

HANSER



Leseprobe

Uwe Lämmel, Jürgen Cleve

Künstliche Intelligenz

ISBN: 978-3-446-42758-7

Weitere Informationen oder Bestellungen unter

<http://www.hanser.de/978-3-446-42758-7>

sowie im Buchhandel.

2

Darstellung und Verarbeitung von Wissen

Die Repräsentation und die Verarbeitung von Wissen sind Kernthemen der künstlichen Intelligenz. Wissen wird dabei im Folgenden als Information verstanden, die in bestimmten Situationen wertvoll ist und für Entscheidungen genutzt werden kann. Wissen ist Information, die für die Lösung eines Problems hilfreich ist.

Um dem Computer Wissen zugänglich zu machen, ist eine Repräsentation in einer Form nötig, die der Computer „versteht“. Mit „Verstehen“ ist nicht gemeint, dass er tatsächlich weiß, was Begriffe wie *reich*, *1 kg* oder *GmbH* bedeuten. Er muss diese Information nur in geeigneter Weise verarbeiten können.

Der Wunsch nach der Möglichkeit einer formalisierten und automatisierten Wissensdarstellung und Wissensverarbeitung ist sehr alt. Bereits Aristoteles entwickelte mit seiner **Syllogistik** eine Form der formalisierten Wissensverarbeitung.

Gottfried Wilhelm Leibniz (1646–1716), der auf sowohl philosophischen als auch mathematischen Gebieten tätig war, entwickelte ein Konzept, mit dem man im Prinzip alles ausrechnen kann. Das leibnizsche Programm sah vor, dass man sämtliches Wissen, was man über einen Problembereich hat, formalisiert. Man benötigt dazu eine Sprache, in der dieses Wissen formuliert werden kann. Anschließend sollte es möglich sein, mit diesem Wissen zu „rechnen“. Dies erfordert einen Mechanismus, der Informationen in dieser Sprache verarbeiten kann. Leibniz bezeichnete dies als:

- *lingua characteristica*
- *calculus ratiocinator*.

Mit den Worten der künstlichen Intelligenz aus heutiger Sicht benötigt man

- eine formale Sprache zur **Wissensrepräsentation (WR)**,
- einen Verarbeitungsmechanismus zum automatischen Schließen (**Wissensverarbeitung, WV**).

Um ein Problem zu lösen, wird typischerweise in diesen Schritten vorgegangen:

1. Charakterisierung des Gegenstandsbereiches
2. Symbolische Repräsentation der Objekte
3. Eingabe des Wissens in den Computer
4. Fragen stellen
5. Interpretieren der Antworten

Zunächst wird das Wissen über einen Gegenstandsbereich gesammelt. Dabei muss darauf geachtet werden, dass nur problemrelevantes Wissen herangezogen wird. Soll ein Kunde bezüglich seiner Kreditwürdigkeit betrachtet werden, dürfte seine Haarfarbe völlig uninteressant sein. Relevant ist ebenfalls der *Detailierungsgrad* einer Information. Will man die Sicherheiten eines Kunden beurteilen, spielen Besitzgüter unterhalb einer bestimmten Grenze keine Rolle. Der zweite Schritt ist die *formale Darstellung* des gesammelten Wissens in einer geeigneten Sprache. Geeignete Darstellungsformen können beispielsweise mathematische Relationen, eine Logik, aber natürlich auch eine passende Programmiersprache sein.

Das Wissen über den Problem- oder Gegenstandsbereich wird nun dem Computer übergeben. Darauf aufbauend erfolgt die Lösung des Problems, indem beispielsweise Fragen über die Kreditwürdigkeit eines Kunden X gestellt werden. Dabei wird versucht, eine zu prüfende Annahme aufgrund des vorhandenen Wissens nachzuweisen.



