

# Grundlagen der künstlichen Intelligenz: Hausaufgabe 1

Tom Nick - 340528  
Niklas Gebauer - 340942

## Aufgabe 1

- a) **Zustandsraum:**  $(a, b, c)$  wobei  $a \in \{0, \dots, 100\}, b \in \mathbb{N}, c \in \{A, B, C, Z, i, j\}$   
wobei  $a$  die aktuelle Position,  $b$  die benötigte Zeit und  $c$  den Ladezustand beschreibt.

**Anfangszustand:**  $(A, 0, 100)$

**Zielzustand:**  $(Z, b, c)$  wobei  $c \geq 50$ ,  $b$  hat keine Einschränkung. **Aktionen:**

1.

$\text{fahren}(\text{start}, \text{ziel}, \text{zeit}, \text{energie}) : (a, b, c) \rightarrow (x, y, z)$

mit

$$\begin{aligned} a &= \text{start} \wedge x = \text{ziel} \wedge \\ z &= c - \text{energie} \wedge z \geq 0 \wedge y = b + \text{zeit} \wedge \\ (\text{start}, \text{ziel}, \text{zeit}, \text{energie}) &\in \{(A, Z, 170, 95), (Z, A, 170, 95), \\ &\quad (A, i, 100, 50), (i, A, 100, 50), \\ &\quad (i, Z, 200, 100), (Z, i, 200, 100), \\ &\quad (i, j, 100, 50), (j, i, 100, 50), \\ &\quad (i, B, 80, 45), (B, i, 80, 45), \\ &\quad (j, Z, 80, 40), (Z, j, 80, 40), \\ &\quad (j, C, 25, 20), (C, j, 25, 20), \\ &\quad (Z, C, 20, 10), (C, Z, 20, 10)\} \end{aligned}$$

2.

$\text{laden}(\text{zustand}) : (a, b, c) \rightarrow (x, y, z)$

mit

$$\text{zustand} \in \{i, j\} \wedge y = b + 200 \wedge z = 100 \wedge \text{zustand} = a = x$$

- b) **Verzweigungsgrad:** maximal 3

**Tiefe:** 6 wenn man sich beim Suchen intelligent anstellt. Wobei das bedeutet, dass wir einen Knoten nur 2x besuchen wenn im zweiten Besuch des Knotens die Ladung größer ist als beim ersten Besuch.

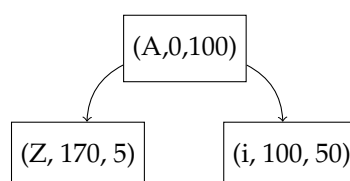
- c) Da wir den schnellsten Weg Finden wollen wäre **Branch and Bound** am besten, wir haben keine Heuristiken, also kein  $A^*$ , wir haben aber Pfadkosten also keine BFS/DFS.

d)

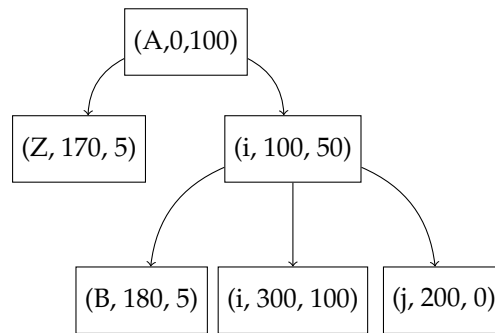
1.

$(A, 0, 100)$

2.

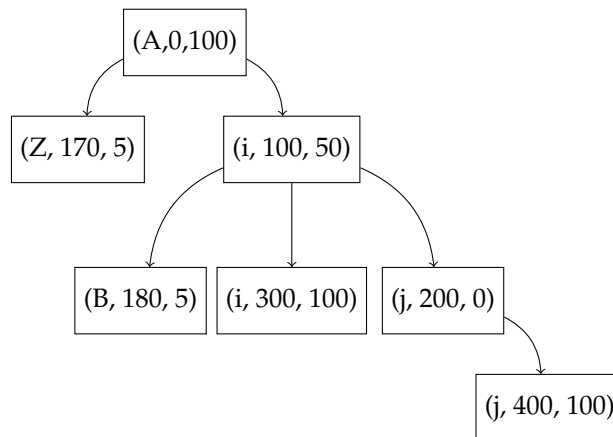


3.

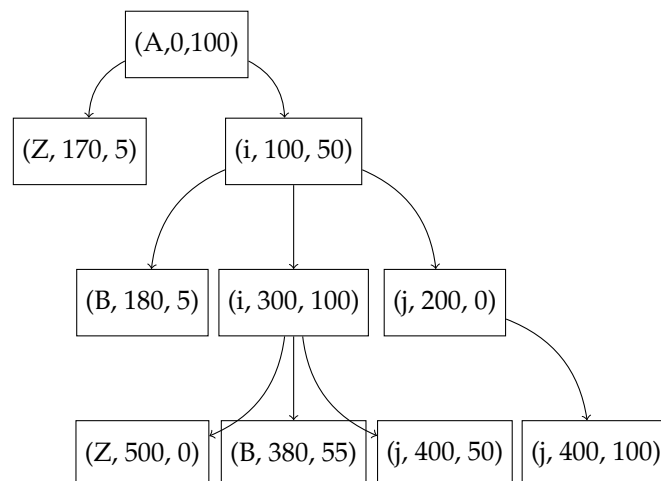


4. Die dritte und vierte Expansion von dem Knoten (Z,170,5) und (B,180,5) bewirken keine Veränderung des Baumes, da Sie keine Nachfolger haben. (Energie reicht nicht aus um zu einem anderen Knoten zu fahren)

5.



6.



e) Algorithmen der dynamischen Programmierung werden genutzt um optimale Lösungen zu finden, in dem Beispiel hier wollen wir den kleinstmöglichen Weg von A nach Z finden, wobei die verbleibende Energie größer gleich 50 ist. Dafür werden ALLE möglichen Wege untersucht, viele werden jedoch von vornerein ausgeschlossen (eine Art intelligentes Brute Force). In diesem Beispiel benutzen wir Branch and Bound. Dieser Algorithmus untersucht als nächsten Knoten immer den mit den niedrigsten insgesamten Pfadkosten, dadurch werden gleichzeitig viele unnütze Wege, Schleifen... ausgeschlossen.

**Aufgabe 2**

**Aufgabe 3**

**Aufgabe 4**