

Vorlesung Künstliche Intelligenz Wintersemester 2006/07

Teil I: Überblick

Kap.1: Themen und Techniken der KI

Diese Folien basieren z. T. auf Material von Andreas Abecker, FZI
Karlsruhe, siehe <http://www.andreas-abecker.de/>

Was ist Künstliche Intelligenz?



The Arithmetical Machine produces effects which are nearer to thought than all the actions of animals. But it does nothing which would enable us to attribute will to it, as to animals.

Blaise Pascal

The brain happens to be a meat machine.

Marvin Minsky

The only way to know that a machine is thinking is to be that machine and feel oneself thinking.

Alan M. Turing

At the end of this century the way of words and general educated opinion will have altered so much that one will be able to speak of “machines tinking” without expecting to be contradicted.

Alan M. Turing, 1938



“... Die Bezeichnung “Künstliche Intelligenz” entstand als Übersetzung von “artificial intelligence”. Dieser Begriff war Mitte der 50er Jahre in den USA geboren worden. Die Übersetzung von “artificial” als “künstlich” bedeutet zugleich “unecht”, “gekünstelt”, “Schein-”. Die Programme, die in der KI entstehen, verhalten sich also für den Betrachter als ob sie Intelligenz besäßen. Auch das Wort “intelligence” besitzt im Englischen eine weitergehende Bedeutung als das Wort “Intelligenz” im Deutschen, nämlich denkbezogene Information, Einsicht und Verständnis ... “

aus: Informatik Duden

Was ist Künstliche Intelligenz?



Elaine Rich: Forschung darüber, wie Rechner Dinge machen, die Menschen derzeit noch besser beherrschen.

Barr & Feigenbaum: Teil der Informatik, in welchem intelligente Systeme entworfen werden, d.h. Systeme, die die menschliche Intelligenz „simulieren“.

Buchanan & Shortliffe: Teil der Informatik, in welchem symbolische, nicht-algorithmische Problemlösungen entwickelt werden.

Buchanan: Teil der Informatik, mit symbolischer statt numerischer Repräsentation von Wissen und mit heuristischen Verarbeitungstechniken.

Was ist Künstliche Intelligenz?



GOLIATH





Schwache KI-These:

Computer können so handeln, als wären sie intelligent.

Starke KI-These:

Hypothese des physikalischen Symbolsystems (physical symbol systems hypothesis - PSSH, [Newell, Simon 1976]): Menschliche wie maschinelle Intelligenz beruhen auf der Fähigkeit zur Symbolverarbeitung.

Aus der PSSH folgt, daß intelligente Leistungen von der Besonderheiten der physikalischen Implementation unabhängig sind. Künstliche und natürliche Intelligenz sind prinzipiell ununterscheidbar

*„The brain happens to be a
meat machine.“*
[Marvin Minsky]



Def.: Intelligentes Verhalten ist die Fähigkeit, menschliche Performanz in allen kognitiven Bereichen zu erreichen, so dass ein Interviewer getäuscht wird.

Turing Test (1950): Ein Mensch kann (nach 30 Minuten freiem Dialog) nicht entscheiden, ob sein Gegenüber ein Mensch oder eine Maschine ist (Kommunikation nur über Terminal).

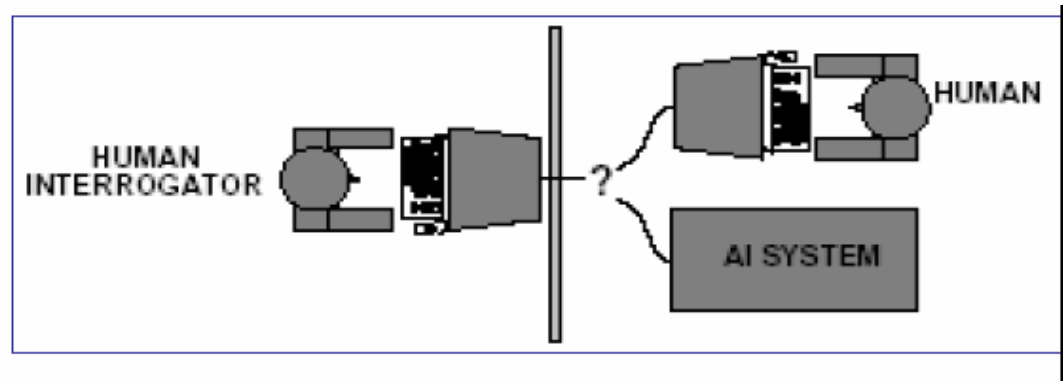
Erforderliche Fähigkeiten:

- Sprachverarbeitung

Wissensrepräsentation

Schlussfolgerungsfähigkeit

Lernfähigkeit



Der Turing-Test



Der Britische Mathematiker Alan Turing schlug bereits im Jahr 1950 einen empirischen Test zur Überprüfung intelligenten Verhaltens einer Maschine durch Vergleich mit einem Menschen (Imitationsspiel):

- Eine Person befragt einen Menschen und ein künstliches intelligentes System. Der Tester kommuniziert nur indirekt über ein Textmedium.
- Der Tester soll allein aus den Antworten auf seine Fragen entscheiden, wer Mensch und wer Computer ist.
- Wenn er das nicht kann, dann - so argumentiert Turing - kann man annehmen, daß die Maschine intelligent ist.

Der Tester kann jede beliebige Frage stellen. Durch die indirekte Kommunikation soll vermieden werden, daß der Interviewer durch äußeres Erscheinen oder mechanische Eigenschaften (z.B. synthetische Stimme) beeinflußt wird.

Erweiterung zu totalem Turing Test für Umgang mit physikalischen Objekten:

- Bildverarbeitung
- Robotik



- Seit 1990 (gestiftet von Loebner, vergeben in Cambridge/USA)
<http://www.loebner.net/Prizef/loebner-prize.html>
- Ablauf: Eine Jury kommuniziert mit 8 Programmen sowie 2 Menschen und soll beides auseinanderhalten.
- Preise:
 - Gold: 100000 \$ für natürlichsprachlichen Dialog,
 - Silber: 25000 \$ für schriftlichen Dialog,
 - Bronze: 2000 \$ für bestes Programm



Schwache Methoden:

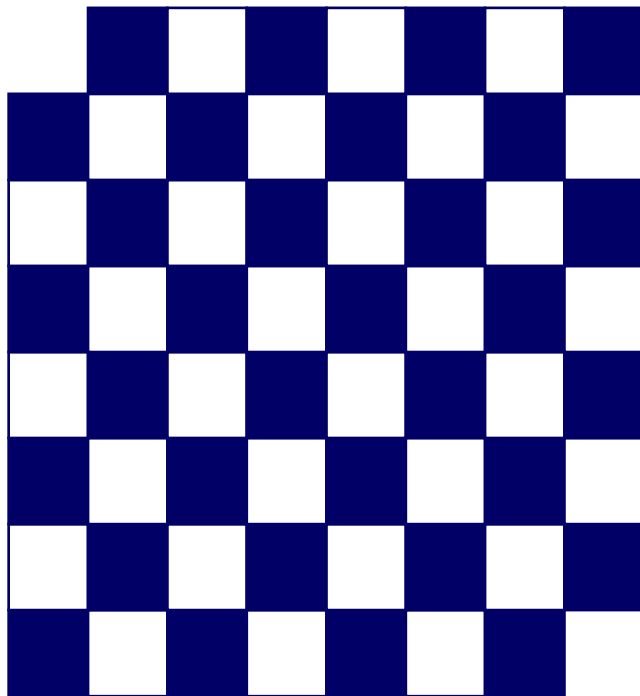
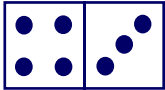
Allgemeine Suchstrategien, die für eine große Klasse von Problemen anwendbar sind, z.B. Hill-climbing, A*-Algorithmus, means-ends analysis

Starke Methoden:

Auf spezielle Problemklassen zugeschnittene Problemlöser, z.B. Planungsverfahren, Diagnosestrategien (werden vor allem in Expertensystemen eingesetzt)

