**Grundlagen der Künstlichen Intelligenz**

Prof. Dr. M. Helmert
Dr. M. Wehrle, S. Sievers
Frühjahrssemester 2015

Universität Basel
Fachbereich Informatik

# Übungsblatt 4

**Abgabe: 27. März 2015**

**Aufgabe 4.1** (1.5+1.5+1 Punkte)

Betrachten Sie das folgende Suchproblem auf einem quadratischen Grid mit einer Länge und Breite von jeweils $2n + 1$ Zellen ($n \in \mathbb{N}$). Ein Agent befindet sich im Anfangszustand $S$ auf der Position mit den Koordinaten $(x, y) = (n + 1, n + 1)$ und hat das Ziel, Position $G$ mit den Koordinaten $(x, y) = (n + 1, 1)$ zu erreichen. Hierzu hat der Agent Aktionen zur Verfügung, um sich jeweils eine Zelle nach oben, unten, rechts oder links bewegen, wenn eine solche Nachbarzelle existiert (auf Randpositionen sind entsprechende Bewegungen verboten). Um das Ziel zu erreichen soll Breitensuche betrachtet werden.

| | 1 | ... | n | n+1 | | | |
|---|---|---|---|---|---|---|---|
| 1 | | | | G | | | |
| ... | | | | | | | |
| n | | | | | | | |
| n+1 | | | | S | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

(a) Wie viele Suchknoten muss Breitensuche *ohne* Duplikateliminierung (Foliensatz 10, Folie 21 in der Druckversion der Folien) hierzu mindestens in die Open-Liste einfügen, um das Ziel $G$ ausgehend von $S$ zu erreichen? Geben Sie die Anzahl in Abhängigkeit von $n$ an und begründen Sie Ihre Antwort.

(b) Wie viele Suchknoten muss Breitensuche *mit* Duplikateliminierung (Foliensatz 10, Folie 25 in der Druckversion der Folien) mindestens in die Open-Liste einfügen, um das Ziel $G$ ausgehend von $S$ zu erreichen? Geben Sie die Anzahl in Abhängigkeit von $n$ an und begründen Sie Ihre Antwort.

(c) Vergleichen Sie die Anzahl der Suchknoten in der letzten komplett erzeugten Suchebene für Grids mit $n = 10$ und $n = 20$. Diskutieren Sie das Ergebnis.

**Aufgabe 4.2** (2+1+1 Punkte)

Betrachten Sie das 8-Puzzle (analog zum 15-Puzzle aus der Vorlesung) im folgenden Anfangszustand `init`.

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | |
| 7 | 8 | 6 |

Wie beim 15-Puzzle seien die möglichen Aktionen folgendermassen modelliert: Das leere Feld kann nach oben (*up*), links (*left*), rechts (*right*) und unten (*down*) verschoben werden, wenn es sich nicht bereits am entsprechenden Rand befindet.
Der einzige Zielzustand sei folgendermassen gegeben.

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | |

(a) Wenden Sie tiefenbeschränkte Suche mit einer maximalen Tiefe von 3 auf den Anfangszustand an (d.h., simulieren Sie den Aufruf `depth_limited_search(init(), 3)`). Geben Sie hierzu den Suchbaum und die Expansionsreihenfolge der Knoten an. Die Reihenfolge, in der Nachfolgezustände berechnet werden, sei durch *up* < *left* < *right* < *down* gegeben (d.h., für einen gegebenen Zustand wird stets zuerst dessen Nachfolger bzgl. *up* und als letztes dessen Nachfolger bzgl. *down* von `succ` berechnet).

(b) Was passiert, wenn keine Tiefenbeschränkung gegeben ist? Begründen Sie Ihre Antwort.

(c) Nehmen Sie an, die Reihenfolge der Aktionen sei nun im Vergleich zu a) invertiert, d.h. *down* < *right* < *left* < *up*. Simulieren Sie für diese Reihenfolge noch einmal den Aufruf `depth_limited_search(init(), 3)`. Diskutieren Sie das Ergebnis im Vergleich zu a).

