

Grundlagen der Künstlichen Intelligenz

Wintersemester 25/26

Vorlesung 11

Suche, Reasoning, Planning, Logik, Wissen

Meiste andere Vorlesungen

Reflex

- Direktes Mapping von Input zu Output

Reflex

- Direktes Mapping von Input zu Output



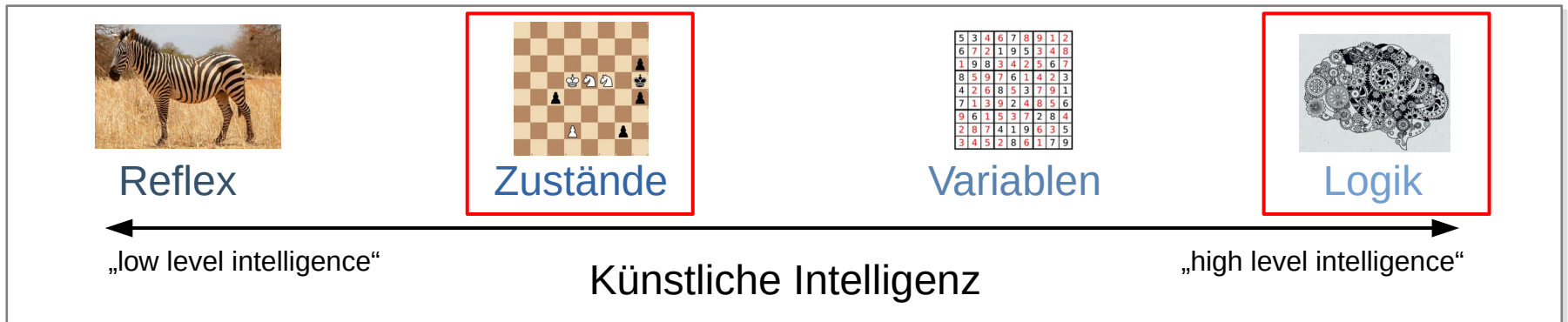
Anwendungen

- Vorhersage (Klassifikation, Regression)
- Computer Vision: Klassifikation, Objekterkennung, ...
- Sprache: Speech-to-text, Übersetzung, ...

Heute

Suche

- Rationale Planung mit klarem Ziel
- Welt modelliert als Zustände



Anwendungen

- Spiele (Schach, Go, Pacman, ...)
- Allgemein: Planung, Strategie
- Robotik: Bewegungsplanung
→ Planung durch Zustände und Aktionen

Übersicht

Agenten die planen und handeln

- Rationalität
- Agentendesign

Suchprobleme

- Umgebungen mit Zustandsräumen
- Zustandsräume definieren

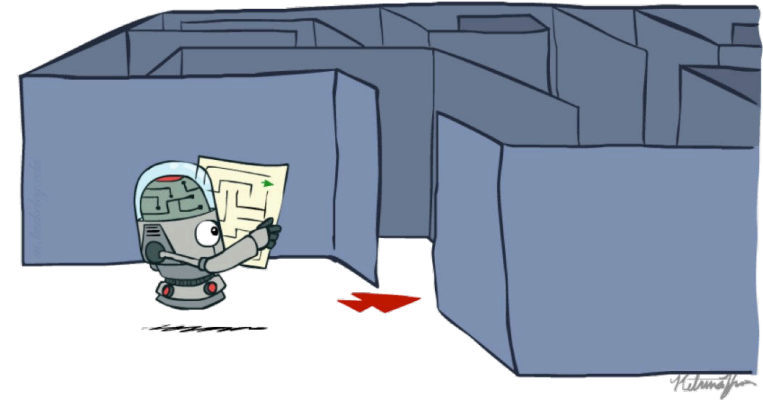
Uninformierte Suchverfahren

- Tiefensuche („Depth-First Search“)
- Breitensuche („Breadth-First Search“)
- Uniform-Cost Suche („Uniform-Cost Search“)

Heuristische Suchverfahren

- A* Tree Suche

Logik- und Wissens-Repräsentationen



Teil 2

Agenten die planen und handeln

- Rationalität
- Agentendesign

Suchprobleme

- Umgebungen mit Zustandsräumen
- Zustandsräume definieren

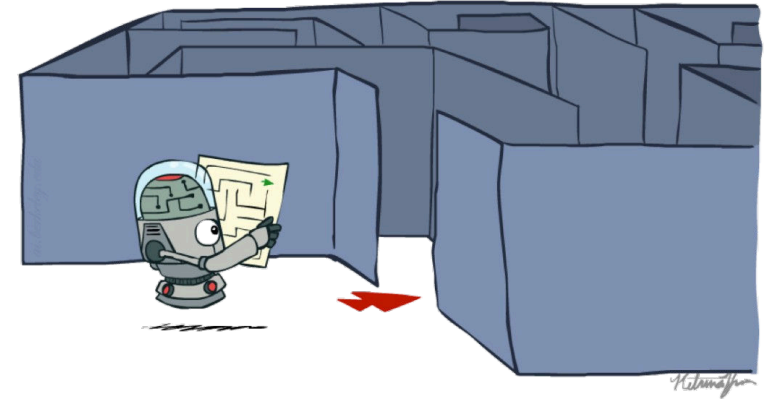
Uninformierte Suchverfahren

- Tiefensuche („Depth-First Search“)
- Breitensuche („Breadth-First Search“)
- Uniform-Cost Suche („Uniform-Cost Search“)

Heuristische Suchverfahren

- A* Tree Suche

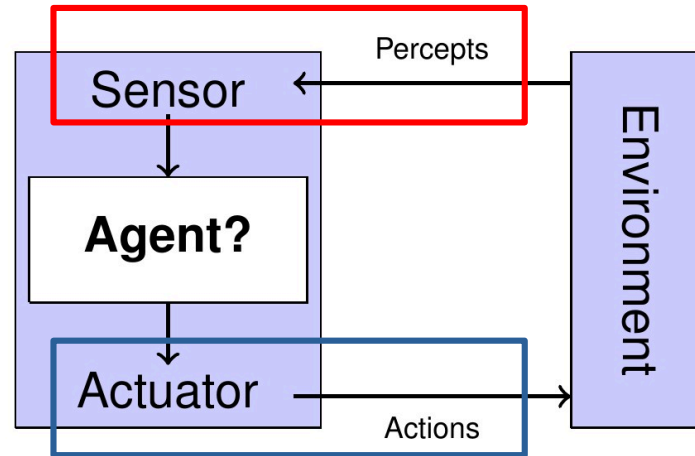
Logik- und Wissens-Repräsentationen



Autonomie

Ein **autonomer** Agent ...

