Inferenzmethoden

apl. Prof. Dr.-Ing. habil.

Rainer Knauf

Fachgebiet Künstliche Intelligenz

Fakultät für Informatik & Automatisierung

Technische Universität Ilmenau

Zuse-Bau, Raum 3060 (Sekretariat)

Tel. 03677 69-1445, 0361 3733867, 0172 9418642

rainer.knauf@tu-ilmenau.de

Auf der Seite

<u>http://www.tu-ilmenau.de/ki/lehre/</u>

sind downloadbar:

1. das Skript zur Vorlesung
2. <u>diese</u> Foliensammlung als PowerPoint-Präsentation
3. eine Sammlung von Übungsaufgaben

Inhaltsübersicht

1. Einführung
2. Architektur und Realisierungskonzepte
   - 2.1 Wissensrepräsentationsformen der angewandten KI, zugehörige Inferenzmethoden und technische Realisierungskonzepte:
     Prädikatenkalkül, Semantische Netze, Frames, Produktionssysteme, ...
   - 2.2 Inferenzmethoden in der Logik
     - 2.2.1 Deduktion
       Wesen & Eigenschaften der Folgerungsrelation, Folgern nach ROBINSON (Erweiterung auf Klauseln, Systematisierung der Suche einer Resolutionsfolge) Natürliche Inferenz nach GENTZEN
     - 2.2.2 Induktion
       Generierung von Klassifikationsregeln (u.a. ID3), Induktive Inferenz im PK1 (u.a. Anti-Unifikation), BAYES'sches Lernen
     - 2.2.3 Abduktion
   - 2.3 Anforderungen an sonstige Komponenten
3. Ausgewählte aktuelle Forschungsthemen

Inhaltsverzeichnis

(Inhaltsverzeichnis)

1 Einführung in die Künstliche Intelligenz (KI)

Ziel : Mechanisierung von Denkprozessen

Grundidee *(nach G.W. Leibniz)*

1. lingua characteristica — Wissensdarstellungssprache
2. calculus ratiocinator — Wissensverarbeitungskalkül

Teilgebiete der KI

Fokus der Vorlesung

- Wissensrepräsentation
- maschinelles Beweisen (Deduktion)
- Lernen (Induktion)
- Wissensbasierte Systeme
- KI - Sprachen
- Wissensverarbeitungstechnologien (Suchtechniken, fallbasiertes Schließen, Multiagenten-Systeme)
- Sprach- und Bildverarbeitung

Wissensbasierte Systeme: ***Was zeichnet sie aus?***

1. **Wissensbasis**: Wissensdarstellungen (bei „Bedeutungs-Schwängerung“ Wissen verkörpernde Symbolfolgen) werden explizit (getrennt von Verarbeitungskomponenten) abgelegt.
2. **Inferenz**: Nicht explizit abgelegte Wissensdarstellungen werden durch (typischerweise nichtdeterministische) Inferenzverfahren abgeleitet.
3. **Erklärungsfähigkeit**: Die (Kette von) Schlussfolgerungen kann auf Anforderung erklärt werden.
4. **Lernfähigkeit**: Eine Eigenschaft von (Künstlicher) Intelligenz die Gabe, aus dem eigenen Agieren und dem resultierenden Feedback seine Wissensbasis systematisch zu verfeinern.

2 Architektur und Realisierungskonzepte

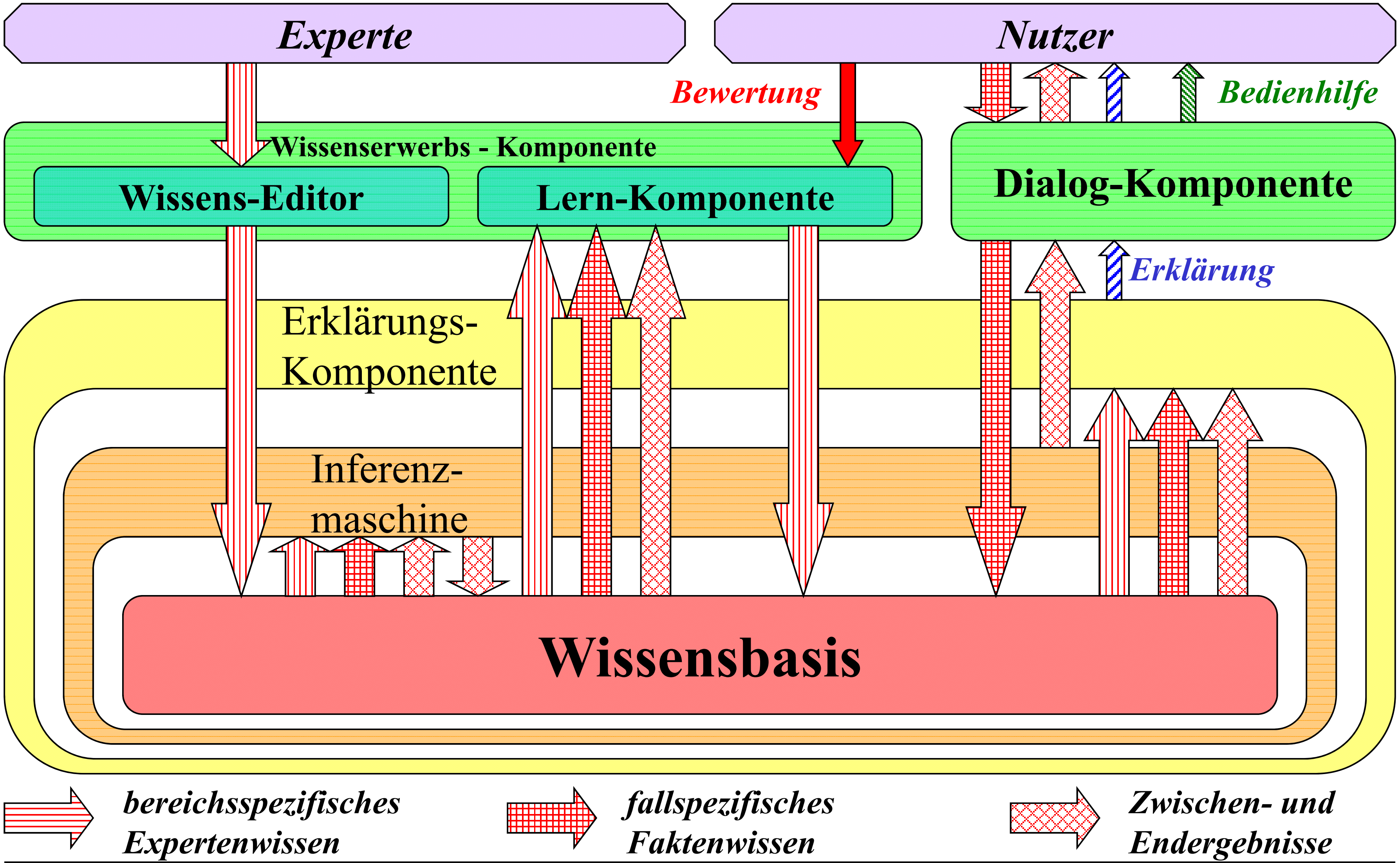
Wissensbasierte Systeme:

Welche Typen dargestellten Wissens muss man unterscheiden?

1. Bereichsspezifisches Expertenwissen
   - *beim Systementwurf zu formulierende "statische Wissensbasis"*
   - *Inhalt bleibt bei der Systemkonsultation unverändert*
   - *nur Lernverfahren können diese ändern*
2. Fallspezifisches Faktenwissen
   - *ergibt sich durch die Nutzereingaben während der Konsultation*
3. Zwischen- und Endergebnisse
   - *durch Inferenz abgeleitetes Wissen*

Welche Ebenen dargestellten Wissens muss man unterscheiden?

1.	Nutzerebene	*problemorientiert*
2.	implementierungsorientierte Ebene	*„tool"-orientiert*
3.	Systemebene	*system-immanent*


Experte
Nutzer
Bewertung
Bedienhilfe
Wissenserwerbs - Komponente
Wissens-Editor
Lern-Komponente
Dialog-Komponente
Erklärung
Erklärungs-
Komponente
Inferenz-
maschine
Wissensbasis
bereichsspezifisches
Expertenwissen
fallspezifisches
Faktenwissen
Zwischen- und
Endergebnisse

2.1 Wissensdarstellung

Deklarative (**deskriptive**) Repräsentationsideen basieren auf der Annahme, dass man Wissen unabhängig davon betrachten kann, wie es verarbeitet wird.

Beides, Wissen und dessen Verarbeitung, wird sauber getrennt.

