

GKI - Hausaufgaben 1

Tao Xu, 343390 - Mitja Richter, 324680 - Björn Kapelle, 320438 - Marcus Weber, 320402

Aufgabe 1

1.a)

Zustandsraum:

$S = (a_1, a_2, a_3, a_4, a_5, z), a_i \in \{0, 1, 10, 100\}$ und $z \in [0, 127]$

wobei a_i für die jeweils zu bearbeitende Aufgabe steht und 0, 1, 10, 100 für unbearbeitet(0), Alfons(1), Bernd(10) und Christine(100) (als bearbeitende Personen). Z stehen für die noch aufzuwendende Bearbeitungszeit von Alfons, Bernd und Christine. Es wird also ein 6-stelliges 7bit-Array benötigt.

Startzustand:

$S_0 = (0, 0, 0, 0, 0, 0)$

Zielzustand:

$S_Z = (a_1^Z, a_2^Z, a_3^Z, a_4^Z, a_5^Z, z^Z)$, mit $a_i^Z \neq 0$ und $z^Z = 0$

Aktionen: Die Auswahl des Studenten der die nächste noch nicht bearbeitete Aufgabe bearbeiten soll. Dargestellt mithilfe der Überföhrungsfunktion $wahl$ mit dem Parameter Student (s).

$wahl(s) : S = (a_1, a_2, a_3, a_4, a_5, z) \rightarrow S' = (a'_1, a'_2, a'_3, a'_4, a'_5, z'), s \in \{0, 1, 10, 100\}$

sodass gilt wenn $s \in \{1, 10, 100\}$ (d.h. wenn ein Student für eine Aufgabe ausgewählt wird):

sei a_j das erste $a_i \neq 0$ für mit $<$ geordneten i und $/$ die Ganzzahldivision

- $\forall a'_i, i \neq j. a'_i = a_i$
- $a'_j = s$ (Belegen der Aufgabe mit Alfons(1), Bernd(10) oder Christine(100))
- $z' = s + z$ (Bilden der neuen Bearbeitungszeit)
- $\frac{z'}{100} \leq 1$ (Christine darf nicht zwei Aufgaben gleichzeitig machen)
- $\frac{(z' \bmod 100)}{10} \leq 2$ (Bernd darf nicht zwei Aufgaben gleichzeitig machen)
- $((z' \bmod 100) \bmod 10) \leq 4$ (Alfons darf nicht zwei Aufgaben gleichzeitig machen)
- $a'_1 \neq 100$ (Christine kann Aufgabe 1 nicht)
- $a'_3 \neq 100 \wedge a'_3 \neq 10$ (Christine und Bernd können Aufgabe 3 nicht)
- $a'_4 \neq 100$ (Christine kann Aufgabe 4 nicht)

und wenn $s = 0$ (d.h. wenn kein neuer Student ausgewählt wird und stattdessen eine Zeiteinheit vergeht):

- $\forall a'_i. a'_i = a_i$
- $z' = z - \text{sgn}(\frac{z}{100}) \cdot 100 - \text{sgn}(\frac{(z \bmod 100)}{10}) \cdot 10 - \text{sgn}((z \bmod 100) \bmod 10)$ (sgn steht hier für Signumfunktion; in diesem Schritt vergeht die Hälfte der durchschnittlichen Bearbeitungszeit)

Pro Zug wird entweder ein Student ausgewählt, der die nächste Aufgabe übernimmt, oder eine Zeiteinheit (in diesem Fall die Hälfte der durchschnittlichen Bearbeitungszeit) vorangeschritten. Damit die zweite Möglichkeit nur in Frage käme, wenn keine validen Belegungen erster Möglichkeit vorhanden sind, kann man die Aktionskosten der zweiten Möglichkeit entsprechend hoch setzen. So könnte man den Algorithmus auch terminieren lassen, indem man zusätzlich zulässige Höchstkosten definiert. Oder indem man z gegen 0 prüft und auf valide Belegungen erster Möglichkeit testet. In der Form ohne diese Zusätze terminiert der Algorithmus erst, wenn er eine valide Belegung für die a_i gefunden hat und $z = 0$.