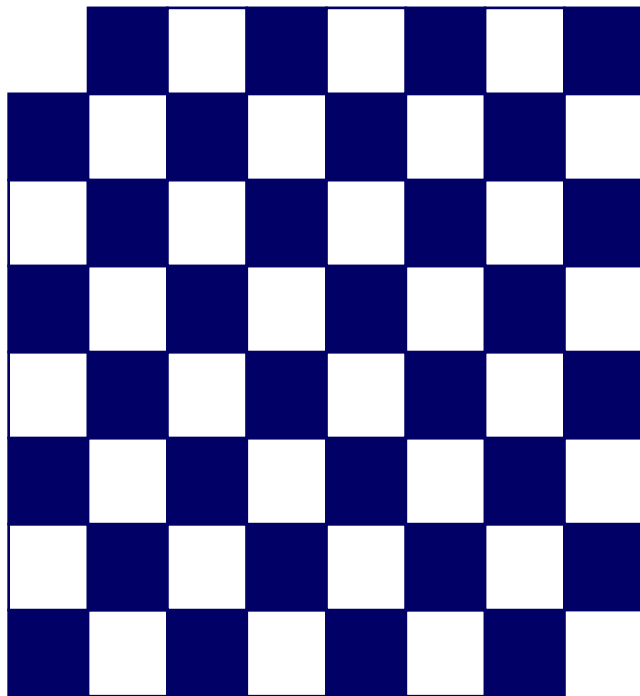
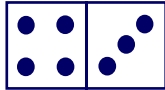


Plazierung von Dominosteinen auf einem Schachbrett



Gegeben ist ein Schachbrett, bei dem zwei gegenüber-liegende Eckfelder fehlen.

Frage: Kann man das Schachbrett so mit Dominosteinen bedecken, dass alle Felder belegt sind?



Lösung 1: Vollständige Suche

Teste alle Möglichkeiten, Dominosteine auf das Brett zu legen. Stop, sobald alle Felder belegt sind oder falls alle Möglichkeiten fehlgeschlagen sind.

Lösung 2: Heuristiken

Wähle eine Strategie, so dass nur erfolgversprechende Lösungen probiert werden.

Lösung 3: Wissen

Das Brett hat 30 weiße und 32 schwarze Felder. Jeder Stein bedeckt genau ein schwarzes und ein weisses Feld.



- Philosophie: Logik, Schlussfolgerungsmethoden, Gehirn als physisches System, Grundlagen des Lernens, der Sprache, der Rationalität
- Mathematik/Informatik: Formale Repräsentationen und Beweise, Algorithmen, Berechenbarkeit, Entscheidbarkeit, Komplexität, Wahrscheinlichkeit
- Psychologie: Anpassung, Phänomene der Wahrnehmung und der Muskelsteuerung, Experimentelle Techniken
- Wirtschaftswissenschaften: Formale Theorie rationaler Entscheidungen
- Linguistik: Wissensrepräsentation, Grammatik
- Neurowissenschaften: physische Gegenstände für mentale Aktivitäten
- Kontrolltheorie: homöostatische Systeme, Stabilität, einfache Entwürfe für optimale Agenten



1942 - 1956: Vorlaufphase

1952 - 1969: Früher Enthusiasmus, große Erwartungen

1966 - 1974: Relativierung durch Realitätsbezug

1969 - 1979: Wissensbasierte Systeme

1980 - 1988: Beginn der wissensbasierten Industrie

1985 - 1995: Rückkehr der Neuronalen Netze

seit 1995 Agenten



Cullach & Pitt 1943: Integration dreier Quellen zum Vorschlag eines Netzes künstlicher Neuronen:

- Physiologie der Neuronen
- Aussagenlogik
- Theorie der Berechenbarkeit (Turing)

1951: Minsky (mit Edmonds) bauen ersten Neuronalen Computer

1950: Turing: "Computing Machinery and Intelligence":

einschl. Turing Test, Lernen

Dartmouth-Workshop 1956 (Newell, Simon, Minsky, Mc Carthy et al.):

Prägung des Begriffes "Künstliche Intelligenz".



Anfangs wurden die (relativ primitiven) Computer nur als Rechenmaschinen betrachtet ("eine Maschine kann niemals X tun). KI-Forscher erregten viel Aufmerksamkeit, indem sie für ein X nach dem anderen Programme entwickelten.

- Logic Theorist (Newell & Simon, CMU): Beweisen mathematischer Theoreme.
- General Problem Solver (Newell & Simon, CMU): Allgemeine Problemlöse-Architektur.
- Geometry Theorem Prover (Gelernter, IBM): Benutzung einer numerischen Repräsentation von Diagrammen beim Beweisen.
- Checker-Program (Samuel, IBM): Lernfähiges Spielprogramm
- LISP (Mc Carthy, MIT, 1958).
- Time-Sharing (Mc Carthy, MIT, 1958).
- Advice Taker (Mc Carthy, MIT, 1958).



Explizite Wissensrepräsentation

- Mikrowelten (Minsky, MIT)
 - SAINT (Slage, 1963): Integral-Rechnung
 - ANALOGY (Evans, 1968): IQ-Tests: Geometrische Analogie
 - STUDENT (Bobrow, 1967): Lösen natürlichsprachlicher schulmathematischer Textaufgaben
- Einheitliche Domäne: Klötzchen-Welt (MIT):
 - Bildverarbeitung (Huffman, 1971)
 - Bildverarbeitung und Constraint-Propagierung (Waltz, 1975)
 - Sprachverarbeitung (SHRDLU, Winograd, 1972)
 - Lerntheorie (Winston, 1970)
 - Planung (Fahlman, 1974)
- Neuronale Netze
 - Perceptron-Konvergenz-Theorem (Rosenblatt, 1962).

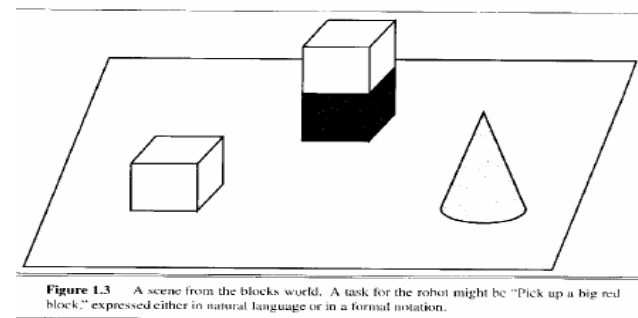
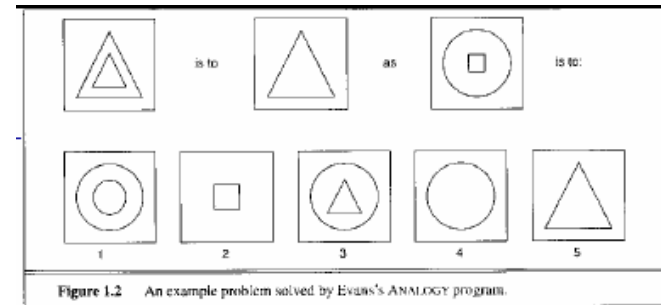


Figure 1.3 A scene from the blocks world. A task for the robot might be "Pick up a big red block," expressed either in natural language or in a formal notation.



- 1966 Abbruch der US-Forschungsförderung für Sprachübersetzung (gefördert seit Sputnik-Schock 1957)
- Nichterfüllen der Prophezeiung von Simon (1958), dass in 10 Jahren ein Computer Schachweltmeister und ein wichtiges neues mathematisches Theorem maschinell bewiesen worden sei.
- Entdecken der Skalierungsprobleme: "Wenn ein Programm ein Problem im Prinzip lösen kann, heißt dies nicht, dass es für einen praktischen Erfolg erforderliche Mechanismen enthält."
- Fehlschlag bei Experimenten zur Maschinen-Evolution (heute: Genetische Algorithmen) trotz tausender CPU-Stunden.
- Lighthill-Bericht (1973): Abbruch der KI-Forschungsförderung in Großbritannien (mit Ausnahme von zwei Unis)
- Die von Minsky & Papert (1969) gezeigten theoretischen Grenzen von Perceptrons führen zu einem weitgehenden Ende der Forschungsförderung im Bereich neuronaler Netze



"In the knowledge lies the power" (Feigenbaum) - und nicht in universellen ("schwachen") Problemlösungsmethoden.

DENDRAL (Stanford, 1969): Interpretation von Massenspektrogrammen zur Aufklärung von Molekül-Strukturformeln.

MYCIN (Shortliffe, Stanford, 1976): "Großvater der Expertensysteme"; Antibiotikumtherapie; Trennung Regeln & Inferenz

Minsky (1975) fasst viele Ideen der Zeit in dem Konzept der "Frames" zusammen.