- ... **nimmt** seine Umgebung mit **wahr**: **Input**
- ... **agiert** mit der Umgebung: **Aktionen**



Kreislauf

- Aktionen beeinflussen die Umwelt ...
- ... was sich auf den **Input** auswirkt ...
- ... was sich auf die **Aktionen** auswirkt ...

Rationalität

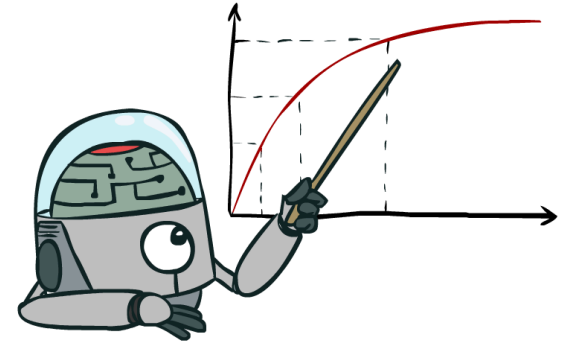
Ein **rationaler Agent** wählt Handlungen, die seinen erwarteten **Nutzen** (“utility”) **maximieren**.

Suche

- Agenten haben ein Ziel und Kosten
- Beispiel: Erreiche das Ziel mit geringsten Kosten

Reinforcement Learning

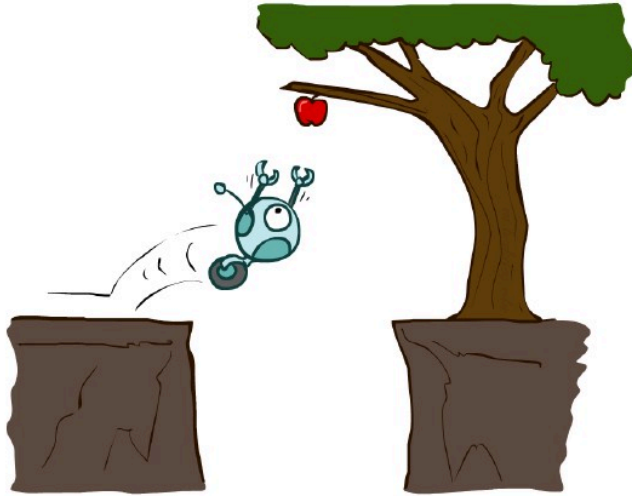
- Agenten haben einen numerischen Nutzen und Belohnungen
- Beispiel: Maximiere die Gesamtbelohnung
- Belohnung: Größter Gesamtgewinn oder erwarteter Gesamtgewinn



Maximiere den Nutzen!

Planende Agenten und Reflex-Agenten

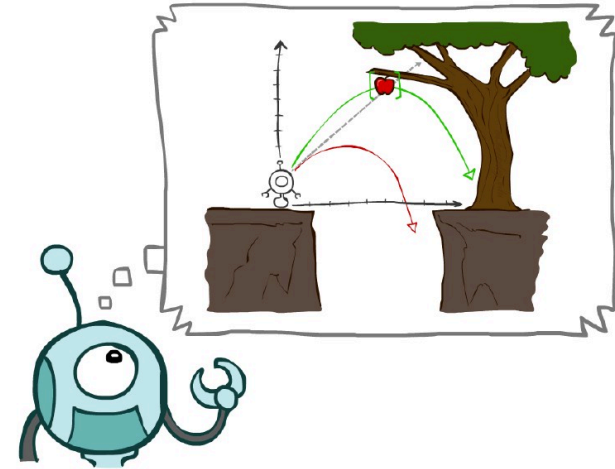
Reflex



Reflex Agenten ...

- ... wählen eine Handlung auf Grundlage der aktuellen Wahrnehmung (plus ggf. Erinnerung).
- ... bedenken nur, wie die Welt **ist**.
- ... denken **nicht** an die zukünftigen Konsequenzen ihres Handelns.

Planung



Planende Agenten...

- ... entscheiden auf der Grundlage von (hypothetischen) Handlungsfolgen.
- ... müssen ein Modell für Auswirkungen von Handlungen haben und wie die Welt **sein würde**.
- ... müssen ein Ziel formulieren.

Teil 2

Agenten die planen und handeln

- Rationalität
- Agentendesign

Suchprobleme

- Umgebungen mit Zustandsräumen
- Zustandsräume definieren

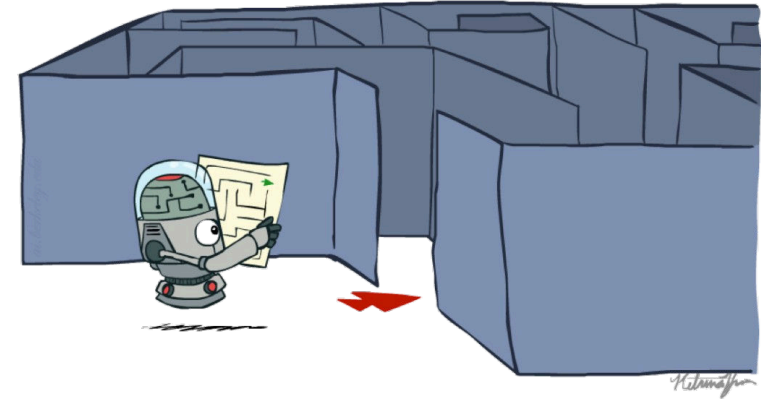
Uninformierte Suchverfahren

- Tiefensuche („Depth-First Search“)
- Breitensuche („Breadth-First Search“)
- Uniform-Cost Suche („Uniform-Cost Search“)