BILD 2.1 Vorgehen bei einer Problemlösung

Dieses Vorgehen ist zunächst nicht KI-spezifisch. In der klassischen Programmierung wird das Wissen über ein Problem direkt in den Lösungsalgorithmus hineinprogrammiert. In der KI dagegen wird das Wissen *getrennt* von der Verarbeitungskomponente dargestellt. Ein ähnliches Konzept findet man bei den Datenbanken, wo Information (Datensätze) und Verarbeitung getrennt verwaltet werden.

Auf eine typische Architektur eines wissensbasierten Systems am Beispiel eines Expertensystems wurde bereits im Abschnitt 1.3 eingegangen:

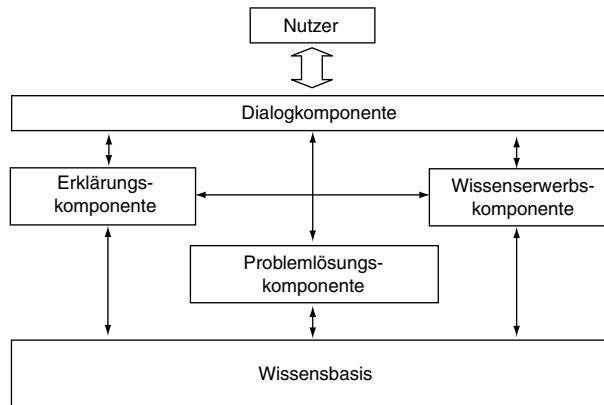


BILD 2.2 Struktur eines Expertensystems

Hier findet man das Architekturprinzip der **Trennung von Wissen und Verarbeitung** wieder. Wählt man beispielsweise die Logik als Wissensrepräsentations-Sprache, so besteht die

Wissensbasis aus logischen Aussagen. Die Problemlösungskomponente muss dann einen Logik-Beweiser enthalten.

Welche Vorteile bietet eine solche Architektur? Warum ist die Trennung von Problemlösungskomponente und Wissensbasis sinnvoll?

Es kann durchaus der Fall eintreten, dass das Wissen im Laufe der Zeit revidiert werden muss, beispielsweise dadurch, dass ein Zusammenhang, eine bestimmte logische Regel als nicht korrekt erkannt wird. Betrachtet man ein System zur Kreditwürdigkeitsprüfung, so kann es passieren, dass eine Person als kreditwürdig bewertet wurde, was sich – leider erst später – als falsch herausstellte. Hat man beispielsweise die Kreditwürdigkeit nur davon abhängig gemacht, dass ein regelmäßiges Einkommen in einer bestimmten Mindesthöhe vorliegt:

regelmäßiges_Einkommen → kreditwürdig

so stellt man schnell fest, dass zumindest eventuelle Schulden geprüft werden müssen:

regelmäßiges_Einkommen und geringe_Schulden → kreditwürdig

In einem konventionellen Programm muss nun die entsprechende Stelle im Programm gefunden und modifiziert werden. Anschließend muss das System vollständig oder teilweise neu erstellt werden. Hat man das Wissen aber *explizit* und von der Verarbeitung getrennt repräsentiert, so muss nur die entsprechende Regel ergänzt oder modifiziert werden. Die Verarbeitungskomponente ist davon nicht betroffen. Die *Modifikation* kann sogar zur Laufzeit automatisch geschehen.

Da eine Änderung der Wissensbasis die Verarbeitungskomponente nicht betrifft, lässt sich sogar die gesamte Wissensbasis auswechseln. Das gleiche Programm kann also für unterschiedliche Anwendungen verwendet werden. Entfernt man die Wissensbasis eines Expertensystems, so spricht man in diesem Zusammenhang auch von einer **Expert system shell** (ESS). Die Verwendung einer ESS für unterschiedliche Anwendungen setzt natürlich voraus, dass die gewählte Sprache zur Wissensrepräsentation für beide Anwendungen gleich ist.