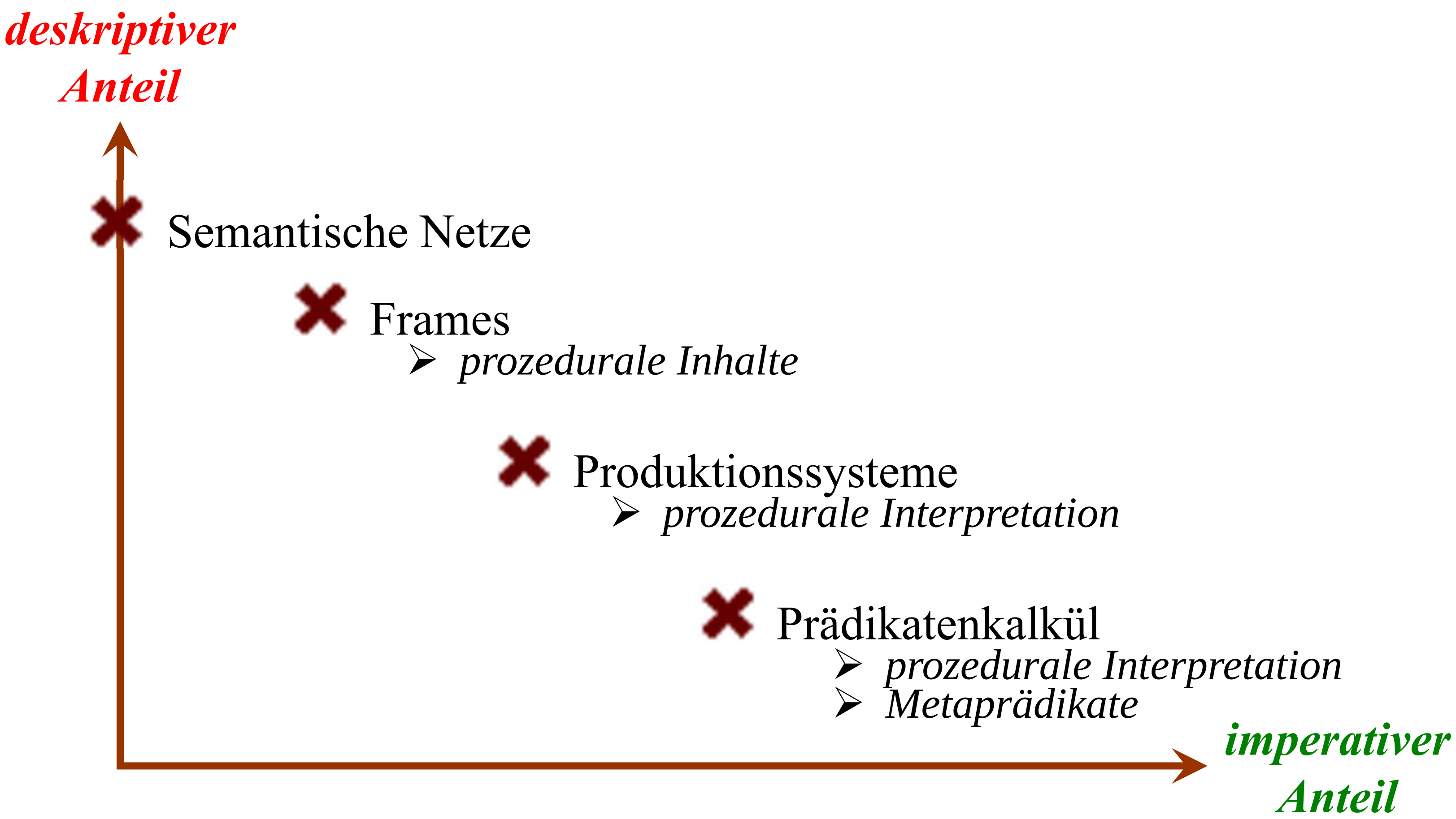
- Wissenserwerb konzentriert sich auf Inhalte
- einmalige Speicherung trotz Verwendung in verschiedenen Kontexten
- verschiedenen „Modelle“ mit gleicher Inferenzmaschine bearbeitbar
- Modifizierung der Wissensbasis ohne Nebeneffekte möglich
- keine effiziente Verarbeitung

Prozedurale (**imperative**) Repräsentationsideen hingegen betonen den Verarbeitungsaspekt.

Es wird auch Wissen über die Verarbeitung von Wissen (Metawissen) repräsentiert und die Trennung verwaschen.

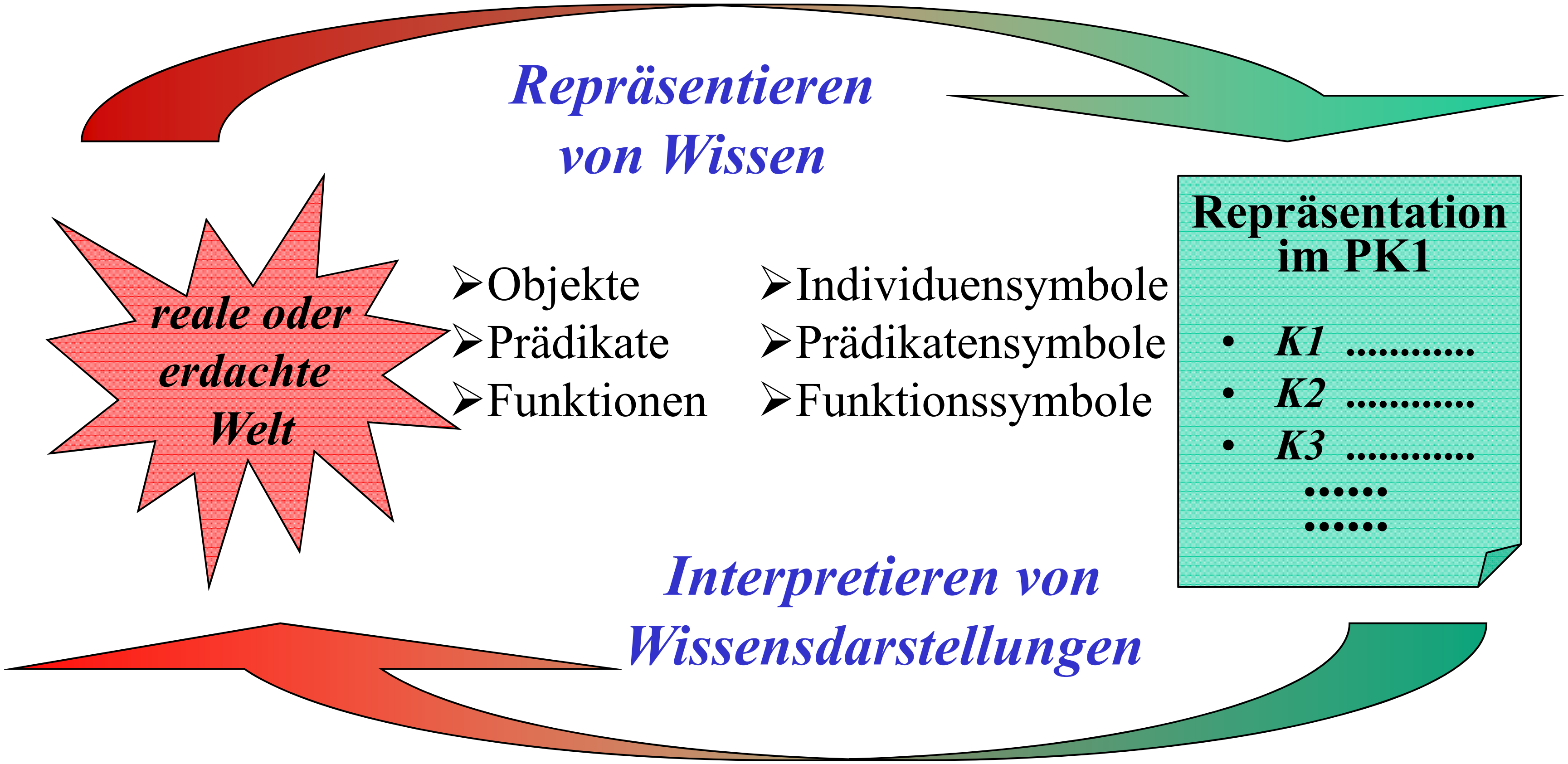
- effiziente Verarbeitung
- schlechte/keine Les-, Editier- und Interpretierbarkeit der Wissensbasis

Praktizierte deklarative (deskriptive) Repräsentationsformen sind oft um prozedurale (imperative) Komponenten erweitert:



2.1.1 Prädikatenkalkül der ersten Stufe (PK1)

Der PK1 als Wissensdarstellungskalkül



... zur Erinnerung (KI, 4. Semester):

- **Korrektheit**: $\forall H\,(\,M \vdash H \;\rightarrow\; M \models H\,)$
- **Vollständigkeit**: $\forall H\,(\,M \models H \;\rightarrow\; M \vdash H\,)$
- **Ein PK n ist für n>1 nicht vollständig.**

  m.a.W.:

  - *Für „echte" Erweiterungen des PK1 (welche seine Ausdrucksfähigkeit erhöhen) gibt es keine vollständigen Inferenzverfahren.*

... und trotzdem mitunter sinnvoll:

Erweiterungen des PK1

2.1.1.1 Erweiterungen des PK1

Erweiterung # 1: Sortenlogik

<u>*Idee*</u>

Die Objektmenge (das Universum) wird in Teilmengen jeweils gleicher Sorte partitioniert, z.B. Personen, Gegenstände, Zahlen, ...