Heuristische Suchverfahren

- A* Tree Suche

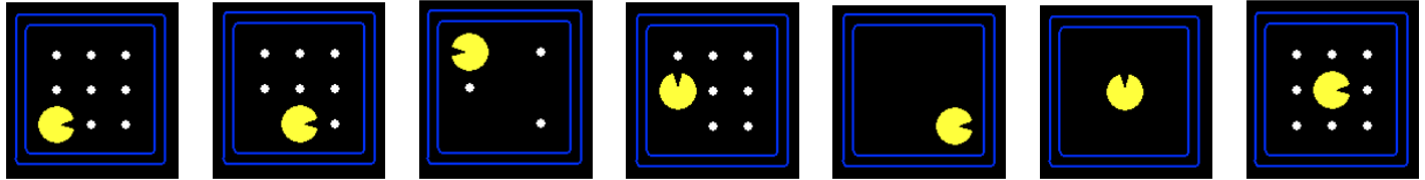
Logik- und Wissens-Repräsentationen



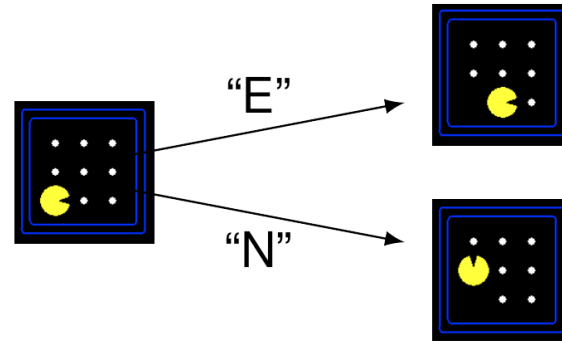
Suchprobleme

Suchprobleme bestehen aus

- Zustandsraum



- Nachfolgerfunktion (mit Aktionen, Kosten)

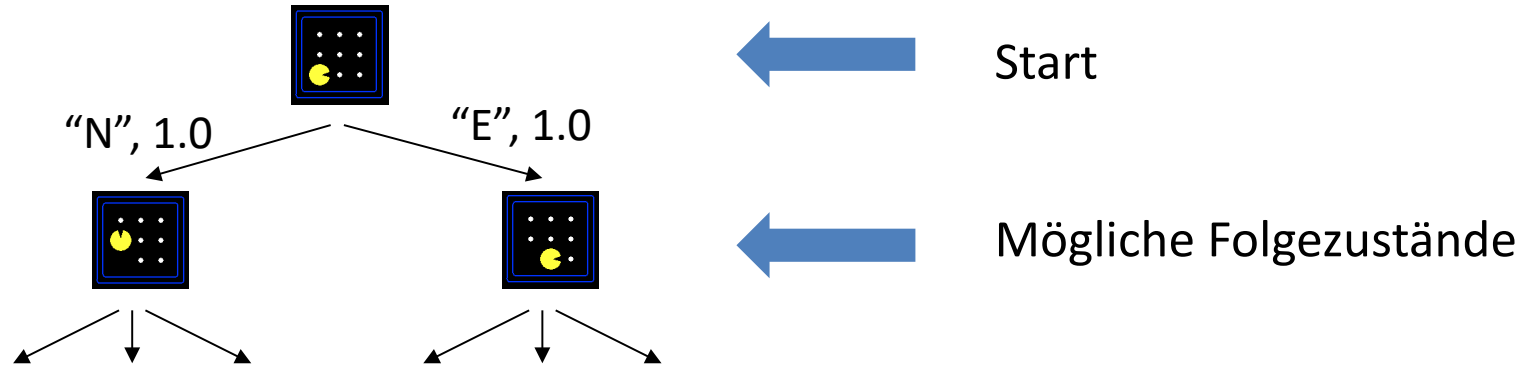


- Ein Startzustand und ein **Zieltest**: $\text{State} == \text{Goal} ?$

Lösung von Suchproblemen

- Abfolge von Aktionen (ein Plan), die den Startzustand zu einem Zielzustand (Zieltest ist positiv) überführt.

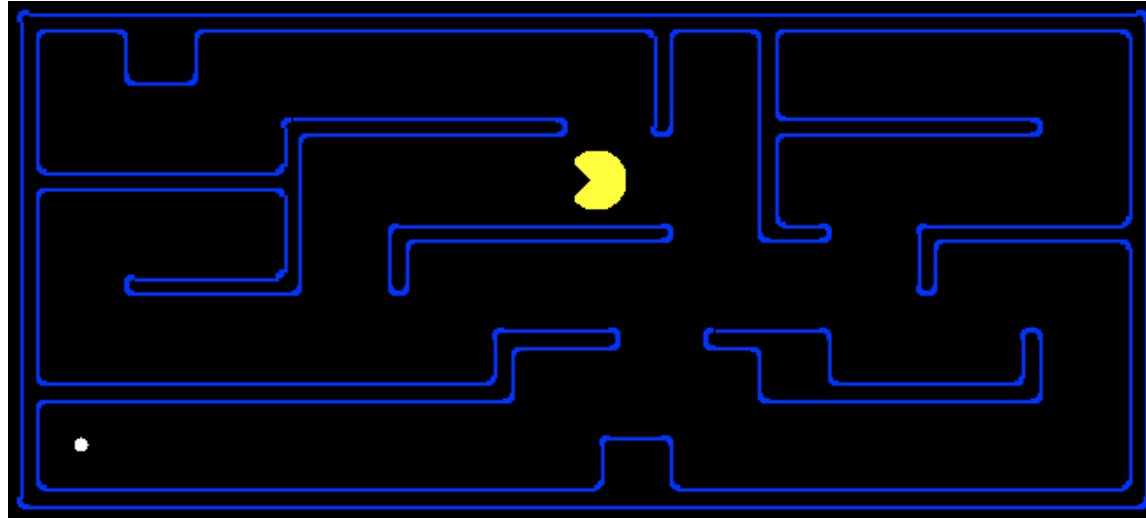
Zustandsraumdarstellung als **Suchbaum**



Ein Suchbaum

- Ein "Was-wäre-wenn"-Baum von Pfaden
- Der Wurzelknoten ist der Startzustand
- Nachfolger entsprechen den möglichen Folgezuständen
- Wichtig: Knoten sehen aus wie Zustände, entsprechen aber in Wirklichkeit Pfaden, um diese Zustände zu erreichen
 - Mehrere identische Knoten im Suchbaum möglich: Diese entsprechen unterschiedlichen Pfaden um diesen Zustand zu erreichen.

Wie finde ich das Ziel? Was ist der kürzeste Weg?



Kennt ihr Strategien/Algorithmen?

