



OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

FAKULTÄT FÜR
INFORMATIK

Intelligente Systeme

Wissensbasierte Systeme

Prof. Dr. R. Kruse C. Braune C. Doell

{kruse, braune, cmowes}@iws.cs.uni-magdeburg.de

Institut für Wissens- und Sprachverarbeitung

Fakultät für Informatik

Otto-von-Guericke Universität Magdeburg

Übersicht

1. Wissensbasierte Systeme

2. Wissensrepräsentation und Schlussfolgern

3. Wissensmodellierung

Wissensrepräsentation und Schlussfolgern

aus der *kognitiven Psychologie* ist bekannt, dass menschliches Problemlösen und Lernen folgende Dinge involviert:

Repräsentation und Ausnutzung von *mehreren Arten von Wissen*

Kombination von *verschiedenen Methoden zum Schlussfolgern*:

regelbasiertes Schließen, modellbasiertes Schließen,
bedingungsbasiertes (constraint-based) Schließen

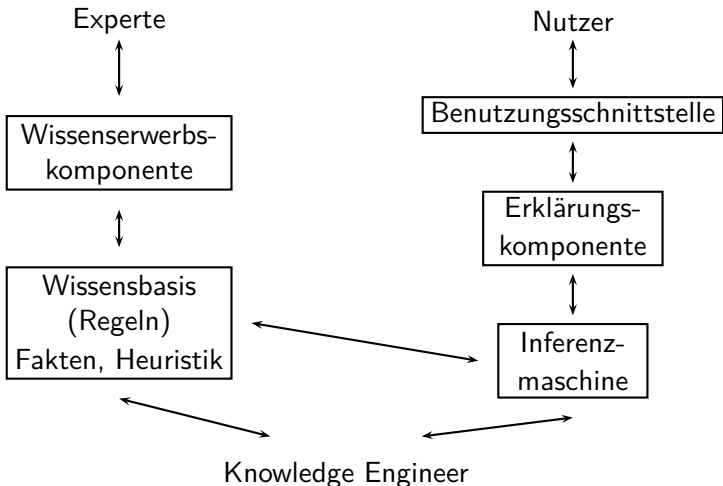
Fragen:

Wie können die reale Welt und das Schlussfolgern mit dem Computer modelliert werden?

Wie kann man Kenntnisse der Technologien gewinnen, die zum Entwurf von wissensbasierten Systemen gebraucht werden?

wissensbasiertes System = System, dass mittels Wissen schlussfolgern kann

Wissensbasierte Systeme



Beispiel: „Darlehensvergabe“

<i>OK</i>	Darlehen sollte bewilligt werden.
<i>COLLAT</i>	Sicherheit für Darlehen ist genügend.
<i>PYMT</i>	Antragsteller kann Darlehenszahlungen tilgen.
<i>REP</i>	Antragsteller hat gute finanzielle Reputation.
<i>APP</i>	Beurteilung der Sicherheit ist ausreichend größer als Größe des Darlehens.
<i>RATING</i>	Antragsteller hat gute Bonität.
<i>INC</i>	Einkommen des Antragstellers ist größer als seine Ausgaben.
<i>BAL</i>	Antragsteller hat ausgezeichnete Bilanz.

Beispiel: „Darlehensvergabe“

Regeln zur Entscheidungsfindung:

$\text{COLLAT} \wedge \text{PYMT} \wedge \text{REP} \Rightarrow \text{OK}$

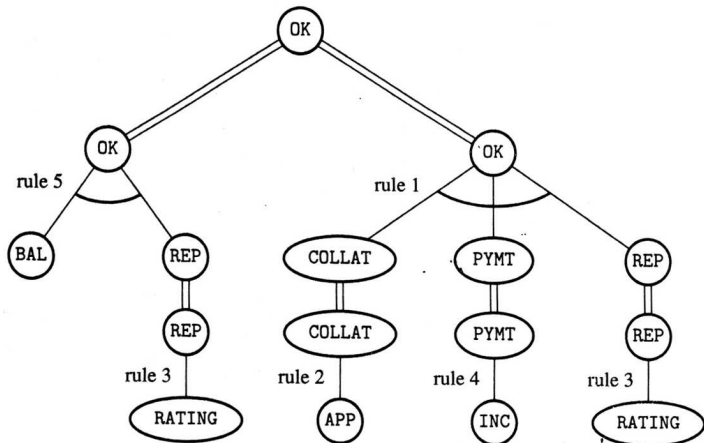
$\text{APP} \Rightarrow \text{COLLAT}$

$\text{RATING} \Rightarrow \text{REP}$

$\text{INC} \Rightarrow \text{PYMT}$

$\text{BAL} \wedge \text{REP} \Rightarrow \text{OK}$

Entscheidungsbaum (Und/Oder-Baum)