1. Variablen und Individuensymbole (Konstanten) sind von einer Sorte S, welche als Index notiert wird: X_S, c_S, ...
2. Jedem n - stelligen Funktionssymbol f ist ein $(n+1)$ - Tupel von Sorten $[S_1, \dots, S_{n+1}]$ zugeordnet.
   Es werden nur Terme der Gestalt $f(t_1, \dots, t_n)$ zugelassen, bei denen t_i von der Sorte S_i ist und die Sorte des gesamten strukturierten Terms S_{n+1} ist.
3. Jedem n - stelligen Prädikatensymbol p ist ein n - Tupel $[S_1, \dots, S_n]$ zugeordnet.
   Es werden nur Atomformeln $p(t_1, \dots, t_n)$ zugelassen, bei denen t_i von der Sorte S_i ist.

Durch Einführung von 1-stelligen Sortenprädikaten $p_s(t)$, welche einen Term t auf ***wahr*** abbilden, gdw. er von der Sorte S ist, kann die Sortenlogik auf die sortenfreie Logik reduziert werden:

- für variablenfreie Ausdrücke $\Phi(t_S)$ gilt
  - $\mathbf{\Phi(t_S) \equiv \Phi(t) \wedge p_S(t)}$

- für Variablen X_S in Ausdrücken $\Phi(X_S)$ gilt
  - $\mathbf{\exists\, X_S\ \Phi(X_S) \equiv \exists\, X\ (\ \Phi(X) \leftarrow p_S(X)\ )}$
  - $\mathbf{\forall\, X_S\ \Phi(X_S) \equiv \forall\, X\ (\ \Phi(X) \leftarrow p_S(X)\ )}$

- für HORN-Klauseln bedeutet letzteres
  - $\mathbf{\forall\, X_S\ (\ A(X_S) \leftarrow B(X_S)\ )\ \equiv\ \forall\, X\ (\ A(X) \leftarrow B(X) \wedge p_S(X)\ )}$

Erweiterung # 2: Prädikatenlogik mit Gleichheit

<u>*Idee*</u>

- *Die Menge der variablenfreien Terme (Grundterme) wird in Teilmengen partitioniert, welche jeweils dasselbe Objekt bezeichnen.*
- *Durch einen besonderen Resolutionsschritt, welcher verschiedene syntaktische Charakterisierungen für dasselbe Objekt substituiert, wird die semantische Identität bei der Inferenz berücksichtigt.*
- *Die semantische Identität wird durch ein 2-stelliges (infix notiertes) Prädikatensymbol* $\equiv$ *notiert.*
- *Die Definition der Äquivalenz, der Allgemeingültigkeit und der Kontradiktorizität von Aussagen sowie des Folgerns berücksichtigt auch die semantische Identität von Termen.*

1. Für jeden Term t gilt: $t \equiv t$
2. Wenn $(t_1 \equiv t_1') \wedge (t_2 \equiv t_2') \wedge \ldots \wedge (t_n \equiv t_n')$, so gilt
   - $f(t_1, \ldots, t_n) \equiv f(t_1', \ldots, t_n')$
   - $ag(\ p(t_1, \ldots, t_n) \rightarrow p(t_1', \ldots, t_n')\ )$

Paramodulationsregel (*für HORN-Klauseln*)

- Sei $\{ K_1 , \ldots , K_n \}$ eine Menge von Regeln und Fakten (Klauseln).
- Sei $H \equiv H_1 \wedge \ldots \wedge H_m$ eine Frage (Hypothese).
- Sei $(s \equiv t) \leftarrow B_1 \wedge \ldots \wedge B_p$ eine der o.g. Klauseln.
- Es gebe Termeinsetzungen ϑ_1 und ϑ_2 in die Variablen von s und einem in einem der H_i (etwa H_l) vorkommenden Term r , so dass $\vartheta_1(s) \equiv \vartheta_2(r)$.

$$M' \equiv \bigwedge_{i=1}^{n} K_i \wedge \neg \underbrace{(\bigwedge_{i=1}^{m} H_i)}_{H}$$

ist kontradiktorisch (*kt* M'), gdw. M' nach Ersetzen von H durch

$$\bigwedge_{i=1}^{l-1} \vartheta_2(H_i) \wedge \vartheta_2(H_l') \wedge \bigwedge_{k=1}^{p} \vartheta_1(B_k) \wedge \bigwedge_{i=l+1}^{m} \vartheta_2(H_i)$$

noch immer kontradiktorisch ist.

H_l' entsteht aus H_l durch Substitution von r durch t .

Hintereinanderanwendung von Resolution und Paramodulation

Der Wissensbasis werden die Eigenschaften der Gleichheitsrelation beigefügt:

- $\forall X \,(X \equiv X \leftarrow \text{true})$
- $\forall X \, \forall Y (X \equiv Y \leftarrow Y \equiv X)$
- $\forall X \, \forall Y \, \forall Z \,(X \equiv Z \leftarrow X \equiv Y \wedge Y \equiv Z)$

$$M \equiv \bigwedge_{i=1}^{n} K_i \wedge \neg \underbrace{\left(\bigwedge_{i=1}^{m} H_i\right)}_{H}$$

ist kontradiktorisch (*kt M*), gdw. durch wiederholte Anwendungen der Resolutionsmethode und/oder der Paramodulationsregel die leere Klausel $\mathit{false} \leftarrow \mathit{true}$ abgeleitet werden kann.

2.1.1.2 Die Praxis: Logische Programmierung und deren Beschränkung auf HORN-Logik

1. Über HORN-Klauseln gibt es ein korrektes und vollständiges Ableitungs-verfahren.
   - $\{K_1, ..., K_n\} \models H$, gdw. $\{K_1, ..., K_n\} \vdash_{ROB} H$

2. Die Suche nach einer Folge von Resolutionsschritten ist algorithmisierbar.
   - Das Verfahren „Tiefensuche mit Backtrack“ sucht systematisch eine Folge, die zur leeren Klausel führt.
   - Rekursive und/oder metalogische Prädikate stellen dabei die Vollständigkeit in Frage.

3. Eine Menge von HORN-Klauseln mit nichtleeren Klauselköpfen ist stets erfüllbar; es lassen sich keine Widersprüche formulieren.
   - $K_1 \wedge K_2 \wedge ... \wedge K_n \neq \text{false}$

Die systematische Erzeugung von (HORN-) Klauseln

1. Erzeugung der Verneinungstechnischen Normalform (VTNF)

$\neg\neg A \equiv A$

$\neg (A \wedge B) \equiv \neg A \vee \neg B$

$\neg (A \vee B) \equiv \neg A \wedge \neg B$

$\neg (A \rightarrow B) \equiv A \wedge \neg B$

$\neg (A \leftrightarrow B) \equiv (A \wedge \neg B) \vee (\neg A \wedge B)$

$\neg \forall X\, A(X) \equiv \exists X \neg A(X)$

$\neg \exists X\, A(X) \equiv \forall X \neg A(X)$

2. Erzeugung der Pränexen Normalform (PNF)

$\nabla X\, A(X) \circ B \equiv \nabla X\, (A(X) \circ B)$

- *falls X nicht in B vorkommt*
- $\nabla \in \{ \forall, \exists \}, \quad \circ \in \{ \wedge, \vee \}$

$A \rightarrow \nabla X\, B(X) \equiv \nabla X\, (A \rightarrow B(X))$

$\forall X\, A(X) \rightarrow B \equiv \exists X\, (A(X) \rightarrow B)$

$\exists X\, A(X) \rightarrow B \equiv \forall X\, (A(X) \rightarrow B)$

3. Erzeugung der SKOLEM'schen Normalform (SNF)

Notation aller existenzquantifizierten Variablen als Funktion derjenigen allquantifizierten Variablen, in deren Wirkungsbereich ihr Quantor steht.
Dies ist keine äquivalente - , wohl aber eine die Kontradiktorizität erhaltende Umformung.