Teil 3

Agenten die planen und handeln

- Rationalität
- Agentendesign

Suchprobleme

- Umgebungen mit Zustandsräumen
- Zustandsräume definieren

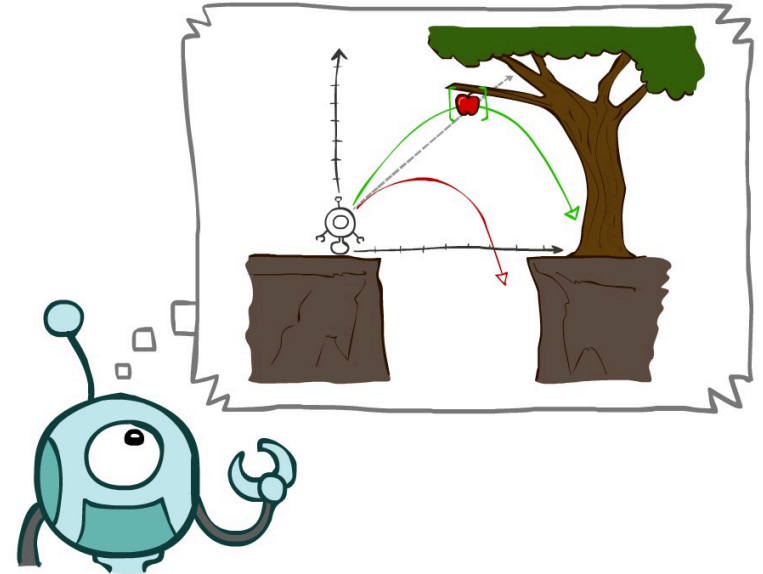
Uninformierte Suchverfahren

- Tiefensuche („Depth-First Search“)
- Breitensuche („Breadth-First Search“)
- Uniform-Cost Suche („Uniform-Cost Search“)

Heuristische Suchverfahren

- A* Tree Suche

Logik- und Wissens-Repräsentationen



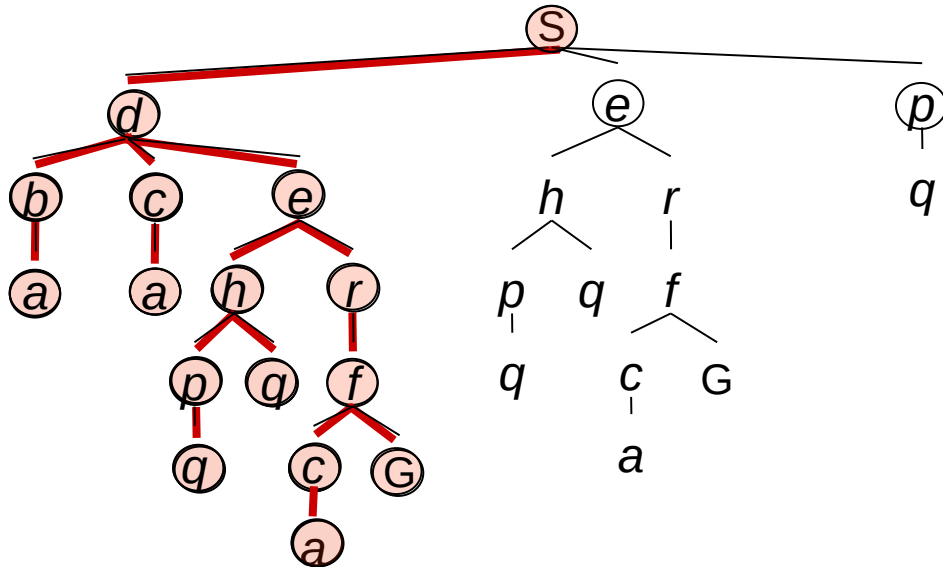
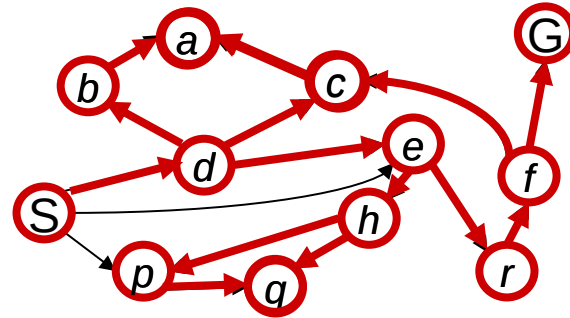
Depth-First Suche (Tiefensuche)

Strategie

- Zuerst den tiefsten und neusten Knoten erweitern

Umsetzung

- Fringe ist ein LIFO-Stack (Last-In First-Out)



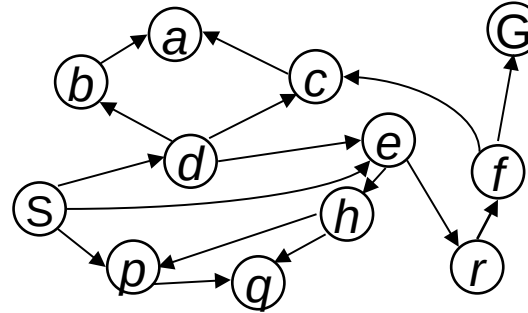
Anmerkung

- Knoten in umgekehrter alphabetischer Reihenfolge zum Fringe hinzufügen:
- → "p", "e", "d" in der ersten Runde
- LIFO-Stapel: Expandiere dann zunächst "d"

Breadth-First Suche (Breitensuche)