Ein weiterer Vorteil der expliziten Darstellung des Wissens ist die damit verbundene Erhöhung der *Transparenz*. Es ist wesentlich leichter möglich, das benutzte Wissen zu verstehen. Es ist ebenfalls einfacher, den Ablauf einer Verarbeitung zu verfolgen.

Man beachte, dass eine Problemlösung immer von dem vorhandenen, *expliziten* Wissen ausgeht. Aufgrund dieses Wissens wird versucht, eine Hypothese zu überprüfen.



Beispiel 2.1 Kreditwürdigkeit

Seien diese Aussagen bekannt:

regelmäßiges_Einkommen und geringe_Schulden → kreditwürdig
regelmäßiges_Einkommen
geringe_Schulden

Neben der bereits diskutierten Regel ist auch bekannt, dass ein regelmäßiges Einkommen gegeben ist und nur geringe Schulden vorliegen. Man sieht sofort, dass die Kreditwürdigkeit gemäß der Regel erfüllt ist. Implizit steckt also die Kreditwürdigkeit in den gegebenen Aussagen.

Aufgabe einer Wissensverarbeitung ist es nun, dieses implizite Wissen abzuleiten. Problemlösen ist das *Explizieren von implizitem Wissen*. Die gewählte **WissensrepräsentationsSprache** und der zugehörige **Verarbeitungsmechanismus** müssen gewissen Kriterien genügen:

- Die WR-Sprache muss die Darstellung des Wissens ermöglichen.
Will man darstellen, dass es morgen *eventuell* regnet, reicht eine WR-Sprache, die klassische Logik unterstützt, nicht aus. Stattdessen muss eine Sprache gewählt werden, die unsichere Aussagen darstellen kann.
- Der Verarbeitungsmechanismus muss das Wissen verarbeiten können. Wählt man als WR-Sprache die Aussagenlogik, so braucht man einen aussagenlogischen Beweiser.
- Die Verarbeitung muss effizient möglich sein. Ein aussagenlogischer Beweiser muss eine Antwort nicht irgendwann, sondern innerhalb akzeptabler Zeiten liefern.
- Die Qualität der Antwort muss den Erfordernissen des Problems entsprechen. Vielfach genügt eine suboptimale Lösung, eine Lösung, die „gut genug“ ist.

Im Folgenden werden Konzepte zur Darstellung von Wissen auf einem Computer behandelt. Dies wird immer in Zusammenhang mit möglichen Verarbeitungsmechanismen geschehen.

In welcher Form kann man Wissen auf einem Rechner repräsentieren? Die von uns Menschen hauptsächlich genutzte WR-Sprache ist die natürliche Sprache. Allerdings ist sie wenig geeignet, als Sprache zur Repräsentation von Wissen auf dem Computer zu dienen. Dies liegt im Wesentlichen an der großen Vielfalt, die unsere Sprachen bieten. Ein und derselbe Sachverhalt lässt sich im Deutschen auf unterschiedlichste Art darstellen. Die Aussagen „Heute ist Montag“, „Montag ist heute“, „Der heutige Tag ist Montag“ und „Heute ist der Tag vor dem Dienstag“ sind alle äquivalent. Dies ist für eine effiziente Verarbeitung sehr ungünstig, da vor der eigentlichen Verknüpfung des Wissens eine Normalisierung der Aussagen erfolgen muss. Dies ist eines der Probleme, die bei der automatischen Verarbeitung natürlicher Sprache gelöst werden müssen.

Bevor auf einige WR-Formalismen, die in diesem Kapitel ausführlicher behandelt werden, eingegangen wird, werden zunächst entsprechende Wissensarten betrachtet, die wir Menschen benutzen.