1.b)

Der Verzweigungsgrad beträgt 4, da wir für die erste Aufgabe 3 Studenten auswählen, oder eine Zeiteinheit verstreichen lassen können.

Ohne andere wie in (1.a) beschriebenen Terminierungsformen ist die maximale Tiefe des Baumes unendlich, da man immer eine Zeiteinheit verstreichen lassen kann.

1.c)

Tiefensuche eignet sich nicht da die maximale Tiefe des Baumes unendlich beträgt. Breitensuche würde dagegen eine (nicht unbedingt optimale) Lösung finden. Best-First-Search (als eigentlich informierte Suche) könnte mithilfe der in 1.b) vorgeschlagenen Aktionskosten schneller zu einer Lösung kommen, die aber nicht unbedingt optimal ist.

1.d)

A^* für dieses Problem zu verwenden ist problematisch, da es schwierig ist eine Heuristik zu finden die den optimalen Wert nicht zu weit unterschätzt. So kann man zum Beispiel die Aktionskosten für den Fall, dass $s \in \{1, 10, 100\}$ auf die jeweiligen benötigten Zeiteinheiten 1, 2 und 4 und die Aktionskosten für den Fall dass $s = 0$ auf den höchsten bisherigen Aktionspreis, nämlich 4 setzen. Wenn man jetzt den Wert, den man erhält wenn man errechnet wie viele Zeiteinheiten man mindestens benötigen würde, wenn alle Studenten so schnell wie der schnellste wären (in diesem Fall 2 Zeiteinheiten; 3 Studenten im ersten Durchgang 2 im zweiten), so kommt man hiermit zwar auf eine optimale Lösung, muss aber in ungünstiger Anfangskonstellation einen großen Teil des Baumes durchsuchen.

Aufgabe 5

Bei den Aufgaben a) bis c) ist die Zulässigkeit verschiedener Metriken als Heuristik zu zeigen. Eine Heuristik h ist zulässig, wenn $0 = h(X) = h^*(X)$ für alle Knoten X , wobei h^* die tatsächlichen Kosten bezeichnet.

5.a)

Wir betrachten als heuristische Funktion h die euklidische Metrik. Die euklidische Metrik ist definiert als $d_{eukl}(x, y) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$, wobei $x := (x_1, y_1)$ und $y := (x_2, y_2)$.

Da $(x_2 - x_1)^2 + (y_2 - y_1)^2 \geq 0, \forall x_1, x_2, y_1, y_2 \in \mathbb{R}$, ist die Wurzel dieses Terms definiert und da diese wiederum ein nichtnegatives Ergebnis ausgibt gilt: $0 \leq h(X)$ für alle möglichen zweidimensionalen Koordinaten (Längen- und Breitengrade). Zudem beschreibt die euklidische Metrik die Luftlinienentfernung zwischen zwei Punkten, somit gilt auch: $h(X) \leq h^*(X) \Rightarrow$ die euklidische Metrik ist zulässig.

5.b)

Wir betrachten als heuristische Funktion h die Maximum-Metrik.

Da $|x_2 - x_1| \geq 0$ und $|y_2 - y_1| \geq 0, \forall x_1, x_2, y_1, y_2 \in \mathbb{R}$, ist $\max(|x_2 - x_1|, |y_2 - y_1|)$ ebenfalls nichtnegativ. Somit gilt: $0 \leq h(X)$ für alle möglichen zweidimensionalen Koordinaten (Längen- und Breitengrade).

Zudem gilt: $d_{max}(x, y) \leq d_{eukl}(x, y)$ (siehe d.) und somit gilt auch $h(X) \leq h^*(X) \Rightarrow$ die Maximum-Metrik ist zulässig.

5.c)

Wir betrachten als heuristische Funktion h die Manhattan-Metrik.