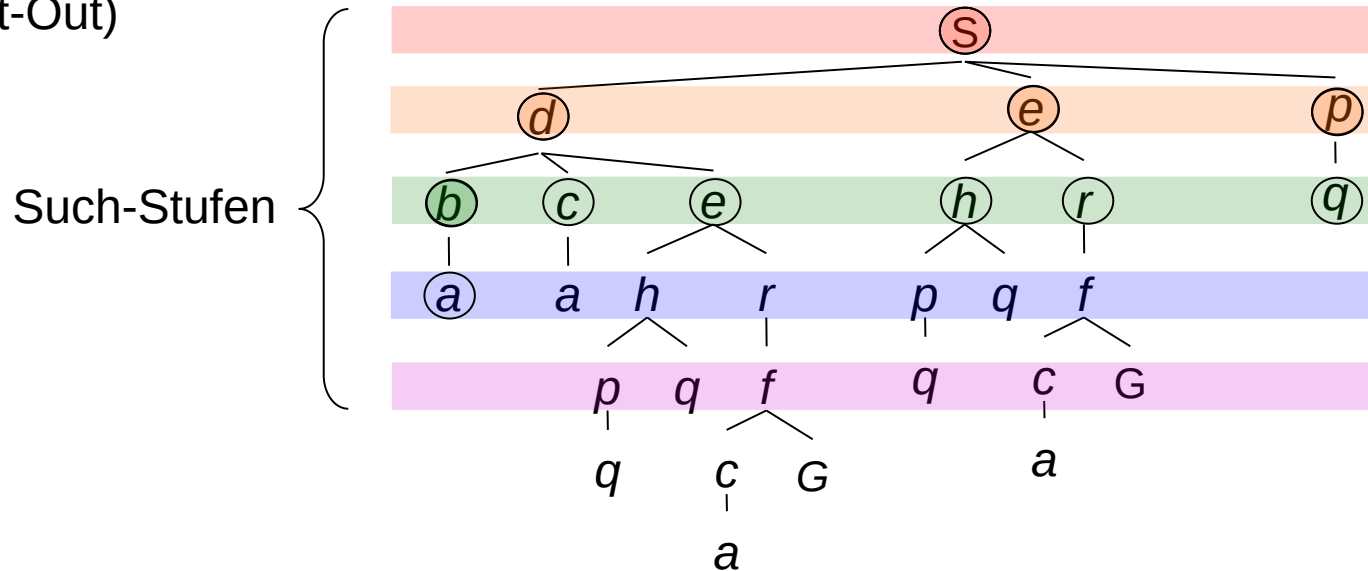
Strategie

- Zuerst den flachsten Knoten erweitern



Umsetzung

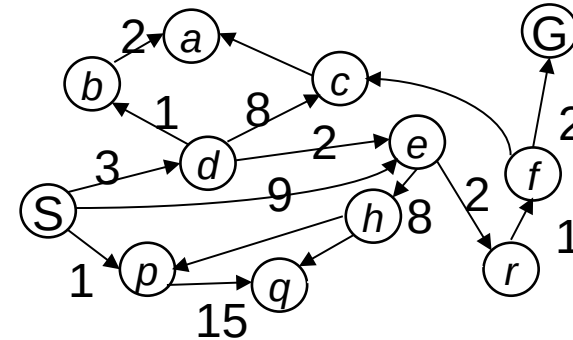
- Fringe ist ein FIFO-Stack (First-In-First-Out)



Uniforme Kostensuche (Uniform Cost Search, UCS)

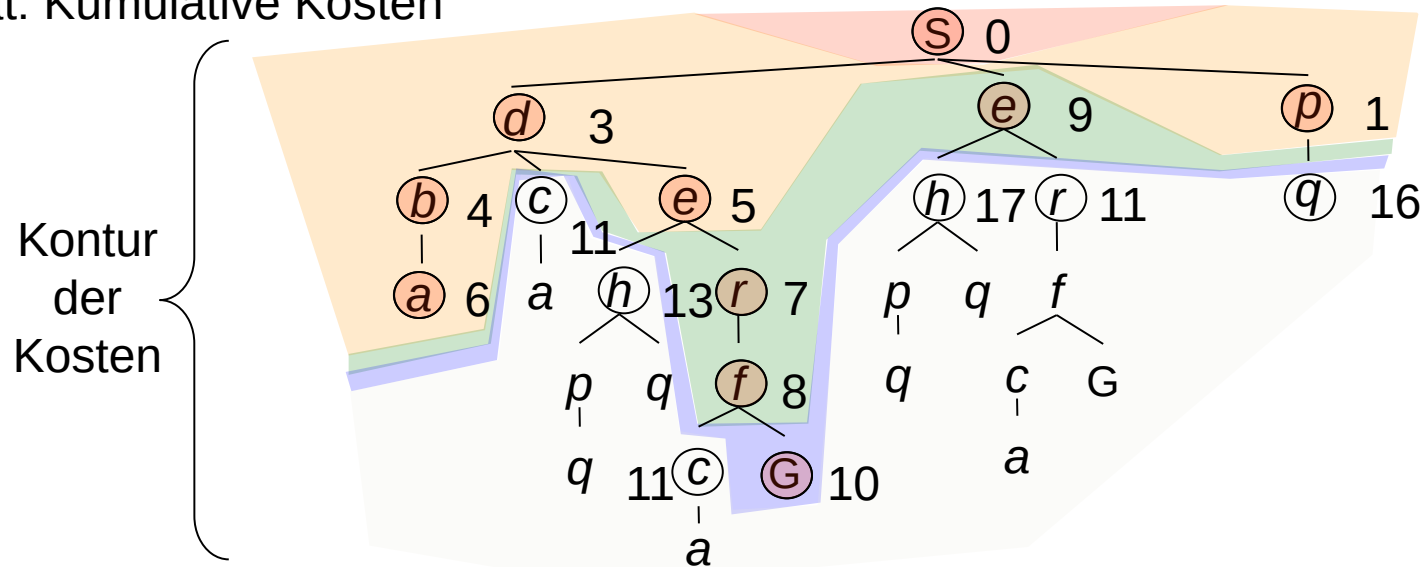
Strategie

- Zuerst einen Knoten mit niedrigsten Kosten erweitern



Umsetzung

- Fringe ist ein Prioritäts-Stack
- Priorität: Kumulative Kosten



Teil 4

Agenten die planen und handeln

- Rationalität
- Agentendesign

Suchprobleme

- Umgebungen mit Zustandsräumen
- Zustandsräume definieren

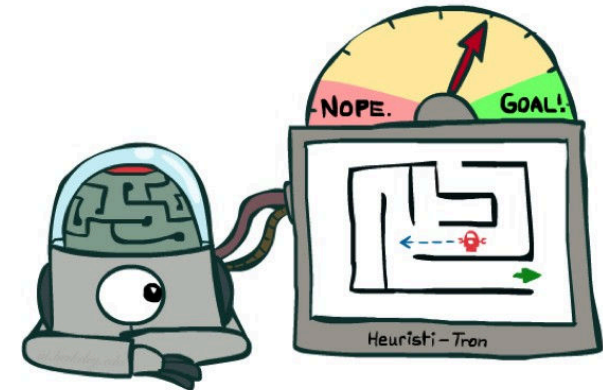
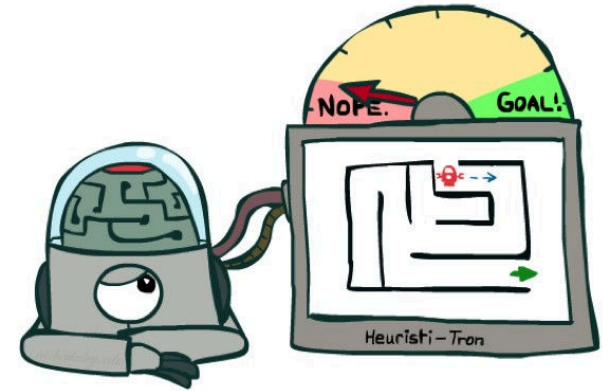
Uninformierte Suchverfahren

