeint ist, erkennt man besonders gut am Nash Gleichgewicht (B,L) .

- 1 würde natürlich am liebsten A wählen, in der Hoffnung, 2 werde dann R spielen und 1 würde eine Auszahlung von 2 erhalten.
- 2 droht aber damit, auf A mit L zu antworten, so dass es für 1 günstiger ist, B zu wählen und wenigstens eine Auszahlung von 1 zu erhalten.
- Würde 1 aber ungeachtet der Drohung doch A wählen, wäre es für 2 besser statt die Drohung auszuführen und L zu spielen, was ihm eine Auszahlung von 0 einbrächte, lieber entgegen seiner Ankündigung R zu wählen und eine Auszahlung von 1 zu erhalten.
- Ist 1 von der Rationalität von 2 überzeugt, wird er der Drohung also nicht glauben.

### Wie sind TPG definiert?

Es stellt eine Verfeinerung des NGs dar, d.h.: jedes teilspielperfekte Gleichgewicht ist auch ein NG. Ein NG ist teilspielperfekt, wenn es ein NG in jedem Teilspiel von  $G$  induziert.

### Wie heißt der Satz?

Kuhn: Jedes endliche extensive Spiel mit perfekter Information hat ein TPG

### Wieso nur bei endlichen Spielen?

*Habe die 2 Gegenbeispiele aus dem Skript aufgemalt und erklärt*

### Was hat das ganze mit Minimax und NSS zu tun?

- Das ist die Erweiterung von NSS und dem Minimax-Algorithmus aus der KI -VL
- Rückwärts Induktion ist Verallgemeinerung des Minimax-Algorithmus

Ähnlichkeiten Kuhn zu Minimax:

Lösung wird gefunden, indem der Baum von unten nach oben abgesucht wird und in jedem Knoten der optimale Spielzug gewählt wird und die Werte nach oben weiter-propagiert werden.

Unterschiede: Im Fall der Rückwärts Induktion sind mehr als 2 Spieler möglich und der Nutzen (payoff) ist nicht nur eine einzelne Zahl sondern ein ganzes Nutzenprofil.

### Welche Erweiterungen hatten wir da? Was passiert mit dem Satz von Kuhn?

Zufallszüge: Kuhn gilt, man muss mit Erwartungswert rechnen

*Prof. Nebel wollte wissen, woher der Zufall kommt, nach Klärstellen der Frage habe ich was von exogen gesagt, also von außen und er war zufrieden*

Simultane Züge: Kuhn gilt nicht mehr, Gegenbeispiel: Matching Pennies

*Prof. Nebel sieht Prof. Burgard an und meint: Ach ja, da hatten wir was neues in der VL, das muss sie dir nun erklären:*

## **Was ist denn Mechanismus Design?**

- Teilbereich der Spieltheorie, bei dem es um die Synthese von Spielen geht.
- Ziel ist es, eine soziale Entscheidung als Lösung eines Spiels zu implementieren, also Spiele so zu definieren, dass deren Lösungen (Gleichgewichte) gerade die gewünschten Ausgänge sind.

Beispiele für soziale Entscheidungen sind Wahlen, Auktionen, Festlegungen von Policies.

- Nicht immer werden die Präferenzen der beteiligten Personen ehrlich angegeben
- Mechanismusdesign implementiert die Bestimmung der sozialen Entscheidungen in einer strategischen Umgebung, in der die Präferenzen der Teilnehmer nicht öffentlich sind.

## **Was ist eine soziale Entscheidung?**

Aggregiert alle Präferenzen der Wähler in einzelne soziale Entscheidung:

- Eine soziale Wohlfahrtsfunktion aggregiert alle Präferenzen von allen Wählern in eine totale soziale Ordnung der Kandidaten:  $F : L^n \rightarrow L$
- Eine soziale Entscheidungsfunktion aggregiert alle Präferenzen von allen Wählern in eine einzelne soziale Entscheidung für einen Kandidaten:  $F : L^n \rightarrow A$
- $L$  ist die Menge der linearen Ordnungen auf  $A$ , sprich alle Elemente können paarweise verglichen werden

## **Wie lautet der Satz von Arrow?**

Jede soziale Wohlfahrtsfunktion über einer Menge von mehr als zwei Alternativen, die totale Einstimmigkeit und UIA erfüllt, ist diktatorisch.

## **Was ist ein Diktator?**

Die soziale Präferenz ist immer die des Diktators, ungeachtet der Präferenzen aller anderen Wähler

## **Was heißt strategisch manipulierbar?**

Eine soziale Entscheidungsfunktion ist manipulierbar, wenn ein Wähler  $i$ , der  $b$  vor  $a$  präferiert,  $b$  erzwingen kann, wenn er statt seiner wahren Präferenz  $<_i$  eine davon verschiedene Präferenz  $<'_i$  angibt.

$f$  heißt anreizkompatibel, wenn  $f$  nicht manipulierbar ist.

## **Was besagt das Erweiterungslemma?**

Falls  $f$  eine anreizkompatible, surjektive und nicht-diktatorische soziale Entscheidungsfunktion ist, so ist ihre Erweiterung  $F$  eine soziale Wohlfahrtsfunktion, die Einstimmigkeit, Unabhängigkeit von irrelevanten Alternativen und nicht-diktatorische Entscheidung erfüllt.

Satz von Gibbard-Satterthwaite ist das Analogon zum Satz von Arrow für soziale Entscheidungsfunktionen

## **Was folgt aus dem Satz von Gibbart & Satterthwaite?**

Er scheint alle Hoffnungen zu zerstören, man könnte anreizkompatible soziale Entscheidungsfunktionen entwerfen.

## **Wie konnten wir das umgehen?**

Man kann das Modell ändern. Es gibt zwei gängige Möglichkeiten, die Hinzunahme von Geld und die Einschränkung der zulässigen Präferenzrelation

*Bei Prof. Nebel sollte man auf jeden Fall Ahnung haben wovon man spricht und/oder die Definitionen sauber lernen, er fragt da gerne mal nach (nicht immer, aber wenn es ihm gerade in*

*den Sinn kommt, z.B. ob die Definition bei strikter Dominanz für alle Strategien gilt). Es ist in jeder Hinsicht sehr freundlich und geduldig, stellt seine Fragen ruhig und präzise und erklärt sie bei Unverständnis gerne noch einmal.*

### **Advanced AI Techniques:**

#### **Was ist denn der Unterschied von Reinforcement Learning zu Spieltheorie?**

Wir haben nur einen Agenten

#### **Was ist denn RL?**

Man beobachtet, wie der Agent durch Erfolg und Misserfolg mit Belohnung durch die Interaktion mit der Umgebung lernen kann.

#### **Wieso ist es so wichtig? Was hat man denn hier für Annahmen, die man in der Spieltheorie nicht hat? Wieso ist RL besser als Minimax?**

In der Spieltheorie nimmt man an, dass der Gegner optimal spielt. Im RL kann man lernen die Schwächen eines suboptimalen Gegners auch auszunutzen.

#### **Was suchen wir im RL? Was ist gegeben?**

Man versucht anhand der beobachteten Belohnungen die optimale Strategie zu finden. Die optimale Strategie maximiert den erwarteten Gesamtgewinn.

#### **Was ist Bellmann? (also so hat er nicht gefragt, aber an irgendeiner Stelle habe ich Bellman erklärt)**

Wertefunktionen im RL erfüllen bestimmte rekursive Bedingungen. Die Gleichung beschreibt eine Konsistenz zwischen dem Zustand und allen möglichen Nachfolgezuständen und einer Strategie.

#### **Was ist DP?**

Man benutzt die Wertefunktion um nach guten Strategien zu suchen.

*Dann habe ich lang und breit Policy evaluation und Policy Improvement erklärt, wie beides funktioniert und daran dann die Policy iteration mit diesen 2 Schritten indem man Bellman als Zuweisung benutzt also einen vollständigen policy evaluation Schritt hingeschrieben, dann denn policy improvement Schritt und erklärt, dass das endet wenn die Policy stabil ist. Auch dass der Policy evaluation -Schritt mehrere sweeps durch den Zustandsraum braucht und das dann bei der value iteration abgeschnitten wird, weil die optimale Policy oft vor der Konvergenz der Wertefunktion erreicht ist.*

#### **Was ist der Unterschied zu MC?**

Bei MC: Kein bootstrapping, also Verletzung der Markov-Eigenschaft nicht so schlimm, aktualisiert Werte erst am Ende der Episode, also nur für episodische tasks geeignet, braucht kein vollständiges Umgebungsmodell, lernt also direkt aus der Erfahrung, kann man einfach und effizient auf Untermengen anwenden

#### **Was ist GPI?**

Generelle Idee zweier interagierender Prozesse, die sich der Policy und der Wertefunktion annähern. Der eine Prozess nimmt die Strategie als gegeben und ändert die Wertefunktion, dass sie zur policy passt, der andere Prozess nimmt die Wertefunktion als gegeben und verbessert die Strategie damit. Die Prozesse konkurrieren in dem Sinne, dass sie immer die Basis des anderen ändern, aber insgesamt arbeiten sie zusammen an einer gemeinsamen Lösung. Wenn sie sich gegenseitig nicht mehr ändern, sind sie optimal.

