

T1	T2	$\sum T_i$	UE	Σ	N

Prüfung (Exam)
 VU Einführung in die Künstliche Intelligenz, 184.735, 2023S
 26.09.2023

Name: **Big Schnozer**
 Matrikelnummer (Student ID):
 Kennzahl (Study Code):

Bitte leserlich mit Füllfeder oder Kugelschreiber schreiben. *Kein Bleistift!*
 (Please give readable answers and use a fountain or ball pen. *No lead pencil!*)

Für die Multiple-Choice Fragen: Jede richtige Antwort zählt positiv, jede falsche Antwort negativ!
 (Multiple-Choice Questions: Correct answers give positive points, but wrong answers give negative points!)

Teil (Part) 1

40 Punkte (points)

a) Modellieren Sie ein neuronales Netz mit maximal einem inneren Layer, welches zwei binäre Eingangssignale I_1, I_2 sowie drei Ausgangssignale O_1, O_2, O_3 besitzt und dessen Signale sich wie in der folgenden Wahrheitstabelle verhalten.

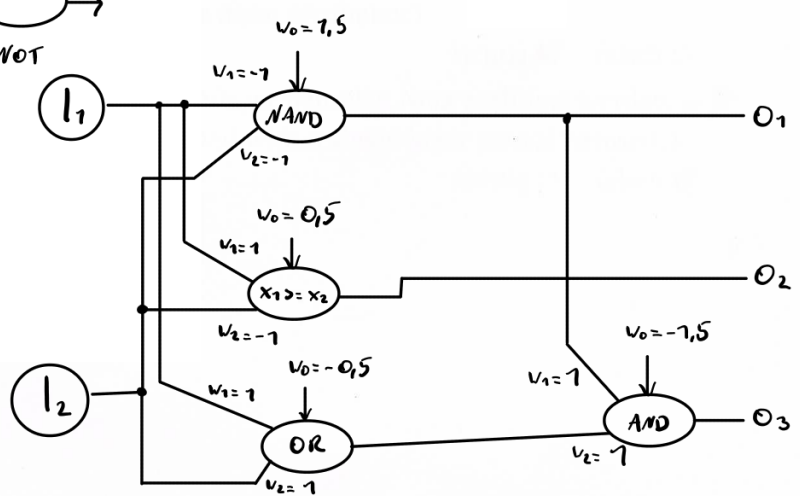
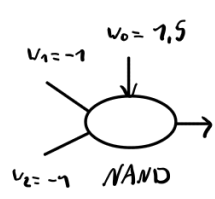
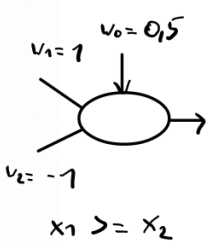
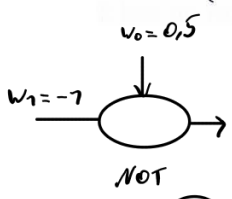
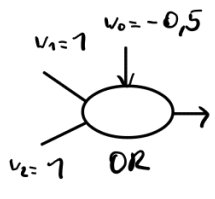
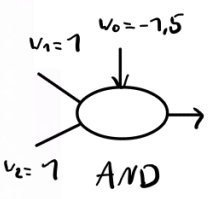
(Construct a neural network with at most one inner layer, which has two binary input signals I_1, I_2 and three output signals O_1, O_2, O_3 . The signals should behave according to the entries of the following truth table.)

I_1	I_2	NAND	\geq	xOR
I_1	I_2	O_1	O_2	O_3
0	0	1	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	1	0

Verwenden Sie für jedes Neuron folgende Aktivierungsfunktion:
 (For each neuron, use the following activation function:)

$$g(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

6 Punkte (points)



X NOT -> see appendix 1

b) Nennen Sie die 4 Komponenten der PEAS Description eines Agents. Geben Sie ein Beispiel für jede Komponente.

(Name the 4 components of the PEAS description of an agent. Give an example for each component.)