Modi der Verwendung

Anfragemodus:

Fakten werden aus Datenbank geholt oder Benutzer wird gefragt

Erklärungsmodus:

System beantwortet Fragen, wie Entscheidung gefällt wurde

Beispiele:

Nutzer: „Warum glaubst du, dass *das Darlehen bewilligt werden soll?*“

System: „Weil *die Sicherheit für das Darlehen genügend ist, und der Antragsteller die Darlehenszahlungen tilgen kann, und der Antragsteller eine gute finanzielle Reputation hat.*“

Anfrage- und Erklärungsmodus

Nutzer: „Warum glaubst du, dass *der Antragsteller eine gute finanzielle Reputation hat?*“

System: „Weil die Schufa gesagt hat, dass *der Antragsteller eine gute Bonität hat.*“

Nutzer: „Warum hast du mich gefragt, ob *das Einkommen des Antragstellers größer ist als seine Ausgaben* oder nicht?“

System: „Weil ich versucht habe, herauszufinden ob *der Antragsteller die Darlehenszahlungen tilgen kann* oder nicht.“

Nutzer: „Warum versuchst du herauszufinden, ob *der Antragsteller die Darlehenszahlungen tilgen kann* oder nicht?“

System: „Weil es eines der Kriterien zum Herausfinden ob *das Darlehen bewilligt werden sollte* oder nicht.“

Nutzer: „Warum versuchst du herauszufinden, ob *das Darlehen*

Wissensbasierte Systeme

einzelne Regel = (im Idealfall) Wissensseinheit

Menge von Regeln zusammen mit Ausführungsstrategie = Programm für Lösung eines Problems oder Problemklasse

Prolog-Programm: *Regelbasiertes System*

regelbasierte Systeme: historisch erste Formen von KI-Systemen

lange Zeit als prototypisch für *Expertensysteme* angesehen

inzwischen: nicht jedes Expertensystem benutzt Regeln als Hauptinferenzmechanismus

Regelmodule: wichtige Bestandteile von Expertensystemen

Regelsysteme gewinnen erneut Bedeutung

z.B. bei Kodifizierung von *Geschäftsprozess-Regeln*

XCON/R1

Beispiel für ein Konfigurationssystem

XCON/R1: Werkzeug für Konfiguration von *DEC Vax-Computern*

seit ca. 1980 entwickelt und löst folgende Aufgaben:

- Identifikation fehlender Komponenten,
- Platzierung der Komponenten bezüglich Bussen, Schnittstellen und Gehäusen

implementiert in *OPS5* (vorwärtsverkettetes regelbas. System)

eine der ersten erfolgreichen kommerziellen Anwendungen von regelbasierten Expertensystemen

hatte großen Einfluss auf industrielles Interesse an Expertensystemen

wurde ständig weiterentwickelt: heute ca. 10.000 Regeln

löst routinemäßig Aufgaben mit 100–200 Komponenten

Beispiel einer XCON/R1 Regel

IF

the most current active context is distributing massbus devices, and there is a single-port disk drive that has not been assigned to a massbus, and there are no unassigned dual-port disk drives, and the number of devices that each massbus should support is known, and there is a massbus that has been assigned at least one disk drive and that should support additional disk drives, and the type of cable needed to connect the disk drive to the previous device on the massbus is known

THEN

assign the disk drive to the massbus.

Beispiel: Implementierung über Prolog

„Roboterarm bewegt sich, wenn Batterie i.O. ist und Gewicht nicht zu schwer ist“

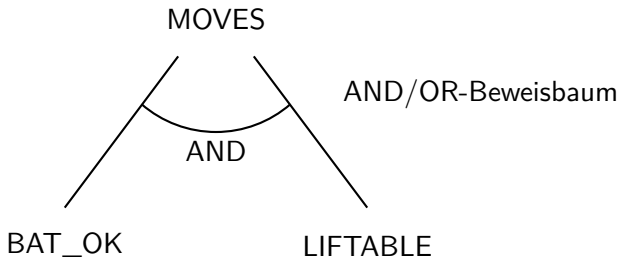
Programm entspricht der Aussage:

$$(BAT_OK \wedge LIFTABLE \wedge (BAT_OK \wedge LIFTABLE \Rightarrow MOVES)) \Rightarrow MOVES$$