(Anm. Die korrekte Übersetzung von Policy lautet Taktik, ich habe aber oft Strategie oder Policy gesagt, also ein gewisses "switchen" zwischen den Sprachen stellt kein Problem dar)

### Eine letzte Frage: Was ist das exploration-exploitation Dilemma?

Man kann nicht immer ausbeuten, man kann nicht immer explorieren, aber beides muss getan werden um ein optimales Ergebnis zu erzielen. Das Dilemma ist nun, dass man beides nicht mit einem Schritt durchführen kann.

(Glaube diese beiden "wirklich" letzten Fragen waren die finale Entscheidung für die bessere Note)

**In der realen Welt haben wir ja nicht immer solche Eigenschaften wie im RL gefordert ist?**  
Nein, da kennt der Agent den Zustand in dem er sich befindet nicht. Das nennt sich dann POMDP.

### Und löst man das nun anders als vorher, oder wie macht man das?

Nein, nicht arg viel anders, denn man kann POMDP für einen Zustandsraum auf die Lösung eines MDPs reduzieren, man verwendet dann den zugehörigen Glaubenzustand. Allerdings wurden POMDPs bisher nur auf kleinen Zustandsräumen angewendet, denn die Anzahl der linearen Bedingungen wächst mit jeder Iteration exponentiell.

(Das war Gott sei Dank alles, was er wissen wollte, den mehr hätte ich nicht sagen können)

### Und wie heißt dann die Gleichung?

(geraten, und zwar gut ;-))

Das ist die Recursive Bayes Filtering Gleichung.

Gleichung aufgeschrieben und kurz erklärt (Sensor-Modell, Aktionsmodell, dass man Zustand x in dynamischen System schätzen möchte)

Bei Prof. Burgard sollte man sich auch auf jeden Fall ein bisschen über den Bezug von RL und Spieltheorie kümmern (also zumindest bei dieser Kombination oder bei der Prüfung in AAIT) und die ganzen Prinzipien ebenfalls gut verstanden haben sowie die Definitionen und Formeln sicher können. Auch er stellt seine Fragen ruhig, aber manchmal war mir nicht ganz klar worauf er hinaus wollte, aber er gibt gerne eine kleine Hilfestellung wenn man nicht direkt antworten kann. Für die 1,0 reicht es nicht, nur den RL-Teil zu lernen, POMDP und Recursive Bayes Filtering sollte man sich dazu zumindest mal kurz angesehen haben. Was auf jeden Fall sitzen muss ist Bellmann, alle Backup-Schritte von den einzelnen Verfahren und alle anderen Grundbegriffe wie GPI oder Markov-Eigenschaft.

Zur Vorbereitung empfele ich den regelmäßigen Besuch der Vorlesung und das gründliche Durcharbeiten der Vorlesungs-Aufzeichnungen und des Skripts in Spieltheorie. Wichtige Definitionen und Formeln muss man auf jeden Fall sicher auswendig können. Die Übungsblätter habe ich mir zur Vorbereitung auf diese Prüfung nicht ein einziges Mal angesehen, dafür aber andere Prüfungsprotokolle und mir auch selber alle möglichen Fragen ausgedacht, die dran kommen könnten und sie vor allem auch beantwortet. Ohne einen gewissen Lernaufwand ist bei dieser Kombination eine gute Note sicher nicht drin. Außerdem legt schon die Länge des Protokolls dar, dass ich doch ziemlich viel zu sagen hatte, was den Vorteil bietet, dass dann weniger Fragen kommen. Mit genauerem Nachfragen zu dem Erzählten muss man auf jeden Fall immer rechnen.

Ach ja, es wurde mir gesagt, dass ich Mut zur Lücke bewiesen habe, weil ich zu den LCP-Ungleichungen überhaupt nichts vernünftiges zustande gebracht habe, aber habe dennoch eine 1,0 bekommen, also während der Prüfung von solchen Kleinigkeiten nicht aus der Ruhe bringen lassen.

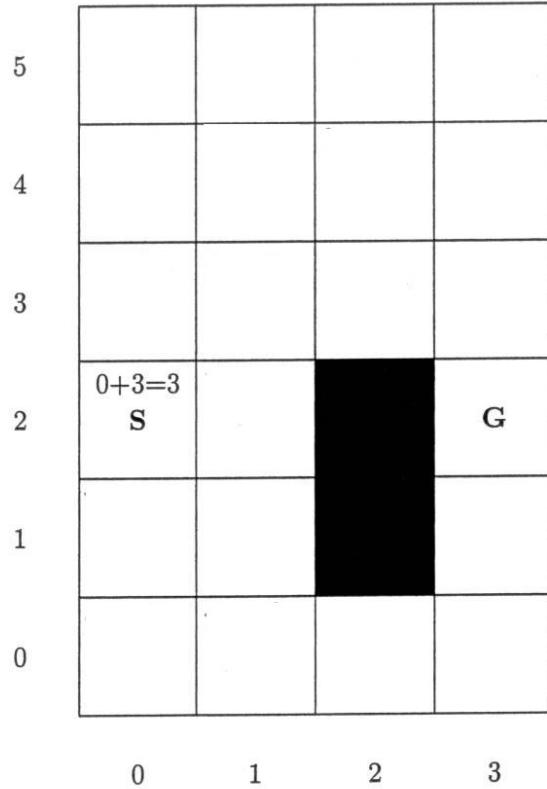
Ich wünsche allen viel Glück bei Ihrer Prüfung!

Name: \_\_\_\_\_

**Question 1 (8+4 marks)**

**HEURISTIC SEARCH / HEURISTISCHE SUCHE**

A robot has to move from position  $S = (0, 2)$  to position  $G = (3, 2)$ . Only horizontal and vertical movements (no diagonal ones) by one position per step are allowed. Positions marked black are inaccessible.  $A^*$  is employed to construct a search tree. Assuming unit cost, the heuristic is the Manhattan distance from the current position to  $G$  (ignoring the obstacles).



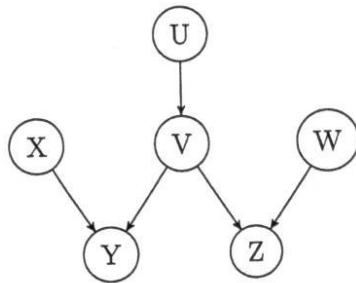
- (a) Compute the  $f$  values of each node in the search tree, and mark the corresponding values in form of " $g + h = f$ " in the grids (exemplary done for cell  $(0,2)$ ). The cells which are not visited ( $A^*$  would not calculate the  $f$  value) in the search should be left empty.
- (b) Correct or wrong? For each of the following statements, decide whether or not it is correct. For every correct answer you get 1 point, for every wrong answer you get -1 point. Thus, if unsure, don't give an answer. In total, the lowest possible score is 0.
- 1) In the above search problem, the Manhattan distance is an admissible heuristic.
  - 2) Uniform cost search and  $A^*$  (assuming an admissible heuristic) both are complete and optimal.
  - 3) The space complexity of breadth-first search is lower than that of depth-first search.
  - 4) A search problem consists of five parts: The state space, initial situation, actions, goal test and path costs.

Name: \_\_\_\_\_

**Question 2 (3+2+2+5 marks)**

**CONDITIONAL INDEPENDENCE**

- (a) Given the unconditional probabilities  $P(A \wedge B) = 0.3$ ,  $P(\neg A) = 0.6$ , and  $P(B) = 0.5$ , are  $A$  and  $B$  independent? Justify your answer.
- (b) Given the joint probability  $P(C \wedge D) = 0.21$  and the unconditional probability  $P(D) = 0.5$  calculate the conditional probability  $P(C | D)$ .
- (c) Rewrite the joint probability distribution  $P(U, V, W, X, Y, Z)$  using the conditional dependencies expressed by the following network:



- (d) Determine, which of the following conditional independence statements follow from the structure of the Bayesian network ( $Ind(A, B | C)$  denotes that  $A$  is conditionally independent of  $B$  given  $C$ ).

statement	correct	not correct
$Ind(V, X   U)$		
$Ind(V, Z   U)$		
$Ind(U, Z   V)$		
$Ind(Y, Z   X, V)$		
$Ind(U, V   Y, Z)$		

Name: \_\_\_\_\_

**Question 3 (5+5 marks)**

FIRST-ORDER LOGIC

- (a) Consider the following set of formulas:

$$\Theta = \left\{ \begin{array}{l} \forall x \forall y (f(x) = f(y) \Rightarrow x = y) \\ \forall x \forall y (P(x, y) \Rightarrow \neg(x = y)) \\ \forall x P(f(x), x) \end{array} \right\}$$

Specify an satisfying interpretation  $\mathcal{I} = \langle \mathcal{D}, \cdot^{\mathcal{I}} \rangle$  with domain  $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$ .