4. Erzeugung der Konjunktiven Normalform (KNF)

Durch systematische Anwendung des Distributivgesetzes

$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$$

lässt sich aus der SNF $\forall X_1 \ldots \forall X_n\, A(X_1, \ldots, X_n)$ *stets die äquivalente KNF*

$$\forall X_1 \ldots \forall X_n \left((L_1^{\,1} \vee \ldots \vee L_1^{\,n_1}) \wedge \ldots \wedge (L_m^{\,1} \vee \ldots \vee L_m^{\,n_m}) \right)$$

erzeugen. Die L_i^k *sind unnegierte oder negierte Atomformeln und heißen positive bzw. negative Literale.*

5. Erzeugung der Klauselform (KF)

Jede der Elementardisjunktionen $(L_i^j \vee \ldots \vee L_i^{jk})$ *der KNF kann man als äquivalente Implikation (Klausel)*

$$(L_i^j \vee \ldots \vee L_i^{jm}) \leftarrow (L_i^{jm+1} \wedge \ldots \wedge L_i^{jk})$$

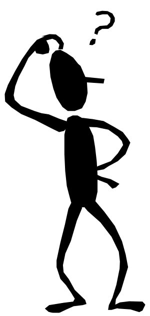
notieren, indem man alle positiven Literale $L_i^j, \ldots, L_i^{jm}$ *disjunktiv verknüpft in den* DANN-*Teil (Klauselkopf) und alle negativen Literale* $L_i^{jm+1}, \ldots, L_i^{jk}$ *konjunktiv verknüpft in den* WENN-*Teil (Klauselkörper) notiert.*

6. Sind die Klauseln aus Schritt # 5 HORN ?

In dem Spezialfall, dass alle Klauselköpfe dabei aus genau einem Literal bestehen, war die systematische Erzeugung von HORN-Klauseln erfolgreich; anderenfalls gelingt sie auch nicht durch andere Verfahren.

Heißt das etwa, die HORN-Logik ist eine echte Beschränkung der Ausdrucksfähigkeit ?

Richtig, das heißt es.



☺ *im Abschnitt 2.2.2 lernen wir eine Resolutionsmethode für Klauseln kennen*

☹ *... deren Algorithmisierbarkeit allerdings an der „kombinatorischen Explosion“ der Resolutionsmöglichkeiten scheitert.*

2.1.2 Semantische Netze

Repräsentation

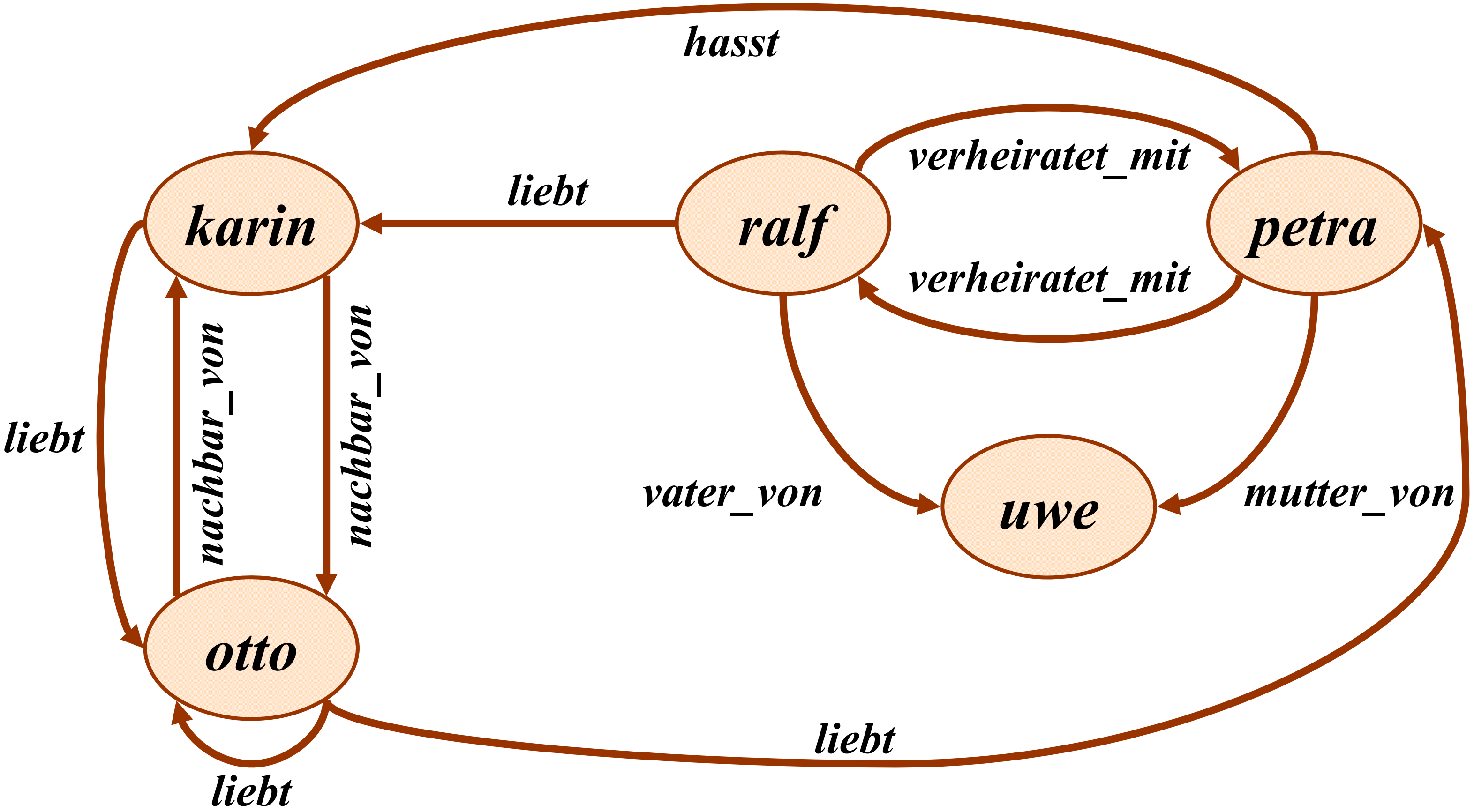
K:	Menge von Objekten (Entities)
V:	Menge 2-stelliger Relationen über **K** : $\mathbf{V} \subseteq \mathbf{K} \times \mathbf{K}$

gerichteter Graph $\mathbf{G = \{K,V\}}$ mit

K:	Menge von Knoten
V:	Menge gerichteter Kanten (Verbindungen)

Interpretation

´n Beispiel aus der Nachbarschaft:



Implementierungsvariante # 1:

jede Kante ...

$$K_i \xrightarrow{\textit{relation}} K_j$$

... als Fakt

$$\textit{relation}(\,K_i\,,\,K_j\,)\,.$$

Implementierungsvariante # 2:

jede Kante ...

$$K_i \xrightarrow{\textit{relation}} K_j$$

... als Fakt

$$\textit{kante}(\,\textit{relation}\,,\,K_i\,,\,K_j\,)\,.$$

- Beide Varianten der Realisierung Semantischer Netze sind „Teilsprachen“ von PROLOG (2- bzw. 3-stellige Fakten ohne Variablen).
- PROLOG ist eine „Teilsprache“ des PK1 (HORN-Klauseln des PK1)

$\Longrightarrow$ **Semantische Netze sind ausdrucksschwächer als der PK1**

Inferenz über Semantischen Netzen

Mittels „Metawissen" über die Eigenschaften der Relationen, z.B.

- Die Relation ***is_a*** (***a_kind_of***) ist transitiv:
  - $\forall X\ \forall Y\ \forall Z\ (is\_a(X,Z) \leftarrow is\_a(X,Y) \wedge is\_a(Y,Z))$
- Für die Relationen $\in$ (Element von) und ***is_a*** gilt:
  - $\forall X\ \forall Y\ \forall Z\ (is\_a(Z,X) \leftarrow is\_a(Y,X) \wedge \in(Z,Y))$
- Für die Relationen ***part_of*** (Teil von), ***has*** (hat) und ***is_a*** gilt:
  - $\forall X\ \forall Y\ (part\_of(X,Y) \leftrightarrow has(Y,X))$
  - $\forall X\ \forall Y\ \forall Z\ (has(X,Z) \leftarrow is\_a(X,Y) \wedge has(Y,Z))$
- . . .
- . . .