4 Punkte (points)

Performance measure - profit, number of deliveries made

Environment - streets

Actuators - steering system

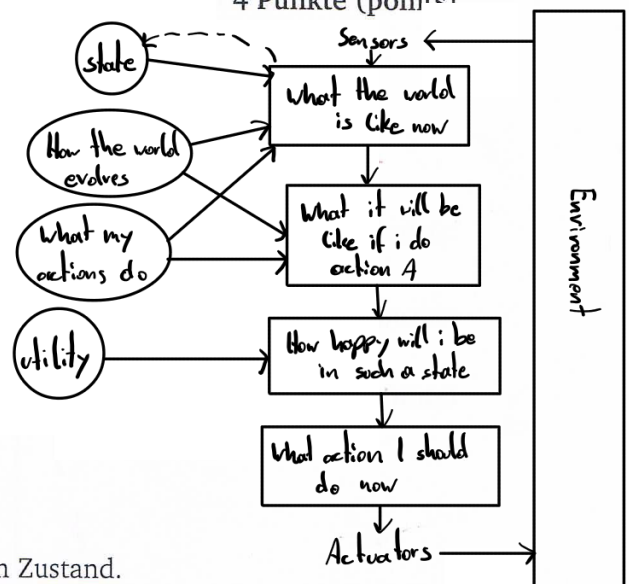
Sensor - GPS, video

c) Erklären Sie das Konzept eines nutzenbasierten Agenten und zeichnen Sie ein Diagramm, welches den Aufbau des Agenten verdeutlicht.

(Explain the concept of a utility-based agent and draw a diagram which explains the structure of the agent.)

4 Punkte (points)

Like a goal-based agent, but also adds utility measurement.
Uses utility function to assess goals. => Acts not only on goals, but also on the best way to achieve a goal.



d) Kreuzen Sie Zutreffendes an:

(Check the correct answers:)

(i) Ein Recurrent Neural Network hat keinen internen Zustand.

(A Recurrent Neural Network has no internal state.)

richtig falsch

(ii) Entscheidungsbäume können jede beliebige Funktion aus den Eingabeattributen darstellen.

(Decision trees can express any function of the input attributes.)

richtig falsch

(iii) Die logische Äquivalenz kann mit einem single-layer neuronalen Netz realisiert werden.

(The logical equivalence function can be realized with a single-layer neural network.)

richtig falsch

RNN:

bi-directional networks.

Delay of input, via internal state

(iv) Es wird angenommen, dass rationale Agenten allwissend sind.

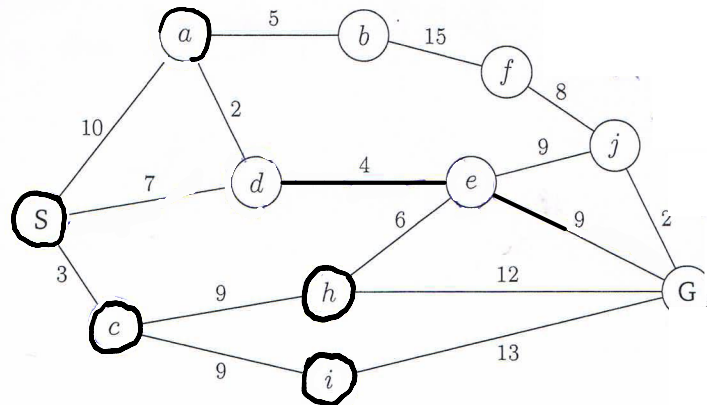
(It is assumed that rational agents are omniscient.)

richtig falsch

6 Punkte (points)

e) Gegeben sei folgendes Suchproblem mit Startknoten S und Zielknoten G .

(Consider the following search problem with start node S and goal node G .)



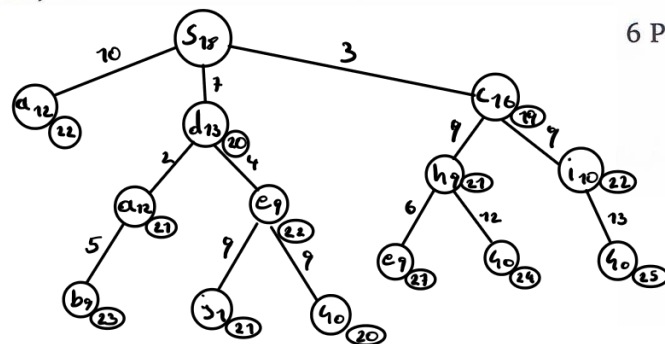
(Lösen Sie das Suchproblem mittels A^* -Suche. Nutzen Sie folgende Heuristikfunktion:)

Solve the search problem using A^* search. Use the following heuristic function:

$$h(S) = 18, h(a) = 12, h(b) = 9, h(c) = 16, h(d) = 13, \\ h(e) = 9, h(f) = 8, h(h) = 9, h(i) = 10, h(j) = 1, h(G) = 0.$$

Geben Sie die finale Lösung und ihre Kosten an. Geben Sie außerdem den Wert der *Evaluierungsfunktion* für jeden expandierten Knoten an.