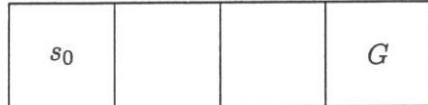
- (b) Transform the following formula to Skolem normal form:

$$\forall x \forall y (P(x, y) \Rightarrow \exists z (P(x, z) \wedge P(z, y)))$$

Name: \_\_\_\_\_

**Question 4 (4+6+4 marks)**

MDP



Consider the markov decision process (MDP) with the four states  $s \in S$  that correspond to the cells of the grid world depicted above. In each of the cells the agent can select one of the actions West or East. The action West moves the agent one cell to the left with a probability of 0.8. With a probability of 0.2 the action West fails, leaving the agent in the same cell. The action East has a probability of 0.8 for moving the agent one cell to the right, and 0.2 for failing. If the action West is used in the leftmost cell (state  $s_0$ ), the agent stays in place with probability 1. Furthermore, once it's reached, the goal state  $G$  cannot be left by the agent. The immediate reward is  $R(s) = -1$  for state  $s \in S \setminus \{G\}$  other than the terminal state  $G$ , and  $R(G) = 0$  for a step within the terminal state  $G$ . Hint: By definition, the utility  $U$  of the terminal state  $G$  always is  $U(G) = 0$ . Furthermore, the rewards in this problem are additive!

- (a) *Value Iteration Algorithm.* How could a non-optimal utility function  $U^t : S \mapsto \mathbb{R}$  be improved using the **value iteration** algorithm? Either write down the exact formal definition for calculating the updated utility  $U^{t+1}(s)$  of a state  $s$  or explain (exactly!) the iterative update rule using natural language.
- (b) *Applying Value Iteration.* Consider the following sub-optimal utility function  $U^t$ . Do one iteration of the Value Iteration algorithm updating all states  $s \in S$  and filling the improved utilities  $U^{t+1}(s)$  into the right figure. Write down the necessary calculations for each of the three states to update.

$U^t$				$U^{t+1}$			
-5	-2	-10	0				0

- (c) *Policy Improvement.* Consider the utilities  $U^{\pi^t}(s)$  under a policy  $\pi^t$  as depicted in the upper-left grid of the figure below. Do a *Policy Improvement* step based on this utility function and fill in the resulting policy  $\pi^{t+1}$  in the right grid of the upper row. Finally fill in the optimal policy into the remaining grid in the lower row (on the next page).

$U^{\pi^t}$				$\pi^{t+1}$			
$-\infty$	+2	-10	0				0

$\pi^*$			
			0

# PRÜFUNGSPROTOKOLL SPIELTHEORIE & KI

Diplomhauptprüfung Nebenfach Informatik (Hauptfach Mathematik)

Vorlesungen **Künstliche Intelligenz** (gehört bei Prof. Nebel) und **Spieltheorie** (gehört bei Prof. Burgard).

Prüfer: **Prof. Dr. Nebel**, Note: 1,0 16.12.2011, Peterhans Hendel

**Bemerkung:** Die Reihenfolge der Prüfungen war frei wählbar, ich habe mit Spieltheorie begonnen. Das Prüfungsprotokoll wurde nachträglich aus dem Gedächtnis verfasst und ich kann mich nicht mehr an jede Zwischenfrage erinnern aber die grobe Struktur müsste stimmen. Ob meine Antworten wirklich richtig waren, habe ich nicht überprüft, aber wenn Prof. Nebel etwas beanstandet hat, habe ich das angemerkt.

SPIELTHEORIE >20MIN

## Was ist Spieltheorie?

Spieltheorie ist die Analyse strategischer Entscheidungssituationen.... (Praktisch den ersten Absatz des Vorlesungsskriptes mehr oder weniger Wortgetreu vorgetragen, die Frage scheint sehr häufig gestellt zu werden).

## Welche Forderungen müssen wir außerdem an die Agenten stellen?

Die Agenten handeln rational und sie gehen davon aus dass Ihre Gegenspieler ebenfalls rational handeln. 2/3 Average Spiel als Beispiel dafür genannt warum dies notwendig ist, um die Methoden der Spieltheorie erfolgreich verwenden zu können.

## Wie sieht den so ein strategisches Spiel aus?

$(N, (A_i), (u_i))$  aufgeschrieben und erklärt. (Kurze Diskussion, ob die Menge der Aktionen abzählbar sein muß)

## An was sind wir hier interessiert?

Gleichgewichten. Definition von NG aufgeschrieben.

## Gibt's solche Gleichgewichte immer?

Nein, aber für endliche Spiele in gemischten Strategien gibt es immer welche. Spiele in gemischten Strategien und Gleichgewichte erklärt, dann Satz von Nash, dazu zunächst die alternative Definition von Gleichgewichten aufgeschrieben und erklärt. Satz von Nash angegeben.

## Beweisen sie mal

Kakutami mit Voraussetzungen und Oberhemistetigkeit aufgeschrieben und die Beweisskizze aus dem Skript erklärt. Für den Rest des Beweises war Prof. Nebel glücklicherweise mit "... und dann muß man nur noch die Voraussetzungen überprüfen und ist fertig" zufrieden.

## Wie lassen sich in einem Nullsummenspiel die Gleichgewichte berechnen?

Lineares Programm in Standardform aufgeschrieben und Zeile für Zeile erläutert wie man aus einem Nullsummenspiel ein LP macht.

## Wie funktioniert das im allgemeinen Fall?

Komplementäres Lineares Programm aufgeschrieben und erklärt wo die Bedingungen herkommen (Support-Lemma

etc.) und dann ungefragt den dazugehörigen Lösungsalgorithmus erklärt.

## Wie sieht es mit der Laufzeit aus?

Laufzeit für das Lösen des LP mittels Simplex angegeben und daraus die Laufzeit für CLP abgeleitet. Außerdem Härte für andere Probleme genannt wie "Hat ein Spiel ein Auszahlungsprofil von mindestens x".

## Garantiert der Satz von Nash Gleichgewichte in allen Arten von endlichen Spielen ? (Oder so ähnliche Frage)

Nein, nicht in extensiven Spielen mit perfekter Information. Gegenbeispiel für leere Drohung aus dem Skript aufgezeichnet und erklärt. Unterschied zwischen Strategischen Spielen und Extensiven Spielen kurz in Worten zusammengefasst. Satz von Kuhn als Analogon zum Satz von Nash für extensive Spiele genannt.

## Wie beweist man den Satz von Kuhn?

Definition für Teilspielperfekte Gleichgewichte aufgeschrieben und kurz das Ein-Schritt-Abweichungslemma erwähnt dann, den Beweis in Worten gegeben. Irgendwie habe ich dann noch anhand der Beispiele aus dem Skript erklärt weswegen die Endlichkeit-Voraussetzung bei Kuhn existiert.

KÜNSTLICHE INTELLIGENZ <10MIN

## Das ist eine gute Brücke zur KI, erinnert sie was sie eben gemacht haben an etwas aus der KI?

Ja, Min-Max-Suche. Min-Max Suche erklärt, Alpha Beta Pruning und den Zusammenhang zum Einschrittabweichungslemma kurz mehr oder weniger richtig umrissen. (Zum Glück hatte ich den ersten Teil dieser Frage vorher in einem anderen Prüfungsprotokoll gesehen, sonst wäre ich da ohne Hilfe sicher nicht drauf gekommen)

## Können sie sonst noch etwas zur MinMax Suche erzählen?

Evaluation Function (Gewichtete lineare Funktionen) und für was man die benötigt am Beispiel von Schach erklärt, Probleme mit der Min-Max-Suche, Komplexität der Suche am Beispiel von Goo und Schach. (Ich glaube für Goo habe ich einen falschen durchschnittlichen Verzweigungsgrad angegeben, war aber wohl egal.)

**Min-Max bei “Unsicherheit”?**

Min-Max mit Erweiterung durch Monte Carlo Methode und “Average over all Outcomes” erklärt.

**Die Vorlesung behandelte ja auch andere Gebiete wie z.B. Logik, welche Methoden hatten wir zum überprüfen von Erfüllbarkeit?**

Davis Putnam, Resolution, Wahrheitstabelle, GSAT.

**Ok, nehmen wir DP, wie funktioniert der DP-Algorithmus?**

Den DP-Algorithmus komplett in Pseudocode aufgeschrieben und erklärt was in welchem Schritt passiert. Eigenschaften von DP: konstruiert ein Modell wenn eines existiert, komplett, korrekt, terminiert, ist anderen Verfahren überlegen.

**Laufzeit von DP?**

Laufzeit exponentiell wegen Splitting Rule.

**Laufzeit von DP für Horn Clauses?**

Polynomiell. Dann habe ich noch versucht aus dem Steg zu erklären, warum das so ist aber die Erklärung schien Prof. Nebel nicht so ganz zu überzeugen.

**Bemerkung:** Herr Prof. Nebel unterbricht einen nicht, solange man mit seiner Antwort halbwegs in der Nähe dessen liegt was er hören möchte. Für die Erklärung der LP, CLP und den Beweis des Satzes von Nash gingen gefühlt über 15 Minuten drauf, und daher blieb dann nicht mehr so viel Zeit für die hinteren Kapitel im Spieltheorieskript und den KI-Teil der Prüfung. Ich musste am Anfang ein paar Definitionen exakt aufschreiben und habe das dann für den Rest der Prüfung immer ungefragt getan, wenn ich ganz sicher war keinen Fehler zu machen. Die Prüfungsatmosphäre war sehr angenehm und ich kann für die Fächerkombination Spieltheorie und KI Prof. Nebel als Prüfer sehr empfehlen.

