Universität Bielefeld

9 Ausgewählte Themen

30. Vorlesung: Wiederholung und Zusammenfassung Teil 2

Methoden der Künstlichen Intelligenz

Ipke Wachsmuth WS 2000/2001

Universität Bielefeld

Methoden der KI WS 2000/2001

4 Logik und Inferenz
- Schlußfolgern im Prädikatenkalkül; Deduktion; Skolemisierung
- Unifikation; Vorwärts- und Rückwärtsverkettung ; Goal Trees
- Indexing; Assoziative Netzwerke: Inferenz durch Graphsuche

5 Spezielle Schlußverfahren
- Abduktion und Hypothesenbildung; Induktion und Lernen
- Probabilistisches Schließen; Nichtmonotones Schließen
- Temporales und räumliches Schließen

6 Einführung in Expertensysteme
- Regelbasiertes Programmieren und Expertensysteme
- Wissensmodellierung: Rapid Prototyping; modellbasiert / KADS
- Diagnostische Problemlösungsmethoden

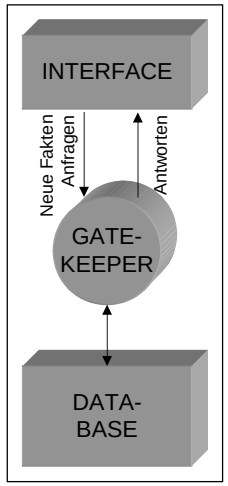
Universität Bielefeld

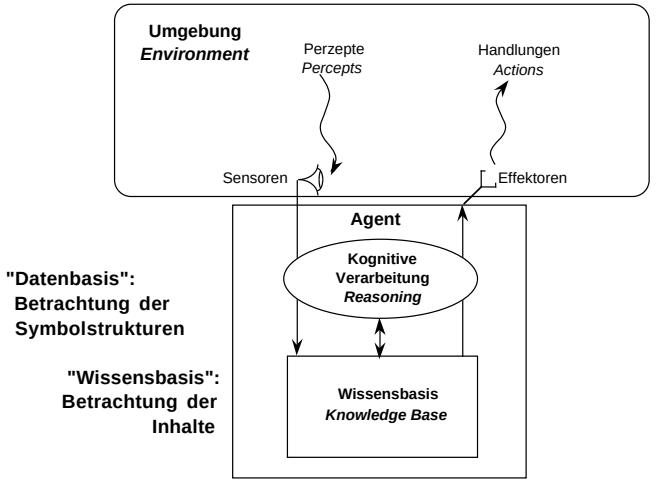
Logik und Inferenz: Übersicht

- Ausgangspunkt: Ausstattung eines Agenten mit Schluß-folgerungsfähigkeiten
- Basis-Aufbau mit DATABASE und GATEKEEPER und Basisoperationen `assert`, `retract`, `query`
- Verschiedene deduktive Inferenzverfahren, insbesondere für variablenhaltige Formeln der PL
- Vorbereitung der Formeln für maschinelle Verarbeitung (insbesondere Skolemform) durch Theorembeweiser
- Indexing-Verfahren; Inferenz durch Graphsuche in assoziativen Netzen

Universität Bielefeld

Ziel: Schlußfolgernder Agent





Universität Bielefeld

Übersicht Inferenz-regeln

(I) Modus Ponens
Aus p und (if p q) inferiere q

(II) Universelle Einsetzung
Aus (forall (-*vars*-) p) inferiere p mit allen Vorkommen jeder Variable durch den gleichen Term eingesetzt

(III) Unifikations-Inferenz
Aus p' und (if p q) inferiere q'
wobei p mit p' unifizieren muß
und die resultierende Substitution auf q angewandt wird, wodurch man q' erhält.

(IV) Subsumtions-Inferenz
Aus p subsumiert q schließe
q folgt aus p

(V) Allgemeine Resolutionsregel
Aus (or n_1 (not m')) und (or m n_2) inferiere (or n_1' n_2')

mit unifizierender Substitution analog zu (III)

Universität Bielefeld

Formelvorbereitung – Verfahren

Quantorenelimination, für maschinelle Formelverarbeitung

1. Bestimme, welche Variablen existenz- und welche allquantifiziert sind (den "wirklichen" Quantorentyp).
2. Ersetze jede existenzquantifizierte Variable durch eine Skolemfunktion. Die Argumente dieser Funktion sind alle diejenigen allquantifizierten Variablen, in deren Skopus der Existenzquantor liegt.
3. Falls zwei verschiedene allquantifizierte Variablen gleiche Namen haben, benenne eine davon „brandneu" um (Standardisierung).
4. Ersetze schließlich jede allquantifizierte Variable v durch eine mit "?" markierte match-Variable $?v$.
   (Die Allquantifizierung bleibt implizit dadurch gegeben, daß eine solche Variable mit allem "matcht".)

