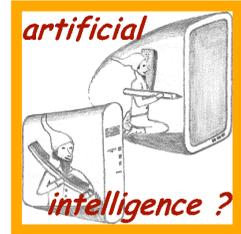




**Ruprecht-Karls-Universität Heidelberg**  
**Psychologisches Institut**

**Computermodelle des Denkens und Problemlösens**



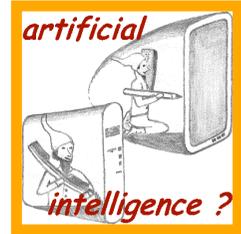
**Seminar: Denken und Problemlösen**  
**Dozent: Prof. Dr. Joachim Funke**

Referent:  
Philip Furley

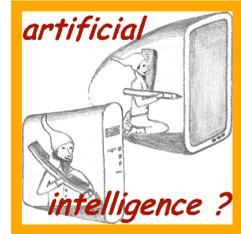


**Ruprecht-Karls-Universität Heidelberg**  
**Psychologisches Institut**

**Computermodelle des Denkens und Problemlösens**



# Gliederung



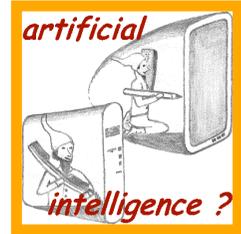
# Einleitung

**Ziel: Gemeinsamkeiten und Unterschiede der Sichtweisen der Künstlichen Intelligenz und der Psychologie aufzuzeigen**

- Begründung der KI 1956
  - KI als ingenieurwissenschaftliche Disziplin
  - KI als formalwissenschaftliche Disziplin
  - KI als kognitionswissenschaftliche Disziplin



**Grundannahme: Kognition wird als Berechnungsprozess aufgefasst**



# Einleitung

- Problem: Umsetzung von Prozessmodell in Computermodell
- Möglichkeit: → Kognitive Architekturen

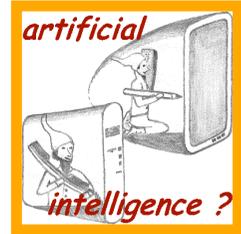
Explizite Vorgabe von elementaren Mechanismen der Informationsverarbeitung, von denen angenommen wird, dass sie über alle möglichen Aufgaben hinweg stabil bleiben

## Produktionssysteme

- ACT
- Soar

## KI-Systeme

- Prodigy
- IPAL



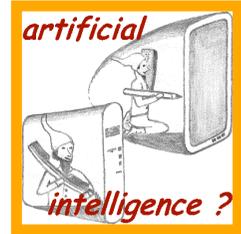
# Computermodelle des Problemlösens

## Problemlösen als Suche im Problemraum

- Menge klar definierter Problemzustände...
- und Operatoren

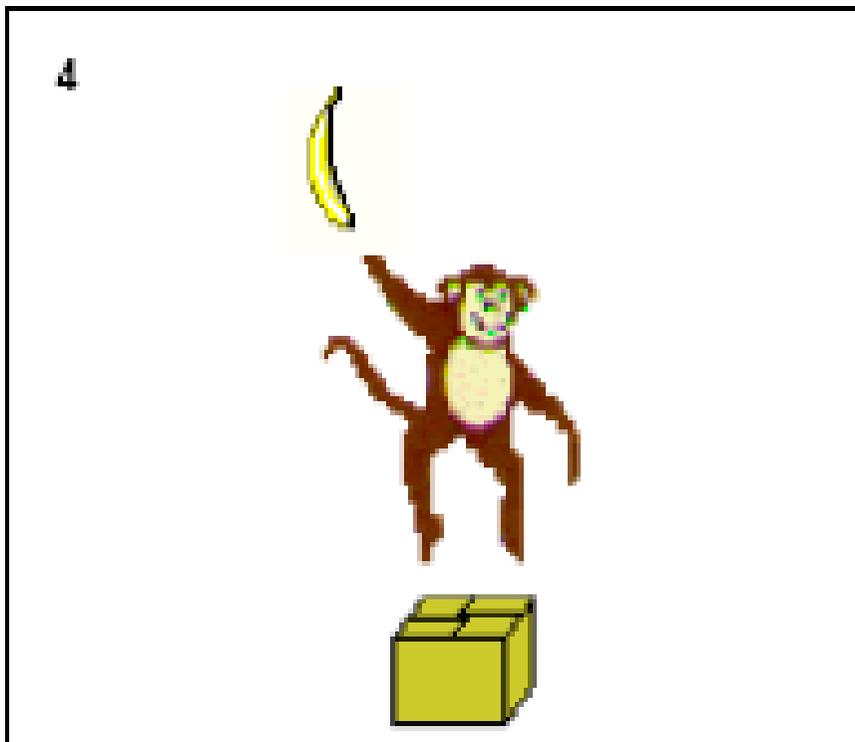
→ Meisten Problemlöseverfahren der KI zur Lösung von Problemen dieser Art

Nach Nilsson (1971): „*Ein Problem ist gegeben durch einen Anfangszustand, ein Problemlöseziel und eine Menge von Aktionen oder Problemlöseoperatoren*“



# Computermodelle des Problemlösens

## Das „Affe-Banane“ Problem



Problem:

Anfangszustand

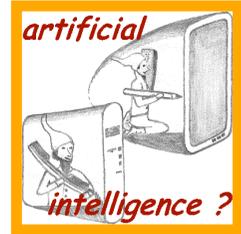
Problemlöseziel

Problemlöseoperatoren

Zur Kiste gehen

Kiste in Pos. bringen

Auf Kiste steigen



# Computermodelle des Problemlösens

## Das „Affe-Banane“ Problem

Anfangszustand: (links, rechts, boden, nein)

Ziel: (→, →, → ja)

Bewegung im Raum:

(links, links, boden, nein) → (mitte, links, boden, nein)  
 (links, links, boden, nein) → (rechts, links, boden, nein)  
 (mitte, links, boden, nein) → (links, links, boden, nein)  
 (mitte, links, boden, nein) → (rechts, links, boden, nein)  
 (rechts, links, boden, nein) → (links, links, boden, nein)  
 (rechts, links, boden, nein) → (mitte, links, boden, nein)  
 (links, mitte, boden, nein) → (mitte, mitte, boden, nein)  
 (links, mitte, boden, nein) → (rechts, mitte, boden, nein)  
 (mitte, mitte, boden, nein) → (links, mitte, boden, nein)  
 (mitte, mitte, boden, nein) → (rechts, mitte, boden, nein)  
 (rechts, mitte, boden, nein) → (links, mitte, boden, nein)  
 (rechts, mitte, boden, nein) → (mitte, mitte, boden, nein)  
 (links, rechts, boden, nein) → (mitte, rechts, boden, nein)  
 (links, rechts, boden, nein) → (rechts, rechts, boden, nein)  
 (mitte, rechts, boden, nein) → (links, rechts, boden, nein)  
 (mitte, rechts, boden, nein) → →  
 (rechts, rechts, boden, nein) → →  
 (rechts, rechts, boden, nein) → (links, rechts, boden, nein)  
 (rechts, rechts, boden, nein) → →  
 (mitte, rechts, boden, nein)

Schieben der Kiste:

(links, links, boden, nein) → (mitte, mitte, boden, nein)  
 (links, links, boden, nein) → (rechts, rechts, boden, nein)  
 (mitte, mitte, boden, nein) → (links, links, boden, nein)  
 (mitte, mitte, boden, nein) → →  
 (rechts, rechts, boden, nein)  
 (rechts, rechts, boden, nein) → (links, links, boden, nein)  
 (rechts, rechts, boden, nein) → →  
 (mitte, mitte, boden, nein)

Steigen auf Kiste:

(links, links, boden, nein) → (links, links, kiste, nein)  
 (mitte, mitte, boden, nein) → (mitte, mitte, kiste, nein)  
 (rechts, rechts, boden, nein) → (rechts, rechts, kiste, nein)

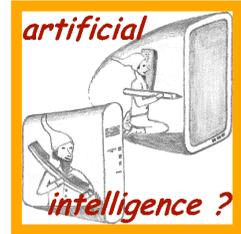
Greifen der Banane:

(mitte, mitte, kiste, nein) → (mitte, mitte, kiste, ja)



**Ruprecht-Karls-Universität Heidelberg**  
**Psychologisches Institut**

**Computermodelle des Denkens und Problemlösens**



# Computermodelle des Problemlösens

## Das „Affe-Banane“ Problem

Anfangszustand:  $\text{pos}(\text{Affe}, \text{Links}), \text{auf-boden}, \text{pos}(\text{Kiste}, \text{Rechts}), \text{pos}(\text{Banane}, \text{Mitte})$

Ziel:  $\text{hat-banane}$

Raumpositionen:  $\text{ort}(\text{Links}), \text{ort}(\text{Mitte}), \text{ort}(\text{Rechts})$

### **Operatoren:**

#### **GehevonNach (x,y):**

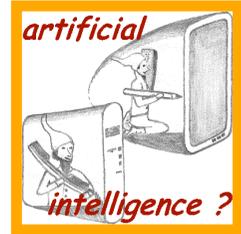
Anwendungsbedingung:  $\text{ort}(x), \text{ort}(y), \text{pos}(\text{Affe}, x), \text{pos}(\text{Kiste}, x), \text{auf Boden}$

Auswirkung:  $\text{ADD pos}(\text{Affe}, y); \text{DEL pos}(\text{Affe}, x)$

#### **SteigeAufKiste(x):**

Anwendungsbedingungen:  $\text{ort}(x), \text{pos}(\text{Affe}, x), \text{pos}(\text{Kiste}, x), \text{auf Boden}$

Auswirkung:  $\text{ADD auf Kiste}; \text{Del auf Boden}$



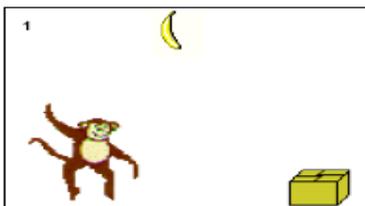
# Computermodelle des Problemlösens

## ADD-DEL Listen

### Aktueller Zustand

Pos(Affe, Links)  
Auf-boden  
Pos(Kiste, rechts)  
Pos(Banane, mitte)

Ort(links)  
Ort(mitte)  
Ort(rechts)



### Operatoranwendung

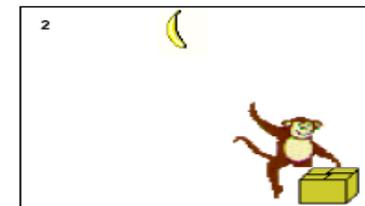
GeheVonNach(Links, Rechts)

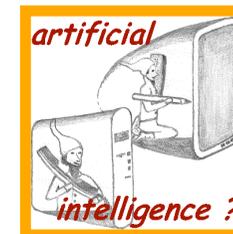


### Folgezustand

~~Pos(Affe, Links)~~  
Auf-boden  
Pos(Kiste, rechts)  
Pos(Banane, mitte)  
Pos(Affe, rechts)

Ort(links)  
Ort(mitte)  
Ort(rechts)





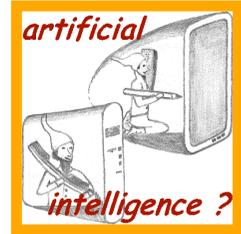
# Computermodelle des Problemlösens

## Problemraum

Die Formulierung eines Problems durch Anfangszustand, Zielzustand und Operatoren definiert einen Problemraum (Newell & Simon, 1972)

