

Künstliche Intelligenz

Prof. J. Fürnkranz

Technische Universität Darmstadt — Wintersemester 2014/15

Termin: 20. 2. 2015

Name:

Vorname:

Matrikelnummer:

Fachrichtung:

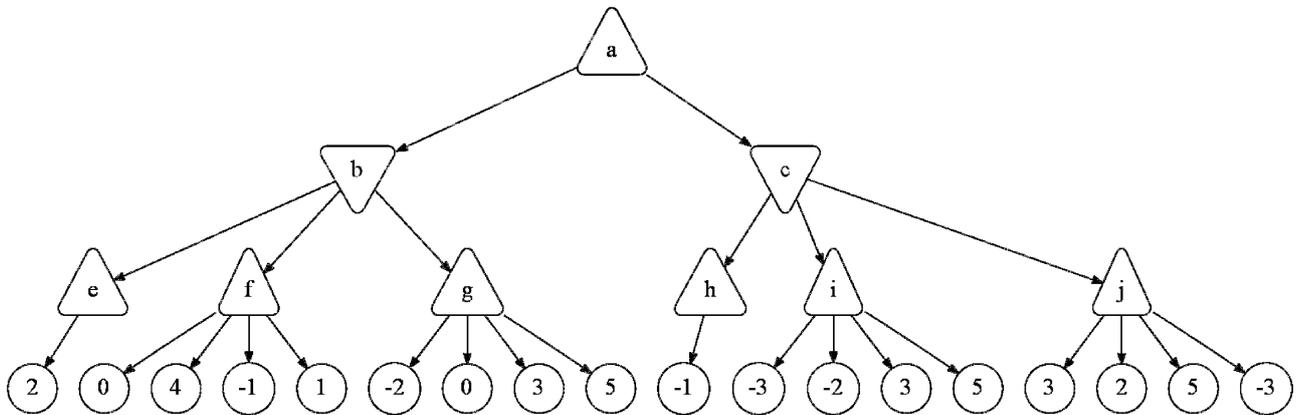
Punkte: (1) (2) (3) (4) (5) Summe:

- **Aufgaben:** Diese Klausur enthält auf den folgenden Seiten 5 Aufgaben zu insgesamt 100 Punkten. Jede Aufgabe steht auf einem eigenen Blatt. Kontrollieren Sie *sofort*, ob Sie alle sechs Blätter erhalten haben!
- **Zeiteinteilung:** Die Zeit ist knapp bemessen. Wir empfehlen Ihnen, sich zuerst einen kurzen Überblick über die Aufgabenstellungen zu verschaffen, und dann mit den Aufgaben zu beginnen, die Ihnen am besten liegen.
- **Papier:** Verwenden Sie nur Papier, das Sie von uns ausgeteilt bekommen. Bitte lösen Sie die Aufgaben auf den dafür vorgesehenen Seiten. Falls der Platz nicht ausreicht, vermerken sie dies bitte und setzen die Lösung auf der letzten Seite fort. Brauchen Sie zusätzlich Papier (auch Schmierpapier), bitte melden.
- **Fragen:** Sollten Sie Teile der Aufgabenstellung nicht verstehen, bitte fragen Sie!
- **Abschreiben:** Sollte es sich herausstellen, dass Ihre Lösung und die eines Kommilitonen über das zu erwartende Maß hinaus übereinstimmen, werden beide Arbeiten negativ beurteilt (ganz egal wer von wem in welchem Umfang abgeschrieben hat).
- **Ausweis:** Legen Sie Ihren *Studentenausweis* und *Lichtbildausweis* sichtbar auf Ihren Platz. Füllen Sie das Deckblatt sofort aus!
- **Hilfsmittel:** Zur Lösung der Aufgaben ist ein von Ihnen selbst handschriftlich beschriebenes DIN-A4-Blatt erlaubt. Gedruckte Wörterbücher sind für ausländische Studenten erlaubt, elektronische Hilfsmittel (Taschenrechner, elektronische Wörterbücher, Handy, etc.) sind verboten! Sollten Sie etwas verwenden wollen, was nicht in diese Kategorien fällt, bitte klären Sie das *bevor* Sie zu arbeiten beginnen.
- **Aufräumen:** Sonst darf außer Schreibgerät, Essbarem, von uns ausgeteiltem Papier und eventuell Wörterbüchern nichts auf Ihrem Platz liegen. Taschen bitte unter den Tisch!