Universität Bielefeld

Übersicht Unifikation

Unifikation:
das Substituieren von Variablen so, daß zwei Ausdrücke (Terme oder Formeln) gleich werden.

Unifikator:
Die Substitution θ, die zwei Ausdrücke (Terme oder Formeln) gleich macht. (D.h. ein Unifikator ist eine spezielle Substitution!)

Subsumtion:
Eine Formel p subsumiert eine Formel q, wenn q aus p durch eine Variablensubstitution hervorgeht. (Formeln, die sich gegenseitig subsumieren, heißen Varianten.)

Der MGU ist *eindeutig* (bis auf Varianten)

Allgemeinster Unifikator (MGU):
Diejenige unifizierende Substitution, die am wenigsten Spezialisierungen vornimmt. Man kann zeigen: ***Für jede unifizierbare Menge von Formeln existiert ein MGU!***

Universität Bielefeld

Vorwärts- & Rückwärtsverkettung

Forward Chaining

Aus p' und (if p q) inferiere q'

wobei p mit p' unifiziert mit MGU θ und $q' = q\theta$

- Inferenz zur Assertion Time
- "Die Assertion resolviert mit der Implikation."

Backward Chaining

(if p q) sei assertiert.

Wenn nach q' gefragt wird (als goal) und q, q' haben MGU θ wird $p' = p\theta$ als subgoal aufgeworfen.

- Inferenz zur Query Time
- "Das goal resolviert mit der Implikation."

◆ Rückwärtsverkettung: zielorientiertes Inferenzverfahren
(bei konjunktiven (sub-)goals: in Verbindung mit Goal Tree-Suche)

Universität Bielefeld

Theorembeweiser – Goal Tree

Allgemein:

Gegeben das goal `(Show:q')` **mit** `(if p q)` **in DATABASE und** q, q' **haben MGU** θ .

Produziere subgoal `(Show:`$p\theta$`)` **und falls das eine Antwort** ψ **hat, ist** $\theta \cup \psi$ **eine Antwort auf das ursprüngliche goal.**

„Antwortsubstitution“

Bei konjunktiven (sub)goals z.B. `(if (r1 r2) q)`**:**

Bei konjunktiven Goals sind Antworten für ein Konjunkt zu finden, die auch Antworten für die anderen Konjunkte sind.

Algorithmus: Theorembeweiser ("Deductive Retriever")

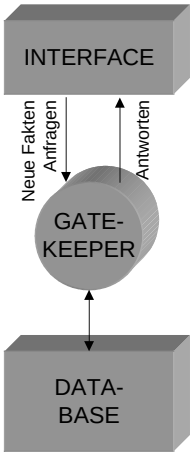
---> Goal Tree-Suche

Die UND-Knoten eines Goal Trees enthalten *constraints:* Randbedingungen an die Lösungen für jede ihrer Komponenten.

Hier sind das die folgenden: Die Variablen-bindungen für gleich benannte Variablen in den Teillösungen müssen identisch sein!

Universität Bielefeld

Indexing-Verfahren



Frage:

- Wie organisiert man geschickt den Zugang zu Einträgen in DATABASE und deren maschinelle Benutzung?

Zwei Basisansätze:

- Indexieren prädikatenlogischer Formeln in DATABASE und dafür geeignete Inferenzalgorithmen in GATEKEEPER
- DATABASE mit assoziativen Netzwerken und Inferenz durch Graphsuche-Algorithmen in GATEKEEPER

Universität Bielefeld

Spezielle Schlußverfahren...

Abduktives Schließen

Gegeben daß (if p q) **und** q **gilt**

schließe, daß p **gilt.**

Kein legaler Schluß in der Logik, aber oft benutzt, um Hypothesen zu generieren.

Induktives Schließen

Gegeben daß `(Pa), (Pb), ...` **gilt**

schließe, daß `(forall(x) Px))` **gilt.**

Kein legaler Schluß in der Logik, ist aber eine der Grundlagen von Lernen.

Probabilistisches Schließen (z.B.)

$A \rightarrow B$ (70%)

Die Regel gilt in 70% der Fälle.

Nichtmonotones Schließen (z.B.)

$A \rightarrow B$ UNLESS C

Die Regel gilt, außer wenn eine der Ausnahmen aus C zutrifft.