(Provide the final solution and its cost. In addition, give the value of the *evaluation function* for each expanded node.)



6 Punkte (points)

path = $S \rightarrow d \rightarrow e \rightarrow g$

f) Wann dominiert eine zulässige Heuristik h_1 eine andere zulässige Heuristik h_2 ? Angenommen, h_1 ist konsistent; ist dann h_2 auch immer konsistent? Begründen Sie Ihre Antwort.

(When does an admissible heuristics h_1 dominate another admissible heuristics h_2 . Assuming that h_1 is consistent, is then h_2 also always consistent? Justify your answer.)

4 Punkte (points)

• h_1 dominates h_2 , if for every node $h_1(n) \geq h_2(n)$

• No the dominating aspect, does not entail consistency.

Counter example:

$h_1(n) = 10$	$h_1(n) \leq c(n, n') + h_1(n')$
$h_2(n) = 9$	$10 \leq 1 + 9$
	$10 \leq 10 \checkmark$
$h_1(n') = 9$	
$h_2(n') = 1$	$h_2(n) \leq c(n, n') + h_2(n')$
$c(n, n') = 1$	$9 \leq 1 + 1$
	$9 \leq 2 \quad \times$

g) Erklären Sie die Begriffe *Completeness*, *Optimality*, *Time complexity*, und *Space complexity* im Kontext von Suchalgorithmen.

(Explain the terms *Completeness*, *Optimality*, *Time complexity*, and *Space complexity* in the context of search algorithms.)

4 Punkte (points)

Completeness:

If the algorithm finds a solution (should one exist).

Optimality:

If the algorithm finds the best (lowest-cost) solution.

Time Complexity:

Number of nodes generated/expanded.

Space Complexity:

Maximum number of nodes in memory.

h) Kreuzen Sie Zutreffendes an:

(Check the correct answers:)

DFS is never
optimal

- (i) Depth-first search ist optimal wenn der Suchbaum endlich ist
(Depth-first search is optimal if the depth of the search tree is finite.)
richtig falsch
- (ii) *Random-restart hill climbing* wird bei genügend Zeit immer das globale Optimum finden.
(*Random-restart hill climbing* will always find the global optimum given enough time.)
richtig falsch
- (iii) Für *konsistente* Heuristikfunktionen h ist der A^* -Algorithmus optimal. (For *consistent* heuristic functions h the A^* -Algorithm is optimal.)
richtig falsch
- (iv) Wenn die Kosten jeder Kante gleich sind, ist Breadth-first search optimal.
(If the costs of every edge are identical, Breadth-first search is optimal.)
richtig falsch

6 Punkte (points)

Teil (Part) 2

40 Punkte (points)

- a) Betrachten Sie das folgende kryptoarithmetische Puzzle, wobei jedem Buchstaben eine Ziffer derart zugeordnet werden soll, sodass je zwei Buchstaben nicht dieselbe Ziffer besitzen:

(Consider the following cryptarithmic puzzle in which every letter is assigned a digit such that no two letters have the same digit:)

$$\begin{array}{r}
 S U N \\
 + S K I N \\
 \hline
 B U R N
 \end{array}$$

Beschreiben Sie das Puzzle als ein *Constraint-Satisfaction Problem* (CSP), d.h. geben Sie die Menge der Variablen, die Constraints und die Domänen jeder Variablen an.

(Describe the puzzle as a *constraint-satisfaction problem* (CSP), i.e., describe the set of variables, the constraints, and the domains of each variable.)