**Aufgabe 4.3** (2+2 Punkte)

Bestimmen Sie für die folgenden Heuristiken, welche der Eigenschaften *sicher*, *zielerkennend*, *zulässig* und *konsistent* auf sie zutrifft. Begründen Sie Ihre Antwort.

(a) Für das "Missionare und Kannibalen"-Problem:

$$h_1(\langle m, c, b \rangle) := \max\{m + c - b, 0\}$$

Zur Erinnerung: Zustände sind als Zahlentripel $\langle m, c, b \rangle \in \{0, 1, 2, 3\} \times \{0, 1, 2, 3\} \times \{0, 1\}$ repräsentiert, wobei $m$ die Anzahl der Missionare, $c$ die Anzahl der Kannibalen und $b$ die Anzahl der Boote am *falschen* Flussufer beschreibt. Beachten Sie, dass das Boot nur maximal 2 Personen aufnehmen kann.

(b) Für das Blocks-World-Problem: Seien $x_1, \ldots, x_n$ die existierenden Blöcke. Dann sei

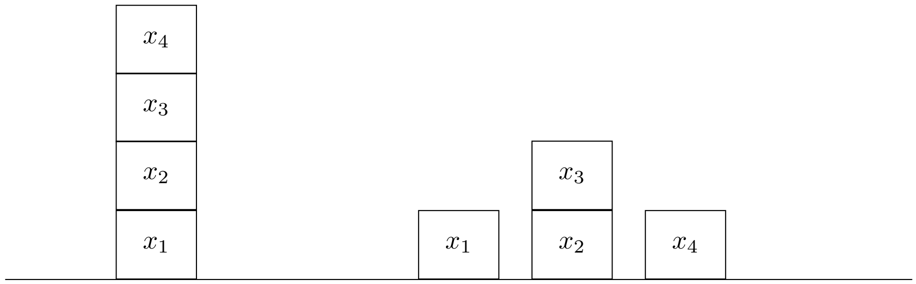
$$h_2(s) := \sum_{i=1}^{n} f(x_i),$$

wobei die Funktion $f$ folgendermassen definiert ist:

$$f(x_i) := \begin{cases} 1 + |\{x_j \mid x_j \text{ liegt irgendwo über } x_i\}|, & \mathit{goalpos}(x_i) \neq \mathit{pos}(s, x_i) \\ 0, & \text{sonst} \end{cases}$$

Der Ausdruck $\mathit{goalpos}(x_i) \neq \mathit{pos}(s, x_i)$ drückt hierbei aus, dass Block $x_i$ im Zustand $s$ auf einem Block $y$ liegt, im Ziel jedoch auf einem anderen Block $z \neq y$ liegen muss (wie in der Vorlesung beschrieben dürfen $y$ und $z$ auch der Tisch sein). Die Heuristik $h_2$ zählt also alle Blöcke, die sich in $s$ noch nicht in der Zielposition befinden, und für jeden dieser Blöcke $x$ zusätzlich die Anzahl der Blöcke, die sich in $s$ im gleichen Stapel irgendwo oberhalb von $x$ befinden. Die Intuition von $h_2$ ist, dass für jeden falsch platzierten Block auch die Blöcke oberhalb dieses Blocks umgeordnet werden müssen.

*Beispiel:*



Der initiale Zustand $s_0$ befindet sich links, der Zielzustand rechts. Der Wert $h_2(s_0) = 0 + 3 + 0 + 1 = 4$: Die Blöcke $x_1$ und $x_3$ liegen bereits auf dem Block, der vom Zielzustand vorgegeben wird, daher $f(x_1) = f(x_3) = 0$. Im Gegensatz dazu liegt $x_2$ fälschlicherweise auf $x_1$ (statt auf dem Tisch), und in $s_0$ liegen 2 Blöcke über $x_2$, daher gilt $f(x_2) = 3$. Analog gilt $f(x_4) = 1$.

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