Erweiterungen des Wissensrepräsentationskonzepts

Ziel: Steigerung der Ausdrucksfähigkeit

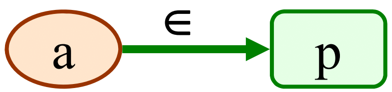
Erweiterung # 1

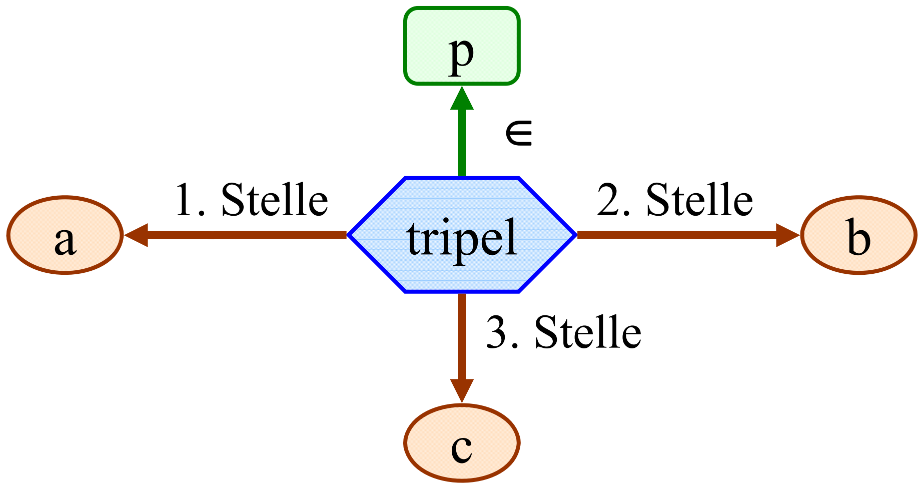
Ziel: Repräsentation von Relationen (Prädikaten) beliebiger Stelligkeit

Idee: auch die Relationen und Stelligkeiten werden als Knoten notiert

Ansatz: 1-stellige Prädikate — z.B. ***p(a)***

mehrstellige Prädikate — z.B. ***p(a,b,c)***





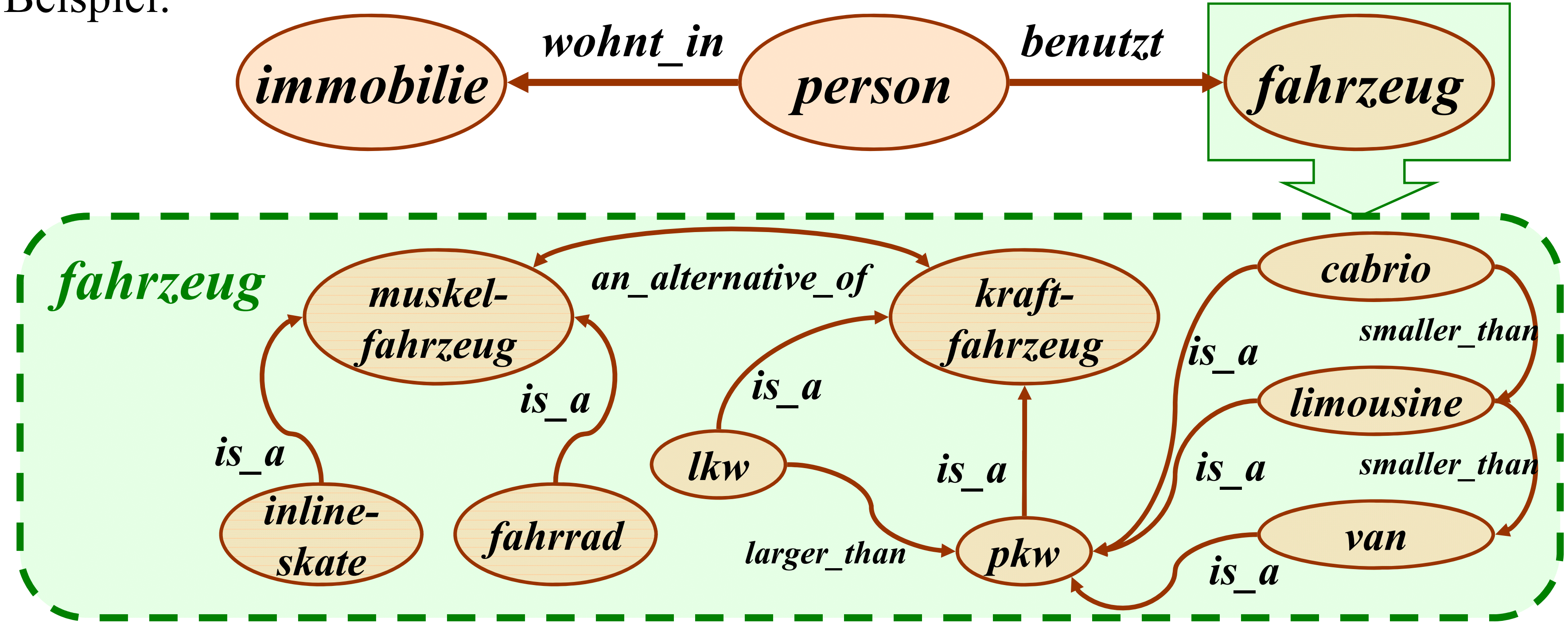
Erweiterung # 2

Ziel: Darstellung von Objekthierarchien

Idee: Einführung von Abstraktionsebenen

Ansatz: Knoten eines Netzes können (Sub-) Netze repräsentieren

Beispiel:



Vorteile

☺ **objektzentrierte Darstellung**
Alle explizite Information über ein Objekt konzentriert sich um den jeweiligen Knoten.

☺ **inferenzunterstützende Darstellung**
Die Kanten definieren Zugriffspfade.

☺ **Anschaulichkeit**

Nachteile

☹ **keine formale Semantik**

☹ **kein allgemeingültiger Verarbeitungskalkül**
Die Verarbeitung (Inferenz) erfolgt auf der Basis domänenspezifischen Wissens.

☹ **Akquisition und Hinterlegung domänenspezifischen Wissens ist für die Inferenz nötig**

Eine moderne Anwendung

- **Storyboards** zum Management von Lernprozessen
  - *Vortrag zur CACS 2010 Singapore*
    - ⇨ CACS 2010 presentation
  - Beispiele:
    - ⇨ Complete degree course on Information Environment at Tokyo Denki Univ.
    - ⇨ Intelligent Systems at Univ. of Central Florida
    - ⇨ Inference Methods at Ilmenau Univ. of Technology

2.1.3 Frames

Annahmen

1. **Das Wahrnehmen, Abbilden, Modellieren und Denken des Menschen ist von Erwartungen getrieben.**
   Beim Nennen eines Begriffs werden sofort Vorstellungen und Eigenschaften des bezeichneten Objekts damit assoziiert.
2. **Es gibt eine Objekthierarchie, in welcher Eigenschaften vererbt werden.**
   Diese werden einem Objekt a priori unterstellt; nur Spezifisches oder Abweichendes bedarf der Nennung.

Beispiel: ***Hund***

Erwartung:

Tier, 4 Pfoten, Schnauze, Schwanz, Bellfähigkeit, Beißgefahr, ausgeprägter Geruchssinn, ...

Spezifisches:

typische Notationsform

Framename
Oberkonzepte
Unterkonzepte
Slotname_1: Slotwert_1 $\text{Merkmal}_1{}^1$: $\text{Ausprägung}_1{}^1$ ... $\text{Merkmal}_n{}^1$: $\text{Ausprägung}_n{}^1$
••• ••• •••
Slotname_m: Slotwert_m $\text{Merkmal}_1{}^m$: $\text{Ausprägung}_1{}^m$ ... $\text{Merkmal}_n{}^m$: $\text{Ausprägung}_n{}^m$

Beispiele

Personenkraftwagen
Oberkonzepte: *Kraftfahrzeuge*
Unterkonzepte: *Cabrio, Limousine, Van*
Motor: *kraftstoffgetrieben* Kraftstoff: default *Benzin* Zylinderzahl: >= 2 Hubraum: > 1000 cm^3
Getriebe: default *Schaltgetriebe* Anzahl der Vorwärtsgänge: default 5 Anzahl der Getriebenen Räder: default 2
Anzahl der Räder: default 4

Limousine
Oberkonzepte: *Personenkraftwagen*
Unterkonzepte: *Compact, Mittelklasse, ...*
Gesamtmasse: >= 2 t
Sitzplätze: default 5
Anzahl der Räder: default 4

Ein Implementierungsvorschlag

- ***frame(FrameName , Liste_der_Oberkonzepte , Liste_der_Unterkonzepte)***
- ***slot(FrameName , SlotName , SlotWert)***
- ***merkmal(FrameName , SlotName , Merkmal , Auspraegung)***

Vererbungsregeln (= Inferenz)

slot(Frame , Slot , Wert) :-
        frame(OberFrame , _ , UnterFrames) ,
        member(Frame , UnterFrames) ,
        slot(OberFrame , Slot , Wert) .
merkmal(Frame , Slot , Merkmal , Auspraegung) :-
        frame(OberFrame , _ , UnterFrames) ,
        member(Frame , Unterframes) ,
        merkmal(OberFrame , Slot , Merkmal , Auspraegung) .