PRÜFUNGSVORBEREITUNG  
SOMMERSEMESTER 2013

---

## **CHEATSHEET**

### **FOUNDATIONS OF ARTIFICIAL INTELLIGENCE**

---

6. September 2013

N.R.  
BSc Informatik, 4. Fachsemester

Sommersemester 2013

# Inhaltsverzeichnis

<b>1 Suchen und Planung</b>	<b>3</b>
1.1 Depth-Search . . . . .	3
1.2 A* . . . . .	3
1.3 Heuristiken . . . . .	3
1.3.1 Optimistische Heuristiken (admissible) . . . . .	3
1.4 Constraint-Satisfaction-Problem . . . . .	4
1.5 Arc Consistency, Kantenkonsistenz . . . . .	4
1.6 Handlungsplanung & STRIPS . . . . .	4
1.6.1 Operatoren . . . . .	4
1.7 Progression Planning: Forward Search . . . . .	5
1.7.1 Regression Planning: Backward Search . . . . .	5
1.8 Relaxierung . . . . .	5
1.9 FF-System . . . . .	5
1.10 Bellman-Equation und Utility . . . . .	6
1.11 Value/Policy Iteration . . . . .	6
<b>2 Entscheidung</b>	<b>6</b>
2.1 Information Gain . . . . .	6
2.2 Decision Trees . . . . .	7
<b>3 Logik</b>	<b>7</b>
3.1 Resolution & DPLL . . . . .	7
3.2 PNF, SNF . . . . .	7
3.3 Herbrand . . . . .	8
3.3.1 Herbrand-Universum . . . . .	8
3.3.2 Herbrand-Struktur . . . . .	8
3.4 Unifizierung . . . . .	8
<b>4 Stochastik</b>	<b>8</b>
4.1 Bayes . . . . .	8
4.2 Normalisierung . . . . .	9
4.3 Bayes'sche Netze . . . . .	9

# 1 Suchen und Planung

## 1.1 Depth-Search

Einfachste Form der Suche. Findet Lösung, indem Baum in kompletter Tiefe besucht wird, bevor abzweigende Pfade aufgegriffen werden. Kann durch Variationen, wie die *beschränkte Tiefensuche*, oder daraus folgend die *iterative Tiefensuche*, verbessert werden.

## 1.2 A\*

Informierter Suchalgorithmus, der den kürzesten Pfad zwischen Knoten innerhalb eines Graphen mit positiven Kantengewichten finden will. Verwendet eine Schätzfunktion (Heuristik), um zielgerichtet zu agieren.

Findet immer die optimale Lösung.

**Algorithmus** Wendet alle potentiellen Aktionen an und berechnet für jeden resultierenden Knoten einen Wert  $f(x)$ , der angibt, wie lange der Pfad von Start nach Ziel unter Verwendung des betrachteten Knotens  $x$  im günstigsten Fall ist. Anschließend wird die Aktion ausgeführt, die  $f(x)$  minimiert:

$$f(x) = g(x) + h(x)$$

$g(x)$  bezeichnet dabei die Knoten vom Start aus zu  $x$ .  $h(x)$  steht für die geschätzten Kosten zum Ziel (Heuristik).

## 1.3 Heuristiken

Heuristiken sind Schätzfunktionen, die Alternativen in Suchproblemen bewerten, um die lohnenswerteste Aktion zu identifizieren.

Heuristiken sind dabei so konstruiert, dass sie schnell ausführbar sind und ausschließlich als Schätzwerte dienen. Tatsächliche Informationen über das konkrete Problem und die damit verbundenen Kosten können nicht extrahiert werden.

### 1.3.1 Optimistische Heuristiken (admissible)

Optimistische Heuristiken sind so definiert, dass sie niemals die tatsächlichen Kosten zur Zielerreichung überschätzen. Der Schätzwert ist also niemals höher als die geringstmöglichen Kosten.

## 1.4 Constraint-Satisfaction-Problem

CSP's sind Probleme, in denen ein Zustand über einer Menge von Objekten bestimmte Constraints oder Limitierungen erfüllen muss. CSP's werden dabei als Ansammlungen von endlichen Constraints über Variablen definiert, die über bestimmte *constraint satisfaction* Methoden (Bsp. Backtracking) gelöst werden.

## 1.5 Arc Consistency, Kantenkonsistenz

Die Kantenkonsistenz ist eine Möglichkeit CSP's zu lösen. Dabei wird zuerst der CSP-Graph erstellt, der die verschiedenen Variablen durch binäre Constraints verbindet. Anschließend wird jede Variable überprüft, ob für jeden Wert aus der gegebenen Domäne Werte der anderen Variablen gefunden werden können, die die Constraints verbinden. Ist dies nicht der Fall, so wird der Wert aus der Domäne gelöscht. Ist schließlich kein Konflikt mehr vorhanden spricht man von *Kantenkonsistenz*.

Für Graphen in Kantenkonsistenz kann schließlich sehr einfach eine valide Variablenbelegung gefunden werden.

## 1.6 Handlungsplanung & STRIPS

STRIPS (Stanford Research Institute Problem Solver) stellt eine formale Möglichkeit dar Probleme, die logisch formuliert werden können, zu lösen.

STRIPS erzeugt dabei ein Vokabular  $\mathcal{S}$  in Prädikatenlogik (Prädikate und Funktionssymbole) und stellt Atome  $\Sigma_{\mathcal{S}}$  bereit. STRIPS-Zustände  $S$  stellen schließlich eine Teilmenge von  $\Sigma_{\mathcal{S}}$  dar und beschreiben eine complete theory dar (per CWS, also eine Closed World Assumption).

Planungsaufgaben können nun also als 4-Tupel  $\Pi = \langle \mathcal{S}, \mathbf{O}, \mathbf{I}, \mathbf{G} \rangle$  dargestellt werden, wobei

- $\mathbf{O}$  eine Menge an Operatoren ist
- $\mathbf{I} \subseteq \Sigma_{\mathcal{S}}$  den Anfangszustand beschreibt
- $\mathbf{G} \subseteq \Sigma_{\mathcal{S}}$  die Zielspezifikation ist.

### 1.6.1 Operatoren

Operatoren in STRIPS sind 3-Tupel  $o = \langle para, pre, eff \rangle$ . Dabei gilt:

- $para \subseteq \mathbf{V}$
- $pre \subseteq \Sigma_{\mathcal{S}, \mathbf{V}}$
- $eff \subseteq \Sigma_{\mathcal{S}, \mathbf{V}} \cup \neg \Sigma_{\mathcal{S}, \mathbf{V}}$  (Elementweise Negierung)

- Alle Variablen in *pre* und *eff* sind in *para* aufgeführt.

Jede Operation hat also eine Vorbedingung, die erfüllt sein muss, um sie auszuführen. Der tatsächliche Effekt, der durch die Ausführung entsteht ist in *eff* definiert und wird in positive (*eff*<sup>+</sup>) und negative (*eff*<sup>-</sup>) Teileffekte eingeteilt.

## 1.7 Progression Planning: Forward Search

Bei Progression Planning wird beim Ausgangszustand begonnen. Es wird nun nicht-deterministisch eine anwendbare Aktion ausgewählt und der Folgezustand berechnet. Der Plan  $\delta$  wird um die Operation erweitert und ein Test wird ausgeführt, um herauszufinden, ob ein Zielzustand erreicht ist. Ist dies nicht der Fall, wird erneut nicht-deterministisch gewählt und das Prozedere fortgeführt, bis ein Ziel erreicht ist.

Anstatt nicht-deterministisch zu wählen kann selbstverständlich auch eine Heuristik, oder allgemein eine Strategie verwendet werden.

### 1.7.1 Regression Planning: Backward Search

Das Regression Planning stellt die Umkehrung des Progression Plannings dar. Anstatt beim Initialzustand zu starten, wird am Ziel begonnen. Es wird dann eine Aktion gesucht, deren Effekte der Information des Zielzustandes entsprechen und der Folgezustand über die Precondition der Operation gebildet.

Hier kann anstatt von nicht-deterministischen Mitteln ebenfalls eine Strategie verwendet werden.

## 1.8 Relaxierung

Bei der Relaxierung werden negative Effekte gestrichen. Daraus resultiert ein deutlich einfacher zu berechnendes Planungsproblem, welches als Heuristik verwendet werden kann.

## 1.9 FF-System

Variante des Planungsgraphen, bei der die Zieldistanzen nach jedem Suchzustand über Relaxierungen neu berechnet werden. Es werden allerdings nur Nachfolgezustände berücksichtigt, die über die relaxierte Lösung auch generiert würden.

Die Strategie stellt dabei *Enforced Hill-Climbing* dar.

## 1.10 Bellman-Equation und Utility

Die Bellman-Equation

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s, a, s') U(s')$$

wird verwendet, um die Utility eines bestimmten Zustandes zu bestimmen. Sie berechnet dabei also die maximalen Ausbeuten, die erreicht werden können in Bezug auf Folgezustände und Übergangswahrscheinlichkeiten und verrechnen diese mit Discountwert  $\gamma$  und dem direkten Reward  $R(s)$  des Zustandes selbst.

