

# Exam 2 *Introduction to Artificial Intelligence*, WS 14/15, U Stuttgart

Prof. Dr. Marc Toussaint

September 11, 2015

Name:

Matrikelnummer:

**DO NOT OPEN THE EXAM BEFORE THE  
ANNOUNCEMENT!**

**DIE KLAUSUR NICHT VOR DER ANSAGE ÖFFNEN!**

- There are 45 points in total. The 7 questions are with points: 6, 3, 3, 6, 12, 9, 6.
- You have 60 minutes.
- Write your name on all sheets.
- Put your *Studentenausweis* next to you, so we can check it during the exam.
- **You may only use your pen and scratch paper – no other materials (no textbooks, script, or mobiles) are allowed.**
- **Write all your answers in the boxes!** By default we will not grade or correct anything outside the boxes. The free space and backside of pages is for your use as scrap paper.

- 
- Es gibt insgesamt 45 Punkte. Die Punkte der 7 Fragen sind: 6, 3, 3, 6, 12, 9, 6.
  - Sie haben 60 Minuten.
  - Schreiben Sie Ihren Namen auf jedes Blatt der Klausur.
  - Legen Sie Ihren Studentenausweis neben sich, damit wir ihn während der Klausur einsehen können.
  - **Sie dürfen ausschließlich Kugelschreiber/Tintenschreiber nutzen. Andere Hilfsmittel, wie Bücher, Skript oder Mobiltelefone/elektronische Geräte sind nicht erlaubt.**
  - **Schreiben Sie alle Antworten in die vorgesehenen Boxen!** Alles außerhalb der Boxen wird, im Normalfall, nicht korrigiert oder beachtet. Der freie Raum und die Rückseite der Seiten dient Ihnen als Schmierpapier.

**Question 1 — Tree search (6Pts)**

Assuming a tree search with

- maximum branching factor  $b$  of the search tree,
- depth  $d$  of the least-cost solution,
- maximum depth  $m$  of the state space.

What is the *time* complexity of the following tree search methods (neglecting a +1 or -1 in the expressions is ok):

- breadth first search?
- depth first search?
- iterative deepening search?

Which of the three is *complete* (i.e., finds a solution). Answer [Y/N].

- breadth first search?
- depth first search?
- iterative deepening search?

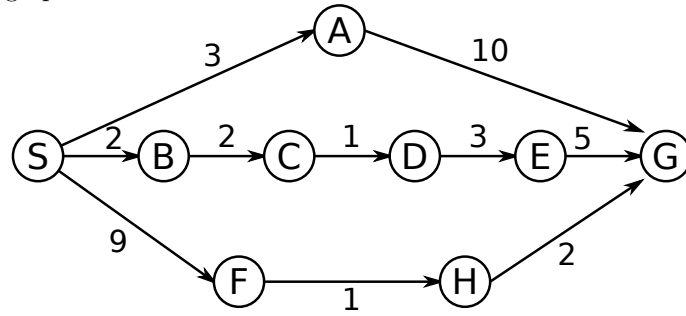
---

Wir betrachten eine Baumsuche mit maximalem Verzweigungsgrad  $b$ , Tiefe  $d$  der optimalen Lösung, und maximaler Tiefe  $m$  des Zustandsraums. Was ist die Speicherkomplexität von a) Breitensuche, b) Tiefensuche, c) iterativer Tiefensuche.

Welche der drei Folgenden ist optimal (d.h., berechnet die optimale Lösung unter der Annahme, dass jeder Schritt die Kosten 1 hat.) d) Breitensuche, e) Tiefensuche, f) iterative Tiefensuche.

**Question 5 — A\* (12Pts)**

Consider the following state graph:



Edges are labeled with the cost  $c(s, s')$  for a state transition; the start state is  $S$ ; the goal state is  $G$ . Assume a *constant* heuristic  $\forall s \neq G : h(s) = 1$  with value 1 for all states except for the goal state, where  $h(G) = 0$ .

Provide below the *priority queue* in every iteration of A\*. (More precisely: write the state of the queue *before* the goal test and expand operation.) In each box, use the notation “ $X:g+h$ ” for an entry of the queue, where  $X$  denotes the state,  $g$  the cost-so-far, and  $h$  the heuristic. Left entries should have higher priority (lower  $g + h$ ).

Step 0:	S:0+1		
Step 1:			
Step 2:			
Step 3:			
Step 4:			
Step 5:			
Step 6:			
Step 7:			
Step 8:			

In step 7, an  $H$ -node is inserted into the priority queue *before* the previously inserted, non-optimal  $G$ -node. Provide equations and inequalities which prove  $f(H) \leq f(G_{\text{non-opt}})$  for an admissible heuristic, where  $G_{\text{non-opt}}$  is such a non-optimal  $G$ -node.

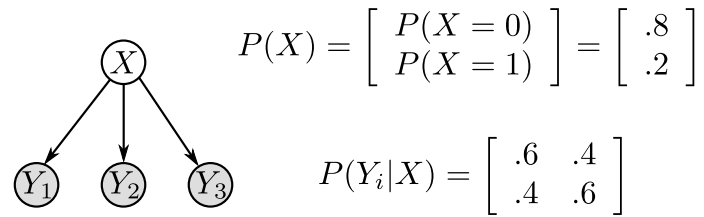
Im gegebenen Zustandsgraph sind an den Kanten die Kosten  $c(s, s')$  für einen Übergang zwischen zwei Zuständen gegeben.  $S$  ist der Start-Zustand;  $G$  der Ziel-Zustand. Die Heuristik ist für alle Zustände konstant 1, außer für den Ziel-Zustand  $G$ , wo sie 0 ist:  $\forall s \neq G : h(s) = 1, h(G) = 0$ .

Schreiben Sie *priority queue* in jeder Iteration von A\* nieder. Genauer: Schreiben Sie den Zustand der Queue *vor* der *goal test* und *expand*-Operation nieder. Verwenden Sie für einen Eintrag in der Queue die Notation “ $X:g+h$ ”, wobei  $X$  der Name des Zustands,  $g$  die cost-so-far, und  $h$  die Heuristik bezeichnet. Links sollten Einträge höherer Priorität stehen.

In Schritt 7 wird ein  $H$ -Knoten “*vor*” den bereits gefundenen nicht-optimalen  $G$ -Knoten in die *priority queue* eingegliedert. Schreiben Sie Gleichungen und Ungleichungen die beweisen, dass  $f(H) \leq f(G_{\text{non-opt}})$ ; wobei  $G_{\text{non-opt}}$  ein solcher nicht-optimaler  $G$ -Knoten ist und die Heuristik *admissible* ist.

**Question 7 — Inference for Naive Bayes (6Pts)**

Given the graphical model

where  $Y_1 = 1$ ,  $Y_2 = 1$ , and  $Y_3 = 0$  are observed/fixed. Compute

$$P(X | Y_{1,2,3} = (1, 1, 0)) \propto$$

The expression may neglect the normalization factor.

---

In dem gegebenen Graphischen Modell sind  $Y_1 = 1$ ,  $Y_2 = 0$ , and  $Y_3 = 1$  beobachtet/fixiert. Berechnen Sie  $P(X | Y_{1,2,3} = (1, 0, 1))$ ; Sie dürfen die Normalisierungskonstante vernachlässigen.