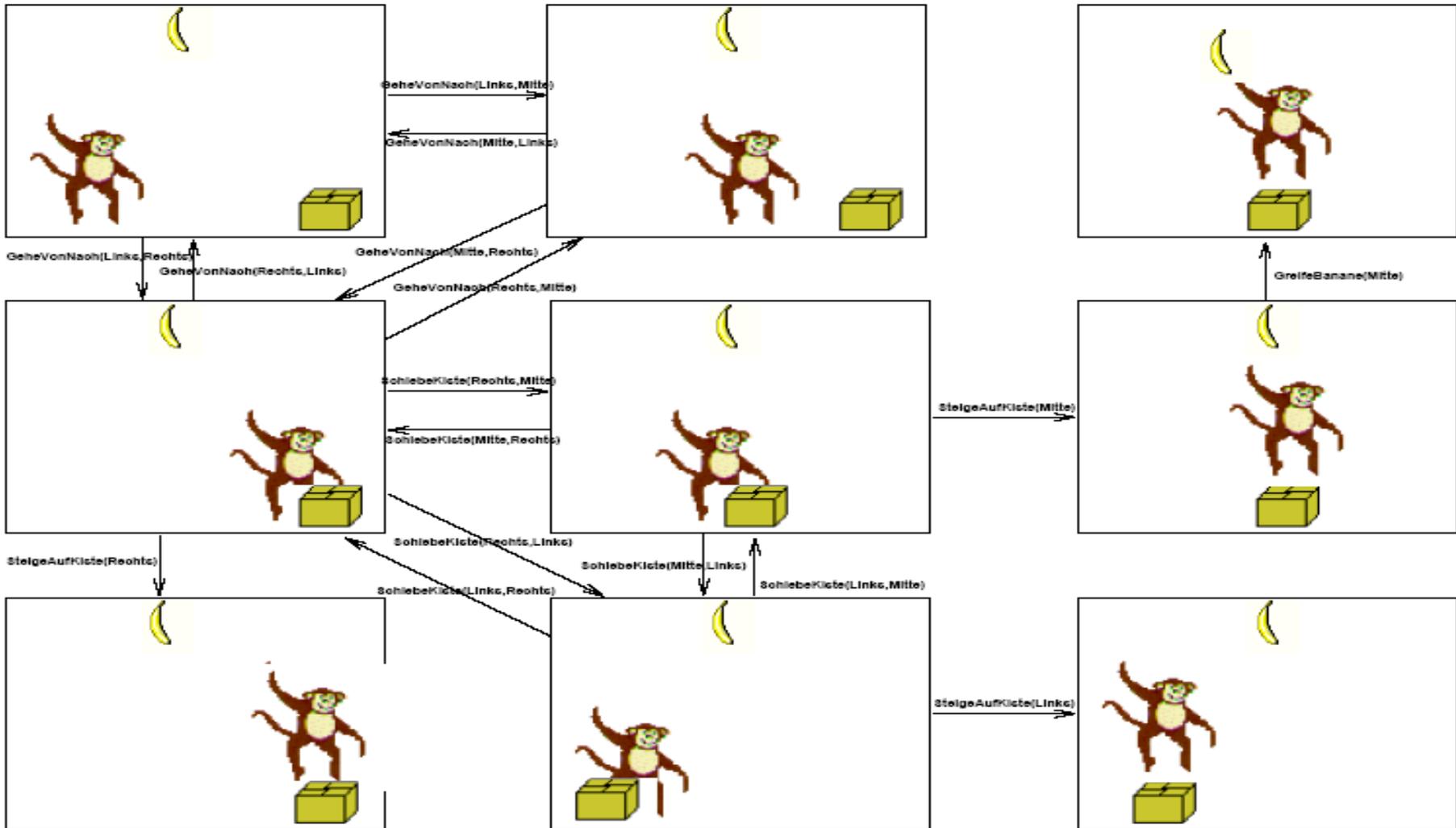
Allgemein ist eine Problemlösung definiert als eine Abfolge von Operatoren, die einen gegebenen Anfangszustand in den gewünschten Zielzustand überführt

1. GeheVonNach (Links, Rechts)
2. SchiebeKiste (Rechts, Mitte)
3. SteigeAufKiste (Mitte)
4. GreifeBanane (Mitte)



Computervermodelle des Denkens und Problemlösens

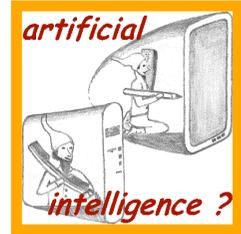
Problemraum





**Ruprecht-Karls-Universität Heidelberg**  
**Psychologisches Institut**

**Computermodelle des Denkens und Problemlösens**



# Suchstrategien

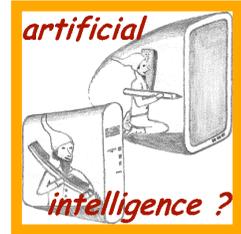
Auf einen gegebenen Zustand sind im allgemeinen mehrere Operatoren anwendbar → welcher angewandt wird, wird durch eine Suchstrategie gesteuert.