Universität Bielefeld

Konzeptlernen durch Induktion

Lernen aus Beispielen und Gegenbeispielen:

- Als Musterfall wurde das Lernen des Konzepts „arch“ (Bogen) in einem semantischen Netz betrachtet.
- Durch jedes weitere Beispiel/Gegenbeispiel werden schon induzierte Beschreibungen weiter differenziert.
- Eine besondere Rolle spielen dabei „knapp verfehlte“ Gegenbeispiele („near misses“).
- Einsatz verschiedener Induktionsheuristiken

- **Require-link**
- **Forbid-link**
- **Climb-Tree**

Universität Bielefeld

Unsicherheit, Unvollständigkeit

- In der klassischen Logik kann nur ausgedrückt werden, daß eine Aussage wahr oder falsch ist, jedoch nicht, daß man eine Aussage für wahrscheinlich hält oder über ihr Zutreffen nichts weiß.
  - In Anwendungen kommen solche Fälle häufig vor (Unsicherheit und Unvollständigkeit des Wissens).
  - Wissensrepräsentation und Schlußverfahren müssen entsprechend erweitert werden.
- Hauptansätze:
  - probabilistisches Schließen
  - nichtmonotones Schließen

Universität Bielefeld

Probabilistisches Schließen

u.a. nach dem Bayes-Satz (Voraussetzungen beachten!)

Berechnung der wahrscheinlichsten Diagnose D_i unter Annahme der Symptome S_1 ... S_m aus:

- den A-priori-Wahrscheinlichkeiten P(Di) einer Menge von n Diagnosen und
- den bedingten Wahrscheinlichkeiten P(Sj/Di) (der statistischen Häufigkeit des Auftretens der Symptome bei geg. Diagnose Di)

(zu berechnen für jedes i)

$$P(D_i / S_1 \& \ldots \& S_m) = \frac{P(D_i) * P(S_1 / D_i) * \ldots * P(S_m / D_i)}{\sum_{j=1}^{n} P(D_j) * P(S_1 / D_j) * \ldots * P(S_m / D_j)}$$

Universität Bielefeld

Nichtmonotones Schließen

hier nur: pragmatischer Ansatz („Belief Revision“)

Man nimmt an, daß eine Formel gilt, solange ihre Negation nicht bewiesen wurde:

- „unbeweisbar“ wird ersetzt durch „bisher nicht bewiesen“
- Ableitbarkeit relativ zum Problemlösungszustand definiert
- Abhängigkeiten der einzelnen Schlüsse von Vorbedingungen werden in einem Abhängigkeitsnetzwerk aufgehoben
- falls sich später herausstellt, daß eine Annahme falsch war: Schluß zurücknehmen (verschiedene Ansätze, u.a. JTMS, ATMS)

–> Truth-Maintenance-Systeme
(erheblich effizienter als nichtmonotone Logiken)

Universität Bielefeld

Temporales und räumliches Schließen

Ansätze zur Behandlung von Zeit

- Resultatbezogene Darstellung: Situationskalkül
- Tokenbezogene Darstellungen:
  - intervallbasierte
  - punktbasierte
- Beispiele: Overlap Chaining, TSA Zeitkartenverwaltung (TMM), Allens Kalkül

FRAME-Problem !

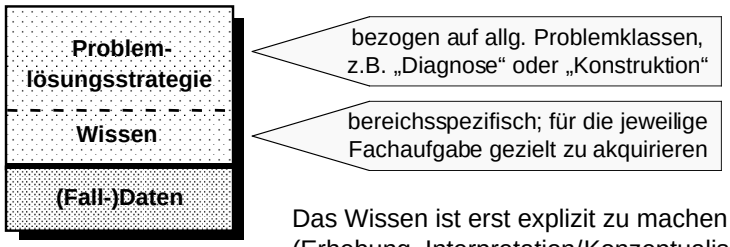
(nur kurz angesprochen)

- Felder räumlichen Schließens
- Beispiel Wegsuche
- Pfadorientierter Ansatz
- Hierarchische Wegsuche
- Formorientierter Ansatz

Universität Bielefeld

Einführung in Expertensysteme

Wissensbasiertes System: ein Problemlösesystem, in welchem explizites Wissen eines menschlichen Problemlösers ("Experten") eingebettet ist, um die Komplexität des Suchraums zu kontrollieren.



Das Wissen ist erst explizit zu machen!
(Erhebung, Interpretation/Konzeptualisierung, Formalisierung, Operationalisierung...)

Universität Bielefeld

Rapid Prototyping

Idee: Frühzeitig eine Mini-Version des angestrebten Systems bauen, als Referenz für die weitere Entwicklung. Beschränkung auf das zur Behandlung weniger ausgewählter Fälle notwendige Wissen.

Vorteile: Fokussierung bei der Wissenserhebung und -interpretation; Überzeugungsfaktor gegenüber beteiligtem Experten und dem Management für die Durchführbarkeit des Projekts.