Vorteile

☺ **humanorientierte Darstellung**
Das Ausfüllen von Frames ist simpel.

☺ **humanorientierte Inferenz**
Schlussfolgern auf der Basis von Wissen über übergeordnete Konzepte ist dem Menschen adäquat.

☺ **effiziente Notation und Verarbeitung**

Nachteile

☹ **keine formale Semantik**

☹ **kein allgemeingültiger Verarbeitungskalkül**
Die Verarbeitung muss auf den Anwendungsfall zugeschnitten werden.

☹ **geringes Inferenzpotenzial**
Die Inferenz basiert allein auf Wissen über hierarchische Beziehungen.

über

2.1.4 Objektorientierte Wissensrepräsentationen

kann man sich in anderen Vorlesungen sehr viel detaillierter informieren ...

2.1.5 Produktionssysteme

Grundlagen

- Wissensrepräsentation:
  HORN Klauseln des Aussagenkalküls
- Inferenz:
  Hintereinanderanwendung des Modus Ponens

$$A \leftarrow \bigwedge_{i=1}^{n} B_i$$

$$\{A \leftarrow B, B\} \models A$$

Komponenten

1. Faktenbasis (Menge einfacher Aussagen des Aussagenkalküls)
2. Menge von Produktionsregeln (HORN-Klauseln des Aussagenkalküls)
3. Steuerkomponente, welche zyklisch
   a) die Menge der auf die Faktenbasis anwendbaren Regeln (die **Konfliktmenge**), d.h. diejenigen Regeln, bei denen alle Prämissen auch Element der Faktenbasis sind, ermittelt
   b) nach einer wohldefinierten Strategie (prioritätengesteuert, nach Reihenfolge der Notation, einer Heuristik folgend, ...) daraus eine Regel auswählt (= **Konfliktlösung**) und
   c) die gewählte Regel anwendet, indem sie die Konklusion dieser Regel der Faktenbasis hinzufügt.

Ein prioritätengesteuertes Produktionssystem

- **Fakten:** ***fakt(Fakt)***
- **Produktionsregel:** ***regel(Nr , Prioritaet , Konklusion , PraemissenListe)***

Steuerkomponente: Konfliktmenge ermitteln, Konfliktlösung, Regelanwendung

```
start :-      fakt(schluss).
start :-      konfliktmenge(K) ,
              auswahl(K,RegelNr) ,
              anwenden(RegelNr) ,
              !,  start.
konfliktmenge(L) :-
              bagof(Nr,anwendbar(Nr),L).
anwendbar(Nr) :-
              regel(Nr,_,Konkl,Praemissen),
              not(fakt(Konkl)),
              erfuellt(Praemissen).
```

```
erfuellt( [ ] ).
erfuellt( [K|R] ) :- fakt(K) ,
                     erfuellt(R).
auswahl([K|R],Nr) :- ausw(R,K,Nr).
ausw( [ ] , Nr , Nr ).
ausw( [K|R] , B , Nr ) :-
     regel(K,P1,_,_), regel(B,P2,_,_),
     P1 >= P2, !, ausw(R,K,Nr).
ausw( [ _ |R ], B, Nr) :- ausw(R,B,Nr).
anwenden(Nr) :-regel(Nr,_,Konkl,_),
                    assert(fakt(Konkl)).
```

Vorteile

☺ **wohldefinierte formale Semantik:** Wahrheitswerte

☺ **flexible Inferenz**
daten- / zielgetriebene Inferenz sowie verschiedene Strategien der Konfliktlösung realisierbar

☺ **logische Unabhängigkeit der Regeln**
erleichtert modularen Entwurf der Wissensbasis

Nachteile

☹ **wenig effiziente Inferenz**

- In jedem Zyklus muss das gesamte Regelwerk auf Anwendbarkeit geprüft werden.
- Werden beim „Anwenden“ einer Regel nicht auch etwas aus der Faktenbasis entfernt, nimmt die Kardinalität der Konfliktmenge monoton zu.

2.1.6 Prozedurale Repräsentationen

... kann man auch aus der Sicht der KI betrachten und stellt fest:

- ☹ „Fakten" sind nicht explizit repräsentiert. Sie sind die Daten der Prozeduren.
- ☹ Es gibt keine nichtdeterminierten Prozeduren. Die Reihenfolge der Problemlösungsschritte muss detailliert algorithmisch „vorgedacht" sein.
- ☹ Prozeduren haben ein Fließmuster. Es ist für jede Prozedur verbindlich geregelt, welches Ein- und Ausgabeparameter sind. Das durch relationale Betrachtung des den Prozeduren innewohnende Wissen ist nicht inferierbar.
- ☹ Domänenwissen ist nicht separiert und schwer les- und interpretierbar.
- ☹ Prozeduren sind unflexibel gegenüber Änderungen des Domänenwissens. *Üblicherweise muss man ein Update kaufen und kann nicht einfach eine Regel ändern.*
- ☹ Es besteht die Gefahr unerwünschten Systemverhaltens bei „nicht vorgedachten" Situationen.

☺ All' dies' ist der Preis für die sehr effiziente „Wissensverarbeitung" .

2.1.7 Zusammenfassung: elementare Wissensrepräsentationen der KI

nicht objektbezogene (aber formale) Semantik
hohe Modularität durch logische Unabhängigkeit der Wissenselemente

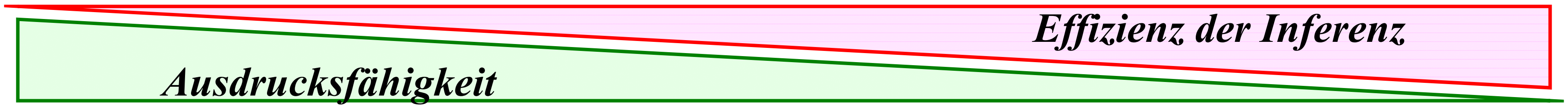
objektbezogene (informale) Semantik
geringe Modularität, Verlust von Wissenselementen macht Restwissen u.U. nicht zugreifbar

PK1
- Objekte, Relationen jeder Stelligkeit, quantifizierte Variablen
- Aussagen über die Relationen
- zielgetriebene (hypothesen-testende) Inferenz

Produktionssysteme
- HORN-Klauseln des Aussagen-kalküls
- daten-getriebene (hypothesen-bildende) Inferenz möglich

Frames
- erwartungs-getrieben
- objektzentriert
- effiziente Inferenz
- nur Aussagen über vorgegebene Rela-tionen inferierbar

Semantische Netze
- objektzentriert
- Objekte und (2-stellige) Relationen, variablenfrei
- effiziente Inferenz
- nur Aussagen über vorgegebene Rela-tionen inferierbar



2.2 Inferenzmethoden in der Logik

Inferenzmethoden = Wissen über die Verarbeitung von Wissen (Metawissen)

- Wissen über Wissensabstraktion

  Wann darf man aus speziellen Fällen allgemeine Aussagen inferieren?

  Methoden der induktiven Inferenz
- Wissen über Analogien

  Wann darf man neues Wissen durch Analogiebetrachtungen inferieren?

  Methoden der analogen Inferenz
- Wissen über Wissensintegrität

  Wie kann man durch die Kenntnis von Integritätsbedingungen (Konsistenz-betrachtungen in formalen Beschreibungen, Invarianzen und Gesetz-mäßigkeiten der Domäne, ...) neues Wissen erschließen?

  Methoden der Verifikation
- Wissen über Logik

  Aussagen über Aussagen: Tautologie, Kontradiktorizität, Folgern, Entscheid-barkeit, Monotonie, ...