- Tiefensuche („Depth-First Search“)
- Breitensuche („Breadth-First Search“)
- Uniform-Cost Suche („Uniform-Cost Search“)

Heuristische Suchverfahren

- A* Tree Suche

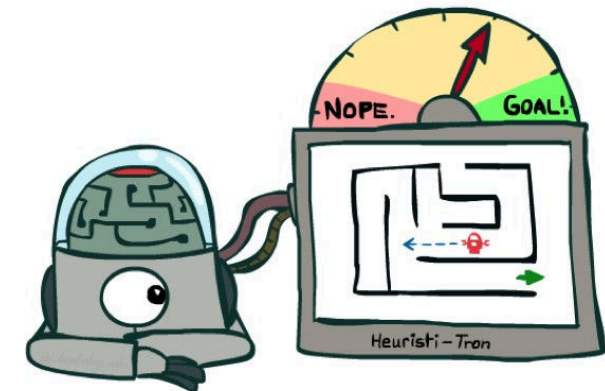
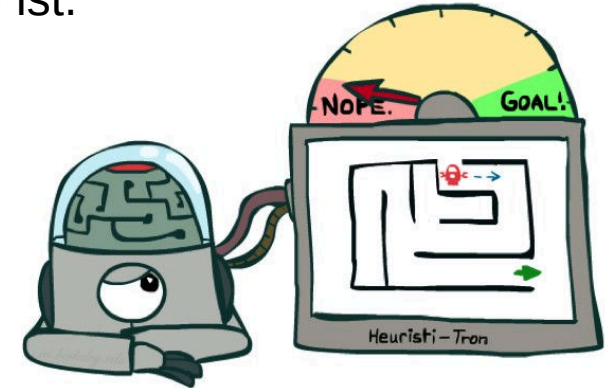
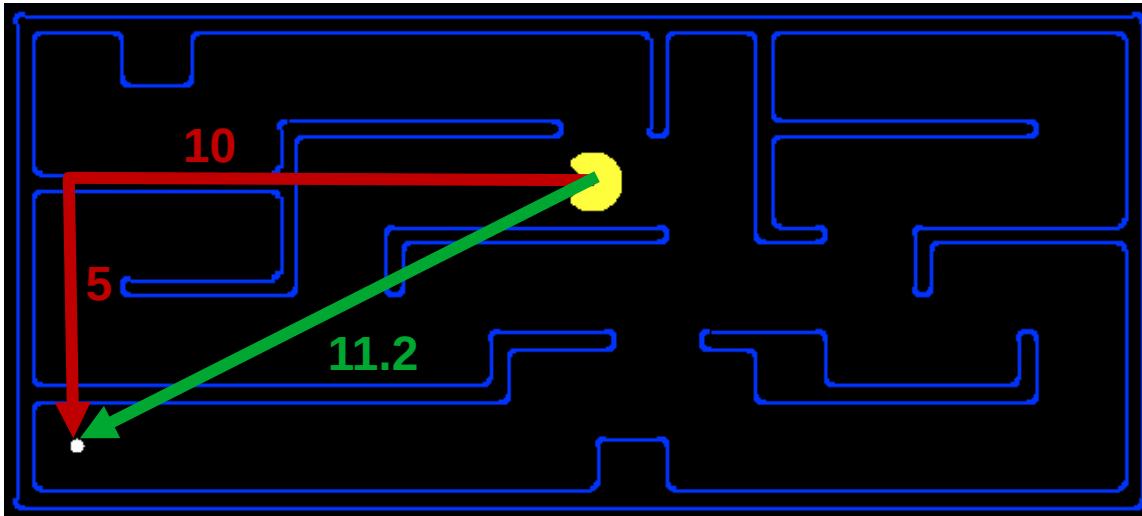
Logik- und Wissens-Repräsentationen



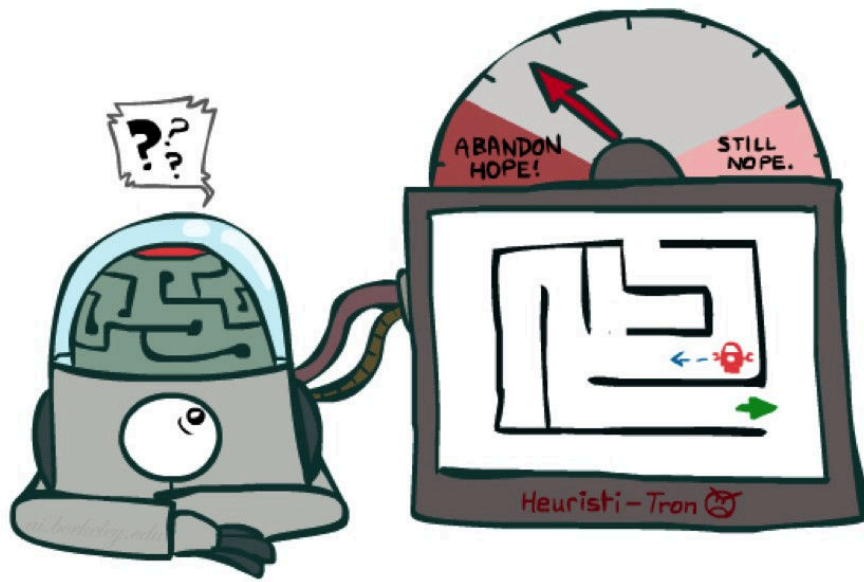
Such-Heuristiken

Eine Heuristik $h(x)$ ist ...