Auch bei der Sprachverarbeitung wird die Bedeutung von Weltwissen (stereotype Situationen, Skripts) - im Gegensatz zu den bisher syntaktisch orientierten Programmen wie SHRDLU betont (Roger Schank, Yale)

LUNAR (Woods, 1973): Erstes extern genutztes natürlichsprachliches Anfragesystem für Datenbank über Mondgestein.

Beginn der wissensbasierten Industrie (1980-88)



R1/XCON (Mc Dermott/DEC 1982): Erstes kommerziell erfolgreiches Expertensystem zur Konfiguration von Computern bei DEC (Einsparungen von ca. 40.000.000 \$ pro Jahr)

Entstehen zahlreicher KI-Firmen mit meist hybriden Expertensystem-Werkzeugen und Beratung (Teknowledge, Carnegie Group, Inference, Intellicorp) & spezieller KI-Hardware (LISP-Maschinen)

Fifth Generation Project (Japan, 1981). 10-Jahres-Projekt zur Entwicklung von intelligenten Maschinen.

Nachziehen der USA (MCC-Konsortium) und Europa (Alvey-Bericht wiederbelebt die KI-Forschung in Großbritannien, BMFT startet Verbundprojekte in Deutschland).

KI-Umsatz stieg 1980-88 von einigen Millionen auf \approx 2 Mrd. \$.

1988-93 Rückschlag wegen zu hoher Erwartungen ("KI-Winter";).



Das Buch von Rumelhard & McClelland "Parallel Distributed Processing" 1986 verbreitet neue Erfolge mit dem wiederentdeckten Backpropagation-Algorithmus (der bereits 1969, im gleichen Jahr in dem das kritische Buch von Minsky und Papert erschienen ist, von Bryson & Ho publiziert wurde).

→ siehe Vorlesung „Knowledge Discovery“



Betonung der Interoperabilität, auch unter nicht-idealen Bedingungen
(z.B. Wahrnehmung ist nie perfekt)

Erfolge in vielen Teilbereichen verstärken wieder den Blick auf das
"Ganze,,

Internet: interessante Umgebung für Software-Agenten



KI ist etabliert. Meiste Informatik-Abteilungen haben KI-Professur, wissenschaftliche Techniken werden in Industrie breit genutzt.

Relative Selbständigkeit anwendungsorientierter Teilbereiche:

Wissenssysteme (Diagnostik, Konfigurierung; Scheduling, Planung, Simulation; Wissensportale)

Sprachverarbeitung (Hohe Qualität bei gesprochener Sprache; einfaches Verstehen geschriebener Texte - shallow parsing)

Bildverarbeitung (relativ selbstständig)

Robotik (relativ selbstständig; Robocup, Aibo, Marsroboter)

Betonung logischer Grundlagen (z.B. Hidden Markov Models, Bayessche Netze, logikbasierte Wissensrepräsentationen):

- "Normative versus empirische Expertensysteme".
- Agenten; Verbreitung über Internet



Schachprogramme haben Weltmeisterniveau erreicht

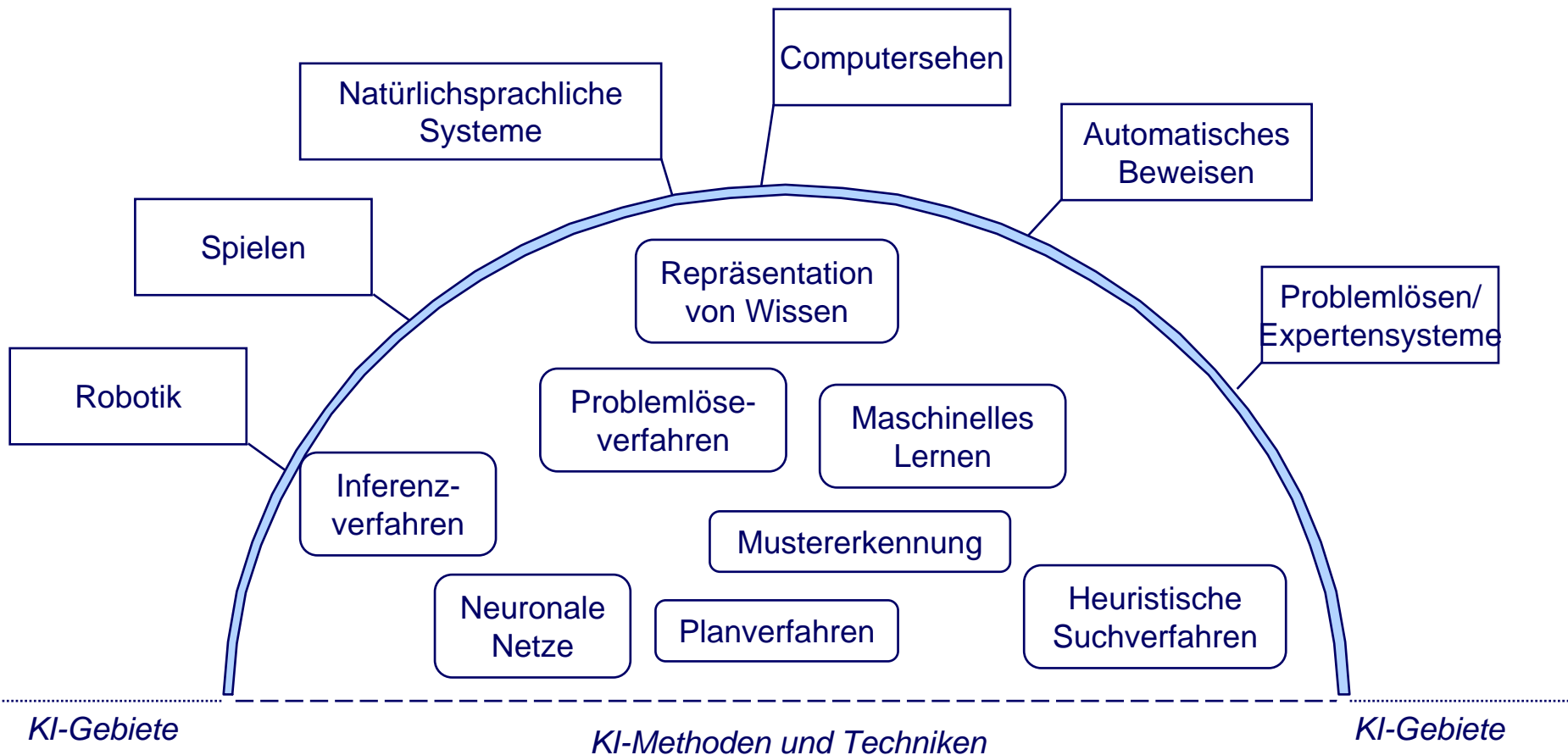
Natürlichsprachliche Auskunftssysteme (z.B. Reisebuchung)

Erfolgreiche Expertensysteme in vielen Bereichen

Automatische Planung und Scheduling in der Raumfahrt

Automatischer Autofahrer

Roboter auf dem Mars; für Hüftoperationen





METHODEN der KI

heuristische Suche

Planen

Lernen

Repräsentieren von
Wissen

Schlußfolgern

Problemlösemethoden

Meta-Reasoning

Konnektionismus

Mustererkennung

ANWENDUNGEN der KI

Verarbeitung natürlicher
Sprache (NLP)

Expertensysteme (XPS)

Deduktionssysteme

Robotik / Wahrnehmung

Computersehen /
Bildererkennung

Spiele :-)

Multiagentensysteme

Dokumentanalyse



Ziel ist das Verstehen geschriebener oder gesprochener Sprache

Anwendungen: Spracheingabe (Bagger, Textverarbeitung),
Übersetzung gesprochener & geschriebener Sprache,
Sprachgenerierung (techn. Dokumentation, Synthesizer),
komplexe Dialogsysteme (Fahrplanauskunft,
Hotelreservierung, Textverstehen

Probleme: low-level Phänomene (Nuscheln, Melodie etc.),
“ungrammatisches Sprechen”, Mehrdeutigkeiten,
Kontextabhängigkeit, non-verbale Elemente, Hintergrund- &
Weltwissen

Ansätze: komplexe Grammatiken, maschinelles Lernen, große
Lexika, aufgabenspezifisches Hintergrundwissen, non-standard
Logiken, (Common-Sense Reasoning CYC)



Natürlichsprachliche Systeme behandeln zwei Aufgaben:

- Verstehen eines Textes
- Generieren von Sätzen

Verstehen ist die Transformation von einer Ausgangsdarstellung (z.B. ein Satz in deutscher Sprache) in eine Zieldarstellung, die eine Menge von Aktionen ermöglicht.

