

Search.01: Problemformalisierung, Zustandsraum

Ausgangslage

- Am linken Ufer stehen 3 Elben** und 3 Orks.
- Es gibt ein Pferd, das höchstens zwei Wesen gleichzeitig tragen kann.
- Das Pferd kann nicht allein über den Fluss fahren.

Ziel ist es, alle sechs Wesen sicher ans rechte Ufer zu bringen.

Regel (Sicherheitsbedingung)

Auf keiner Seite des Flusses darf es jemals mehr Orks als Elben geben, solange dort noch mindestens eine Elf steht.

Sonst würden die Orks die Elben angreifen.

Das bedeutet:

- Wenn Elben auf einer Seite sind, dürfen sie dort nicht in der Minderheit sein.
- Wenn keine Elben auf einer Seite sind, ist es egal, wie viele Orks dort sind.

Aktionen (mögliche Überfahrten)

Das Pferd kann über den Fluss fahren und dabei eine oder zwei Personen mitnehmen.

Mögliche Kombinationen:

- 1 Elf
- 2 Elben
- 1 Ork
- 2 Orks
- 1 Elf und 1 Ork

Nach jeder Fahrt befindet sich das Pferd am anderen Ufer.

Startzustand

Linkes Ufer: 3 Elben, 3 Orks, Pferd links

Rechtes Ufer: 0 Elben, 0 Orks

Zielzustand

Linkes Ufer: 0 Elben, 0 Orks

Rechtes Ufer: 3 Elben, 3 Orks, Pferd rechts

Aufgabe

Finde eine Reihenfolge von Überfahrten,
bei der die Sicherheitsregel (niemals mehr Orks als Elben) nie verletzt wird
und am Ende alle am rechten Ufer sind.

Search.02: Suchverfahren

Tiefensuche

- [A]
- [AB, AC, AE]
- [ABF, AC, AE]
- [ABFI, AC, AE]
- [ABFIJ, AC, AE]
- [ABFIJ]

Breitensuche

Queue	Besuchte Knoten
[A]	[A]
[AB, AC, AE]	[A, B]
[AC, AE, ABF]	[A, B, C]
[AE, ABF, ACG, ACH]	[A, B, C, E]
[ABF, ACG, ACH, AEJ]	[A, B, C, E, F]
[ACG, ACH, AEJ, ABFI]	[A, B, C, E, F, G]
[ACH, AEJ, ABFI]	[A, B, C, E, F, G, H]
[AEJ, ABFI]	[A, B, C, E, F, G, H, J]
[AEJ]	

A*

- [A 0+100=100]
- [AB 85+200=285, AC 217+170=387, AE 173+460=633]
- [AC 387, AE 633, ABF 285+80+10=375]
- [AC 387, AE 633, ABFI 375+250+0=625]
- [AE 633, ABFI 625, ACG 387+186+400=973, ACH 387+103+537=1027]
- [AE 633, ACG 973, ACH 1027, ABFIJ 625+84+0=709]
- [ACG 973, ACH 1027, ABFIJ 709, AEJ 633+502+0=1135]
- [ABFIJ 709]

Vergleich

Bei der Tiefensuche befinden sich maximal 3 Einträge gleichzeitig in der Datenstruktur. Die Hauptschleife wird 5-mal durchlaufen, da in jedem Schritt ein Element entnommen, untersucht und weiterentwickelt wird, bis das Ziel erreicht ist.

Bei der Breitensuche wächst die Datenstruktur auf bis zu 4 Einträge an. Die Hauptschleife wird 8-mal ausgeführt, weil alle Knoten Ebene für Ebene durchsucht werden, bevor das Ziel gefunden wird.

Beim A*-Algorithmus liegt die maximale Anzahl der Einträge in der Datenstruktur bei 4. Durch die Heuristik werden weniger unproduktive Wege verfolgt, sodass die Hauptschleife nur 7-mal durchlaufen wird, bis das Ziel erreicht ist.