Beziehungs-Wissen spielt in unserem Alltag eine große Rolle. Beziehungen zwischen Personen (X ist mit Y verheiratet) oder zwischen Objekten (der Motor ist ein Teil vom Auto) sind typische Vertreter. Man bezeichnet dies als **relationales Wissen**. Es widerspiegelt einfache Beziehungen zwischen Objekten. Relationales Wissen kann beispielsweise in relationalen Datenbanken dargestellt werden. Ein Nachteil ist, dass im Allgemeinen nur Fakten, aber keine logischen Abhängigkeiten abgebildet werden können.

Im Alltagswissen arbeiten wir häufig mit der **Vererbung von Eigenschaften**. Weiß man, dass ein Auto einen Motor hat und dass der Trabant ein Auto ist, so schließen wir sofort, dass der Trabant einen Motor hat. Die Eigenschaft „hat Motor“ wird vom Oberbegriff Auto auf den Unterbegriff Trabant vererbt.

Obwohl wir nicht explizit gesagt haben, dass der Trabant einen Motor hat, wissen wir es, da der Trabant eine spezielle Form eines Autos ist und jedes Auto einen Motor hat.

Prozedurales Wissen ist Wissen, das in bestimmten Situationen Aktionen vorschreibt. Viele Eröffnungen beim Schach können prozedural dargestellt werden. Das Tanken eines Autos oder das Betreten eines Restaurants sind ebenfalls feste Abfolgen von Aktionen.

Häufig arbeiten wir auch mit **logischem Wissen**. Wird der FC Hansa Rostock mindestens Tabellenvierter in der 1. Fußballbundesliga, so ist er für einen internationalen Wettbewerb qualifiziert. Dies ist eine logische Implikation. Aus A folgt B : $A \rightarrow B$. Wird Rostock Tabellen-dritter (A), so kann geschlossen werden, dass B gilt.

In diesem Kapitel werden ausgewählte Wissensrepräsentationsformalismen behandelt, die die obigen Wissensarten in unterschiedlicher Art umsetzen.

Die am besten untersuchte Form der Wissensrepräsentation ist die **Logik**. Es werden zwei Formen betrachtet: die **Aussagenlogik** und die **Prädikatenlogik erster Stufe**.

Semantische Netze und **Frames** sind spezielle Formen einer graphisch bzw. objektorientierten Darstellung für Objekte und deren Eigenschaften. Diese Formen der Wissensrepräsentation beinhalten Vererbungstechniken.

Regel-basierte Sprachen setzen das Konzept der bedingten Ausführung von Aktionen um:

WENN in einem Zustand bestimmte Bedingungen erfüllt sind, DANN führe eine Aktion aus.

Im letzten Abschnitt wird auf **vage Aussagen**, deren Darstellung und Verarbeitung eingegangen. Vage Aussagen sind Aussagen wie: „Es ist hell“, „Morgen wird es mindestens zehn Stunden Sonnenschein geben“.

Für das Studium von weiterführenden Konzepten sei auf die Literatur verwiesen ([BoHeSo07], [Ginsberg93], [Görz03]). Diese Konzepte befassen sich beispielsweise mit:

- zeitlichem Wissen,
- unvollständigem Wissen,
- Constraints (Wissensdarstellung mittels einschränkender Bedingungen),
- Fall-basiertem Schließen.

■ 2.1 Darstellung von Wissen mit Hilfe von Logik

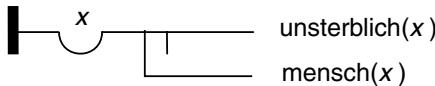
Logik ist nach [Brockhaus]

... die Lehre vom schlüssigen und folgerichtigen Denken, v. a. vom richtigen Schließen, welches dadurch gekennzeichnet ist, dass es zu wahren Voraussetzungen wahre Schlüsse liefert.

Die Logik spielt im menschlichen Alltag eine große Rolle. Wissen in Form von logischen Zusammenhängen nimmt einen zentralen Platz ein.

Zunächst wird die Aussagenlogik eingeführt und gezeigt, wie Aussagen automatisch, also durch einen Computer verarbeitet werden können.