Diese Metrik ist nicht zulässig. Dies zeigen wir durch ein Beispiel an dem man erkennen kann, dass die Manhattan-Metrik für unser Problem die tatsächlichen Kosten überschätzen kann. Dazu seien zwei Städte an den Koordinaten (0,0) und (1,1) die auf direktem Weg (Luftlinie) miteinander durch eine Straße verbunden sind. Die Entfernung zwischen diesen Städten ist demnach $\sqrt{2}$, aber die Manhattan-Metrik gibt $|1 - 0| + |1 - 0| = 2$ als Ergebnis aus. Somit ist die Manhattan-Metrik für unser Beispiel nicht zulässig.

5.d)

Für gegebene reelle Wertepaare $x := (x_1, y_1)$ und $y := (x_2, y_2)$ gilt: $d_{max}(x, y) \stackrel{1.}{\leq} d_{eukl}(x, y) \stackrel{2.}{\leq} d_{man}(x, y)$

Beweis 1.: Ohne Beschränkung der Allgemeinheit sei: $d_{max}(x, y) = \max(|x_2 - x_1|, |y_2 - y_1|) = |x_2 - x_1|$. (Der andere Fall würde genau analog verlaufen)

$$|x_2 - x_1| = \sqrt{(|x_2 - x_1|)^2} = \sqrt{(x_2 - x_1)^2} \leq \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Der vorletzte Schritt ergibt sich aus der Tatsache, dass $(x_2 - x_1)^2 \geq 0$ und somit der Betrag hinfällig ist. Der letzte Schritt ergibt sich aus der Monotonie der Wurzelfunktion und der Tatsache, dass $(x_2 - x_1)^2 + (y_2 - y_1)^2 \geq (x_2 - x_1)^2$.

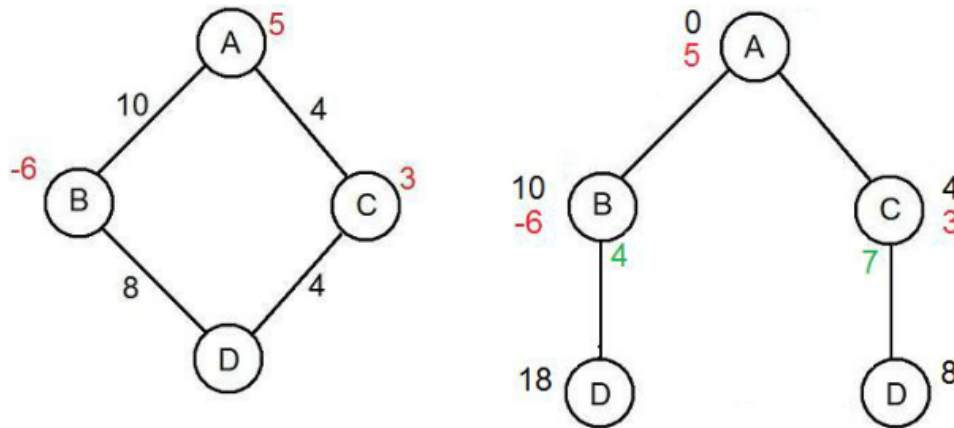
Beweis 2.: Seien $a, b \in \mathbb{R}_0^+$. Dann gilt: $a^2 + b^2 = a^2 + 2ab + b^2 = (a + b)^2$ und somit $\sqrt{a^2 + b^2} \leq \sqrt{(a + b)^2} \stackrel{a, b \geq 0}{=}$ $(a + b)$ (Es wurde wieder die Monotonie der Wurzelfunktion benutzt).

Sei nun $a := |x_2 - x_1| \geq 0$ und $b := |y_2 - y_1| \geq 0$ und damit folgt die Behauptung.

5.e)

Die Manhattan-Metrik ist nicht die geeignetste, da sie nicht zulässig ist und da die euklidische Metrik die Maximum-Metrik dominiert, ist die euklidische Metrik für unser Problem die geeignetste, da ihre Werte näher an den tatsächlichen Kosten liegen.

5.f)



An diesem Beispiel kann man sehen, dass wenn man den Algorithmus durchführen möchte, um einen kürzesten Weg von D nach A zu finden, man alle beiden Wege überprüfen wird. Allerdings würde man dies nicht tun, wenn statt -6 als heuristischer Wert 0 eingesetzt werden würde, da der Gesamtwert nicht mehr 4 sondern 10 wäre und somit in Anbetracht der Gesamtkosten des Pfades A-C-D gar nicht geprüft werden müsste. Bei größeren Beispielen kann dies zu erheblichen Teilpfaden führen, die man weglassen könnte und somit effektiver arbeiten.