## 1.11 Value/Policy Iteration

**Value Iteration** ist ein Verfahren, um Markov-Entscheidungsprobleme zu lösen. Dabei wird für jeden Zustand der Utility-Wert errechnet. Irgendwann konvergieren die so errechneten Nutzen-Werte der Zustände gegen einen Grenzwert, der den optimalen Nutzen beschreibt.

**Policy Iteration** auf der anderen Seite errechnet sich anfangs über ein Gleichungssystem einen Grund-Nutzen für jeden Zustand auf einmal (Policy-Evaluation), um dann wiederholt zu überprüfen, ob die aktuelle Aktion jeden Zustandes immer noch die sinnvollste ist. Gestartet wird dabei mit einer zufällig ausgewählten Start-Policy.

# 2 Entscheidung

## 2.1 Information Gain

$$Gain(a) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - R(a)$$

wobei  $I$  den *Information Content* darstellt, also den Abstand zur endgültigen Aufteilung beschreibt, und  $R(a)$  den *Average Information Content* nach Wahl des Attributs  $a$  repräsentiert:

$$R(a) = \sum_v \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$
$$I(x, y) = x \ln(x^{-1}) + y \ln(y^{-1})$$

Gewählt werden sollte schließlich das Attribut, das den Gain maximiert.

## 2.2 Decision Trees

Entscheidungsbäume sind Möglichkeiten eine Menge von bekannten Daten so einzuteilen, das anhand der so erhaltenen Beispiele, neue Objekte in die eine oder andere Kategorie eingeteilt werden können.

Decision Trees werden dabei im besten Fall über die oben genannte Information-Gain Methode aufgebaut, also stets das Attribut zum Teilen verwendet, das den Gain maximiert. Prinzipiell ist allerdings jeder Baum, der eine Menge von Objekten klar in die Gruppen aufteilt, ein Entscheidungsbaum.

Dadurch beschreibt jeder Entscheidungsbaum also indirekt eine logische Funktion/Formel, anhand derer die Elemente klassifiziert werden.

# 3 Logik

## 3.1 Resolution & DPLL

Resolution ist eine Möglichkeit, Unerfüllbarkeit von Formeln zu beweisen. Dabei wird die Aussagenlogische Formel in CNF überführt (PL1-Formeln in Skolemform, die Quantoren ignoriert → Formel in CNF) und anschließend in Klauselform transformiert. Im nächsten Schritt können dann Resolventen gebildet werden (entweder direkt, oder durch Substitution), sodass sich am Ende die leere Menge ergibt (keine Aussage), oder eine leere Klausel übrig bleibt → Unerfüllbarkeit!

Alternativ dazu wurde die DPLL-Methode entwickelt, die die Resolutionsmethode in *Unit propagation rule*, also die Verwendung einer Klausel, die aus einem einzigen Literal besteht, oder *Splitting Rule*, also die Festsetzung einer Variable entweder auf True, oder False. Splitting Rule ist also eine Form des Backtrackings, der die Resolution leicht berechenbar macht.

## 3.2 PNF, SNF

**Pränexe Normalenform** ist eine prädikatenlogische Formel, in der sämtliche Quantoren am Anfang stehen. Vorgehensweise:

1. Doppelt vorkommende Variablen umbenennen
2. Negierungen so weit wie möglich durchreichen
3. Quantoren an den Anfang schreiben

**Skolem Normalenform** ist eine prädikatenlogische Formel, in der sämtliche Existenzquantoren eliminiert werden und alle Allquantoren am Anfang stehen. Vorgehensweise:

1. Formel in PNF bringen
2. Variablen, die durch Existenzquantoren gebunden sind, durch Funktionssymbole ersetzen. Parameter dieser Funktionssymbole sind die Variablen der vorhergehenden Allquantor-Variablen.

### 3.3 Herbrand

#### 3.3.1 Herbrand-Universum

Das Herbrand-Universum ist die Grundmenge, die der Herbrand-Struktur zu Grunde liegt. Sie wird folgendermaßen aufgebaut:

1. Konstanten der Formel in die Menge aufnehmen
2. Iterativ alle neuen Kombinationen aus bereits gefundenen Termen der Menge und den vorhandenen Funktionssymbolen in die Menge aufnehmen.

Das Herbrand-Universum ist also eine unendliche Menge von Termen.

#### 3.3.2 Herbrand-Struktur

Herbrand-Struktur ist eine Terminterpretation, die auf Basis einer Formel definiert ist. Dabei wird das Universum gebildet und anschließend allen Variablen Terme aus der resultierenden Universum-Menge zugewiesen.

### 3.4 Unifizierung

Nicht übereinstimmende Terme identifizieren und versuchen eine Substitution zu finden, durch die sie äquivalent werden.

## 4 Stochastik

### 4.1 Bayes

$$P(X|Y) = \frac{P(X \wedge Y)}{P(Y)}$$

$$P(X \wedge Y) = P(Y|X)P(X)$$

$$P(X|Y) = \alpha P(X \wedge Y)$$

## 4.2 Normalisierung

Die Normalisierung ist eine Möglichkeit, um von relativen Wahrscheinlichkeiten auf ihre absoluten zu kommen:

$$P(X) = \alpha \langle 0.4, 0.3 \rangle = \frac{1}{0.4 + 0.3} \langle 0.4, 0.3 \rangle = \dots$$

## 4.3 Bayes'sche Netze

Möglichkeit um logische Zusammenhänge in stochastischen Umgebungen effizient zu repräsentieren. Anstatt eine riesige Joint Probability-Tabelle zu verwenden, werden in den Netzen die Informationen in Ereignisse geteilt und diese über Kanten zu einem Graphen verbunden. Jeder Knoten repräsentiert dann im Prinzip eine Teiltabelle.

Inferenz kann schließlich über eine Summierung über alle versteckten Variablen und anschließende Berechnung der Gesamtwahrscheinlichkeit durchgeführt werden.

Der Vorteil gegenüber der Joint-Probability-Tabelle ist Effizienz in der Speicherung und Wartung. Nachteilig ist die Inferenzgeschwindigkeit, da Werte aus der Tabelle einfach abgelesen werden könnten.

---

**FOUNDATIONS OF ARTIFICIAL INTELLIGENCE**  
Reconstruction of 2014's exam (16.09.2014)

---

**Lecturers:** Prof. Dr. Wolfram Burgard  
Prof. Dr. Bernhard Nebel  
Prof. Dr. Martin Riedmiller

**Assistants:** Johannes Aldinger  
Dr. Joschka Boedecker  
Philipp Ruchti

**Permitted exam aids:** none  
**Duration:** 90 Minutes

## **Task 1 – MinMax Algorithm**

**Sadly, we weren't able to reconstruct this task.**

---

The only thing we remember is that the tree had a probabilistic step.

## Task 2 – Decision Trees

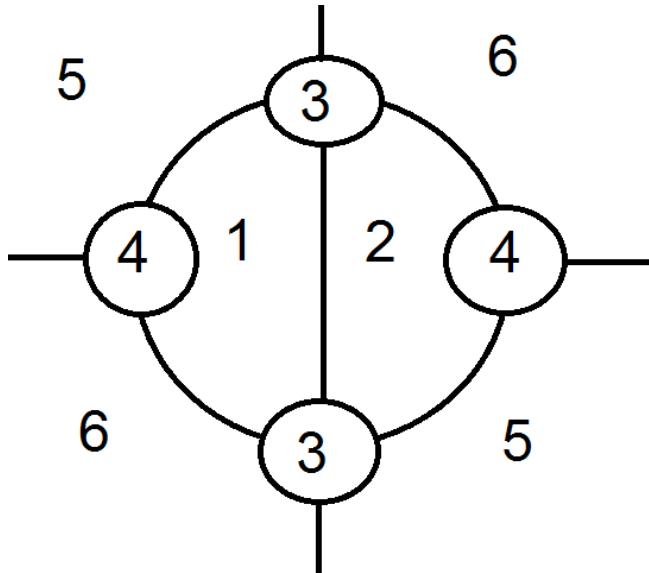
**Sadly, we weren't able to reconstruct this task.**

---

The only thing we remember is that a table was given, one was supposed to calculate the gain values and then build a decision tree and convert it to DNF. The task did have some similarities with Task 6.5 on <http://gki.informatik.uni-freiburg.de/teaching/ss14/gki/exercises/sheet06-english.pdf>.

### Task 3 – Forward Checking

You are a gardener assigned to the following garden:



You can choose one flower-type for each patch (1-6). Available flower-types are A, B, C, D and E. You may not assign the same flower-type to two adjacent patches. Execute the forward checking algorithm using the following tie-breakers:

- most constrained first
- if equally many constraints apply, choose the flower-type earlier in the alphabet for the patch with the smaller number

---

The result should get inserted into a table, one calculation step per line (one column for each patch). The first line contained the first step as an example.

## Task 4 – Allen’s Interval Calculus

- a) Given the following statements

popcorn  $overlaps^{-1}$  movie      credits  $finishes$  movie

Which statements can get deduced between popcorn and credits?

- b) Is  $starts^{-1}$  equal to  $finishes$ ? Explain your solution.