- Uninformierte Suche
- Hill Climbing und Bewertungsfunktion
- Mittel-Ziel-Analyse



**Ruprecht-Karls-Universität Heidelberg**  
**Psychologisches Institut**

**Computermodelle des Denkens und Problemlösens**

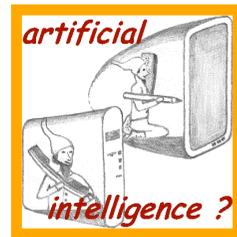


# Uninformierte Suche

- Trial and error → Zufällige Auswahl von Operatoren

## Bsp.: Tiefensuche (Winston, 1992)

- Ordnung der Operatoren
- In jedem Zustand wird erster anwendbarer Operator gewählt
- Sackgasse → Backtracking → nächstmöglicher Operator
- Mechanismus zur Erkennung und Vermeidung von Zyklen



# Tiefensuche für „Affe-Banane-Problem“

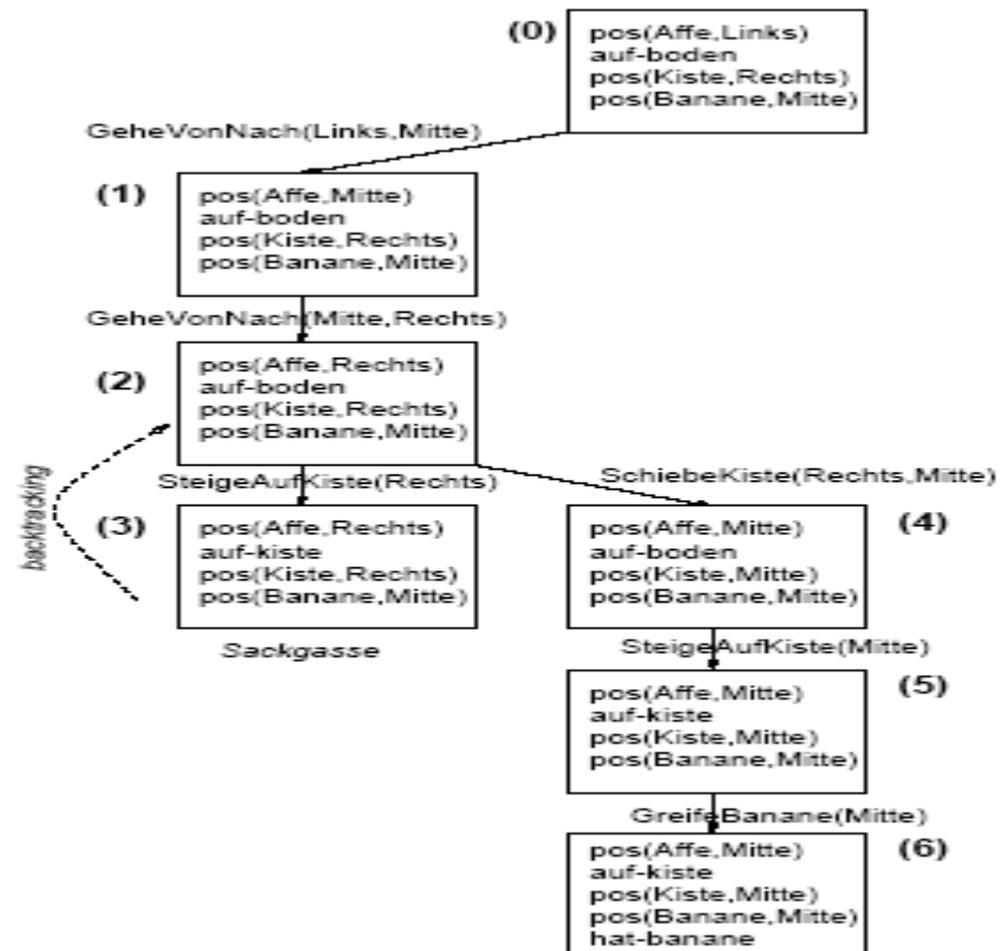
Ordnung auf den Operatoren:  
(siehe Tab. 1)

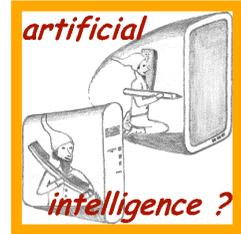
- (1) GreifeBanane,
- (2) SteigeAufKiste,
- (3) SchiebeKiste,
- (4) GeheVonNach

Bevorzugung von Orten:

- (1) Mitte,
- (2) Rechts,
- (3) Links

Die statischen Prädikate `ort(Links)`, `ort(Mitte)` und `ort(Rchts)` werden bei den Zustandsbeschreibungen der Übersichtlichkeit halber weglassen.

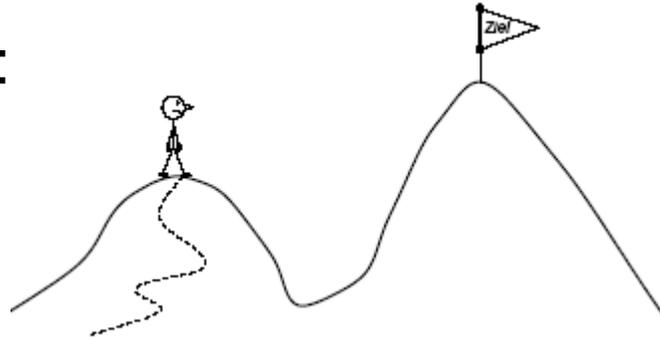




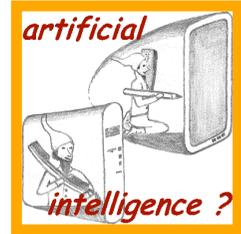
## Hill Climbing und Bewertungsfunktionen

- Am Problemziel ausgerichtetes Suchverfahren
- Basiert auf Unterschiedsreduktion → Suche durch heuristische Bewertung gesteuert
- Allerdings wird jeweils nur aktueller Zustand berücksichtigt

- Problem:



- Voraussetzung: formale Definition der Heuristik

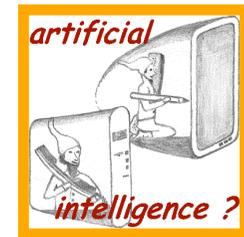


# Mittel-Ziel-Analyse

- Im Rahmen des General Problem Solvers (GPS) (Newell & Simon, 1972) entwickelt

**Grundidee: Vergleich von aktuellem Zustand mit den Problemlösezielen → Auswahl des Mittels (Operator), welcher den Unterschied zu den Zielen am stärksten reduziert.**

- Vorgehen: globales Problemlöseziel → Ableiten von Teilzielen



# Mittel-Ziel-Analyse

Mein Ziel ist, die Banane zu haben.