8 Punkte (points)

Variables = { S, K, I, N, U, B, R }

Domains = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }

Constraints:

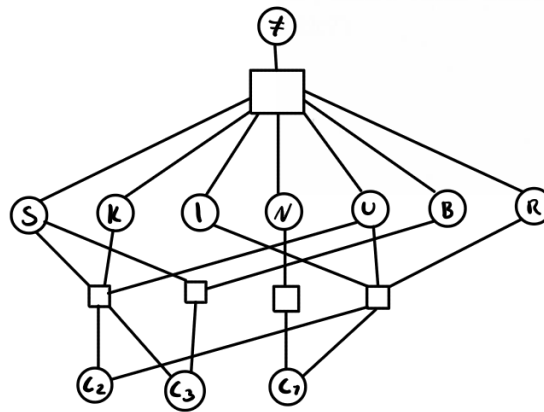
$\neq \text{AllDiff}(S, K, I, N, U)$

$N + N = N + 10 \cdot C_1$

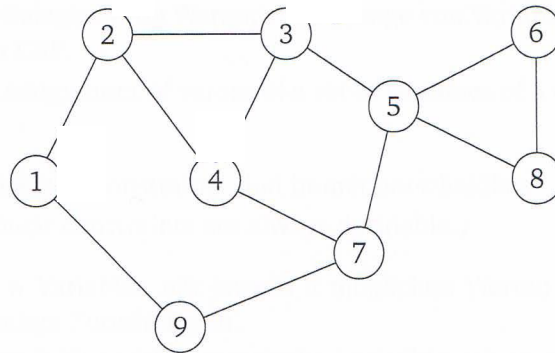
$C_1 + U + I = R + 10 \cdot C_2$

$C_2 + S + K = U + 10 \cdot C_3$

$C_3 + S = B$



- b) Wir betrachten das Dreifärbbarkeitsproblem für folgenden Graphen:
 (Consider the three-colorability problem for the following graph:)



- (i) Ausgehend von der partiellen Zuordnung "3 = grün, 4 = blau", welchen Wert würde der Variablen 5 durch die *Least-Constraining-Value Heuristik* zugeordnet werden?
 (Based on the partial assignment "3 = green, 4 = blue", which value would be assigned to the variable 5 by the *least-constraining-value heuristic*?)

5 - Blue, allows for:
 6 = { Red, Green }
 7 = { Red, Green }
 8 = { Red, Green }

5 - Red, allows for:
 6 = { Blue, Green }
 7 = { Green }
 8 = { Blue, Green }

We choose 5 = Blue

- (ii) Betrachten Sie die partielle Zuordnung "1 = blau, 2 = rot, 3 = grün". Welche Variable würde durch die *Minimum-Remaining-Values Heuristik* als nächstes ausgewählt werden?
 (Consider the partial assignment "1 = blue, 2 = red, 3 = green". Which variable would be assigned next using the *minimum-remaining-values heuristic*?)

4, because it can only be blue.

- (iii) Welche Variable würde durch die *Degree Heuristik* als erstes einen Wert aus ihrer Domain zugeordnet bekommen?
 (Which variable would be assigned a value from its domain first by the *degree heuristic*?)

Node 5, because it has the most constraints
 (4 constraints, $\text{deg}(5) = 4$)

6 Punkte (points)

c) Kreuzen Sie Zutreffendes an:

(Check the correct answers:)

(i) Jede konsistente Belegung von Werten einer Menge von Variablen in einem CSP ist eine Lösung für dieses CSP.

(Each consistent assignment of values of a set of variables of a CSP is a solution of that CSP.) richtig falsch

(ii) CSPs mit nicht-linearen Constraints sind immer entscheidbar.

(CSPs with nonlinear constraints are always decidable.) richtig falsch

(iii) Für ein CSP mit n Variablen mit jeweils d möglichen Werten gibt es maximal $O(d^n)$ mögliche vollständige Zuordnungen.

(For a CSP in n variables with respectively d possible values, there is a total of $O(d^n)$ possible complete assignments.)

richtig falsch

(i) CSP solution = complete + consistent assignment

6 Punkte (points)

consistent assignment = does not violate any constraints complete = Mentions every variable

d) Kreuzen Sie Zutreffendes an:

(Check the correct answers:)

(i) Die Action Description Language (ADL) erlaubt nur positive literale in States.

(The Action Description Language (ADL) allows only positive literals in states.)

richtig falsch

(ii) Kommt in ADL ein Literal in einem Zustand nicht vor, so wird dieses als falsch angenommen.

(If a literal does not occur in a state in ADL, then it is assumed to be false.)

richtig falsch

(iii) Der POP-Algorithmus erzeugt nur konsistente Pläne.