## Task 5 – STRIPS

Consider  $\langle \mathcal{S}, \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$  with

$$\begin{aligned}\mathcal{S} &= \{X, Y, Z, G\} \\ \mathcal{O} &= \{A, B, C, D\} \\ \mathcal{I} &= \{X\} \\ \mathcal{G} &= \{G\}\end{aligned}$$

where the actions  $A, B, C, D$  are given as

$A$	pre : $\neg Y$	post : $\neg X$
$B$	pre : $\neg X$	post : $X, Y$
$C$	pre : $X, Y$	post : $\neg Y, Z$
$D$	pre : $X, Y, Z$	post : $G$

and the plan  $\pi$ , given as

$$\pi = \langle A, B, C, B, D \rangle$$

- a) Execute  $\pi$  as far as possible, and give the set of predicates active after each step of your execution. If  $\pi$  cannot get executed completely, state the reason why the execution failed.
- b) Derive a plan  $\pi'$  from  $\pi$ , which solves the planning task. You may only add one single action.

## Task 6 – A\*-Algorithm

**Sadly, we weren't able to reconstruct the actual graph, its costs, and the actual heuristic function.** You were given a graph with labeled nodes, and costs assigned to each edge. A text described one node as starting point, and another node as target node. Additionally, a heuristic function  $h$  was given as a table (one value for each node in the graph, excluding the target node).

- a) Execute the A\*-Algorithm. While executing, annotate the graph with all  $f$  and  $g$  values. Mark each node with the number of the iteration in which it was expanded.  
Draw the resulting tree.
  - b) What property of the heuristic function is required for  $A^*$  to yield a optimal result? What happens if this property is violated?
- 

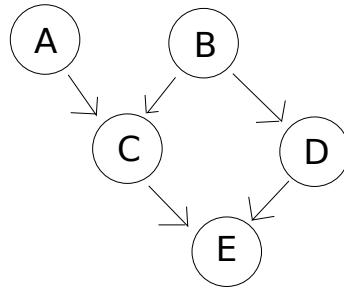
Although we weren't able to completely reproduce this task, we'd like to note an (in our opinion) important fact about the actual graph that was used in the exam:

One node's  $g$  value was updated while applying the algorithm. When drawing the tree while applying the algorithm, this lead to the tree no longer be a tree (as there are two edges towards the updated edge). To preserve the tree, the old edge has to be removed, or the solution wouldn't get accepted.

## Task 7 – Bayesian Networks

Tables with conditional probabilities were given. (**Which sadly couldn't be reconstructed**)

- a) Draw the corresponding Bayesian-Network.
- b) Calculate  $P(D)$ .
- c) Give a definition of the Markov Blanked and indicate the Markov Blanket of D in the following network.



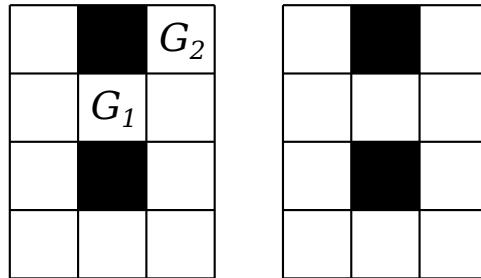
## Task 8 – Satisfiability

- a) Apply the Resolution method to something given (**which is obviously missing in this reconstruction**).
- b) Apply the DPLL algorithm on a given Formula (**which is obviously missing in this reconstruction**). Note: There was one backtrack in the solution.

## Task 9 – Grid World

Consider the grid world, where only horizontal and vertical movements by one position per step are allowed. Positions marked black are inaccessible.

- a) Unfortunately, we couldn't remember this task exactly. The rewards of the end positions  $G_1$  and  $G_2$  were given. One  $G_i$  had a bad reward and should have been avoided. There must have been further information about the rewarding system (e.g. discount) and probabilities to success or fail, when making a move.  
One task was to calculate the *utilites or rewards* of the empty cells on the left. The other task was to fill in the *optimal policy* (obtained by human intuition) on the right.



- b) Apply two steps of *value iteration* on the following grid with given (sub-optimal) utilities. Fill in the improved utilities after each step.

-7			-30			
-2	0		-1			
-9			-4			
-8	-6		-5			

The diagram shows a 4x3 grid with numerical values in the first three columns. The fourth column is empty. The values are: Row 1: -7, empty, empty, -30; Row 2: -2, 0, empty, -1; Row 3: -9, empty, empty, -4; Row 4: -8, -6, empty, -5.

## Gedankenaufschrieb der KI Nachklausur SoSe 2014

### 1. $\alpha$ - $\beta$ Pruning

In nicht beschrifteten Tabellen  $\alpha$  und  $\beta$  eintragen. In das jeweilige  $\Delta$  den v-Wert

### 2. Entscheidungsbaum

Nr	Actor	Genre	Sell-out
1	Mell Gibson	Thriller	yes
2	Tom Cruise	Action	yes
3	Mell Gibson	Thriller	yes
4	Angelina Jolie	Romance	no
5	Mell Gibson	Action	yes
6	Tom Cruise	Romance	no
7	Mell Gibson	Action	yes
8	Angelina Jolie	Thriller	no
9	Angelina Jolie	Romance	no
10	Mell Gibson	Action	yes

(a) Berechne für *Genre* den R-Wert um zu bestimmen, wie viel es den Gain für Sell-out reduziert.

(b) Gebe einen Entscheidungsbaum an. Bei Actor bin ich mir mit der Verteilung nicht mehr 100% sicher. Jedoch hat sie für das Genre Thriller die Mengen geteilt.

### 3. Suchalgorithmen

(a) Ein Graph gegeben, an den Kanten die g-Werte. In einer Tabelle die h-Werte. Man sollte A\* durchführen. F, h und g-Werte an die Knoten. Und die Reihenfolge der Expandierung an Knoten durchnummerieren. Knoten sollten nicht nochmals rückwärts expandiert werden.  
(Wenn von A -> B und nun B expandiert, dann ist A nicht nochmals Kindsknoten)

(b) Unter welchen Bedingungen ist Breitensuche vollständig? Unter welchen Bedingungen ist Breitensuche optimal?

#### 4. CSP

Gegeben war die Belegung  $\alpha$  wie sie in der unten stehenden Tabelle zu sehen ist.

	1	2	3	4	5
v1		X			
v2					
v3					
v4					
v5					

(a) Führe Forwardchecking für Belegung  $\alpha$  aus

	v1	v2	v3	v4	v5
	{2}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}
Fwc					

(b) Führe ARC-Consistency für Belegung  $\alpha$  aus

	v1	v2	v3	v4	v5
ACv1	{2}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}
ACv2	{2}		{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}
ACv3	{2}			{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}
ACv4	{2}				{1, 2, 3, 4, 5}
ACv5	{2}				
	{2}				
	{2}				
	{2}				
	{2}				

**5.** Kann ich mich leider nicht mehr erinnern.

## **6. Byes' Netzwerk**

Man hatte A, B, C, D, E als Knoten und ihre bedingte Abhängigkeiten in Wahrheitstabellen.  
An die Werte kann ich mich nicht mehr erinnern kann.

D	C	$P(E C, B)$
t	t	
t	f	
f	t	
f	f	

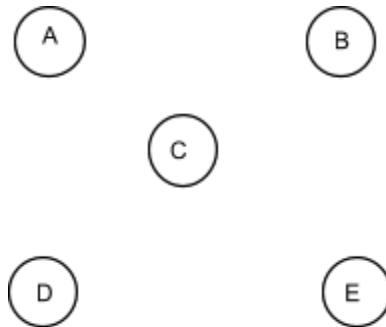
D	$P(A D)$
t	
f	

C	$P(C)$
t	0.2
f	

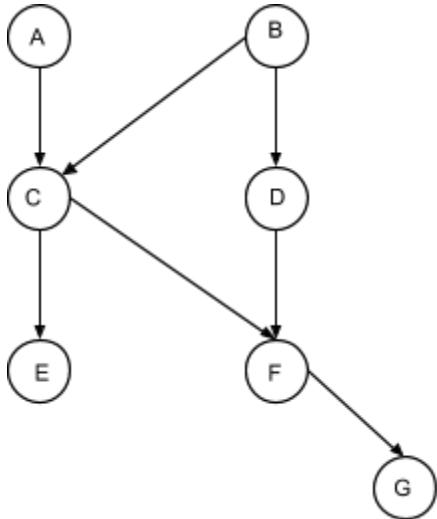
D	C	$P(B D, C)$
t	t	
t	f	
f	t	
f	f	

C	$P(D C)$
t	
f	

**(a)** Zeichne das Baysche Netz anhand der gegebenen Tabellen ein.



**(b)** Berechne Joint Probability von  $P(A, \neg B, C, D, E)$ .



(c) Bedingt Unabhängigkeit ja, nein. Bin mir nicht 100% sicher, jedoch sah Graph so ähnlich aus.

$$\text{Ind}(F, B | D)$$

$$\text{Ind}(E, F | C)$$

$$\text{Ind}(F, A | C, B, G)$$

$$\text{Ind}(C, D | F, B)$$

7. Kann ich mich leider nicht mehr erinnern.

### 8. Unifizierung

Tabelle war gegeben, die man ausfüllen sollte. Ich erinnere mich nicht mehr 100% an T. Es musste mind. 4 mal substituiert werden.