Um die Banane zu haben, muss ich sie greifen (den Operator greifeBanane anwenden).

Um die Banane zu greifen, muss ich

- (a) auf der Kiste stehen (den Operator SteigeAufKiste anwenden) und
- (b) die Kiste muss unter der Banane stehen.

Also ist mein neues Teilziel, die Kiste unter die Banane zu schieben (den Operator SchiebeKiste anwenden).

Um die Kiste unter die Banane zu schieben, muss ich am selben Ort sein wie die Kiste.

Also ist mein neues Teilziel, zur Kiste zu gehen.

Diese Aktion kann ich ausführen. Ich kann also jetzt zur Kiste gehen.

Das Teilziel, zur Kiste zu gehen, ist erreicht.

Jetzt kann ich die Kiste unter die Banane schieben.

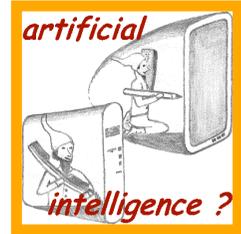
Das Teilziel, die Kiste unter die Banane zu schieben, ist erreicht.

Jetzt kann ich auf die Kiste steigen.

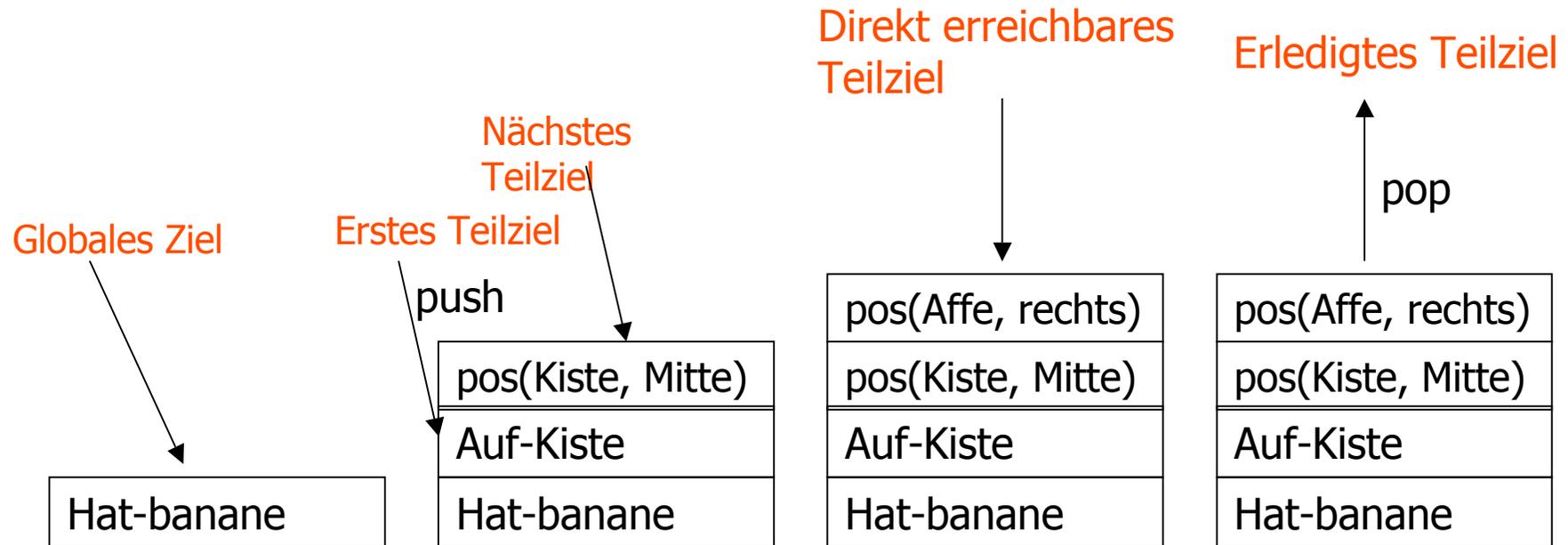
Das Teilziel, auf der Kiste zu stehen, ist erreicht.

Jetzt kann ich die Banane greifen.

Mein Ziel ist erreicht.



# Mittel-Ziel-Analyse

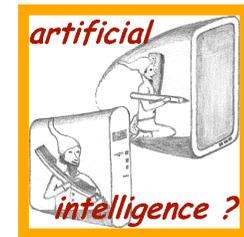


Algorithmus: Transformiere, Reduziere und Wende-An



**Ruprecht-Karls-Universität Heidelberg**  
**Psychologisches Institut**

**Computermodelle des Denkens und Problemlösens**



# Mittel-Ziel-Analyse

Transformiere: Vergleiche den aktuellen Zustand mit dem Ziel

WENN der Zustand das Ziel erfüllt

DANN halte an und melde Erfolg

SONST Reduziere den Unterschied zwischen Zustand und Ziel.

Reduziere: Finde Operator, um den Unterschied zwischen Zustand und Ziel zu verringern

WENN kein solcher Operator verfügbar ist

DANN halte an und melde Fehler

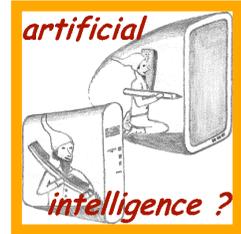
SONST Wende den gefundenen Operator auf den aktuellen Zustand an.