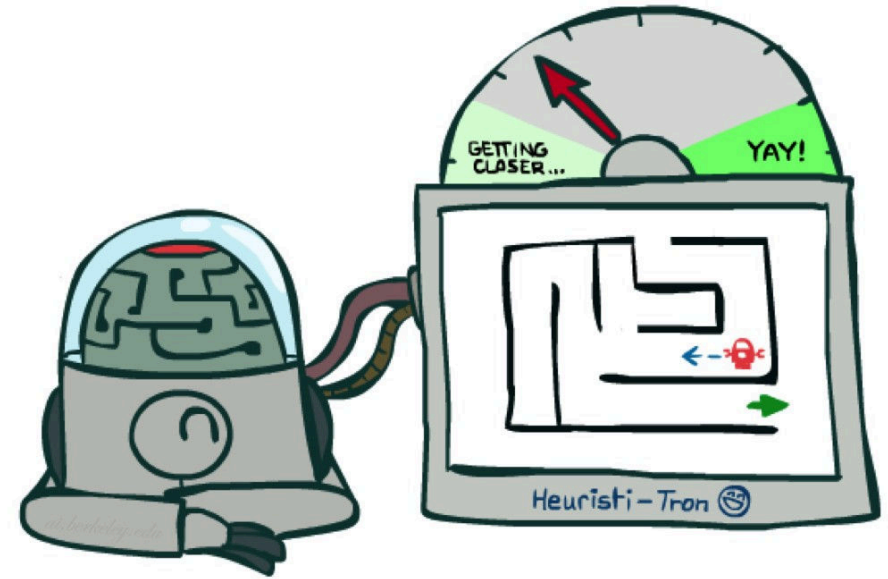
- ... eine Funktion, die schätzt, wie nahe ein Zustand am Ziel ist.
- ... spezifisch für ein bestimmtes Suchproblem.
- Beispiel Wegfindung:
 - **Manhattan-Abstand** zum Ziel
 - **Euklidischer Abstand** zum Ziel



Idee: Zulässigkeit von Heuristiken (Admissibility)



Unzulässige (pessimistische) Heuristiken brechen die Optimalität, indem sie gute Pläne aus dem Fringe ausschließen



Zulässige (optimistische) Heuristiken verhindern schlechte Pläne, unterschätzen aber die tatsächlichen Kosten

A* Suche

- **A*** verwendet sowohl **Rückwärts-Kosten** als auch **(geschätzte) Vorwärts-Kosten**
- **A*** ist **optimal** mit **zulässiger** (optimistischer) Heuristik
- **Design der Heuristik** ist der Schlüssel
 - Häufig werden vereinfachte Probleme verwendet



Teil 5

Agenten die planen und handeln

- Rationalität
- Agentendesign

Suchprobleme

- Umgebungen mit Zustandsräumen
- Zustandsräume definieren

Uninformierte Suchverfahren

- Tiefensuche („Depth-First Search“)
- Breitensuche („Breadth-First Search“)
- Uniform-Cost Suche („Uniform-Cost Search“)

Heuristische Suchverfahren

- A* Tree Suche

Logik- und Wissensrepräsentationen

```
>> Hello! I am your personal assistant!  
>> How can I help you?
```

```
> All students like GKI.  
>> I learned something.
```

```
> Bob does not like GKI.  
>> I learned something.
```

```
> Is Bob a student?  
>> No.
```

Zutaten der Logik

- **Syntax:** Definiert ein Set gültiger Ausdrücke/Formeln

- Beispiel

$\text{rain} \wedge \text{wet}$

→ **Nur Symbole, keine Bedeutung!**

- **Semantik:** Bedeutung (Zuordnung von Werten → „Interpretation“/„Modell“)

- Beispiel

	rain	
	0	1
wet 0		
wet 1		

- **Logisches Schlussfolgern:** Schlussregeln, gegeben Ausdruck f , welches g können wir ableiten?

- Beispiel:

$$\frac{\text{rain} \wedge \text{wet}}{\text{wet}}$$

Knowledge Base: Beispiel

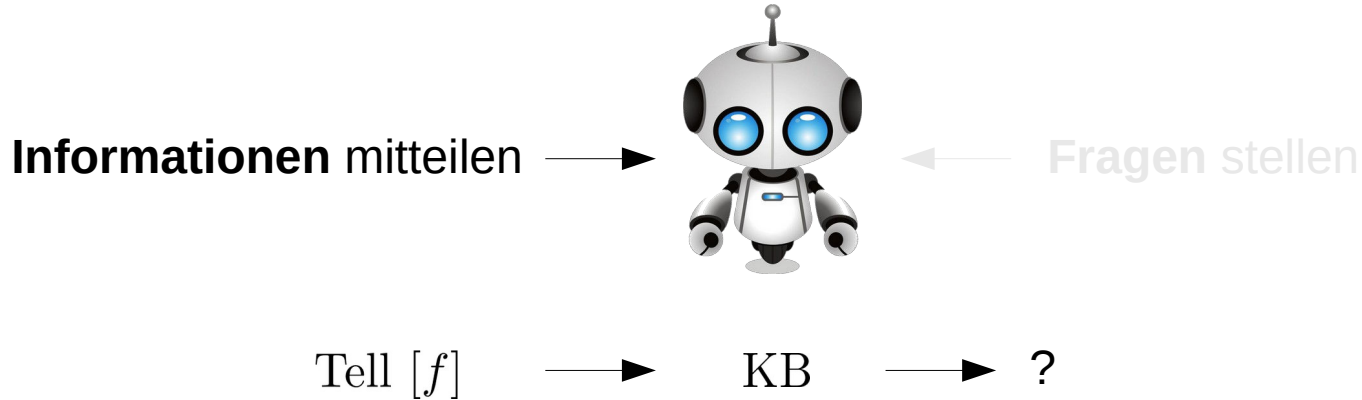
		$\mathcal{M}(\text{Rain})$				$\mathcal{M}(\text{Rain} \rightarrow \text{Wet})$	
			Wet				Wet
			0 1				0 1
Rain	0				0	1	1
	1	1	1		1		1

Schnitt:

$\mathcal{M}(\{ \text{Rain}, \text{Rain} \rightarrow \text{Wet} \})$

		Wet	
		0	1
Rain	0		
	1		1

Szenarien

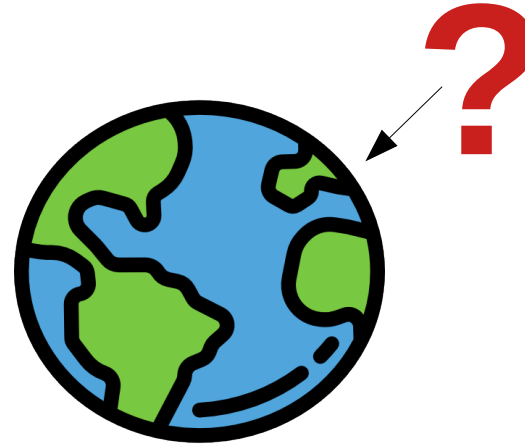


Mögliche Reaktionen

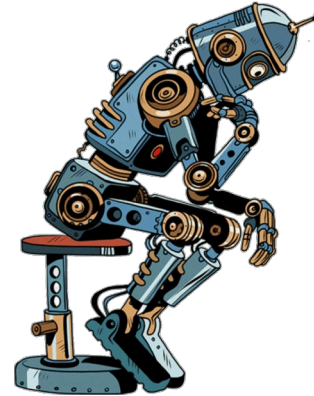
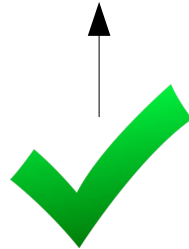
- „Das wusste ich schon.“ → Entailment/Folgerung: $KB \models f$
- „Das glaube ich nicht.“ → Widerspruch: $KB \models \neg f$
- „Ich habe etwas gelernt.“ → Contingency: Update KB