  Methoden der deduktiven Inferenz
- **Ableitungstechnologien zur Umsetzung der o.g. Inferenzmethoden**

- Wissen über Eigenschaften solcher Ableitungsverfahren (= Meta-Metawissen ?)
  - in der induktiven Inferenz:
    Konvergenz (der [Gödel-]nummerierten Hypothesenfolge), Konsistenz, ...
  - in der analogen Inferenz:
    Wissen über die Legitimität von Analogieschlüssen (Vergleichbarkeit der Fälle)
  - bei der Verifikation:
    Konsistenzbedingungen, Invarianzen, Gesetzmäßigkeiten der Domäne, ...
  - in der deduktiven Inferenz:
    Korrektheit, Vollständigkeit
- Wissen über die Komplexität und technische Realisierbarkeit von Ableitungsverfahren
  (= Meta^3-Wissen ?)
- Wissen über Möglichkeiten der Beschränkung der Komplexität
  (= Meta^4-Wissen ?)
  Subsumtion von Aussagen, Stützmengenstrategie, ...

2.2.1 Deduktion

... ist die Kalkülisierung des **Folgerns**:

Sei M eine Menge von Aussagen, A eine Aussage.

A folgt aus M ($M \models A$) , falls jede Interpretation, die zugleich alle Elemente aus M wahr macht (jedes Modell von M), auch A wahr macht.

Für endliche Aussagenmengen $M = \{ A_1 , A_2 , \dots , A_n \}$ bedeutet das:

$M \models A$, gdw. $$ag(\bigwedge_{i=1}^{n} A_i \rightarrow A)$$

bzw. (was dasselbe ist) $$kt(\bigwedge_{i=1}^{n} A_i \wedge \neg A)$$

Wenn A die Menge aller (denkbaren) Aussagen ist, so beschreibt der Folgerungsbegriff eine 2-stellige **Folgerungsrelation** $Fl \subseteq P(A) \times P(A)$:

$$\mathbf{Fl = \{ [\, M , Fl(M) \,] : \; M \in P(A) \;\; , \;\; Fl(M) = \{ H: M \models H \} \}}$$

2.2.1.1 Hülleneigenschaften der Folgerungsrelation

Sei Fl(M)={H: M |= H}. Es gilt

1. **Satz der Einbettung** $\mathbf{M \subseteq Fl(M)}$

   Jede in M enthaltene Aussage ist auch Folgerung aus M.
2. **Satz der Monontonie** $\mathbf{M_1 \subseteq M_2 \;\rightarrow\; Fl(M_1) \subseteq Fl(M_2)}$

   Wenn M_1 eine Teilmenge von M_2 ist, dann ist die Menge der aus M_1 folgerbaren Aussagen eine Teilmenge der aus M_2 folgerbaren Aussagen.

   m.A.W.:

   Das Hinzufügen neuer Aussagen kann nur zur Erweiterung der Menge der Folgerungen führen. Vor dem Hinzufügen folgerbare Aussagen bleiben folgerbar.

Sei Fl(M)={H: M |= H}. Es gilt

3. Satz der Abgeschlossenheit **Fl(Fl(M)) $\subseteq$ Fl(M)**

Die Bildung der Menge der aus Fl(M) folgerbaren Aussagen Fl(Fl(M)) führt zu keiner neuen Folgerung.

... übrigens:

Wegen Einbettung **M $\subseteq$ Fl(M)** *bzw.* **Fl(M) $\subseteq$ Fl(Fl(M))**
und Abgeschlossenheit **Fl(M) $\supseteq$ Fl(Fl(M))**
... gilt: **Fl(M) = Fl(Fl(M))**

Die Eigenschaft, dass Fl(M) ggü. M nichts neues hervorbringt, hat einen Namen:

Definition

Ein Menge M von Aussagen, die mit der Menge der aus ihr folgerbaren Aussagen Fl(M) identisch ist (M = Fl(M)) heißt **deduktiv abgeschlossen**.

Hülleneigenschaften der Folgerungsrelation (3)

Sei $Fl(M)=\{H: M \models H\}$. Es gilt

4. **Endlichkeitssatz**

   $\mathbf{M \models H \;\rightarrow\; \exists\, M^*:\; M^* \subseteq M,\;\; card(M^*) < \infty\;,\;\; M^* \models H}$

   Zu jeder Menge M, aus welcher eine Hypothese H folgt, gibt es eine endliche Teilmenge M, aus der H auch folgt.*

5. **Ableitungstheorem** $\mathbf{M \models (H_1 \rightarrow H_2) \;\rightarrow\; M \cup \{H_1\} \models H_2}$

   Wenn aus M die Regel $H_1 \rightarrow H_2$ folgt, so folgt aus M inklusive Prämisse dieser Regel (H_1) deren Konklusion (H_2).

6. **Deduktionstheorem** $\mathbf{M \cup \{H_1\} \models H_2 \;\rightarrow\; M \models (H_1 \rightarrow H_2)}$

   ... dies gilt auch umgekehrt.

7. **Unvollständigkeitssatz** $\mathbf{\neg\, kt\, M \;\rightarrow\; \exists\, H: \neg(M \models H) \wedge \neg(M \models \neg H)}$

   Für jede widerspruchsfreie Menge M von Aussagen gilt: Es gibt eine Aussage H derart, dass weder sie selbst noch ihre Negation daraus folgt.

M |= H zu zeigen bedeutet definitionsgemäß

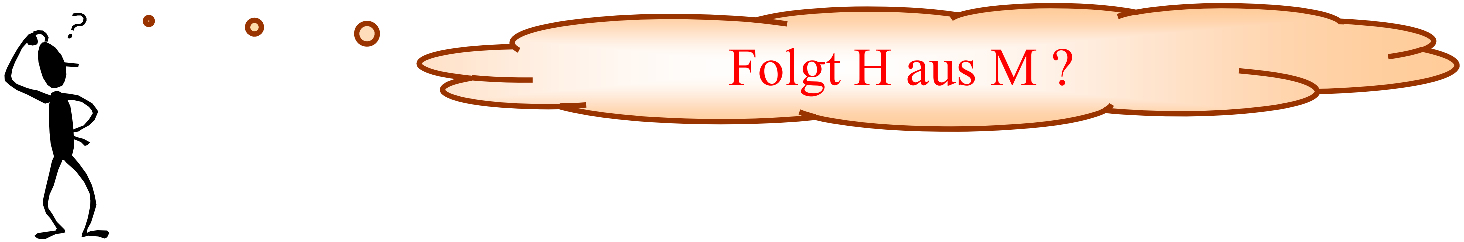
1. die Bildung aller Modelle für **M** (Interpretationen, welche alle $\mathbf{A_i} \in \mathbf{M}$ wahr macht) und
2. die Überprüfung des Warheitswertes für **H** in diesen Modellen

solange bis man

- ein Modell findet, für welches H falsch ist (=> **M |≠ H**) oder
- alle Modelle überprüft hat (=> **M |= H**).

Ein Beispiel, welches sich der Versuchung einer Interpretation entzieht:

$\mathbf{A_1}$	$\forall E\ \forall X\ \forall Y\ \forall Z\ (ulk(Z,verknusung(Y,flup(E,blubb))) \rightarrow ulk(flop(Z),Y))$
$\mathbf{A_2}$	$\forall E\ ulk(flop(flup(E,blubb)),blubb)$
$\mathbf{A_3}$	$\forall E\ \forall X\ \forall Y\ \forall Z\ ((ulk(Z,verknusung(flup(E,X),Y)) \rightarrow ulk(flop(Z),X)) \rightarrow ulk(Y,blubb))$
$\mathbf{A_4}$	$(\forall X\ \forall Y\ (ulk(X,Y) \rightarrow ulk(Y,X)) \ \wedge \forall E\ \forall X\ ulk(flup(E,X),blubb) \rightarrow \forall E\ ulk(blubb,flup(E,blubb)))$
$\mathbf{H}$	$\forall E\ \forall X\ ulk(verknusung(flop(flup(E,X)),blubb),X)$



Eine Interpretation: ***Kellerspeicher***

Symbol	Interpretation
blubb	leerer Stack
verknusung(St_1 , St_2)	derjenige Stack ***St_1+St_2***, der sich beim Auflegen des Stacks ***St_2*** auf den Stack ***St_1*** ergibt
flup(*E* , *St*)	derjenige Stack, der sich beim Auflegen des Elements ***El*** auf den Stack ***St*** ergibt
flop(*St*)	derjenige Stack, der sich beim Entnehmen des obersten Elements vom Stack ***St*** ergibt
ulk(St_1 , St_2)	bildet ***[St_1,St_2]*** auf ***wahr*** ab, gdw. ***St_1*** und ***St_2*** identisch sind, d.h. an gleichen Positionen gleiche Elemente aufweisen

A_1 , A_2 , A_3 , A_4 : wahr. H : wahr.