Beispiele

- Ein System/Mensch hat eine Geschichte „verstanden“, wenn es/er/sie Fragen zu dieser Geschichte korrekt beantworten kann
- Ein Zuginformationssystem „versteht“ die Aussage *„Ich muß so schnell wie möglich nach München“*, wenn es mir die nächste Verbindung liefert.



Probleme beim Verstehen natürlicher Sprache:

- Für eine Aussage gibt es mehrere korrekte Formulierungen
- Ein Satz kann mehrere Bedeutungen haben
- Ein Wort kann mehrere Bedeutungen haben

Verstehen natürlicher Sprache erfordert Wissen über

- die Sprache (Syntax)
- das Themengebiet (Semantik)
- Konventionen des Sprachgebrauchs (Pragmatik)



Syntax: Erkennung der Satzstruktur durch Anwendung grammatischer Regeln

Semantik: Erkennen der Bedeutung syntaktischer Strukturen

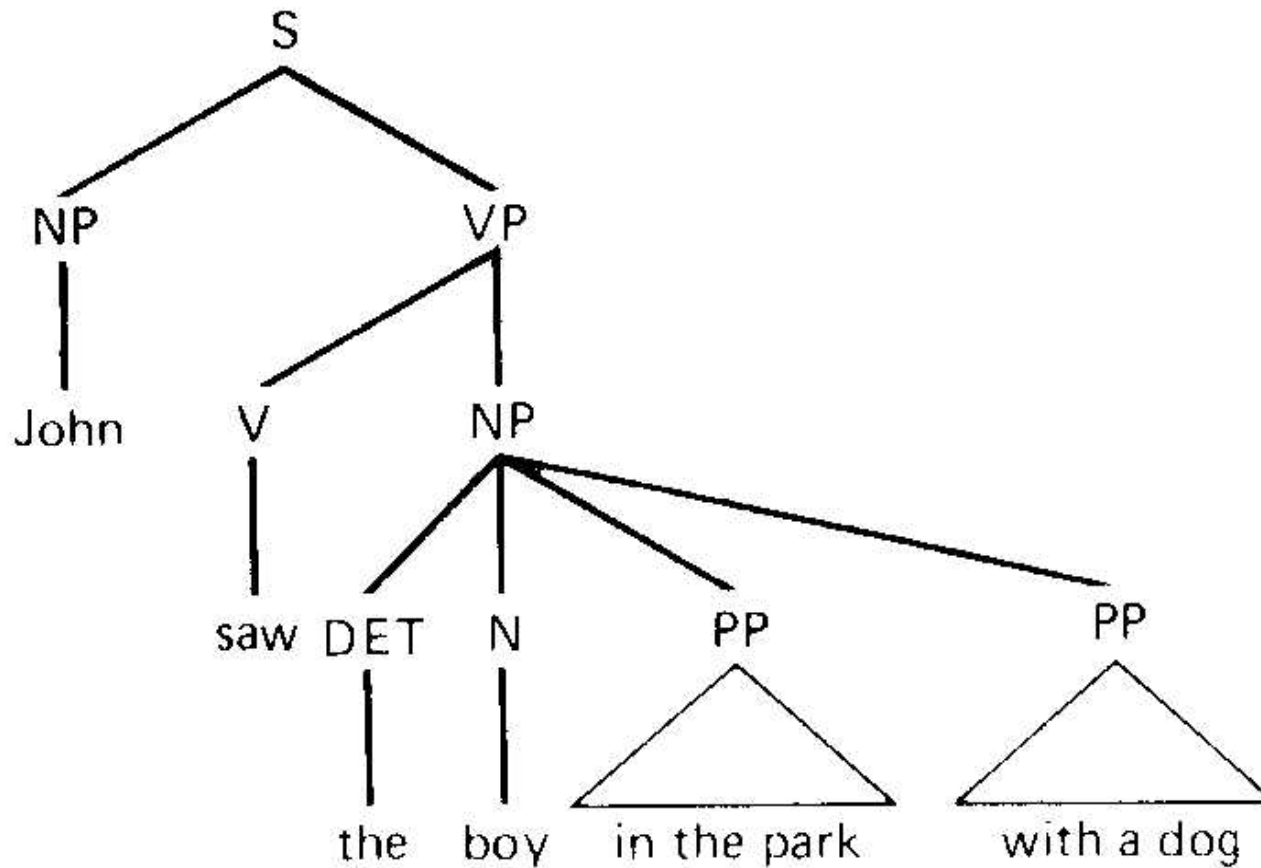
Pragmatik: Festlegung was wirklich gemeint ist, bzw. welche Handlung erfolgen soll

„Wissen sie wieviel Uhr es ist?“

⇒ Aufforderung, die Zeit zu sagen

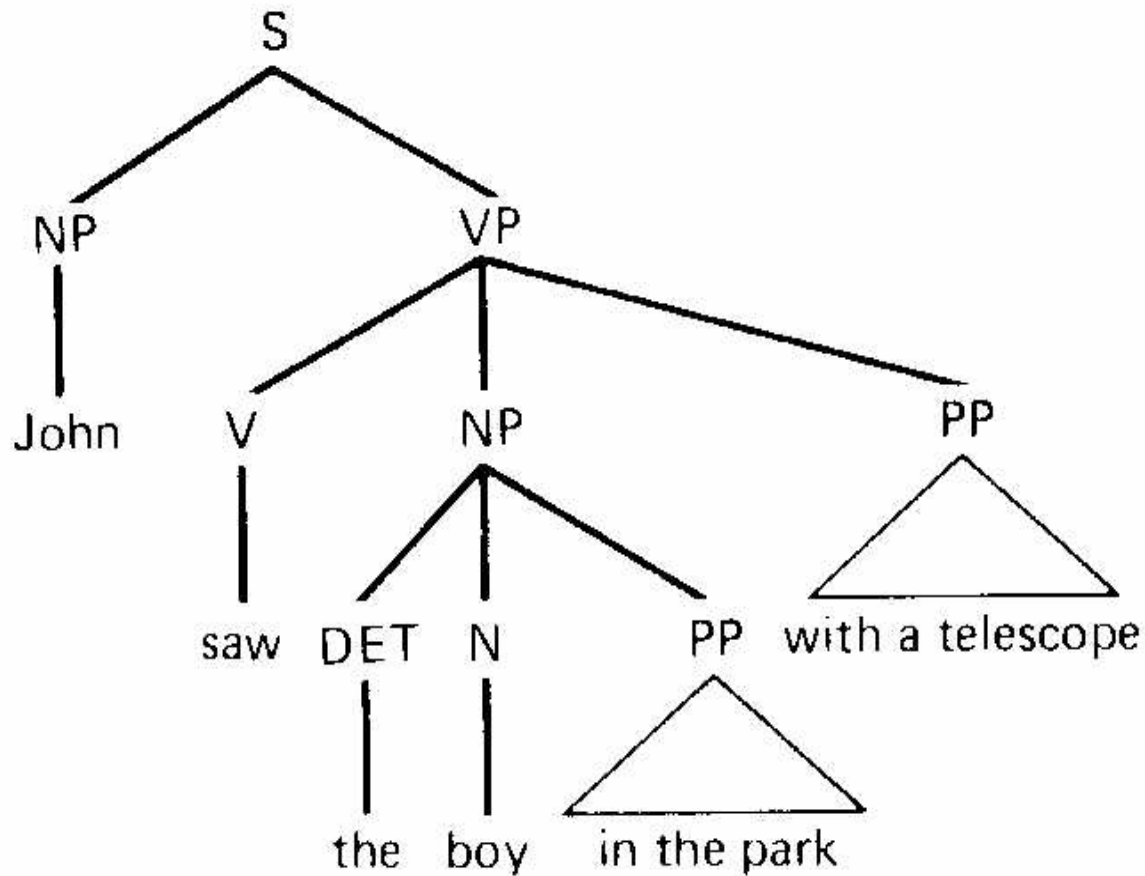


John saw the boy in the park with a dog.



