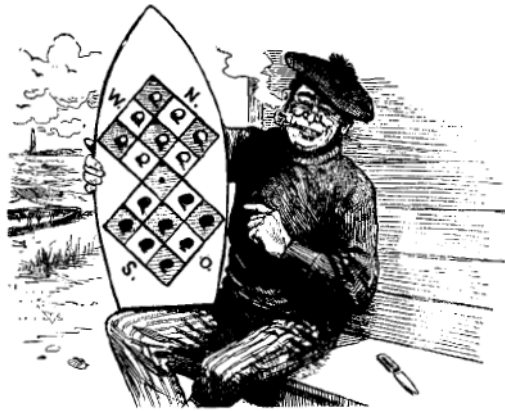


Künstliche Intelligenz (HWS 2016)

Übungsblatt 2 (24 + 5 Punkte)

Christian Meilicke & Manuel Fink
{christian,manuel}@informatik.uni-mannheim.de



Aufgabe 1 (6 Punkte)

Beschreibe folgende Aufgaben als Suchproblem:

- a) Du hast 3 Gefäße, die 12, 8 und 3 Gallonen fassen sowie einen Brunnen mit Wasser. Hiermit sollst du genau eine Gallone Wasser abmessen.
- b) Ein König möchte einen Fluss überqueren und hat einen Blumenkohl, ein Schaf und einen Wolf bei sich. In das Boot passt immer nur einer der Gegenstände und der König, der rudern muss. Natürlich kann der König auch alleine den Fluss überqueren. Solange der König anwesend ist, besteht keine Gefahr für das Schaf und den Blumenkopf. Allerdings frisst der Wolf das Schaf, und das Schaf den Blumenkopf sobald diese alleine gelassen werden (wobei der Wolf schneller als das Schaf ist). Das Ziel besteht darin, alle drei Gegenstände an das andere Ufer zu bringen.

Gib jeweils Anfangszustand, Zielttest, mögliche Aktionen und eine Kostenfunktion präzise genug an, um eine darauf basierende Implementierung zu

ermöglichen. Beschreibe kurz sprachlich, was minimiert werden soll und stelle sicher, dass die angegebene Kostenfunktion hierzu passt. Beispielweise ist in Bezug auf Aufgabenteil a) eine Formulierung wie „Wir versuchen *das Wasser* zu minimieren“ nicht präzise genug; ebenso wenig eine Formulierung wie „den Inhalt eines Gefäßes in ein anderes schütten“, da hieraus nicht hervorgeht, wie viele Gallonen sich danach in dem einen und in dem anderen Gefäß befinden. Dies kann man nur mit Hilfe von Variablen angemessen ausdrücken. Achtung: Es wird nicht nach einer Lösung zu den beiden Problemen gefragt!

Aufgabe 2 (4 Punkte)

Wende eine Breitensuche ohne Duplikateeliminierung auf das in Aufgabe 1 a) beschriebene Problem an! Zeichne hierzu den Suchbaum bis hin zu Ebene 2 (der Startzustand liegt auf Ebene 0)! Zeichne alle möglichen Aktionen ein, inklusive solcher Aktionen, die zu Zuständen führen, die keine Lösung mehr sein können, als auch solche Aktionen, die offensichtlich nichts zum Erreichen einer Lösung beitragen! Markiere alle Zustände, die auf einer früheren Ebene bereits erreicht wurden!

Aufgabe 3 (8 Punkte) [Präsentationsaufgabe]

Das Titelbild zeigt das Spiel Seemanns-Solitär. Bei diesem Spiel sollen die schwarzen und weißen Steine mit möglichst wenigen Zügen die Seiten wechseln. Jeder Stein kann auf ein leeres benachbartes Feld gerückt werden (zwei Felder sind abenachbart, wenn sie eine gemeinsame Seite haben). Außerdem kann ein Stein über einen benachbarten Stein springen, wenn das Zielfeld leer ist, wobei man nicht ums Eck springen kann. Dabei ist die Farbe der Steine egal.

- a) (2 Punkte) Berechne exakt wie viele verschiedene Zustände im Seemanns-Solitär auftreten können.
- b) (2 Punkte) Es gibt eine Lösung des Problems in 47 Zügen. Schätze ab, wie viele Zustände im schlimmsten Fall im Suchbaum *expandiert* werden müssen, wenn man eine Breitensuche anwendet. Vergleiche die geschätzte Zahl mit dem Ergebnis der Berechnung aus Aufgabenteil a). Was bedeutet dies für die zu wählende Suchverfahren?
- c) (2 Punkte) Schildere kurz eine mögliche Implementierung eines Zustandes mit Fokus darauf, dass aktuell mögliche Züge relativ leicht aus einem Zustand bestimmt werden können (d.h. *nicht* durch manuelles Einprogrammieren von 17 Spezialfällen...). Gib hierfür an, wie viel Speicher für einen einzelnen Zustand verwendet wird. Kann man basierend auf dieser Implementierung eine Breitensuche mit Duplikateliminierung durchführen, wenn 4 GB Speicher zur Verfügung stehen?
- d) (2 Punkte) Entwickle und beschreibe zwei möglichst gut geeignete Heuristiken (wobei die eine eine Verbesserung/Erweiterung der anderen sein

kann), die zulässig (admissible) sind und im Rahmen einer A*-Suche eingesetzt werden können! Schätze mit Hilfe dieser Heuristiken jeweils die Kosten, um vom Startzustand zum Zielzustand zu gelangen!

Aufgabe 4 (6 Punkte)

Konstruiere eine Straßenkarte mit den Städten A, B, C, ..., Z! Dabei sollen die Städte als Knoten eingezeichnet werden, wobei die Kanten Straßen zwischen den Städten darstellen. Es geht im folgenden darum, den kürzesten Weg von A nach Z zu finden. Dabei soll als Heuristik die Luftliniendistanz verwendet werden. Entsprechend ist zusätzlich eine Tabelle zu erstellen, in welcher der Luftlinienabstand zwischen den Städten A, ... nach Z vermerkt ist. Im folgenden siehst du ein Beispiel für eine solche Karte und die zugehörige Tabelle.

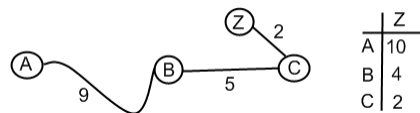


Abbildung 1: Straßenkarte mit Luftlinientabelle

Konstruiere jeweils eine Karte mit Luftlinientabelle, bei der

- die Greedy-Suche schneller eine Lösung findet als die A*-Suche,
- die Greedy-Suche eine suboptimale Lösung findet,
- die A*-Suche aufgrund einer Heuristik, die nicht admissible ist, eine suboptimale Lösung findet (Hierzu muss ein „Fehler“ in die Luftlinientabelle eingebaut werden, der die A*-Suche in die Irre leitet).

Gib hierfür jeweils die Lösungswege an, die durch Greedy-Suche und A*-Suche für die Graphen gefunden werden! Ein bereits besuchter Knoten soll hierbei zu einem späteren Zeitpunkt nicht nochmals betrachtet werden (Duplikateliminiierung).

Bonusaufgabe (5 Punkte)

Implementiere einen Algorithmus, der das Seemanns-Solitär löst und eine optimale Lösung ausgibt! Angabe des erzeugten Lösungsweges in der PDF, Abgabe des Codes in einem Extra-Ordner, der mit hochgeladen werden muss (idealerweise Eclipse Projekt als Archiv exportieren). Korrigiert werden nur Abgaben, die tatsächlich eine korrekte Lösung erzeugen, Teilpunkte können nicht vergeben werden.