Insgesamt zeigt sich: Die Tiefensuche verwendet am wenigsten Speicher und ist schnell im Finden eines (nicht unbedingt optimalen) Ziels, die Breitensuche benötigt den meisten Speicher und mehr Iterationen, während A* einen guten Kompromiss zwischen Speicherbedarf und Effizienz bietet.

Search.03: Dominanz

Begriff: Dominierende Heuristik

Man sagt, eine Heuristik $h_1(n)$ dominiert eine Heuristik $h_2(n)$, wenn für alle Knoten n gilt:

$$h_1(n) \geq h_2(n)$$

und beide zulässig sind (sie überschätzen die tatsächlichen Kosten zum Ziel nicht).

h_1 liefert stets gleich gute oder bessere Schätzungen der Restkosten als h_2 .

Wirkung auf den A*-Algorithmus

Wenn A* eine dominierende Heuristik h_1 verwendet, dann gilt:

- A* mit h_1 erweitert nie mehr Knoten als A* mit h_2 .
- Meistens werden deutlich weniger Knoten untersucht → effizientere Suche.
- Beide finden dieselbe optimale Lösung, solange beide Heuristiken zulässig sind.

Eine dominierende Heuristik spart also Zeit, ohne die Korrektheit oder Optimalität zu gefährden.

Beispiel 1: Pfadsuche auf einem Gitter

Ziel: Von Start S zu Ziel Z laufen (Bewegung in vier Richtungen erlaubt).

- $h_1(n)$ = Manhattan-Distanz (Summe der horizontalen und vertikalen Schritte)
- $h_2(n) = 0$ (triviale Heuristik → entspricht Dijkstra)

Da gilt:

$$h_1(n) \geq h_2(n) \text{ für alle } n$$

→ h_1 dominiert h_2 .

Auswirkung: A* mit h_1 ignoriert viele unnötige Richtungen und ist deutlich schneller, während A* mit h_2 alle Wege gleichberechtigt prüft.

Beispiel 2: Navigationssystem

- $h_1(n)$ = Luftliniendistanz zum Ziel
- $h_2(n)$ = konstante Schätzung (z. B. immer 0 oder 1)

Hier dominiert h_1 klar h_2 .

Das System mit h_1 führt effizienter zum Ziel, da es realistischere Entfernung abschätzt.

Search.04: Beweis der Optimalität von A*

Grundidee

A* wählt immer den Knoten mit dem kleinsten geschätzten Gesamtkostenwert

$$f(n) = g(n) \leq h(n)$$

aus der offenen Liste aus.

Dabei gilt:

- $g(n)$ = tatsächliche Kosten vom Start bis zum Knoten n
 - $h(n)$ = geschätzte (heuristische) Kosten von n bis zum Ziel
 - $f(n)$ = geschätzte Gesamtkosten des günstigsten Weges über n
-

Definition: Zulässige Heuristik

Eine Heuristik $h(n)$ heißt zulässig, wenn sie nie die tatsächlichen Kosten zum nächsten Ziel überschätzt:

$$\text{für alle } n: h(n) \leq h^*(n)$$

wobei $h^*(n)$ die wahren minimalen Kosten vom Knoten n zum Ziel sind.

Das bedeutet:

Die Heuristik ist optimistisch. Sie unterschätzt höchstens, überschätzt aber nie.

Beweisidee (informell)

1. A* wählt Knoten in aufsteigender Reihenfolge von $f(n) = g(n) + h(n)$.
2. Eine zulässige Heuristik garantiert, dass $f(n)$ niemals größer ist als die tatsächlichen Kosten der optimalen Lösung.
3. Sobald A* ein Ziel z auswählt, gilt:
 - $f(z) = g(z) + h(z)$
 - Da $h(z) = 0$, ist $f(z) = g(z)$, also die tatsächlichen Kosten dieses Weges.