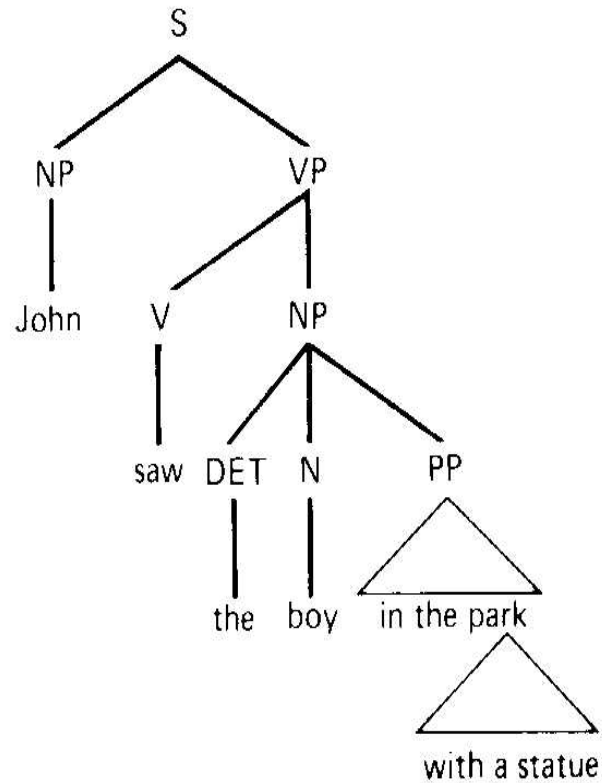


John saw the boy in the park with a telescope.





John saw the boy in the park with a statue.



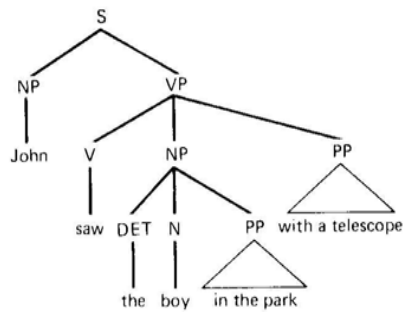
Beispiel: Sprachverarbeitung



Je nach Bedeutung der Wörter, sind unterschiedliche Strukturen korrekt:

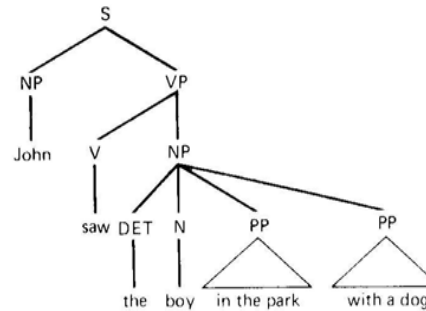
John saw the boy in the park with a telescope.

John saw the boy in the park with a telescope.

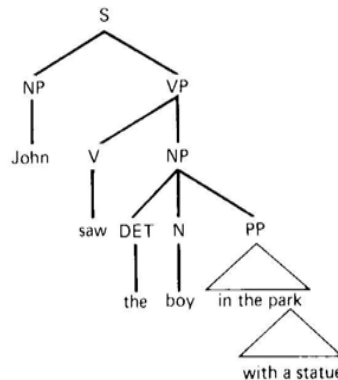


John saw the boy in the park with a dog.

John saw the boy in the park with a dog.



John saw the boy in the park with a statue.



John saw the boy in the park with a statue.

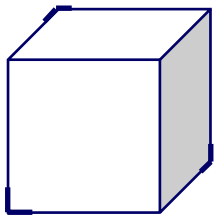
Nach: Hinkelmann



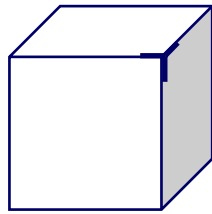
- **Ziel** ist es, visuelle Prozesse der maschinellen Verarbeitung zugänglich zu machen, d.h. Bilderkennung und Bildverstehen
- **Anwendungen:** Industrieroboter, optische Qualitätskontrolle, autonome Transportsysteme, Verkehrsszenenerkennung, Postleitzahlenerkennung, OCR
- für die Maschine aufwendiger als das Führen mathematischer Beweise
- für den Menschen keine bewußte Intelligenzleistung, weil fest verdrahtet in der “feuchten Hardware”
- **Prozeßschritte:** Bilddarstellung, Segmentierung, Objekterkennung, Szenenanalyse
- **Methoden:** Mathematik, neuronale Netze, spezifisches Domänenwissen, Ebenenarchitekturen, erwartungsgesteuerte Wahrnehmung



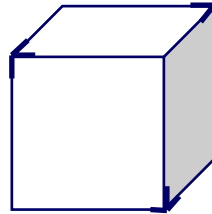
Kantentypen:



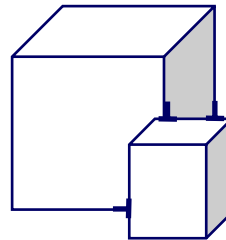
L



Y

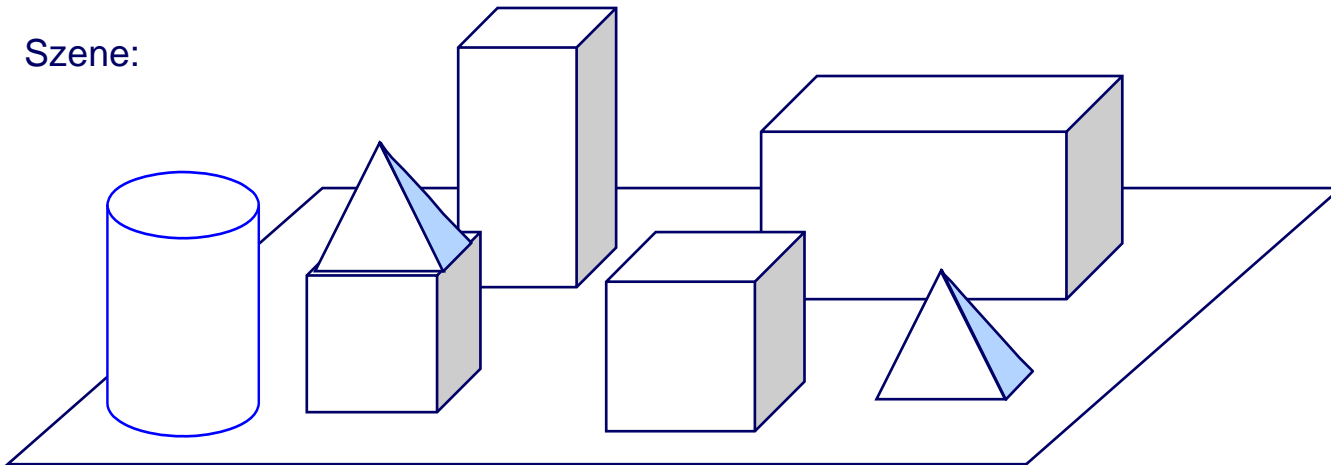


W



T

Szene:





- **Ziel** ist die Verwendung mathematischer Logik zum Folgern aus Fakten und Annahmen
- **Anwendungen:** automatisches und Unterstützung beim Beweisen von mathematischen Sätzen, Entwurf von Inferenzmechanismen für XPS, Programmverifikation, Programmsynthese aus deskriptiver Spezifikation
- **Problem:** extrem große Suchräume, Termination nicht gesichert, Modus der Benutzerinteraktion
- **Ansätze:** taktisches Theorembeweisen, Benutzer greift in die Steuerung ein, Lemmagenerierung

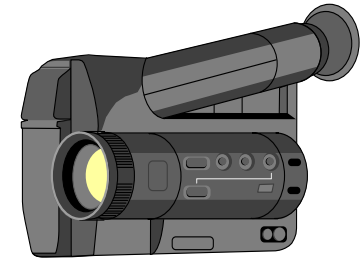
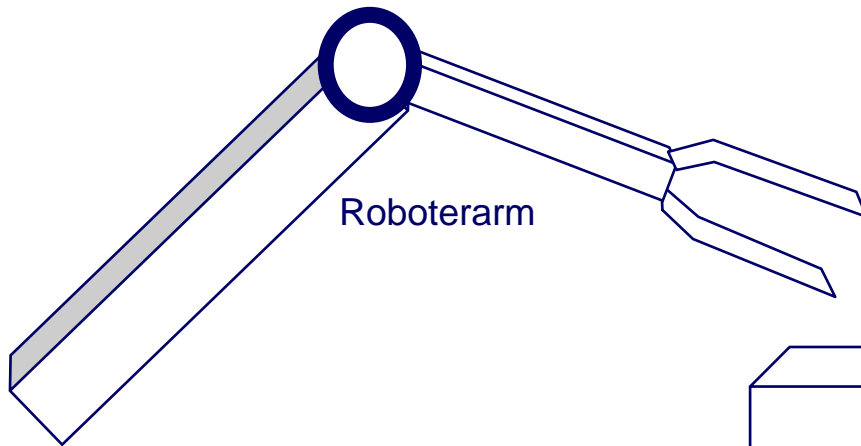


