

Beispiel Missionare + Kannibalen

informale Problembeschreibung:

- ▶ Zu Beginn: 3 Missionare + 3 Kannibalen an einem Flussufer
- ▶ Ziel ist das Übersetzen aller Personen.
- ▶ Es gibt nur ein Boot, welches genau zwei Personen fasst.
- ▶ Alle Personen im Boot steigen am Ufer aus (und dann ggf. wieder ein).
- ▶ Sobald an einer Stelle (Ufer, Boot) mehr Kannibalen als Missionare sind, werden die Missionare gefressen.

formale Modellierung (Beispiel):

- ▶ Zustände: $S \subseteq \{0, \dots, 3\}^4 \times \{-1, 1\}$ mit
 $\forall ((m_s, k_s), (m_z, k_z), u) \in S : m_s + m_z = 3 \wedge k_s + k_z = 3 \wedge \dots$
- ▶ Startzustand: $((3, 3), (0, 0), -1)$
- ▶ Zielzustand: $((0, 0), (3, 3), 1)$
- ▶ Zustandsübergänge
 $((m_s, k_s), (m_z, k_z), u) \in S \rightarrow ((m'_s, k'_s), (m'_z, k'_z), -u) \in S$
mit $\exists n_k, n_m : 1 \leq n_m + n_k \leq 2 \wedge \dots$

Problemlösung durch Suche in Graphen – Beispiele

- ▶ Finden von Wegen in einem Graphen
 - ▶ Aufgabe:
 - ▶ gegeben: Graph G (Tafel)
 - ▶ gesucht: Weg (Pfad) in G von Knoten u zu Knoten v
 - ▶ Lösungsidee: Suche im Graphen
- ▶ Münzenstapelspiel (für eine Person)
 - ▶ Aufgabe:
 - ▶ gegeben: Stapel von n Münzen
 - ▶ gesucht: Zugfolge durch erlaubte Züge (zwei Münzen von einem Stapel nehmen und auf beide Nachbarn verteilen) bis zu einer Situation, in der kein Zug möglich ist
 - ▶ Lösungsidee:
 - ▶ Modellierung als Zustandsübergangssystem
 - ▶ Suche im Graphen
- ▶ 3 Krüge
 - ▶ Aufgabe:
 - ▶ gegeben: 3 volle Krüge mit Volumen 4l, 7l, 9l,
 - ▶ gesucht: genau 6l in einem der 3 Krüge
 - ▶ Lösungsidee: Zustände als Knoten eines Suchbaumes

Darstellung von Aufgabe und Lösung

Aufgabe:

- gegeben:
- ▶ Menge V von Zuständen (evtl. unendlich)
oft beschrieben durch Eigenschaften
 - ▶ Startzustand $s \in V$
 - ▶ Menge $Z \subseteq V$ von Zielzuständen
(oder Eigenschaften der Zielzustände)
 - ▶ mögliche Übergänge zwischen Zuständen
Übergangsrelation $E \subseteq V \times V$

Lösung: Folge von Zuständen (Weg von einem Start- zu einem Zielzustand) (Mitunter interessiert nur der erreichte Zielzustand.)

Wissensrepräsentation: als Graph $G = (V, E)$
(Zustandsübergangssystem):

- ▶ Knotenmenge V : Zustände
- ▶ (gerichtete) Kanten: Zustandsübergänge

Entfaltung des Graphen zu einem Baum:

Pfade im Graphen = Knoten im Baum

Problemlösen durch Suchen

- ▶ formale Darstellung des Problem es als Graph bzw. Baum
- ▶ formale Beschreibung der Lösung als Eigenschaft von
 - ▶ Pfaden im Graphen
 - ▶ Knoten im Baum

Möglichkeiten zum Problemlösen:

- ▶ Pfadsuche im Graphen
- ▶ Knotensuche im Baum

Suche in Graphen

(schon bekannte) Verfahren zur Suche in Graphen (und Bäumen):

- ▶ Tiefensuche (depth-first search):
Suche zuerst in Teilbäumen eines noch nicht besuchten Nachbarn des aktuellen Knotens
- ▶ Breitensuche (breadth-first search):
Suche zuerst in Teilbäumen eines noch nicht besuchten Knotens mit der geringsten Tiefe

Allgemeines Suchverfahren

- Daten: L_a Menge der noch zu expandierenden Knoten
 L_x Menge der expandierten Knoten
 s Startknoten
 φ Anforderungen an Lösung (Zielknoten)

Allgemeiner Suchalgorithmus:

1. $L_a = \{s\}, L_x = \emptyset$
2. solange $\neg L_a = \emptyset$:
 - 2.1 Verschiebe einen auf **festgelegte Art** ausgewählten Knoten u aus L_a in L_x
 - 2.2 Füge alle Nachbarn von u , die nicht in $L_a \cup L_x$ enthalten sind, auf eine **festgelegte Art** in L_a ein
(Abbruch falls ein Nachbar v von u die Bedingung φ erfüllt, also eine Lösung repräsentiert)

prominente Spezialfälle:

- Tiefensuche** ▶ Verwaltung von L_a als **Stack**
▶ Einfügen der Nachbarn an den **Anfang** der Liste L_a
▶ festgelegter Knoten wurde **zuletzt** in L_a eingefügt
- Breitensuche** ▶ Verwaltung von L_a als **Queue**
▶ Einfügen der Nachbarn an das **Ende** der Liste L_a