Gutes Gelingen!

Aufgabe 1 Spielbaum-Suche (2/6/4/3/4/4 = 23 Punkte)

Gegeben sei folgender Spielbaum, bei dem an der Wurzel der MAX-Spieler am Zug ist:



1-a Welchen Zug (b oder c) soll MAX spielen und welches Ergebnis kann er mit diesem Zug erreichen?

1-b Geben Sie alle *Blatt-Knoten* an, die von einer Alpha-Beta-Suche, die die Knoten in der gegebenen Reihenfolge (von links nach rechts) durchsucht, *nicht* betrachtet werden müssen.

Bezeichnen Sie die Blatt-Knoten, indem Sie die Nachfolger der jeweiligen Elternknoten durchnummerieren. Der Knoten f2 bezeichnet also den Blatt-Knoten mit dem Wert 4 (dritter von links).

1-c Geben Sie die jeweiligen Werte für α und β bei folgenden Aufrufen an, die im Laufe der Suche passieren würden (f2 bezeichnet den Knoten mit der Bewertung 4, wie unter b) beschrieben).

1. $\text{MIN-VALUE}(f2, \alpha, \beta)$
2. $\text{MAX-VALUE}(f, \alpha, \beta)$
3. $\text{MIN-VALUE}(c, \alpha, \beta)$

1-d Geben Sie für die folgenden Aufrufe von MIN-VALUE und MAX-VALUE in zwei hypothetischen Knoten s_1 und s_2 die entsprechenden Parameter α und β für die Aufrufe von ALPHA-BETA in einer rekursiven NEGAMAX Implementierung an.

1. MIN-VALUE($s_1, -4, 1$) \rightarrow ALPHA-BETA(s_1, α, β)
2. MAX-VALUE($s_2, -2, \infty$) \rightarrow ALPHA-BETA(s_1, α, β)

1-e Warum müssen bei einer Minimal Window Search Teile des Suchbaumes mehrmals durchsucht werden? Warum kann die Suche dennoch effizienter als eine herkömmliche Alpha-Beta Suche sein?

1-f Hier ist die Formel aus der Vorlesung, mit der UCT (Monte-Carlo Tree Search) einen Nachfolger s des Knotens n zur Expansion auswählt.

$$s_{\max} = \arg \max_{s \in \text{SUCCESSORS}(n)} \text{value}(s) + C \cdot \sqrt{\frac{\log(\#\text{visits}(n))}{\#\text{visits}(s)}}$$

Erklären Sie, wie diese Formel einen Trade-off zwischen Exploration und Exploitation realisiert.

Aufgabe 2 Planning (4/4/5/5/2 = 20 Punkte)

Sie möchten ein Bild eines Aquariums malen. Dazu muss zuerst ein Hintergrund gemalt werden und darauf dann drei Objekte *Nemo*, *Flipper*, und *Arielle* gezeichnet werden.

Zum Erstellen des Bildes auf einem zu Beginn leeren Blatt stehen zwei Aktionen zur Verfügung:

male_hintergrund	
PREC:	leeres_blatt
ADD:	hintergrund
DELETE:	leeres_blatt

zeichne(X)	
PREC:	—
ADD:	gezeichnet(X)
DELETE:	leeres_blatt

2-a Zeichnen Sie einen partiell geordneten Plan, der das oben beschriebene Bild erstellen würde.

2-b Definieren Sie STRIPS-Operatoren für die **Start** und die **Finish** Aktion, die ein partial-order Planner benötigen würde.

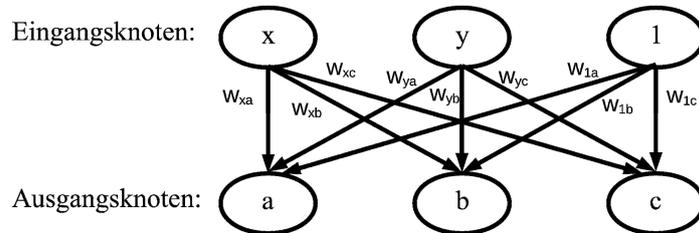
2-c Geben Sie alle kausalen Links an, die für das Finden des Plans aus a) benötigen würden.