(The POP-Algorithm only produces consistent plans.)

richtig falsch

(iv) In STRIPS wird angenommen, dass im Effekt nicht erwähnte Literale unverändert bleiben.

(In STRIPS it is assumed that literals not mentioned in the effect remain unchanged.)

richtig falsch

8 Punkte (points)

(i) STRIPS allows only positive literals
ADL allows positive and negative.

(ii) STRIPS: Unmentioned literals are false
ADL: Unmentioned literals are unknown

(iii) POP = Partial-order Plans

consistent plan:

- No cycles in the ordering constraints
- no conflicts with the causal links

Algorithm only generates consistent plans! Goal test only checks if there are open preconditions

e) Formalisieren Sie in *STRIPS Syntax* eine Aktion *Copy* um eine Datei von einer Festplatte auf eine andere Festplatte zu kopieren.

- Vorbedingung für diese Aktion ist, dass sich die Datei auf der Festplatte befindet und dass die Zielfestplatte leer ist.
- Der Effekt ist, dass sich die Datei auf beiden Festplatten befindet. Weiters ist die Zielfestplatte nach dem Vorgang natürlich nicht mehr leer.
- Die Datei soll der erste, die Quellfestplatte der zweite und die Zielfestplatte der dritte Parameter der Aktion sein.
- Die Parameter sollen in der Vorbedingung entsprechend ihrem Typus geprüft werden.

Verwenden Sie für die Modellierung die Prädikate *HD*, *File*, *Empty* und *On*. Dabei bedeutet *HD(x)*, dass *x* eine Festplatte ist, *On(x,y)*, dass sich *x* auf *y* befindet, *File(x)*, dass *x* eine Datei ist, und *Empty(x)*, dass *x* leer ist.

(Formalize in STRIPS syntax an action *Copy* expressing that a file is copied from one hard disk to another.

- The precondition for this action is that the file on the hard disk must exist and that the target hard disk must be empty.
- The effect is that the file is present on both hard disks. Furthermore, the target hard disk is then non-empty.
- The file should be the first, the source hard disk the second, and the target hard disk the third parameter of this action.
- The parameters should be checked in the precondition according to their type.

Use the predicates *HD*, *File*, *Empty*, and *On* for your formalization. Thereby, *HD(x)* means that *x* is a hard disk, *On(x,y)* means that *x* is present on *y*, *File(x)* means that *x* is a file, and *Empty(x)* means that *x* is empty.)

6 Punkte (points)

Action: $Copy(file, from, to)$

Precondition: $HD(from) \wedge HD(to) \wedge File(file) \wedge$
 $On(file, from) \wedge Empty(to)$

Effect: $On(file, disk2) \wedge \neg Empty(disk2)$

f) Definieren Sie den Begriff einer *Lotterie* in der Entscheidungstheorie.

(Define the notion of a *lottery* in decision theory.)

3 Punkte (points)

A Lottery L is a set of outcomes such that:

• Outcomes S_1, S_2, \dots, S_n can occur with probability p_1, p_2, \dots, p_n

• It can be written as $L = [p_1, S_1; p_2, S_2; \dots; p_n, S_n]$

• Each S_i is either an atomic state or another lottery

g) Geben Sie die Definition des Axioms der *Monotonie* mittels Angabe der entsprechenden Formel.

(Give the definition of the axiom of *monotonicity* by providing its formula.)

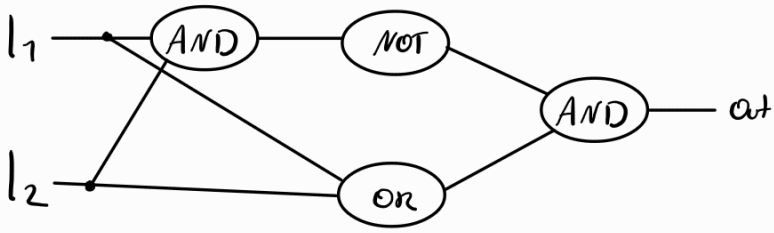
3 Punkte (points)

If two lotteries have the same possible outcomes A and B .

If an Agent prefers A to B , then the agent must prefer precisely the lottery that has a higher probability for outcome A .

$$A \succ B \Rightarrow (p > q \Leftrightarrow [p, A; 1-p, B] \succ [q, A; 1-q, B])$$

1a



XOR can be built like this