	$T\{ \}$	$S\{ \}$	$D\{ \}$	v	t
0	$\{R(x, y, f(A, y)), R(A, g(x), f(x, g(x)))\}$	$\{ \}$			

### 9. Logik 2. Stufe

(a) DPLL. Achtung hier war angegeben, dass wenn Splitting-Rule, dann der Buchstabe, der im Alphabet zu erst kommt, gewählt und zu erst auf TRUE gesetzt werden soll.

(b) Formel in Skolem Normalform überführen

Department of Computer Science

**Exam:** Foundations of Artificial Intelligence - Mock-Exam

**Duration:** 90 minutes

**Permitted exam aids:** Indelible pen (e.g. ball pen, *no* pencil!), nothing else.

---

**Notes:**

- Please fill out this form.
- Please write only on one side of your paper sheets.
- Please write your name and your matriculation number on each paper sheet.
- Please use a new paper sheet for each question.
- Please turn off your mobile phone.

**Withdrawing from an examination:**

In case of illness, you must supply proof of your illness by submitting a medical report to the Examinations Office. Please note that the medical examination must be done at the latest on the same day of the missed exam. More information: [http://www.tf.uni-freiburg.de/studies/exams/withdrawing\\_exam.html](http://www.tf.uni-freiburg.de/studies/exams/withdrawing_exam.html)

**Cheating/disturbing in examinations:**

A student who disrupts the orderly proceedings of an examination will be excluded from the remainder of the exam by the respective examiners or invigilators. In such a case, the written exam of the student in question will be graded as "nicht bestanden" (5.0, fail) on the grounds of cheating. In severe cases, the Board of Examiners will exclude the student from further examinations.

---

Question	Score	Reached score	Comments	Initials
1	6			
2	15			
3	15			
4	6			
5	13			
6	15			
7	20			
<b>Sum</b>	<b>90</b>			

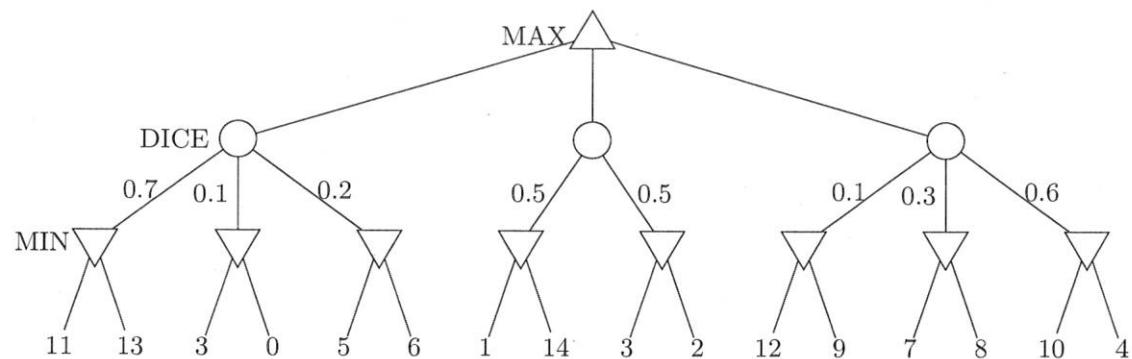
Name and matriculation number:

**Question 1 (6 points)**

BOARD GAMES / BRETTSPIELE

Consider the following game tree for a two-person game with chance nodes:

Betrachten Sie den folgenden Spielbaum für ein 2-Personen-Spiel mit Zufallsknoten:



Determine the utility of all nodes in the game tree.

Bestimmen Sie die Bewertungen der Knoten des Spielbaumes.

Name and matriculation number:

Question 2 (8+7 points)

DECISION TREES / ENTSCHEIDUNGSBÄUME

No Nr.	LeadingActor Hauptdarsteller	Genre Genre	SoldOut? ausverkauft?
1	Matt Damon	Thriller / Krimi	yes / ja
2	Tom Cruise	Action / Action	yes / ja
3	Matt Damon	Action / Action	yes / ja
4	Angelina Jolie	Romance / Liebesfilm	no / nein
5	Matt Damon	Romance / Liebesfilm	no / nein
6	Angelina Jolie	Thriller / Krimi	no / nein
7	Angelina Jolie	Action / Action	yes / yes
8	Tom Cruise	Thriller / Krimi	no / nein
9	Tom Cruise	Romance / Liebesfilm	no / nein
10	Matt Damon	Action / Action	yes / yes

Consider the above record of movie screenings a cinema owner has produced.

Betrachten Sie die von einem Kinobetreiber gesammelten Daten bisheriger Vorführungen.

- (a) We consider the attribute **Genre** as candidate for the first node in a decision tree used to classify whether a movie screening was sold out or not. How much would this attribute reduce the uncertainty about the classification? Justify your answer with detailed calculations. You can make use of the following values:

*Wir betrachten das Attribut **Genre** als Kandidaten für den ersten Knoten eines Entscheidungsbaums der klassifizieren soll, ob eine Vorstellung ausverkauft war oder nicht. Um wieviel würde das Attribut die Unsicherheit der Klassifikation reduzieren? Rechtfertigen Sie Ihre Antwort mit detaillierten Berechnungen. Sie können folgende Werte benutzen:*

$$\log_2\left(\frac{1}{3}\right) \approx -\frac{3}{2}, \log_2\left(\frac{2}{3}\right) \approx -\frac{1}{2}, \log_2\left(\frac{1}{2}\right) = -1, \log_2(1) = 0, 0 \cdot \log_2(0) = 0.$$

Name and matriculation number:

---

- (b) Produce a decision tree, which, by means of the given attributes, correctly classifies whether the given examples have been sold out or not.

*Geben Sie einen Entscheidungsbaum an, der für die gegebenen Beispiele anhand der Attribute korrekt klassifiziert, ob die jeweilige Vorstellung ausverkauft war oder nicht.*

(additional room for answer to Question 2)

Name and matriculation number:

**Question 3 (4+11 points)**

CONSTRAINT SATISFACTION PROBLEMS / CSPs

Consider the 5-queens problem, where 5 pieces have to be placed on a size  $5 \times 5$  board in such a way that no two queens are on the same horizontal, vertical, or diagonal line. The Variables  $v_i \in V = \{v_1 \dots v_5\}$  indicate the position of the  $i$ -th queen with domain  $\text{dom}(v_i) = 1, \dots, 5$  for all variables  $v_i \in V$ . Consider now state  $\alpha = \{v_1 \mapsto 2\}$ .

Betrachten Sie das 5-Damen Problem, bei dem 5 Spielfiguren auf einem  $5 \times 5$  Felder großen Brett so platziert werden sollen, dass sich keine zwei Damen auf der selben horizontalen, vertikalen oder diagonalen Linie befinden. Die Variablen  $v_i \in V = \{v_1 \dots v_5\}$  geben die Position der  $i$ -ten Dame an und haben einen Wertebereich  $\text{dom}(v_i) = 1, \dots, 5$  für alle Variablen  $v_i \in V$ . Betrachten Sie nun den Zustand  $\alpha = \{v_1 \mapsto 2\}$ .

	1	2	3	4	5
$v_1$		♛			
$v_2$					
$v_3$					
$v_4$					
$v_5$					

- (a) Apply forward-checking in  $\alpha$ . Fill the table below with the domains of the missing variables.

Führen Sie Forward-Checking in  $\alpha$  aus. Tragen Sie die Domänen der fehlenden Variablen in die Tabelle ein.

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$
init	{2}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}
FC	{2}				

Name and matriculation number:

- (b) Enforce arc consistency in  $\alpha$ . Specify the domains of the variables after applying arc consistency to all constraints belonging to one variable. (E.g. in the first row you should depict the domain of  $v_2$  after enforcing arc consistency for all constraints containing  $v_2$ .)

*Erzeugen Sie Kantenkonsistenz in  $\alpha$ . Geben Sie hierzu die Wertebereiche der Variablen nach dem Erzeugen der Kantenkonsistenz für alle Constraints an, in welchen die aktuell betrachtete Variable beteiligt ist. (Z.B. in der ersten Reihe ist die neue Domäne von  $v_2$  anzugeben, nachdem Kantenkonsistenz für alle Constraints an denen  $v_2$  beteiligt ist, erzeugt wurde.)*

	1	2	3	4	5
$v_1$		!			
$v_2$					
$v_3$					
$v_4$					
$v_5$					

Additional sketch of the state  $\alpha$  so you don't have to flip pages.

