

# Grundlagen der Künstlichen Intelligenz (CS 205)

Prof. Dr. M. Helmert  
G. Röger  
Frühjahrssemester 2012

Universität Basel  
Fachbereich Informatik

## Übungsblatt 3

**Abgabe: 5. April 2013**

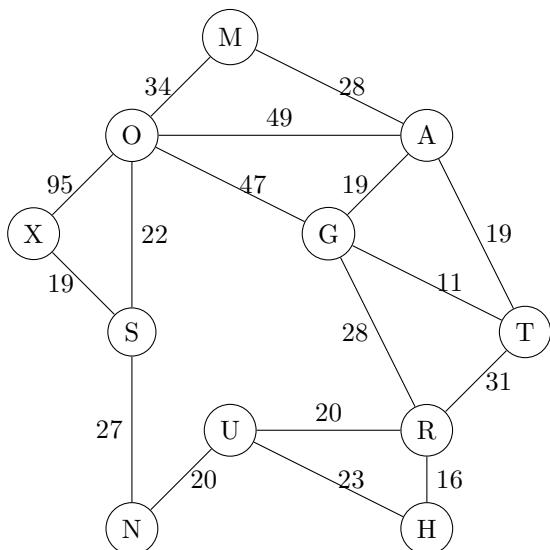
### Aufgabe 3.1 (1+1 Punkte)

Zeigen Sie:

- Breitensuche mit dem Zieltest, wie er bei der allgemeinen Baum- und Graphensuche eingeführt wurde, ist ein Spezialfall von uniformer Kostensuche.
- Uniforme Kostensuche ist ein Spezialfall von A\*.

### Aufgabe 3.2 (4 Punkte)

Betrachten Sie die folgende Strassenkarte:



Die Luftliniendistanzen zwischen Atlantis (A) und den anderen Städten sind in der folgenden Tabelle gegeben:

Stadt	Distanz
G	14
H	50
M	25
N	36
O	42
R	38
S	42
T	14
U	51
X	47

Zeichnen Sie den Suchbaum, der vom A\*-Algorithmus für die Suche eines kürzesten Weges von Hel (H) nach Atlantis (A) erzeugt wird, wenn als Heuristik die Luftliniendistanz nach Atlantis (A) zum Einsatz kommt. Geben Sie an, in welcher Reihenfolge die Knoten expandiert werden, und annotieren Sie jeden Knoten mit seinen f-, g-, und h-Werten.

### Aufgabe 3.3 (2+2 Punkte)

Im Rahmen dieser Aufgabe sollen Sie eine A\*-Suche für das 8-Puzzle aus Aufgabe 2.2 von Blatt 2 implementieren. In der Programmeingabe wird der Anfangszustand angegeben, indem man für jedes der acht Kacheln (der Reihe nach) seine Position angibt, wobei die Positionen folgendermassen nummeriert sind:

0	1	2
3	4	5
6	7	8

Die Eingabe 0 1 2 7 5 8 3 6 beschreibt beispielsweise folgenden Zustand:

1	2	3
7		5
8	4	6

Der einzige Zielzustand ist wie in Aufgabe 2.2 der Zustand

1	2	3
4	5	6
7	8	

Sie können Ihre Implementierung auf Ihre Lösung von Blatt 1 aufbauen. Alternativ stellen wir Ihnen auch eine Basis in Java zur Verfügung, in der bereits der Suchraum des 8-Puzzles implementiert ist.

- (a) Implementieren Sie die Manhattan-Distanz-Heuristik:

Die Manhattan-Distanz eines Zustands  $s$  ist definiert als die Summe der Abstände der Kacheln in  $s$  zur Zielposition. Der Abstand einer Kachel  $1, \dots, 8$  zur Zielposition ist wiederum definiert durch die Summe der Anzahl der waagrechten und senkrechten Zellen, die sie von der Zielposition entfernt ist. Beispiel: Der Abstand der Kachel 4 in obigen Beispielzustand zur Zielposition ist 2, da sie eine waagrechte plus eine senkrechte Zelle von der Zielposition entfernt ist. Die Manhattan-Distanz des Zustands ist  $0 + 0 + 0 + 2 + 1 + 1 + 1 + 1 = 6$ .

- (b) Implementieren Sie den A\*-Algorithmus und testen Sie ihn mit der Manhattan-Distanz-Heuristik. Da diese Heuristik konsistent ist, reicht es, wenn Sie die Variante ohne Reopening umsetzen.

*Hinweise für den Fall, dass Sie auf unseren Code aufbauen:*

- Es handelt sich bei dem Package um einen Eclipse-Export. Vermutlich können Sie den Code dort also einfach importieren, müssen aber eventuell noch das JRE anpassen.
- Für die Implementierung der Heuristik müssen Sie die Repräsentation der 8-Puzzle-Zustände verstehen. Jeder Zustand hat ein Array `positions` der Länge 9, wobei an der  $i$ -ten Position die Nummer der Kachel an Position bzw. 0 für das leere Feld eingetragen ist.
- Sie dürfen die Klasse `Node` um ein Attribut für den Heuristikwert und geeignete Methoden erweitern.

*Die Übungsblätter dürfen in Gruppen von zwei Studierenden bearbeitet werden. Bitte schreiben Sie beide Namen auf Ihre Lösung.*