- **Ziel** ist die Gestaltung weitgehend autonomer und selbstlernender Roboter
- **Anwendungen:** autonome Transportsysteme, Weltraum & Tiefsee, Krankenhaus & Haushalt, Robo-Cup
- **Probleme:** komplexe Sensorik + Motorik + Planen, Echtzeitfähigkeit, “Überraschungen”, Frame-Problem
- **Methoden:** maschinelles Lernen, Schichtenarchitekturen, Planen mit means-end-analysis, Skelettplanen, “kurzgeschlossene Reflexbögen”, reaktives Planen, Kommunikation & Kooperation (Multiagentensysteme)

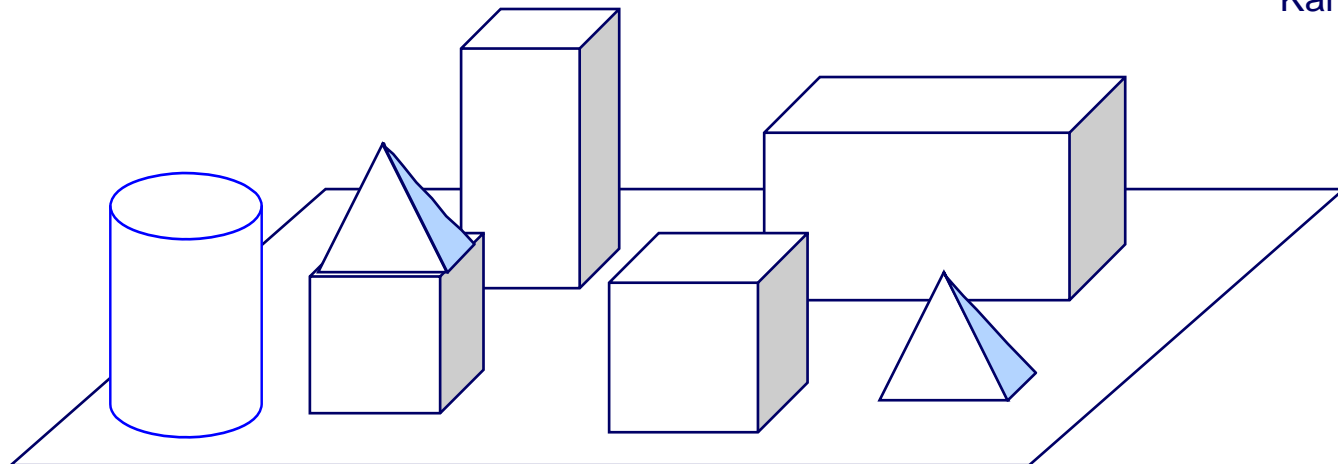


[Winograd, 1972]

„Gib mir die blaue Pyramide auf dem Klotz!“



Kamera



Quelle: Hinkelmann

Beispiel für Planen: Blockswelt



Operatoren (vereinfacht):

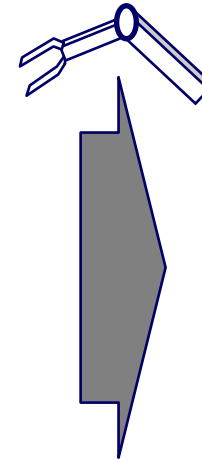
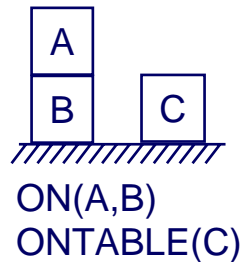
STACK(X,Y): (*lege X auf Y*)
Vorher: HOLDING(X)
CLEAR(Y)
Nachher: ON(X,Y)
HANDEEMPTY

UNSTACK(X,Y): (*hebe X auf, wobei X auf Y steht*)
Vorher: ON(X,Y)
CLEAR(X)
HANDEEMPTY
Nachher: HOLDING(X)
CLEAR(Y)

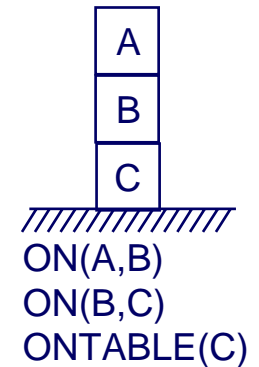
PICKUP(X): *hebe X vom Tisch auf*
Vorher: HANDEEMPTY
ONTABLE(X)
CLEAR(X)
Nachher: HOLDING(X)

PUTDOWN(X): *lege X auf den Tisch*
Vorher: HOLDING(X)
Nachher: HANDEEMPTY
ONTABLE(X)

Start:



Ziel:



Plan:

UNSTACK(A,B)
PUTDOWN(A)
PICKUP(B)
STACK(B,C)
PICKUP(A)
STACK(A,B)

Quelle: Hinkelmann



- **Ziel** ist es, das Spezialwissen und die Schlußfolgerungsfähigkeiten qualifizierter Fachleute bei bestimmten Spezialaufgaben nachzubilden und so zu vergeleichbaren oder besseren Problemlösungen zu kommen
- **Anwendungen** z.B. in technischer oder medizinischer Diagnose oder bei der Arbeitsplanung
- **Probleme:** Wissen kaum explizierbar, allgemeines Weltwissen, jahrelange Erfahrung modellieren, Wissensakquisition, komplexe Schlußfolgerungsstrategien, große Suchräume, vielfältige Wissensarten
- **Methoden:** Wissensrepräsentation & Inferenz, wiederverwendbare Ontologien und Problemlösungsmethoden, Deklarativität



Die beiden Begriffe Expertensystem und wissensbasiertes System werden weitgehend synonym verwendet für Softwaresysteme, die auf Methoden der Künstlichen Intelligenz basieren.

“Ein Expertensystem ist ein Computerprogramm, welches Spezialwissen und Schlußfolgerungsfähigkeiten von menschlichen Experten in einem begrenzten Aufgabengebiet nachbildet und somit Problemstellungen mit einer einem Experten vergleichbaren Leistung löst.”

Expertensysteme gibt es für verschiedene Klassen von Aufgaben:

- Diagnose
- Konfiguration
- Planung
- Konstruktion / Design



Expertensystem :=

Applikation, die Methoden der Künstlichen Intelligenz anwendet:

- stellt **Wissen** dar
 - auf einem eng begrenzten Gebiet
 - problemangepasst
- zieht daraus **Schlüsse**
 - algorithmisch
 - heuristisch
- **erklärt** die Schlüsse
 - unter Bezug auf Falldaten
 - im Dialog mit dem Benutzer



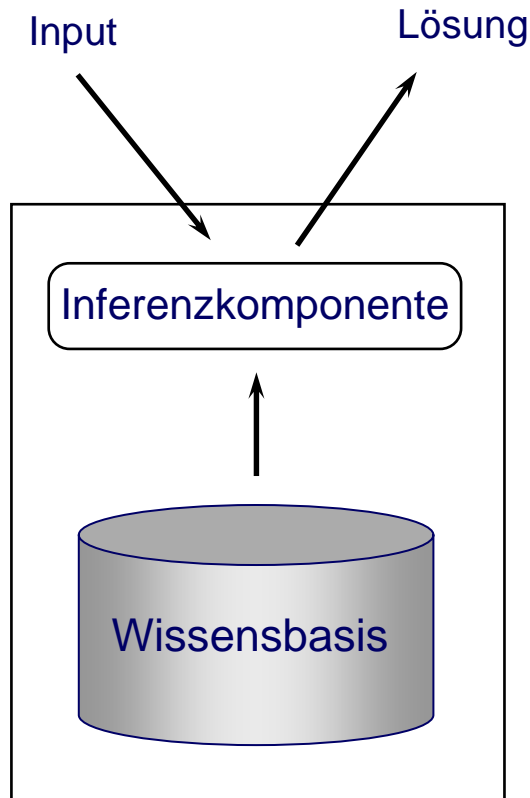
Knowledge Representation Hypothesis:
Ein System, das intelligentes Verhalten zeigt, hat als Komponente eine explizite Repräsentation seines Wissens

Was ist Wissen?