*Weitere Skizze des Zustands  $\alpha$  damit Sie nicht umblättern müssen.*

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$
init	{2}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}
AC - $v_2$	{2}		{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}
AC - $v_3$	{2}			{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}
AC - $v_4$	{2}				{1, 2, 3, 4, 5}
AC - $v_5$	{2}				
	{2}				
	{2}				
	{2}				
	{2}				

Name and matriculation number:

---

**Question 4 (4 + 2 points)**

PLANNING / HANDLUNGSPLANUNG

Consider the following STRIPS-Task  $\Pi = \langle S, O, I, G \rangle$ :

Betrachten Sie folgenden STRIPS-Task  $\Pi = \langle S, O, I, G \rangle$ :

- $S: \{X, Y, Z, G\}$
- $O: \{A, B, C, D\}$  where

$$\begin{array}{ll} A : \text{pre}(A) = \{\neg Y\}, & \text{eff}(A) = \{\neg X\} \\ B : \text{pre}(B) = \{\neg X\}, & \text{eff}(B) = \{X, Y\} \\ C : \text{pre}(C) = \{Y\}, & \text{eff}(C) = \{\neg Y, Z\} \\ D : \text{pre}(D) = \{X, Y, Z\}, & \text{eff}(D) = \{G\} \end{array}$$

- $I: \{X\}$
- $G: \{G\}$

- (a) Derive the states from executing the plan  $\pi = \langle A, B, C, B, D \rangle$  in  $I := s_0$ , as long as actions can be legally applied.

*Leiten Sie die Zustände her, welche bei Ausführung des Plans  $\pi = \langle A, B, C, B, D \rangle$  entstehen, solange die Aktionen legal angewendet werden.*

$$s_0 = \{X\}$$

$$s_1 = App(s_0, A) =$$

$$s_2 = App(s_1, B) =$$

$$s_3 = App(s_2, C) =$$

$$s_4 = App(s_3, B) =$$

$$s_5 = App(s_4, D) =$$

Name and matriculation number: \_\_\_\_\_

- (b) Is the Plan in (a) applicable? If not, can it be fixed by adding an action to the plan?  
State a plan which solves the problem.

*Ist der Plan in (a) anwendbar? Falls nicht, kann er durch Hinzufügen einer Aktion repariert werden? Geben Sie einen Plan an, der das Problem löst.*

(additional room for answer to Question 4)

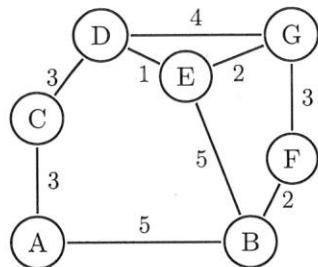
Name and matriculation number:

**Question 5 (10 + 3 points)**

A\* SEARCH / A\*SUCHE

- (a) Draw the search tree generated by the A\* algorithm when searching for a shortest path from Node **A** to Node **G**, using the heuristic distance to node **G** given in the table. Indicate in which order the nodes are expanded and annotate each node with its f, g, and h value. Edges back to parent nodes do not have to be generated. Mark the goal node.

*Zeichnen Sie den Suchbaum, der vom A\*-Algorithmus für die Suche eines kürzesten Weges von Knoten **A** zu Knoten **G** erzeugt wird. Verwenden Sie hierzu den heuristischen Abstand zu Knoten **G** welcher in der Tabelle angegeben ist. Geben Sie an, in welcher Reihenfolge die Knoten expandiert werden, und annotieren Sie jeden Knoten mit seinen f-, g-, und h-Werten. Kanten noten zurück zu Elternknoten müssen nicht generiert werden. Markieren sie den Zielknoten.*



	$h(x)$
A	7
B	3
C	6
D	3
E	2
F	3
G	0

- (b) Is breadth-first search complete? Under which condition does it find the optimal solution?

*Ist Breitensuche vollständig? Unter welchen Bedingungen findet die Breitensuche die optimale Lösung?*

(additional room for answer to Question 5)



Name and matriculation number:

**Question 6** (3.5 + 7.5 + 4 points)

BAYES / BAYES

Consider the following probability tables (all variables are binary, thus they can be either true or false)

*Betrachten Sie die folgenden Wahrscheinlichkeitstafeln (alle Variablen sind binär und können somit entweder wahr oder falsch sein)*

B	D	$\mathbf{P}(A   B, D)$
T	T	0.6
T	F	0.3
F	T	0.2
F	F	0.7

C	$\mathbf{P}(B   C)$
T	0.7
F	0.2

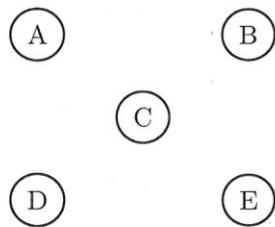
$\mathbf{P}(C)$	
	0.2

B	C	$\mathbf{P}(D   B, C)$
T	T	0.2
T	F	0.7
F	T	0.5
F	F	0.4

B	C	$\mathbf{P}(E   B, C)$
T	T	0.3
T	F	0.4
F	T	0.1
F	F	0.7

- (a) Draw the corresponding Bayesian network. Use the already drawn nodes and add the corresponding edges:

*Zeichnen Sie das zugehörige Bayes'sche Netz. Verwenden Sie hierzu die vorgegebenen Knoten und ergänzen die fehlenden Kanten:*



- (b) Compute the probability  $P(A, \neg B, C, E)$ .

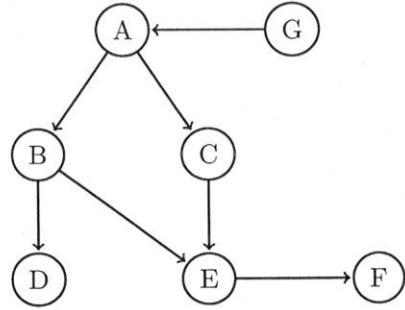
*Berechnen Sie die Wahrscheinlichkeit  $P(A, \neg B, C, E)$ .*

Name and matriculation number:

---

- (c) Determine, which of the following independence statements follow from the structure of the Bayesian network given below.  $\text{Ind}(U, V \mid W)$  denotes that  $U$  is independent of  $V$  given  $W$ :

*Bestimmen Sie, welche der folgenden Unabhängigkeiten aus der Struktur des untenstehenden Bayes'schen Netzes folgen. Hierbei steht  $\text{Ind}(U, V \mid W)$  dafür, dass  $U$  unabhängig von  $V$  gegeben  $W$  ist:*



- $\text{Ind}(B, C \mid A)$
- $\text{Ind}(C, G \mid A)$
- $\text{Ind}(E, A \mid C)$
- $\text{Ind}(E, A \mid B, C, F)$

Name and matriculation number:

**Question 7 (6 + 8 + 6 points)**

LOGIC / LOGIK

- (a) Apply the unification algorithm to the following set of literals: (As usual, we use  $\{P\}$  for predicate symbols,  $\{f, g\}$  for function symbols,  $\{x, y, z\}$  for variables and  $\{A\}$  for constants.)

*Wenden Sie den Unifikationsalgorithmus auf die folgende Literalmenge an: (Wie üblich verwenden wir  $\{P\}$  für Prädikatensymbole,  $\{f, g\}$  für Funktionssymbole,  $\{x, y, z\}$  für Variablen und  $\{A\}$  für Konstanten.)*

$$\{P(x, y, g(A, f(z))), P(g(z, y), y, g(z, y))\}$$

In each step  $k$ , give the values of the set of Terms  $T_k$ , the substitution  $s_k$ , the disagreement set  $D_k$  replaced variable  $v_k$  and the replacement term  $t_k$ .

*Geben Sie für jeden Schritt  $k$  die Werte für die Menge von Termen  $T_k$ , die Substitution  $s_k$ , die "Disagreement" Menge  $D_k$  die ersetze Variable  $v_k$  und der Ersetzungsterm  $t_k$  an.*

$k$	$T_k$	$s_k$	$D_k$	$v_k$	$t_k$
0	$\{P(x, y, g(A, f(z))), P(g(z, y), y, g(z, y))\}$	$\emptyset$			
1					
2					
3					

Name and matriculation number:

---

- (b) Use the Davis-Putnam-Logemann-Loveland (DPLL) Procedure to find a satisfying assignment for the formula  $\phi$ . Write out all steps of the algorithm. If you have to apply a splitting rule, split on variables in alphabetical order, trying *true* first, then *false*. Indicate the satisfying assignment.

*Verwenden Sie die Davis-Putnam-Logemann-Loveland(DPLL)-Prozedur, um eine erfüllende Belegung der Formel  $\phi$  zu finden. Schreiben Sie alle Schritte des Algorithmus auf. Wenn Sie eine Verzweigungs-Regel anwenden müssen, wählen Sie die Verzweigungs-Variablen in alphabetischer Reihenfolge aus, und wählen Sie zuerst wahr, dann falsch. Geben Sie die erfüllende Belegung an.*

$\phi = (\neg A \vee \neg C \vee D) \wedge (A \vee \neg C) \wedge (B \vee C \vee D) \wedge (A \vee D) \wedge (\neg A \vee \neg B) \wedge (B \vee \neg D)$

- (c) Convert the following formula to Skolem normal form.

*Wandeln Sie die folgende Formel in Skolem-Normalform um.*

$$\neg \forall x \exists y P(x, f(y)) \wedge \forall z \exists x (Q(z) \vee R(z, g(x)))$$

(additional room for answer to Question 7)

(

(

Name and matriculation FIRSTNAME LASTNAME, MATRICULATIONNUMBER

Additional room for notes. If you write any solutions here, please point it out at the question.

*Zusätzlicher Platz für Notizen. Falls Sie Lösungen hier aufschreiben, weisen Sie bitte bei der Aufgabe darauf hin.*