Nachteil: Gefahr, daß vor tiefergehender Analyse und Charakterisierung des Problems unadäquate Design-Entscheidungen getroffen werden, die für den weiteren Systemausbau eher hinderlich sind.

Problem: Der konzeptionelle Abstand zwischen den erhobenen Daten und Implementierungen ist möglicherweise zu groß.

Universität Bielefeld

Modellbasierter Ansatz

Vorteile:

- Klare Trennung von Analyse und Implementierung (evtl. von verschiedenen Personen durchführbar)
- nach Analyse relativ vollständige Expertise-Beschreibung
- konzeptuelles Modell transparent und dokumentierbar
- generische Modelle von Aufgabenstrukturen leiten die Interpretation und die Wissensakquisition
- Fehler der Interpretation resultieren nicht in unnötiger Implementierungsarbeit; Änderungsaufwand geringer

Nachteil: Vorteile des Prototyping entfallen, insbesondere kann die Dynamik des Modells nicht direkt überprüft werden.

Universität Bielefeld

Problemanalyse mit KADS

unterstützt durch Bibliothek von **Interpretationsmodellen**

- Bereichsebene

  Beschreibung von Fachbegriffen und Relationen dazwischen (z.B. in Form von Regeln)

- Inferenzebene

  Einteilung der Fachbegriffe und Relationen gemäß ihrer Rolle beim Problemlösen in „Metaklassen“ und „Wissensquellen“

- Aufgabenebene

  Formulierung generischer Problemlösungsstrategien mit den definierten Metaklassen und Wissensquellen

- Strategieebene

  Erfolgsüberwachung und Wechsel zwischen verschiedenen Problemlösungsstrategien

Universität Bielefeld

Metaklassen & Wissensquellen
in der KADS-Methodik

- Metaklassen teilen die auf Bereichsebene identifizierten Fachbegriffe ein (z.B. „Symptome“, „Verdachtsdiagnosen“...)
- Wissensquellen setzen Metaklassen in Beziehung und klassifizieren Inferenzwissen, z.B. als verschiedene Typen von Regeln:
  - Verdachtgenerierungsregeln
  - Verdachtüberprüfungsregeln
  - Indikationsregeln

  } Bereichsrelationen werden auf Inferenzebene nach ihrer Rolle gruppiert!
- Metaklassen und Wissensquellen dienen zur Charakterisierung der Problemlösestrategie auf der Aufgabenebene.

Universität Bielefeld

Problemlösungsmethoden

Problemklassen:

abstrakte Zusammenfassung von Aufgabenstellungen, die sich in Bezug auf die Natur des Problemlöseprozesses ähneln,

etwa:
- Diagnostik
- Konstruktion
- Simulation

(können feiner unterteilt sein)

Problemlösungsmethoden:

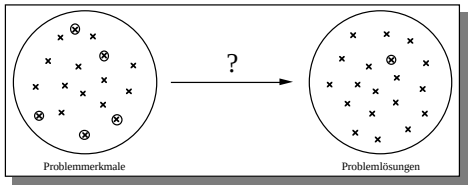
zu den Klassen passende Techniken und algorithmische Vorgehensweisen
- schwache, z.B. Regeln mit Vorwärtsverkettung (weniger interessant)
- starke (zwar weniger flexibel, aber mit wenig Aufwand für ein konkretes System umsetzbar)

Universität Bielefeld

Diagnostische Problemlösung

- Problembereich:
  - explizit gegebene Mengen von Problem-Merkmalen (Symptomen) und Problemlösungen (Diagnosen) und typischerweise unsicherem Wissen über Beziehungen zwischen Symptomen und Diagnosen
- Diagnoseproblem: Teilmenge der Symptome
- Lösung: Eine oder mehrere Diagnosen

Diagnostik ist grundsätzlich eine Klassifikationsaufgabe

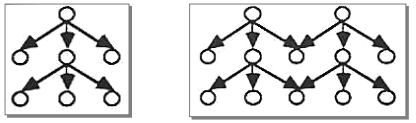


Universität Bielefeld

Problemlösungsstrategien u.a.

Establish-Refine-Strategie
- (strenge Diagnosehierarchien) Zunächst eine Diagnoseklasse durch Rückwärtsverkettung bestätigen, dann verfeinern (d.h. Nachfolger bestätigen).

Hypothesize &Test-Strategie
- aus eingegebenen Symptomen durch Vorwärtsverkettung Verdachtsdiagnosen generieren (Hypothesen), die dann gezielt durch Rückwärtsverkettung überprüft werden (Test).



- Hierarchien sind gut mit der Establish-Refine-Strategie,
- Heterarchien weit besser mit der Hypothesize-and-Test-Strategie auswertbar
  (erfordert zusätzlich Wissen zur Verdachtsgenerierung).