2-d Erklären Sie einen Fall, bei dem bei der Erstellung dieses Plans das Einfügen einer neuen Aktion einen Konflikt mit einem der kausalen Links verursachen könnte, der durch Einführen einer Ordnungsrelation aufgelöst werden muß.

2-e Wie viele total geordnete Pläne können Sie aus diesem partiell geordneten Plan ableiten?

Aufgabe 3 Neuronale Netze (7/4/3/4 = 18 Punkte)

Gegeben sei folgende Struktur eines neuronalen Netzes mit den Eingangsknoten x, y und 1 , wobei an 1 konstant 1 anliegt, und den Ausgangsknoten a, b und c .



Das Netzwerk erlaubt als Eingabe- und Ausgabesignale jeweils nur 0 und 1, wobei 0 für logisch *false* und 1 für logisch *true* steht. Die verwendete Aktivierungsfunktion ist

$$g(x) = \begin{cases} 1 & \text{falls } x > 0 \\ 0 & \text{sonst.} \end{cases}$$

3-a Legen Sie die Gewichte $w_{xa}, w_{xb}, w_{xc}, w_{ya}, w_{yb}, w_{yc}, w_{1a}, w_{1b}, w_{1c}$ des neuronalen Netzes so fest, dass der Ausgang

- $out_a=1$ ist, falls die Anzahl der "1" in den Eingängen x und y **kleiner oder gleich 0** ist, sonst $out_a = 0$,
- $out_b=1$ ist, falls die Anzahl der "1" in den Eingängen x und y **kleiner oder gleich 1** ist, sonst $out_b = 0$,
- $out_c=1$ ist, falls die Anzahl der "1" in den Eingängen x und y **kleiner oder gleich 2** ist, sonst $out_c = 0$.

Wählen Sie **ganzzahlige** Gewichte und tragen Sie Ihre Werte in folgende Tabelle ein:

w von \ nach	a	b	c
x			
y			
1			

Vervollständigen Sie außerdem folgende Tabelle um die Eingangssignale in den Knoten a, b, c . Benutzen Sie hierbei für die Berechnung die oben eingetragenen Kantengewichte.

x	y	Eingangssignale			Ausgangssignale		
		in_a	in_b	in_c	out_a	out_b	out_c
0	0				1	1	1
0	1				0	1	1
1	0				0	1	1
1	1				0	0	1

- 3-b Sie möchten nun an einen der Ausgangsknoten die Funktion “*ist genau 1*” berechnen lassen. Lassen sich für die obige Netzwerkstruktur Gewichte wählen, sodass eine Umsetzung der verlangten Funktion möglich ist? Begründen Sie Ihre Antwort bzw. zeigen Sie ggf. eine mögliche Lösung auf.

- 3-c Nehmen Sie für obiges Netzwerk die Aktivierungsfunktion

$$g(x) = 2x$$

und folgende Gewichte an:

w von \ nach	a	b	c
x	-1	1	2
y	-2	1	0
1	0	0	0

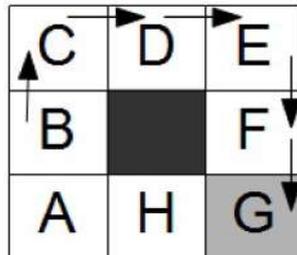
Berechnen Sie mittels Forward-Propagation das Ausgangssignal out_a für das Eingangssignal $(x, y) = (1, 2)$.

- 3-d Nehmen Sie nun ferner an, dass das Netzwerk aus Aufgabe c) für das Eingangssignal $(x, y) = (1, 2)$ die Ausgabe $(a', b', c') = (5, -2, -2)$ liefern soll.

Bestimmen Sie den Fehlerterm Δ_a und berechnen Sie die Gewichtsänderung für das Gewicht w_{xa} für eine Lernrate $\alpha = \frac{1}{2}$.