5.g)

Die Vollständigkeit des Algorithmus ist weiterhin gegeben, da wenn ein Pfad nicht zu einem Zielknoten führt, ein anderer geprüft werden würde und dies so lange bis mindestens ein Punkt erreicht wird an dem Start- und Zielknoten wie gewünscht sind. Sollte kein derartiger Pfad existieren, kann der Algorithmus keinen Pfad finden und von daher kann man solche Fälle vernachlässigen.

Die Optimalität wird ebenfalls nicht beeinträchtigt, da die heuristische Werte nur Schätzwerte sind, die ein schnelleres Finden des besten Weges ermöglichen sollen. Wenn wir negative heuristische Werte zulassen, finden wir den gewünschten Weg evtl. nicht schneller, aber solange wir Heuristiken benutzen, die die tatsächlichen Kosten unterschätzen, erreichen wir trotzdem die beste Lösung.

Dies gilt, weil für die Gesamtkosten der Wege am Ende die heuristischen Werte keinen Einfluss haben. Die Überprüfung anderer - vielleicht besserer - Wege wird gewährleistet durch die Unterschätzung, denn somit wird kein potenzieller Weg durch die heuristischen Werte als „nicht weiter zu überprüfen“ gekennzeichnet.

Aufgabe 6

6.a)

w=weiß, g=grau, s=schwarz

Schritt	A	B	C	D	E
Startbelegung	w	w	w	w	w
Startkonflikte(10)	w(10)g(7)s(7)	w(10)g(6),s(6)	w(10)g(7)s(7)	w(10)g(7)s(7)	w(10)g(7)s(7)
1	w	g	w	w	w
(6)	w(6)g(5)s(4)	w(10)g(6),s(6)	w(6)g(5)s(4)	w(6)g(5)s(4)	w(6)g(5)s(4)
2	s	g	w	w	w
(4)	w(6)g(5)s(4)	w(7)g(4)s(5)	w(4)g(3)s(2)	w(4)g(4)s(4)	w(4)g(3)s(2)
3	s	g	s	w	w
(2)	w(4)g(3)s(2)	w(4)g(2)s(3)	w(4)g(3)s(2)	w(2)g(2)s(2)	w(2)g(2)s(2)
4	s	g	s	w	w
(1)	w(2)g(2)s(1)	w(3)g(1)s(3)	w(2)g(2)s(1)	w(1)g(1)s(1)	w(1)g(1)s(1)

Schritt	F	G
Startbelegung Startkonflikte(10)	w w(10)g(8)s(8)	w w(10)g(8)s(8)
1 (6)	w w(6)g(4)s(4)	w w(6)g(4)s(4)
2 (4)	w w(4)g(3)s(4)	w w(4)g(2)s(2)
3 (2)	w w(2)g(1)s(2)	w w(2)g(1)s(2)
4 (1)	g w(2)g(1)s(2)	w w(1)g(2)s(2)

Das Ergebnis hat immer noch einen Konflikt, d.h. der Algorithmus ist zu keiner Lösung gekommen.

6.b)

Die Tie-Break-Regel muss dahingehend geändert werden, dass sie den den höchsten lexikographischen Folgezustand bevorzugt.

In 6.a) ist das Problem, dass sehr früh A und C belegt werden. Dann ist aber D durch A und B festgelegt und E ist durch B und C festgelegt. Die Constraintverletzung zwischen D und E kann dann nicht mehr in einem Schritt gelöst werden und die lokale Suche bleibt somit in einem lokalen Minimum stecken.

Die geänderte Tie-Break-Regel belegt nun hingegen A und C später und dafür E früher. Somit kommt es zu keinem Konflikt zwischen D und E. Die spätere Belegung von A und C führt zu keinen Problemen, denn deren Konflikte mit F bzw. G können immer in einem Schritt gelöst werden, da F und G nur zwei Constraints haben.