Wende-An: Wende einen Operator auf den aktuellen Zustand an

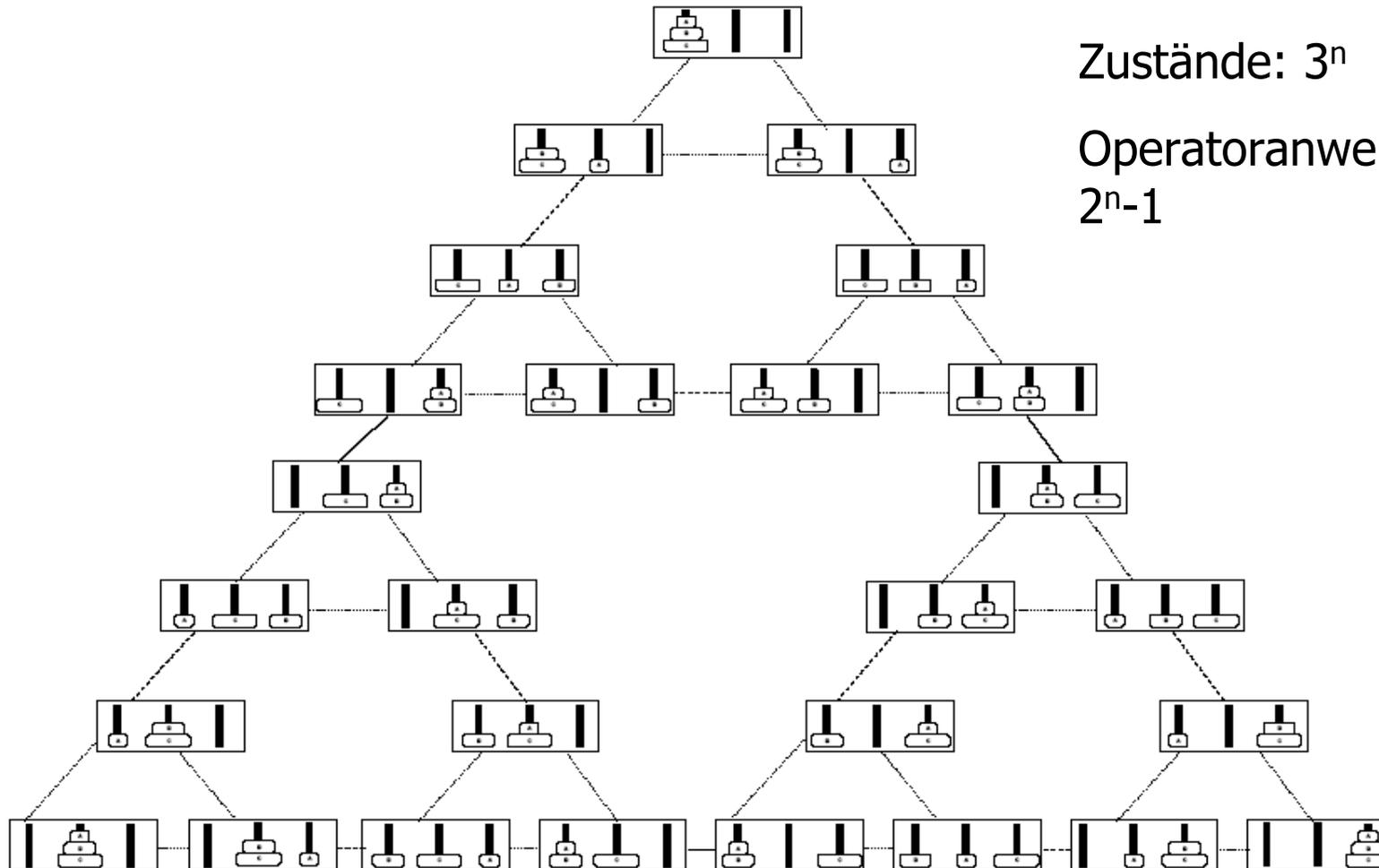
WENN der Operator auf den aktuellen Zustand anwendbar ist

DANN wende den Operator an und Transformiere den dabei entstandenen Folgezustand in das Ziel.

SONST Reduziere den Unterschied zwischen dem Zustand und den Anwendungsbedingungen des Operators.

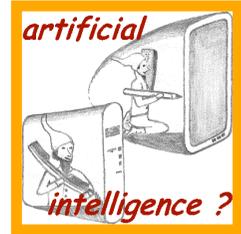


# Rekursive Probleme – Der Turm von Hanoi



Zustände:  $3^n$

Operatoranwendungen:  
 $2^n - 1$



## Rekursive Probleme – Der Turm von Hanoi

- Ermittlung der optimalen Abfolge von Aktionen durch folgende rekursive Vorschrift:

hanoi(n, von, via, nach)=

Wenn  $n=0$

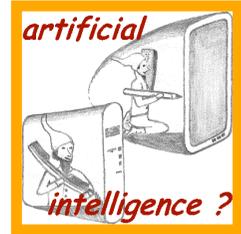
Dann tue nichts

Sonst

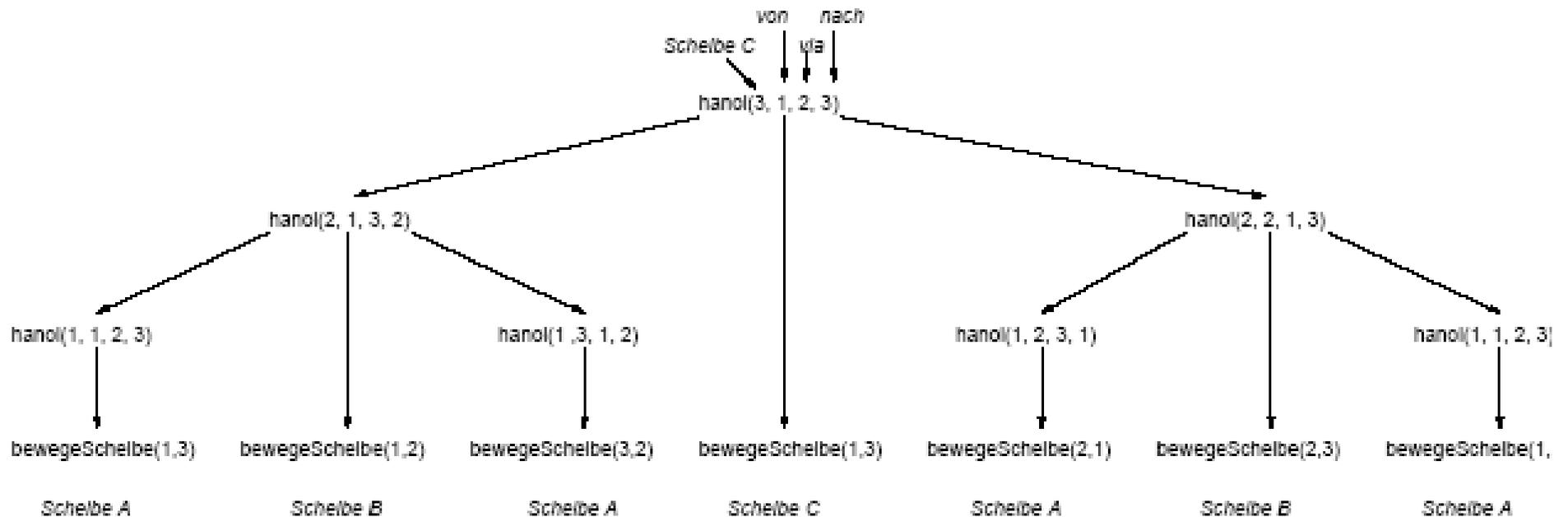
hanoi(n-1, von, nach, via)

bewege(Scheibe(von, nach))

hanoi(n-1, via, von, nach)



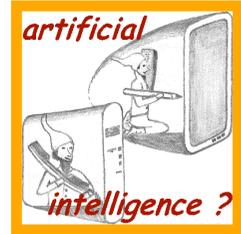
# Rekursive Probleme – Der Turm von Hanoi





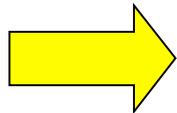
**Ruprecht-Karls-Universität Heidelberg**  
Psychologisches Institut

**Computermodelle des Denkens und Problemlösens**

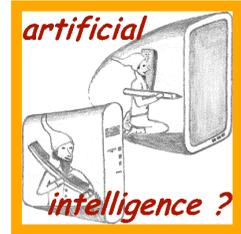


## Rekursive Probleme

Wichtig: Bei Rekursiven Definitionen ist es wichtig, dass es einen Fall gibt, der direkt ermittelt werden kann → da man sonst nie zu einem Abschluss kommen würde



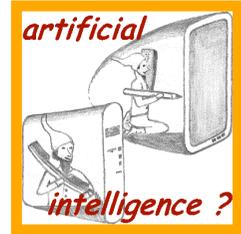
Nur Rekursive Probleme sind algorithmisch zu lösen → von Computern lösbar



# Analoges Problemlösen

**Bisher wurde Problemlösen als Suche im Problemraum modelliert.**

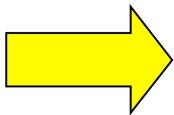
- → In vielen Fällen weist jedoch ein Problem Ähnlichkeit zu einem bereits bekanntem Problem auf.
- → Wird dies erkannt → Anwendung des bekannten Lösungsweg auf neues Problem
- Annahme: Wissensbereiche werden als relationale Strukturen beschrieben (Bsp. Sonnensystem), bestehend aus: Objekten, Attributen und Relationen



# Erwerb von Problemlösefertigkeiten

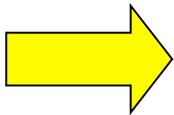
## Learning by doing

1. Verstärkung und Kombination von Regeln

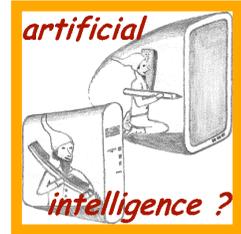


**Lernen als Optimierungsprozess**

2. Erwerb von Problemlösestrategien als Aufbau neuer Regeln oder Schemata



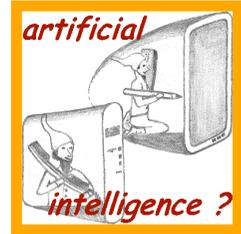
**Aufbau von Wissen über Lösungsstruktur**



# Problemlösen und Wissen

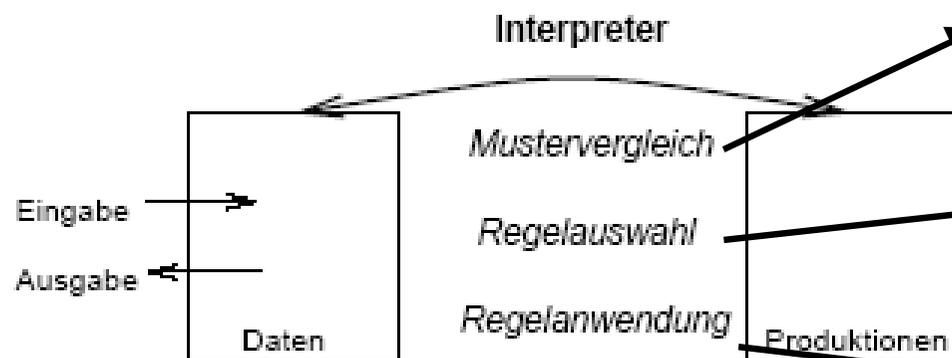
- KI versuchte v.a. in 80er Jahren Expertenwissen in Expertensystemen umzusetzen
  - Mycin → medizinische Diagnose
  - Deep Blue → Schachcomputer

Expertensysteme basieren im Wesentlichen auf formal repräsentiertem Wissen über einen Bereich und ist in der Lage mit Hilfe von Deduktionstechniken Schlüsse aus diesem Wissen abzuleiten.