<code>:- MOVES</code>	Ziel (Anfrage)	
<code>BAT_OK :-</code>	Fakten	
<code>LIFTABLE :-</code>		Prolog beweist
<code>MOVES :- BAT_OK, LIFTABLE</code>	Regeln	

Aussage mit „depth-first“ und „backward chaining“

Beispiel: Schlussfolgern mit Hornklauseln



Anmerkung:

Klausel 1 gegen Klausel 2 oder Klausel 3 geht nicht

aber Klausel 1 vs. Klausel 4 liefert 2 neue Ziele (siehe Baum)

diese können sofort bewiesen werden

Schlussfolgern mit Hornklauseln

Beispiel: Blockwelt

Ist Bauklotz A über C , wenn A auf B und B auf C ?

Prolog:

$:-$ Above(A, C)

On(A, B) $:-$

On(B, C) $:-$

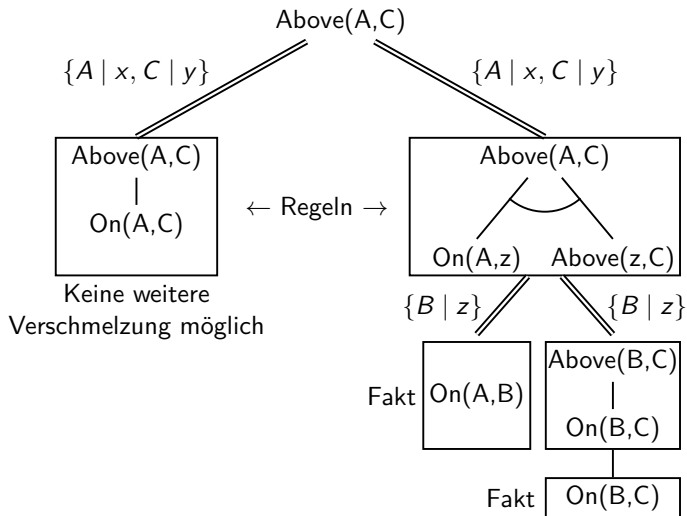
Above(x, y) $:-$ On(x, y)

Above(x, y) $:-$ On(x, z), Above(z, y)

PK1:

$$(\forall x, y, z) \text{ On}(x, y) \Rightarrow \text{Above}(x, y)$$
$$(\forall x, y) (\exists z) \text{ On}(x, y) \wedge \text{Above}(z, y) \Rightarrow \text{Above}(x, y)$$

Schlussfolgeren mit Hornklauseln



Übersicht

1. Wissensbasierte Systeme

2. Wissensrepräsentation und Schlussfolgern

3. Wissensmodellierung

Wonach suchen wir, um eine Maschine zu verkörpern?

Geist

Gedanken

Intelligenz

Rationalität

Neuroanatomie

Wissen

Pyramide des Wissens

Giarratano und Riley (2005)



Weisheit: Nutzen von Wissen in einer vorteilhaften Weise

Metawissen: Regeln über Wissen

Wissen: Regeln über das Nutzen von Informationen

Informationen: potentiell nützlich für Wissen

Daten: potentiell nützliche Informationen

Intellektuelle Ursprünge

vor 2000 Jahren: der **Syllogismus** (Logik von Aristoteles, Konklusionen)

im 17. Jhd.: die **Algebra der Gedanken** (Leibniz: menschliches Schlussfolgern kann auf Berechnungen zurückgeführt werden, regelbasierte Manipulation)

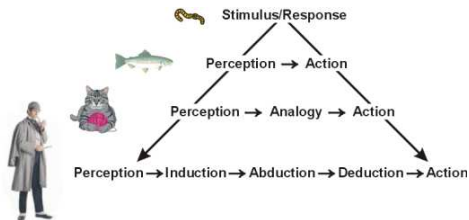
im 19. Jhd.: **Boole'sche Logik**

auch 19. Jhd.: **analytische Maschine** (Babbage, Ada Lovelace)

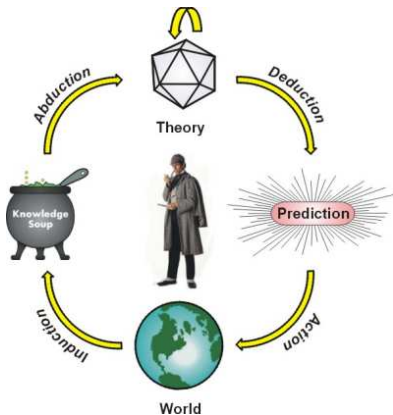
im 20. Jhd.: Turings Ideen über Gedanken und Berechenbarkeit