- Philosophie: Wissen ist wahrer, gerechtfertigter Glaube
- Wissensrepräsentation: Wissen ist Information über die Welt, ohne Anspruch auf deren Wahrheitsgehalt

Wissensbasierte Systeme verarbeiten Symbole, die Informationen über die Welt repräsentieren

Die Wissensbasis wird durch die Inferenzkomponente manipuliert, die neues Wissen ableitet





Im weiteren Sinn ist jede Programmierung Wissensverarbeitung

- Jedes Programm kodiert und verarbeitet Wissen

Im engeren Sinn wird die Wissensverarbeitung der Künstlichen Intelligenz zugeordnet

- Programmiersprachen: Orientierung am Computer
- Wissensrepräsentation: Prinzip der kognitiven Adäquatheit (möglichst natürliche Wissensrepräsentation)



Regeln sind eine klassische, weit verbreitete Programmierform für Expertensysteme:

WENN	(Batterie ok) und (Wert Tankuhr > 0) und (Benzinfilter sauber)
DANN	(Problem = Zündanlage)

- Modellierung menschlichen Problemlöseverhaltens
- Natürliche Ausdrucksform zur Darstellung von Expertenwissen:
 - Erfahrungen basieren auf Problemen, die der Experte früher gelöst hat.
 - Diese Erfahrungen hat der Experte in Faustregeln abstrahiert.
 - Wenn ein Experte mit einem neuen Problem konfrontiert wird, wendet er die passende Faustregel an.



Es gibt verschiedene Arten von Regeln

- Produktionsregeln

Intendierte Bedeutung: *“Falls in der aktuellen Datenbasis die Bedingung erfüllt ist, führe die Aktion aus”*

- Logische Regeln

Intendierte Bedeutung: *“Falls in der aktuellen Datenbasis die Bedingung wahr ist, dann ist auch die Folgerung wahr”*

Unterschied zwischen Regeln und IF-THEN-Anweisung in Programmiersprachen

- Konventionelle Programme haben einen Kontrollfluss: Eine IF-THEN-Anweisung wird nur ausgeführt, wenn das Programm an die entsprechende Stelle kommt
- In regelbasierten Systemen überprüft die Inferenzkomponente zu jedem Zeitpunkt *alle* Regeln, ob ihre Bedingung erfüllt ist
 - Regeln verhalten sich wie **WHENEVER-THEN**



Regelbasis:

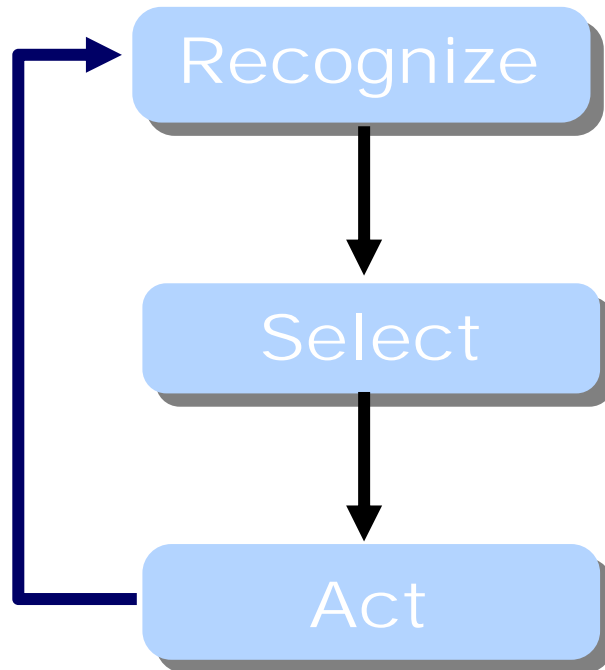
R1	WENN	(Anlasser arbeitet normal)
	DANN	(Batterie ok)
R2	WENN	(Batterie ok)
	UND	(Wert Tankuhr > 0)
	UND	(Benzinfilter sauber)
	DANN	(Problem = Zündanlage)
R3	WENN	(Batterie ok)
	UND	(Tankuhr > 0)
	UND	(NICHT Benzinfilter sauber)
	DANN	(Defekt = Benzinzuleitung)
R4	WENN	(NICHT Scheibenwischer ok)
	UND	(NICHT Licht ok)
	DANN	(Defekt = Batterie leer)
R5	WENN	(NICHT Wert Tankuhr > 0)
	DANN	(Defekt = Tank leer)
R6	WENN	(Problem = Zündanlage)
	UND	(Verteilerdose ok)
	DANN	(Defekt = Zündspule)

Fakten des aktuellen Falls:

Anlasser arbeitet normal
Scheibenwischer ok
Licht ok
Wert Tankuhr > 0
Benzinfilter sauber
Verteilerdose ok

Abgeleitete Fakten:

Batterie ok
Problem =
Zündanlage
Defekt = Zündspule



Datengetriebene Verarbeitung: Wenn alle Bedingungen einer Regel durch Fakten der Wissensbasis erfüllt sind, dann führe die Aktion aus.

- Recognize: Finde alle Regeln, deren Bedingungsteil erfüllt sind.
- Select: Wenn mehr als eine Regel anwendbar ist, wird eine Regel ausgewählt (Konfliktlösung).
- Act: Führe die Aktion der selektierten Regel aus.