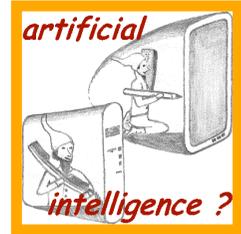
# Produktionssysteme

Ein System, das die Anwendung von Regeln auf Daten steuert heißt Produktionssystem



- **Match:** Suche alle Produktionsregeln, deren Bedingungsteil mit den Daten verträglich ist.
- **Select:** Wähle (nach einer vorgegebenen Regel) eine dieser Regeln aus
- **Apply:** Wende die Regel auf die Daten an

- Bsp: ACT



# Computermodelle des Denkens

Schlussfolgerndes Denken bezeichnet einen Prozess bei dem aus gegebenen Fakten Schlüsse gezogen werden.

Wissensbasis

- Veränderbar

Inferenzmechanismus

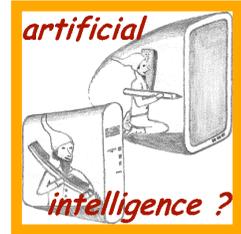
- allgemeingültig

→ Deduktive Inferenz

→ Induktive Inferenz

→ Abduktive Inferenz

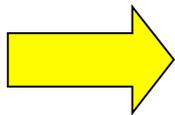
a: Deduktion Alle Menschen sind sterblich Sokrates ist ein Mensch <b>Also ist Sokrates sterblich</b>	Axiom Fakt / Beobachtung <b>Schluss</b>
b: Induktion Sokrates ist Mensch Sokrates ist sterblich <b>Alle Menschen sind sterblich</b>	Hintergrundwissen Beobachtung <b>Hypothese</b>
c: Abduktion Alle Menschen sind sterblich Sokrates ist sterblich <b>Sokrates ist Mensch</b>	Theorie Beobachtung <b>Diagnose</b>



# Grundlagen der logischen Deduktion

In der Psychologie gibt es zahlreiche Belege dafür, dass Menschen ihre Schlüsse nach Gesetzen der logischen Deduktion führen.

- Voraussetzung um Inferenzen auf einer Maschine ablaufen lassen zu können sind s.g. Kalküle.



Kalkül: Menge aus syntaktischen Transformationsregeln, die für eine bestimmte Sprache definiert sind.

Bsp.: Kaffeebohnenproblem

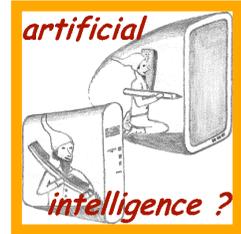
```

W W S S W W S S
  W W S S W S S
    W W S S S S
      W S S S S
        W S S S S
          W S S
            S S
              W

```

Gegeben sind folgende Regeln:

1.  $SW \rightarrow S$
2.  $WS \rightarrow S$
3.  $SS \rightarrow W$

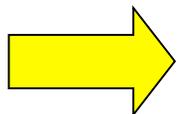


# Logische Repräsentationen

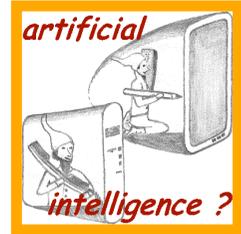
## Eine Logik ist eine formale Sprache

- Ausdrücken wird Bedeutung zugewiesen  
→ Zu einer Logik gehört also eine Menge an bedeutungszuweisenden Regeln
- Für eine Logik muss Beweiskalkül existieren, mit dem aus gegebenen Aussagen, neue Aussagen abgeleitet werden können

Der Hauptgewinn einer formal repräsentierten Logik ist, dass eindeutig festgelegt werden kann, wie logisch korrekte Schlussfolgerungen gezogen werden können.



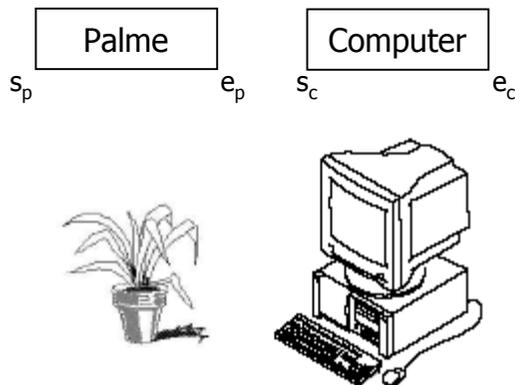
Entwickeln von Algorithmen, welche Schlussfolgern automatisch ablaufen lassen



# Qualitatives Schließen

Menschliches Schließen basiert häufig auf anderen Strategien als dem Anwenden von logischen Regeln.

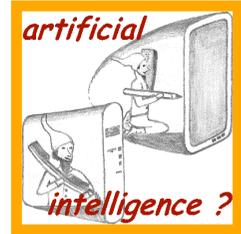
- Häufig scheinen Schlüsse auf der Basis von bildhaften Vorstellungen gezogen zu werden.
- Beleg: Symbol-Distanz-Effekt
  - Verifikation durch Betrachtung eines mentalen Bildes



## Allen-Kalkül

- Objekt als Intervall mit Anfangs- u. Endpunkt
- Informationen des Symbol-Distanz-Effekt können als räumliche Relation bspw. „links-von“ dargestellt werden.

Allen-Kalkül menschlichem Schließen ähnlicher, da über anschauliche Repräsentation



## **Schlussbetrachtung / Fazit**

- Wesentliche Grundlagen für die Modellierung von Denk- und Problemlöseprozesse mit dem Computer sind **Suchverfahren und Logik**
- Formale Grundbausteine der KI-Programme haben **wenig** mit den Prozessen **gemein**, welche menschliches Problemlösen steuern
- KI hilft uns zu verstehen, welche **Arten** der **Intelligenzleistung algorithmisch** zu lösen sind
- Grundkenntnisse in Logik und Suchverfahren → **Entwickeln alternativer Problemlöseverfahren**