



Grundlagen der künstlichen Intelligenz WS 04/05

Albayrak, Jain (AOT) - Obermayer, Martin (NI)

1. Übungsblatt  
Abgabe: 02.11.2005

Aufgabe 1

(1 Punkt)

Formulieren Sie das folgende Problem: Gegeben ist eine Landkarte von Staaten. Färben Sie die Landkarte mit vier Farben ein, so dass die gefärbte Karte folgende Eigenschaften besitzt:

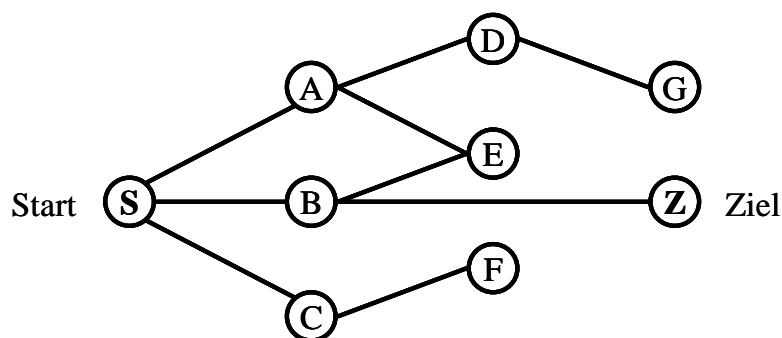
1. Jeder Staat ist mit genau einer Farbe eingefärbt.
2. Benachbarte Staaten besitzen unterschiedliche Färbungen.

Wählen Sie eine Problemformulierung, die hinreichend präzise für eine Implementierung ist.

Aufgabe 2

(4 Punkte)

Betrachte den folgenden Problemgraphen mit Anfangszustand S und Zielzustand Z:



Finden Sie eine Lösung mit Breiten- und Tiefensuche. Die nächste Aktion zur Expandierung eines Teilpfades wird in lexikographischer Reihenfolge der Teilpfade ausgewählt, d.h. Teilpfad <SA> wird vor <SB> expandiert.

### Breitensuche (nach Winston 93)

1. Form a one-element queue consisting of a zero-length path that contains only the root node.
2. Until the first path in the queue terminates at the goal node or the queue is empty
  - a) Remove the first path from the queue; create new paths by extending the first path to all neighbors of the terminal node.
  - b) Reject all paths with loops.
  - c) Append the new paths, if any, to the end of the queue.
3. If the goal node is found, announce success; otherwise announce failure

### Tiefensuche (nach Winston 93)

1. Form a one-element stack consisting of a zero-length path that contains only the root node.
2. Until the first path in the stack terminates at the goal node or the queue is empty
  - a) Remove the first path from the stack; create new paths by extending the first path to all neighbors of the terminal node.
  - b) Reject all paths with loops.
  - c) Push the new paths, if any, to the top of the stack.
3. If the goal node is found, announce success; otherwise announce failure.

a) Die Lösung soll mit beiden Algorithmen nach folgendem Schema simuliert werden:

Iteration	Liste
0	<S>
1	<SA>, <SB>

Geben Sie dabei die Liste (Queue bzw. Stack) immer vor Beginn eines Schleifendurchlaufs (vor Schritt 2a) und am Schluss (nach Schritt 3) an.

- b) Vergleichen Sie beide Verfahren für das obige Problem bezüglich
- Optimalität
  - Zeitkomplexität
  - Speicherkomplexität

### Aufgabe 3

(5 Punkte)

Drei Pilger und drei Räuber befinden sich zusammen mit einem Boot auf der gleichen Seite eines Flusses. Das Boot kann maximal zwei Personen aufnehmen. Wie können alle 6 Personen auf die andere Seite des Flusses transportiert werden, sodass auf keiner Seite des Flusses mehr Räuber als Pilger sind? (Wenn auf einer Seite des

Flusses ausschließlich Räuber sind, dann ist dies ein erlaubter Zustand - denn in diesem Fall finden die Räuber kein(e) Opfer.)

- a) Formulieren Sie das Problem präzise. Zeichnen Sie ein Diagramm des Problemraums aller Zustände, die ausgehend vom Anfangszustand erreichbar sind.
- b) Implementieren und lösen Sie das Problem optimal unter Verwendung eines der Suchalgorithmen in Java oder C/C++. Abzugeben sind
  - Der Quellcode und eine README Datei an [bjj@dai-labor.de](mailto:bjj@dai-labor.de). Die README Datei beschreibt,
    - welcher Suchalgorithmus implementiert wurde
    - wie das Programm kompiliert und gestartet wird.
  - Ein Ausdruck des Quellcodes.

### Organisatorisches:

- **Anmeldung:** Jede Übungsgruppe (4 Teilnehmer) muss sich per Email bis zum 2.11.2005 an [bjj@dai-labor.de](mailto:bjj@dai-labor.de) mit den folgenden Angaben für jeden Teilnehmer anmelden:

Nachname, Vorname, Matrikel-Nr, Email, Studienfach

- Bitte Reihenfolge einhalten und Kommata beachten.
  - Bitte nur eine Mail pro Übungsgruppe
- **Abgabe der Lösungen:** Elektronische Abgaben werden nicht berücksichtigt, außer wenn dies ausdrücklich gefordert wird. Bitte nur ausgedruckte oder handgeschriebene Lösungen abgeben.