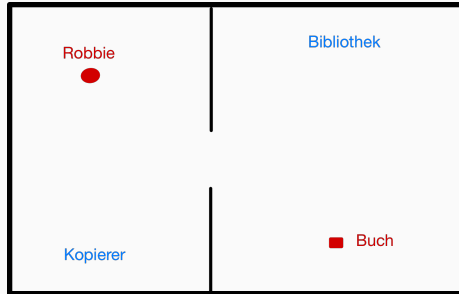


Problemlösen

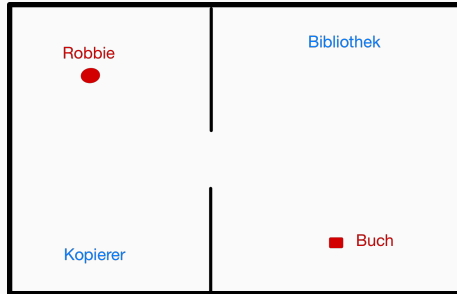
Carsten Gips (HSBI)

Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

Motivation: Roboter in einer Bibliothek



Motivation: Roboter in einer Bibliothek



Aktionen:

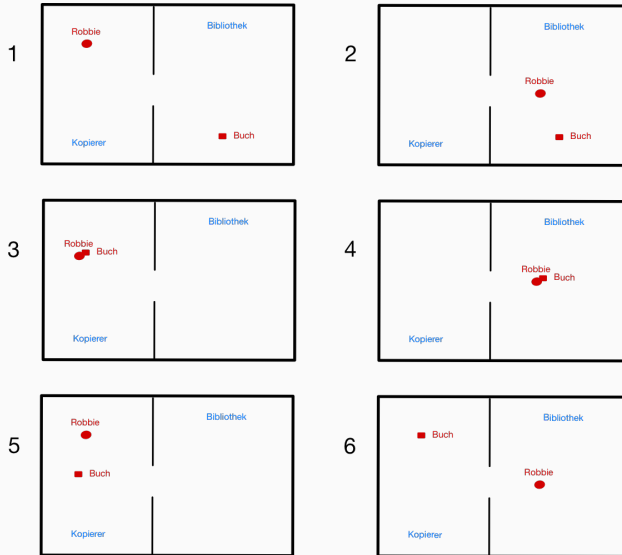
- Right (R)
- Left (L)
- Take (T)
- Drop (D)

Wahrnehmungen:

- In welchem Raum bin ich?
- Habe ich das Buch?

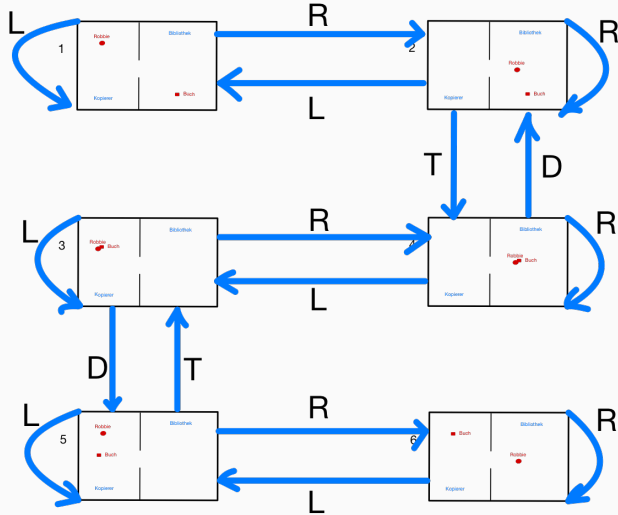
Aufgabe: Das Buch aus der Bibliothek holen und in den Kopiererraum bringen.

Zustände der Bibliotheks-Welt



Problem: Gegeben einen Startzustand, wie komme ich zum Ziel?

Suche im Problemgraphen



Definition Zustand und Aktion

Zustand: (Formale) Beschreibung eines Zustandes der Welt

Aktion: (Formale) Beschreibung einer durch Agenten ausführbaren Aktion

- Anwendbar auf bestimmte Zustände
- Überführt Welt in neuen Zustand (“Nachfolge-Zustand”)

Geeignete Abstraktionen wählen für Zustände und Aktionen!

Definition Problem

Startzustände Menge $S_A \subset S$

Aktionen Menge von Funktionen $op : S \rightarrow S$

Zustandsraum Menge aller Zustände S , die durch (wiederholte) Anwendung von Aktionen von den Startzuständen aus erreichbar sind

Zieltest Funktion $goal : S \rightarrow \{0, 1\}$

Zielzustände Menge $S_E \subseteq S$ mit $\forall x \in S_E : goal(x) = 1$

Kosten Gesamtkosten: $f(n) = g(n) + h(n)$

Definition Problemlösen

Problemlösen Wegesuche im Graph vom Startknoten zu einem Zielknoten

- Spannt den **Suchbaum** auf

Lösung Folge von Aktionen, die Start- in Zielzustand überführen

Suche: Einfache Basisvariante

1. Füge Startknoten in leere Datenstruktur (Stack, Queue, ...) ein
2. Entnehme Knoten aus der Datenstruktur:
 - Knoten ist gesuchtes Element: Abbruch, melde "*gefunden*"
 - Expandiere alle Nachfolger des Knotens und füge diese in die Datenstruktur ein
3. Falls die Datenstruktur leer ist: Abbruch, melde "*nicht gefunden*"
4. Gehe zu Schritt 2

Erweiterung der Suche: Vermeiden von Wiederholungen

1. Füge Startknoten in leere Datenstruktur (Stack, Queue, ...) ein
2. Entnehme Knoten aus der Datenstruktur:
 - Knoten ist gesuchtes Element: Abbruch, melde "*gefunden*"
 - Markiere aktuellen Knoten, und
 - Expandiere alle Nachfolger des Knotens und füge alle unmarkierten Nachfolger, die noch nicht in der Datenstruktur sind, in die Datenstruktur ein
3. Falls die Datenstruktur leer ist: Abbruch, melde "*nicht gefunden*"
4. Gehe zu Schritt 2

Bewertung von Suchalgorithmen

Vollständigkeit Findet der Algorithmus eine Lösung, wenn es eine gibt?

Optimalität Findet der Algorithmus die beste Lösung?

Zeitkomplexität Wie lange dauert es eine Lösung zu finden?

Speicherkomplexität Wieviel Speicher benötigt die Suche?

Größen zur Bewertung:

- **b**: Verzweigungsfaktor
- **d**: Ebene (Tiefe) des höchsten Lösungsknotens
- **m**: Länge des längsten Pfades

- Begriffe “Problem”, “Zustand”, “Aktion”, “Zustandsraum”, “Problemgraph”, “Suchbaum”
- Problemlösen: Suche in Graphen nach Weg vom Start zum Ziel
 - Suche spannt einen Suchbaum auf
 - Unterschiedliche Kostenfunktionen möglich
 - Suchalgorithmen: Einfache Basisvariante, Erweiterung mit Vermeidung von Redundanzen
 - Beurteilung der Suchverfahren: Optimalität, Vollständigkeit, Komplexität



Unless otherwise noted, this work is licensed under CC BY-SA 4.0.