Ziele der Logik-basierten KI

- Das **Wissen** der Welt repräsentieren



- Mit dem Wissen **schlussfolgern**



Die folgenden Folien basieren auf
der Vorlesung von
Prof. Dr. Harald Sack (AIFB, KIT):
Information Service Engineering
<https://www.youtube.com/channel/UCjkkhNSNuXrJpMYZoeSBw6Q>

Formale Wissensrepräsentation

- **Formale Wissensrepräsentation ...**
 - ... ist ein Bereich der künstlichen Intelligenz
 - ... erfasst **eindeutig die Semantik (Bedeutung) von Begriffen, Eigenschaften, Beziehungen und Individuen** spezifischer Wissensdomänen in Form von **strukturierten Daten**
- **Maschinen (Computer)** müssen in der Lage sein, formale Wissensrepräsentationen zu **verstehen**
- „**Verstehen**“: Computer sind in der Lage, die Wissensrepräsentation **korrekt zu interpretieren**

- **Wie machen wir das? Durch eine Ontologie!**

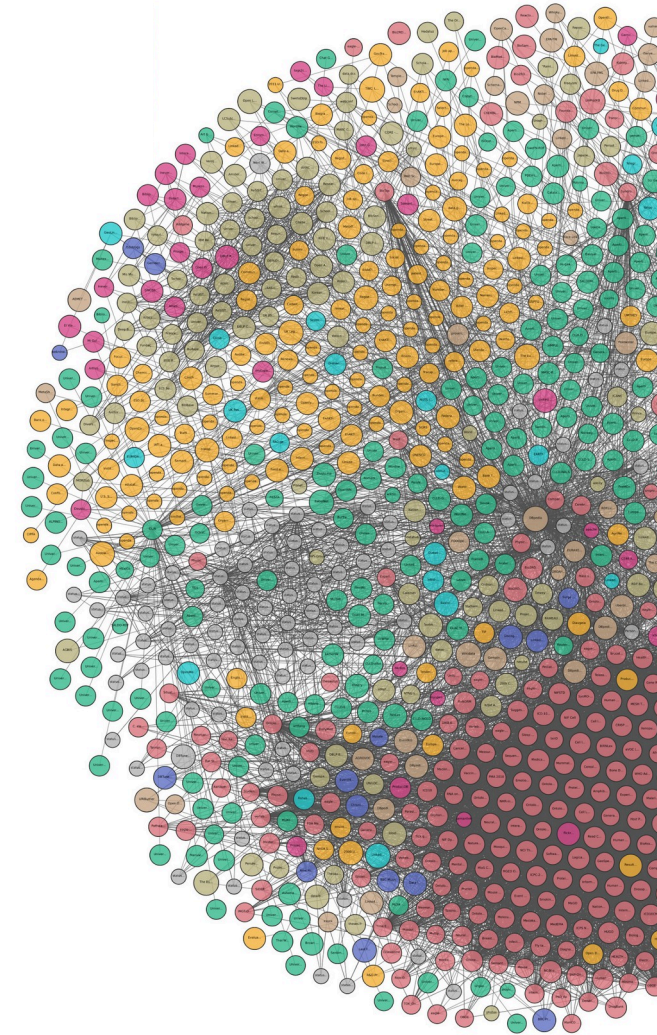


Wie kann man Ontologien darstellen?

- Ontologien lassen sich durch ein abstraktes Modell darstellen
- Dieses Modell umfasst:
 - **Klassen** (um die allgemeinen Begriffe zu beschreiben)
 - **Attribute** (Eigenschaften von Dingen)
 - **Beziehungen** (zwischen Klassen)
 - Beschränkungen/“constraints“ (welche Beziehungen sind erlaubt/verboten)
 - **Individuen/Instanzen**
 - Regeln (prozedurales Wissen, if-then-else)
 - Axiome (alle Aussagen die wir über die Domäne machen wollen)

Wissensgraphen: Definition

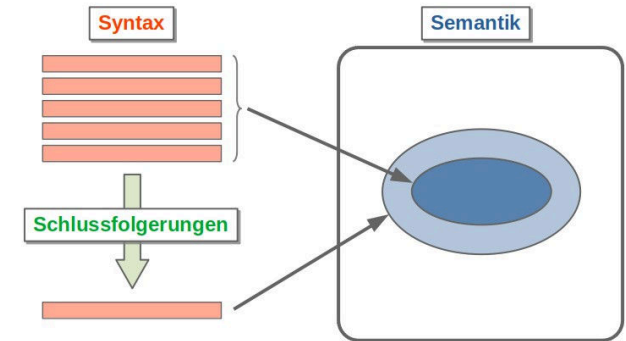
- Ein **Knowledge Graph** besteht aus **Konzepten, Klassen, Eigenschaften, Beziehungen, und Entitätsbeschreibungen**
- Basiert auf **formalen Wissensrepräsentationen** z.B. Resource Description Framework (RDF(S)), Web Ontology Language (OWL)



Zusammenfassung

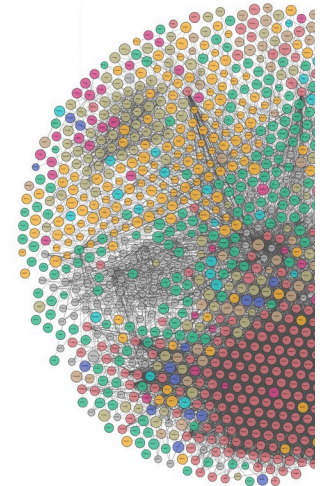
Wie können wir die Welt logisch repräsentieren?

- Syntax von Logik
- Semantik von Logik
- Logische Schlussfolgerungen
- Hier: Beschränkung auf Aussagenlogik



Wie können wir das Wissen der Welt repräsentieren?

- Was ist Wissen?
- Ontologien und ihre Darstellung
- Wissensgraphen („Knowledge Graphs“)



**Vielen Dank für eure
Aufmerksamkeit!**