Danach wird die Prädikatenlogik erster Stufe betrachtet. Der in Wismar geborene Mathematiker und Philosoph Gottlob Frege (1848–1925) gilt als der Begründer der modernen Logik. Mit seiner Begriffsschrift [Frege93] legte er 1878 die Grundlage für den Prädikatenkalkül erster Stufe. Frege baute die Logik auf den drei Operationen *Implikation*, *Negation* und *Allquantifizierung* auf und stellte Aussagen grafisch dar.



Die Aussage wird interpretiert als:

Für alle x gilt: Wenn x ein Mensch ist, dann ist x nicht unsterblich

Die grafische Notation hat sich nicht durchgesetzt, da sie beispielsweise für eine maschinelle Verarbeitung von Wissen nicht geeignet ist. Dennoch hat Frege mit dieser Formalisierung einen, wenn nicht *den* entscheidenden Schritt getan, um die Logik einer automatischen Verarbeitung zugänglich zu machen.

2.1.1 Aussagenlogik

Aussagen der folgenden Form werden von uns im Alltag häufig benutzt:

- *Es regnet.*
 - *Die Firma hat Gewinn erwirtschaftet.*
 - *Der deutsche Fußballmeister 2018 heißt Hansa Rostock.*

Solche Aussagen können nur wahr oder falsch sein, sie sind nicht halbwahr oder wahrscheinlich. Dagegen ist bei der Aussage:

Er ist groß.

durchaus nicht klar, ob sie wahr oder falsch ist. Dies hängt zumeist von der beurteilenden Person ab.

Die klassische Logik betrachtet nur zweiwertige Aussagen.

- Jede Aussage ist entweder wahr oder falsch.
 - Es gibt keine Aussage, die sowohl falsch als auch wahr ist.

Man bezeichnet diese Annahmen als den *Satz vom ausgeschlossenen Dritten* bzw. vom *ausgeschlossenen Widerspruch*.

Es werden nur Aussagen betrachtet, die wahr oder falsch sind. Man muss aber nicht unbedingt den Wahrheitswert kennen. Die GOLDBACHsche Vermutung:

Jede gerade natürliche Zahl lässt sich darstellen als Summe zweier Primzahlen.

ist bis heute nicht bewiesen oder widerlegt. Diese Aussage ist aber in jedem Fall entweder wahr oder falsch.

In der Aussagenlogik werden elementare Aussagen durch **aussagenlogische Variablen** dargestellt.

Man kann Aussagen aber auch verknüpfen.

Sei A diese Aussage:

„Die BMW-Aktie ist um 10 % gestiegen.“

Sei R die Aussage

Der Dollar ist um 1 % gefallen.“

so steht die Formel $A \wedge B$ für die Verknüpfung der beiden Aussagen A und B .

Die BMW-Aktie ist um 10 % gestiegen und der Dollar ist um 1 % gefallen.

Die Aussagen A und B sind **elementare** oder **atomare Aussagen**. Man bezeichnet A und B als **aussagenlogische Variablen**. Die Aussage $A \wedge B$ ist eine **zusammengesetzte Aussage**. Neben dem logischen Und gibt es weitere Möglichkeiten der Kombination von Aussagen. Sie sind in Tabelle 2.1 dargestellt.

TABELLE 2.1 Aussagenlogische Verknüpfungen

Symbol	Bezeichnung	Beispiel	Aussage ist wahr, falls
\wedge	Und	$X \wedge Y$	X und Y wahr sind
\vee	Oder	$X \vee Y$	X oder Y wahr sind
\neg	Negation	$\neg X$	X falsch ist
\rightarrow	Implikation	$X \rightarrow Y$	Y wahr ist, falls X wahr ist
\leftrightarrow	Äquivalenz	$X \leftrightarrow Y$	X genau dann wahr ist, wenn Y wahr ist

Die Definition der Verknüpfungen kann mit Hilfe einer Wahrheitswerttabelle erfolgen (Tabelle 2.2). Als Symbol für **wahr** wird W und als Symbol für **falsch** F verwendet.