1927 (McCarthy): Schlussfolgern kann auf Berechnungen reduziert werden

Kognitive Prozesse



Entwicklung der Kognition



Peirce'sche Logik
des Pragmatismus

Beispiel: Das unordentliche Zimmer

von D.M. Gabbay (2010)

Eine Mutter geht in das Schlafzimmer ihrer Teenager-Tochter. Ihr erster Eindruck ist, dass dort eine große Unordnung herrscht. Es liegt überall irgendetwas herum. Der Eindruck der Mutter ist, dass dies untypisch für ihre Tochter sei, sich so zu verhalten. Was war passiert?
Vermutung: Ihre Tochter hat Probleme mit einem Jungen.

Weitere Analyse: Die Mutter bemerkt ein zusammengebrochenes Regal. Hat ihre Tochter es zerbrochen? Nach weiterer Betrachtung bemerkt sie anhand des Musters der Unordnung, dass das Regal zusammenbrach durch das zu hohe Gewicht und einfach alles verteilte. Aber eigentlich ist das keine Unordnung, sondern ergibt (anziehungskräftegemäß) Sinn.

Beispiel: Das unordentliche Zimmer

verschiedene Modi von Schlussfolgerungen werden hier genutzt:
Schließen mittels neuronaler Netze; Sie erkennt die Unordnung sofort, so wie wir ein Gesicht erkennen.

nicht-monotone Deduktion: Die Mutter schließt aus dem Kontext und ihrem Wissen über ihre Tochter, dass das Mädchen nicht so unordentlich ist. „Was ist passiert?“ fragt sie demnach.

Abduktion: Sie bietet eine sinnvolle Erklärung an, dass das Mädchen Probleme mit einem Jungen hat: Es ist normal in dem Alter.

Deduktion: Dann wendet sie Deduktion an und erkennt, dass die Unordnung von der Schwerkraft herrührt. Diese Deduktion ist nicht mehr ein „Neuronaler Netz“-Eindruck. Es ist eine sorgfältige Berechnung.

Merke: Punkt 4 kann ein „Neuronaler Netz“-Eindruck sein: jemand der viele Fälle von zerbrochenen Regalen sieht, erkennt dieses Muster. Punkt 2 könnte auch ein Bayes-Netz gewesen sein.

Übersicht

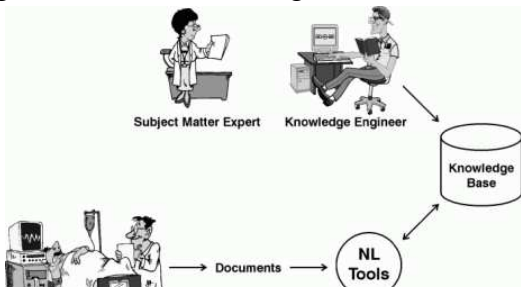
1. Wissensbasierte Systeme
2. Wissensrepräsentation und Schlussfolgern
- 3. Wissensmodellierung**

Knowledge Engineer

seine Aufgabe: Entwurf von effizienten regelbasierten Systemen arbeitet mit den Experten zusammen und beobachtet diese, um Expertisen zu extrahieren und um diese in eine Menge von Regeln zu implementieren

kennt sich aus mit Expertensystemen, aber nicht notwendigerweise in der Anwendungsdomäne

kennt die Möglichkeiten der Technologien und weiß sie zu nutzen



Vergleich zur klassischen Programmierung

Eigenschaft

klass. Programmierung

Expertensystem

gesteuert von...

Regelung und Daten

Lösung durch...

Eingabe

unerwartete Eingabe

Ausgabe

Erklärung

Programmwurf

Modifizierbarkeit

Erweiterung

Anordnung der Aussagen

implizierte Integration

Algorithmus

als richtig angenommen

schwer händelbar

immer richtig

generell sequentiell

strukturiert

schwer

in großen Sprüngen

Inferenzmaschine

explizite Separation

Regeln

unvollständig, falsch

sehr zugänglich

variiert mit dem Problem

opportunistische Regeln

kaum oder keine Struktur

möglich

inkrementell

Vergleich zur klassischen Programmierung

Algorithmen + Datenstrukturen = Programme

Wissen + Inferenz = wissensbasiertes System