Aufgabe 4 Reinforcement Learning (3/3/4/5/3=18 Punkte)

Gegeben sei folgende Umgebung mit 9 Feldern:



Der Agent hat in jedem Feld die Aktionen $\{\text{left, right, up, down}\}$ zur Verfügung, die ihn (deterministisch) auf das linke, rechte, obere bzw. untere Nachbarfeld führen, sofern dieses vorhanden ist. Falls das entsprechende Feld nicht existiert, hat die Aktion keine Auswirkung auf den aktuellen Zustand. Es kann also jede Aktion ausgeführt werden, aber für Übergänge in nicht existierende Felder gilt $s' = s$. Das schwarze Feld in der Mitte kann nicht betreten werden

Im grauen Zielfeld G gibt es keine weiteren Aktionen.

Für jede Aktion erhält der Agent einen (negativen) Reward von -1 außer für Aktionen, die ihn auf das Zielfeld führen. In diesem Fall erhält er einen Reward von 0 .

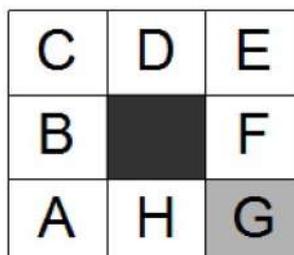
Weiterhin sei eine (partielle) Policy π gegeben, die durch die Pfeile in den jeweiligen Feldern angegeben wird.

Der Discount Faktor (γ) beträgt $\frac{1}{2}$.

4-a Geben Sie für jedes Feld den Wert der Value Function (V^π) für die gegebenen Policy an. Felder, für die keine Policy definiert ist, können Sie ignorieren.

4-b Geben Sie nun die Werte der Q -Function für alle Aktionen des Zustands D an.

4-c Überlegen Sie sich nun eine optimale Policy π^* und Zeichnen Sie diese in das unten stehende Diagramm ein. Wie sieht die dazugehörigen Werte der Value Function (V^*) aus ?



- 4-d Im Folgenden soll ein Roboter-Agent mit Hilfe von Value Iteration eine Policy für die Umgebung lernen. Auf Grund eines bekanntem Defektes wird aber jede Aktion mit einer Wahrscheinlichkeit von 50% ignoriert, d.h. der Roboter verweilt in seinem Zustand. Es gilt wiederum, dass der Roboter bei einer Kollision mit einer Wand auch im selben Zustand verweilt. Modifizieren Sie die Formel für Value Iteration

$$\hat{V}_{i+1}(s) = \max_a (r(s, a) + \gamma \cdot \hat{V}_i(s'))$$

so dass diese stochastische Übergänge $p(s'|a, s)$ berücksichtigen kann. Des Weiteren soll eine Lernrate α verwendet werden können.

Geben Sie ausserdem die Elemente an, über die maximiert wird, also für jedes a den entsprechenden Ausdruck innerhalb des \max_a Operators in ihrer modifizierten Formel. Gegeben sind dabei $s = F$ und die in der Aufgabenstellung spezifizierten Werte für $r(s, a)$. Konkrete Werte für $\hat{V}_i(s)$ sind nicht bekannt, diese müssen also nicht durch Zahlenwerte ersetzt werden.

- 4-e Was bewirkt der Discount-Faktor γ in der Formalisierung eines Reinforcement Learning Problems?

Aufgabe 5 Verschiedenes (4/4/3/6 = 17 Punkte)

5-a Gegeben seien zwei zulässige Heuristiken h_1 und h_2 . Sie bilden nun eine dritte Heuristik

$$h_3 = \alpha_1 \cdot h_1 + \alpha_2 \cdot h_2,$$

wobei $\alpha_1 + \alpha_2 = 1$ und $\alpha_1, \alpha_2 > 0$. Ist h_3 ebenfalls notwendigerweise zulässig? Begründen Sie Ihre Antwort.

5-b Erklären sie kurz in Worten die Grundidee von *Rejection Sampling* zur approximativen Inferenz in Bayes'schen Netzen. Welches Problem dieses Verfahrens wird durch *Likelihood Weighting* gelöst?

5-c Wie funktioniert der *Cross-Over* Operator in einem genetischen Algorithmus?

5-d Erklären Sie die "Physical Symbol Hypothesis". In welcher Form wird diese Hypothese in Searle's Chinese Room Argument abgebildet?