Man beachte, dass das logische Oder das *einschließende Oder* darstellt. $A \vee B$ ist auch wahr, falls beide Teilaussagen A und B wahr sind. Das ausschließende Oder (exclusive or, XOR) entspricht dem Entweder-Oder. Das XOR ist insbesondere falsch, wenn beide Teilaussagen A und B wahr sind.

TABELLE 2.2 Wahrheitswerttabelle aussagenlogischer Verknüpfungen

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$	$A \text{ XOR } B$
W	W	F	W	W	W	W	F
W	F	F	F	W	F	F	W
F	W	W	F	W	W	F	W
F	F	W	F	F	W	W	F

Die Aussage $A \rightarrow B$ steht für:

Wenn A wahr ist, dann ist B zwingend auch wahr.

Man beachte, dass keine Aussage über den Fall, dass A falsch ist, gemacht wurde! In diesem Fall betrachtet man die Aussage folglich als wahr.

Dies kann anhand der Regel

Wenn Hansa Rostock Meister ist, dann spielt Hansa Rostock in der Champions League.

illustriert werden. Diese Aussage ist auch wahr, falls Hansa (leider) nicht Meister ist.

Die Aussage $A \leftrightarrow B$ steht für

A ist genau dann wahr, wenn B wahr ist.

Diese Aussage ist wahr, wenn A, B beide wahr bzw. beide falsch sind.

Aussagen können beliebig verknüpft werden. Die Verknüpfungsoperatoren können nicht nur auf elementare Aussagen angewendet werden, sondern auch auf bereits zusammengesetzte Aussagen. Damit kann man den Begriff einer aussagenlogischen Formel definieren.



Definition 2.1 Aussagenlogische Formeln

Alle Aussagenvariablen sind Formeln.

Sind A und B aussagenlogische Formeln, dann auch:

$A \wedge B$

$A \vee B$

$\neg A$

$A \rightarrow B$

$A \leftrightarrow B$

Das Übersetzen von Alltagsaussagen in die formale Logik bereitet häufig Probleme. Die Tabelle 2.3 stellt typische Satzmuster und deren formale Darstellung gegenüber.

TABELLE 2.3 Alltagssprache und logische Formeln

Satz	Aussagenlogische Form
... und ...	$A \wedge B$
Sowohl ... als auch ...	$A \wedge B$
Aber, jedoch, obwohl	$A \wedge B$
... oder ...	$A \vee B$
Wenn A dann B ; aus A folgt B	$A \rightarrow B$
A , vorausgesetzt dass B gilt	$B \rightarrow A$
A , falls/wenn B	$B \rightarrow A$
A nur dann, wenn B	$A \rightarrow B$
A genau dann, wenn B	$A \leftrightarrow B$
Entweder A oder B	$(A \vee B) \wedge (\neg A \vee \neg B)$
Weder A noch B	$\neg A \wedge \neg B$
A , es sei denn B	$(B \rightarrow \neg A) \wedge (\neg B \rightarrow A)$
Es stimmt nicht, dass ...	$\neg(\dots)$

Wird die Aussagenlogik als Wissensrepräsentationssprache verwendet, interessiert die Frage, ob eine Aussage **wahr** oder **falsch** ist.

Es gibt zum einen Aussagen, die *immer wahr* bzw. *immer falsch* sind; zum anderen gibt es auch Aussagen, deren Wahrheitswert von dem der *Teilaussagen* abhängt.

Die Aussage:

Die SAP-Aktie steigt oder die SAP-Aktie steigt nicht.

ist eine *immer wahre* Aussage, da genau eine der Teilaussagen wahr ist.

Die Aussage:

Die SAP-Aktie steigt und die SAP-Aktie steigt nicht.