Terminierungsbedingungen: Stop, wenn

- keine Regel mehr anwendbar ist oder
- wenn die Aktion "halt" ausgeführt wird

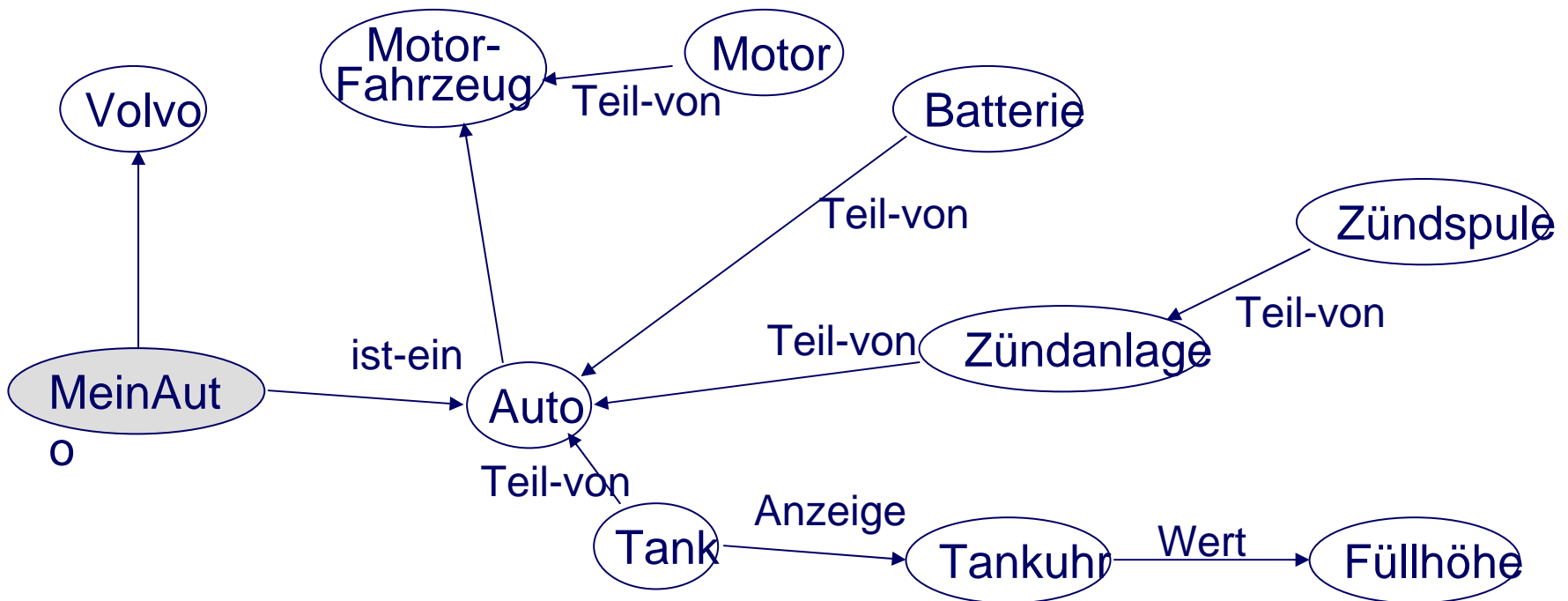
Andere Formen der Wissensrepräsentation

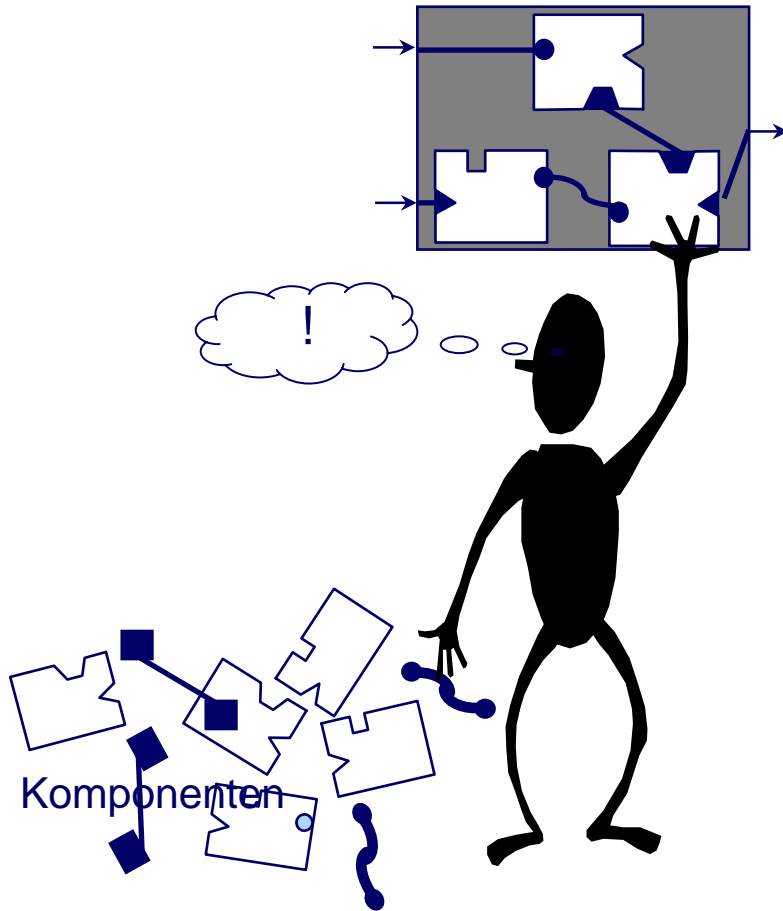


Domänenwissen: Wissen über

- Objekte und
- ihre Beziehungen

Beziehungen zwischen Objekten können teilweise automatisch erkannt und für die Verarbeitung genutzt werden





Konfigurierung: Zusammensetzung eines Systems auf einzelnen Komponenten

- Wissensbasis:
 - Katalog von Komponenten
 - Bedingungen über die Kompatibilität
- Input:
 - Anforderungen an die Lösung

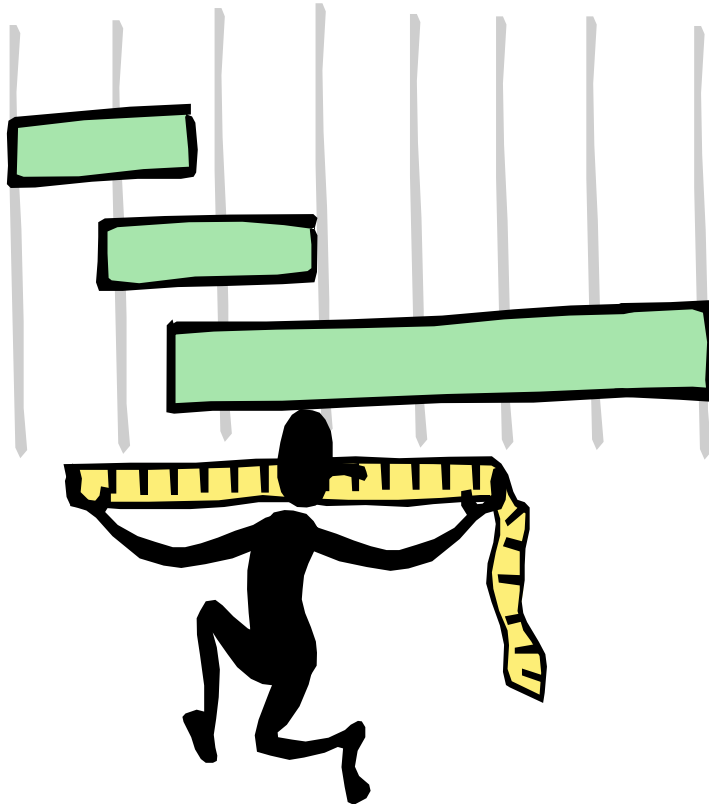
Anwendungsbeispiele:

- R1/XCON: Konfiguration von VAX-Computern
- Konfiguration von Schaltschränken



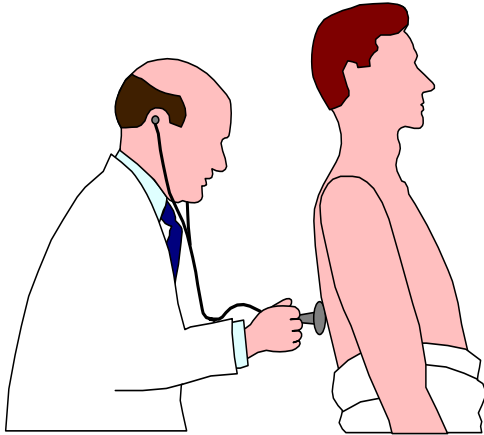
Planen ist die Aufstellung einer Sequenz von Aktionen, die ausgehend von einem Startzustand einen gewünschten Zielzustand erreichen, unter Berücksichtigung aller Laufzeitbedingungen

- Wissensbasis:
 - Definition von Operatoren
- Input:
 - Beschreibung von Start- und Zielsituation
- Lösung
 - Plan: Sequenz von Aktionen



Anwendungsbeispiele:

- Produktionsplanung
- Reiseplanung



Aufgabe der Diagnose: Identifiziere Gründe für das Fehlverhalten eines Systems, damit eine Therapie möglich wird

Anwendungsbeispiele:

- Medizin: Diagnose von Krankheiten
- Automobil: Diagnose von Motorschäden

Wissensbasis

- Modell des Systems
- Regeln über korrektes und fehlerhaftes Verhalten

Input:

- Symptome

Diagnose

Fehlerhafte
Komponenten



Deklarative Programmierung

- Direkte Repräsentation von Expertenwissen

Kein expliziter Kontrollfluss

- Reihenfolge der Regeln spielt keine Rolle
- Regeln sind aktiv sobald ihre Bedingung erfüllt ist
- Konfliktlösung wählt eine der aktiven Regeln aus

Problem

- Wartung grosser Regelsysteme ist aufwendig
- Auswirkungen von Regeländerungen sind nicht direkt erkennbar

Regeln sind nicht für alle Arten von Wissen geeignet. Es gibt weitere Formen der Wissensrepräsentation, die hier nicht weiter erläutert werden.



- Es gibt keine einheitliche Definition von Software Agenten
- Ein Software Agent ist ein Programm, das Menschen assistiert und in ihrem Auftrag arbeitet
- Personen können Arbeiten an Agenten delegieren
- Ein Agent ist
 - autonom
 - zielgerichtet: plant seine Aufgaben im Hinblick auf ein Ziel
 - reagiert auf seine „Umgebung“
 - kann mit anderen Agenten kooperieren um seine Aufgabe zu erfüllen
- Die Grenze zwischen Software-Agenten und anderen Programmen ist fließend



Suchagent im Internet

- Suchen in heterogenen Informationsquellen: Webseiten, Datenbanken, ...
- der Agent kennt die Wünsche und auch die vorhandenen Informationen des Auftraggebers

Adressfinder

- Suchen nach Email-Adressen in unterschiedlichsten Quellen (Verzeichnisse, Anfrage an Kollegen usw.)

Beantwortung von Emails im CRM

- Anfrage interpretieren
- Informationen sammeln und Antwort schicken
- evtl. an Mitarbeiter delegieren

Verwaltung von Terminen und Aufgaben

- Absprache mit Agenten anderer Personen, um einen Termin zu finden

Shopbot: Einkaufsagent

- Suchen von Angeboten mit dem günstigsten Preis
- evtl. Preisverhandlungen führen