4. Angenommen, es gäbe eine bessere Lösung mit geringeren Kosten $C^* < g(z)$.

Dann müsste ein anderer Knoten n mit $f(n) < g(z)$ in der offenen Liste existieren.

Aber A* hätte diesen Knoten zuerst expandiert — Widerspruch!

Daher kann keine bessere Lösung existieren.

→ Die gefundene Lösung ist optimal.

Formeller Beweis

Angenommen, A* findet eine Lösung z mit Kosten $g(z) = C$.

Sei C^* der optimale (minimal mögliche) Lösungspfad.

Da h zulässig ist, gilt für alle Knoten n :

$f(n) = g(n) + h(n) \leq g(n) + h^*(n)$ Insbesondere für alle Knoten auf dem optimalen Pfad gilt:

$$f(n) \leq C^*$$

Bevor A* das Ziel z auswählt, werden alle Knoten mit $f(n) \leq C^*$ expandiert.

Da für den optimalen Pfad jedes $f(n) \leq C^*$ ist, wird der Zielknoten des optimalen Pfades nicht übergangen, bevor A* eine teurere Lösung findet.

Sobald ein Zielzustand mit Kosten $C = C^*$ erreicht ist, stoppt A*.

Somit ist die gefundene Lösung optimal.

Bonus Aufgabe

KI und Robotik: Was heute schon geht – und was noch nicht

Was läuft schon relativ gut

- In der Robotik und Automation sind viele Routine-Aufgaben bereits automatisiert:
z. B. Montagearbeiten in Fabriken, Logistik-Roboter in Lagern, Bildverarbeitung zur Defekterkennung.
- Large Language Models (LLMs) wie ChatGPT werden eingesetzt für Textgenerierung, Übersetzung und als Sprachassistenten.
- Autonomes Fahren (Teilautomation): Systeme wie Spurhalteassistent, adaptiver Tempomat oder Parkassistent funktionieren zuverlässig im Alltag.

Chancen: Effizienzsteigerung, Entlastung bei monotonen oder gefährlichen Aufgaben, Assistenzsysteme für ältere oder eingeschränkte Personen.

Was ist noch offen / ungelöst

- Vollautonomes Fahren ist technisch noch nicht ausgereift:
Wahrnehmungsprobleme bei Dunkelheit, Wetter, unerwarteten Verkehrssituationen.
-LLMs zeigen weiterhin Ungenauigkeiten, Voreingenommenheit und Halluzinationen.

- Gesetzgebung & Ethik: Haftungsfragen bei Unfällen, Datenschutz, Arbeitsplatzverlagerung, Verantwortung für KI-Entscheidungen.
- Kognitive Auswirkungen: Menschen übernehmen zunehmend Formulierungen von KI-Systemen.

Gesellschaftliche Auswirkungen – Chancen & Risiken

Chancen

- Barrierefreiheit: KI-gestützte Assistenzsysteme und Übersetzer erleichtern Teilhabe.
- Produktivität: Weniger Fehler, schnellere Bearbeitung, neue Services.
- Neue Arbeitsfelder: z. B. KI-Trainer, Datenethiker, Prompt-Designer.

Risiken

- Arbeitsplätze: Automatisierung könnte Ungleichheit verstärken.
- Kulturelle Veränderungen: Einheitlichere Sprache durch KI-Einfluss → weniger Vielfalt.
- Manipulation: LLMs können überzeugend, aber auch irreführend argumentieren.
- Haftung: Wer ist verantwortlich bei Fehlentscheidungen oder Unfällen?
- Kognitive Abhängigkeit: Gefahr des Verlernens eigener Denk- und Schreibfähigkeiten.

Fazit / persönliche Einschätzung

Es ist weder alles gut noch alles schlecht. Wir stehen mitten in einem großen Wandel.

KI bietet enorme Chancen, wenn sie verantwortungsvoll eingesetzt wird.

Wichtig ist, dass Gesellschaft, Bildung und Politik mithalten, damit Technik den Menschen unterstützt, nicht ersetzt.