$\{ A_1 , A_2 , A_3 , A_4 \} \models H$ ist denkbar.

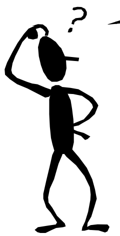
Noch eine Interpretation: *Warteschlange*

Symbol	Interpretation
blubb	leere Warteschlange
verknusung($Schl_1$, $Schl_2$)	diejenige Schlange $\boldsymbol{Schl_1{+}Schl_2}$, die sich durch das verketten der beiden Schlangen $\boldsymbol{Sch_1}$ und $\boldsymbol{Schl_2}$ ergibt
flup(*E* , *Schl*)	diejenige Schlange, die sich durch Anhängen des Elements ***El*** an die Schlang ***Schl*** ergibt
flop(*Schl*)	diejenige Schlange, die sich durch das Streichen des ersten Elements aus der Schlange ***Schl*** ergibt
ulk($Schl_1$, $Schl_2$)	bildet $\boldsymbol{[Schl_1,Schl_2]}$ auf ***wahr*** ab, gdw. $\boldsymbol{Schl_1}$ und $\boldsymbol{Schl_2}$ identisch sind, d.h. an gleichen Positionen gleiche Elemente aufweisen

A_2 , A_4 : wahr, A_1 , A_3 : falsch H : egal

$\{A_1, A_2, A_3, A_4\} \models H$ **ist denkbar.**

- *Interpretationen bilden ist irgendwie langweilig.*
- *Außerdem wird man nie wirklich fertig damit.*
- *Man müsste Folgern syntaktisch charakterisieren!*
- *Da gab's doch 'was ....*



2.2.1.2 Resolution nach ROBINSON

gegeben:

- Menge von Regeln und Fakten M
- negierte Hypothese $\neg H$

$$M = \{K_1, \ldots, K_n\}$$

$$\neg H \equiv \neg \bigwedge_{i=1}^{m} H_i \equiv false \leftarrow \bigwedge_{i=1}^{m} H_i$$

Ziel:

- Beweis, dass $M \models H$

$$\boxed{kt(\bigwedge_{i=1}^{n} K_i \wedge \neg H)}$$

Eine der Klauseln habe die Form $A \leftarrow \bigwedge_{k=1}^{p} B_k$. (*A*, B_k – Atomformeln)

Es gebe Termeinsetzungen ϑ_1 und ϑ_2 in die Variablen von *A* und eines der H_i (etwa H_l), so dass $\vartheta_1(A) \equiv \vartheta_2(H_l)$.

$$M' \equiv \bigwedge_{i=1}^{n} K_i \wedge \neg \underbrace{(\bigwedge_{i=1}^{m} H_i)}_{H}$$

ist kontradiktorisch (*kt M‘*), gdw. *M‘* nach Ersetzen von *H* durch

$$\bigwedge_{i=1}^{l-1} \vartheta_2(H_i) \wedge \bigwedge_{k=1}^{p} \vartheta_1(B_k) \wedge \bigwedge_{i=l+1}^{m} \vartheta_2(H_i)$$

noch immer kontradiktorisch ist.

Na „prima“!?
Jetzt wissen wir also, wie man die zu zeigende Kontradiktorizität auf eine andere – viel kompliziertere (?) – Kontradiktorizität zurückführen kann.

Für $p=0$ und $m=1$ wird es allerdings trivial.

Die sukzessive Anwendung von Resolutionen muss diesen Trivialfall systematisch herbeiführen:

Satz von ROBINSON

$$M' \equiv \bigwedge_{i=1}^{n} K_i \wedge \neg H$$

ist kontradiktorisch (*kt M‘*), gdw. durch wiederholte Resolutionen in endlich vielen Schritten die negierte Hypothese $\neg H \equiv false \leftarrow H$ durch die leere Klausel $false \leftarrow true \equiv$ ersetzt werden kann.

Substitution

- Eine (Variablen-) Substitution ϑ einer Atomformel A ist eine Abbildung der Menge der in A vorkommenden Variablen X in die Menge der Terme (aller Art: Konstanten, Variablen, strukturierte Terme).
- Sie kann als Menge von Paaren *[Variable,Ersetzung]* notiert werden:
  $\vartheta = \{[x,t]: x \in X, t = \vartheta(x)\}$
- Für strukturierte Terme wird die Substitution auf deren Komponenten angewandt:
  $\vartheta(f(t_1, ..., t_n)) = f(\vartheta(t_1), ... , \vartheta(t_n))$
- Verkettungsoperator $\circ$ für Substitutionen drückt Hintereinander-anwendung aus:
  $\delta \circ \vartheta(t) = \delta(\vartheta(t))$
- Substitutionen, die zwei Terme syntaktisch identisch machen, heißen **Unifikator**:
  ϑ unifiziert zwei Terme s und t (oder: heißt Unifikator von s und t), falls $\vartheta(s) \equiv \vartheta(t)$ ist.

Genügt „irgendein“ Unifikator?

ein Beispiel

$$K_1 : p(A,B) \leftarrow q(A) \wedge r(B)$$

$$K_2 : q(c) \leftarrow true$$

$$K_3 : r(d) \leftarrow true$$

$$\neg H : false \leftarrow \underbrace{p(X,Y)}_{H_1^0}$$

Unifikatoren für H_1^0 und Kopf von K_1:

$$\vartheta^1 = \{[X,a],[Y,b],[A,a],[B,b]\}$$

$$\vartheta^2 = \{[X,a],[Y,B],[A,a]\}$$

$$\vartheta^3 = \{[X,A],[Y,b],[B,b]\}$$

$$\vartheta^4 = \{[A,X],[B,Y]\}$$

… *u.v.a.m.*

- Obwohl $\{K_1, K_2, K_3\} \models H$, gibt es bei Einsetzung von ϑ^1, ϑ^2, und ϑ^3 keine Folge von Resolutionsschritten, die zur leeren Klausel führt.
- Bei Einsetzung von ϑ^4 hingegen gibt es eine solche Folge.
- Die Vollständigkeit des Inferenzverfahrens hängt von der Wahl des „richtigen“ Unifikators ab.
- Dieser Unifikator muss möglichst viele Variablen variabel belassen. Unnötige Spezialisierungen versperren zukünftige Inferenzschritte.

Ein solcher Unifikator heißt **allgemeinster Unifikator** bzw. most general unifier (*m.g.u.*).

Allgemeinster Unifikator

Eine Substitution ϑ heißt allgemeinster Unifikator (most general unfier ***m.g.u.***) zweier Terme s und t ($\vartheta = m.g.u.(s, t)$), gdw.

1. die Substitution ϑ ein Unifikator von s und t ist und
2. für jeden anderen Unifikator σ von s und t eine nichtleere und nicht identische Substitution τ existiert, so dass $\sigma = \tau \circ \vartheta$ ist.

graphisch betrachtet:

$$\begin{tikzcd} \bullet \arrow[r, "\vartheta"] \arrow[dr, "\sigma"'] & \bullet \arrow[d, "\tau"] \\ & \bullet \end{tikzcd}$$

Der Algorithmus zur Berechnung des ***m.g.u.*** *zweier Terme* s *und* t *verwendet* **Unterscheidungsterme**:

Man lese s und t zeichenweise simultan von links nach rechts. Am ersten Zeichen, bei welchem sich s und t unterscheiden, beginnen die Unterscheidungsterme s^* und t^* und umfassen die dort beginnenden (vollständigen) Teilterme.

Algorithmus zur Bestimmung des allgemeinsten Unifikators 2er Terme

input: s , t **output**: Unifizierbarkeitsaussage, ggf. $\vartheta = m.g.u.(s,t)$

$i := 0$; $\vartheta_i := \emptyset$

$s_i := s$; $t_i := t$

▶ s_i und t_i identisch?

ja ⇒ **s und t sind unifizierbar**, $\vartheta := \vartheta_i = m.g.u.(s,t)$ *(fertig)*

nein ⇒ Bilde die Unterscheidungsterme s_i^* und t_i^*

s_i^* oder t_i^* Variable?

nein ⇒ **s und t sind nicht unifizierbar** *(fertig)*

ja ⇒ sei *(o.B.d.A.)* s_i^* eine Variable

$s_i^* \subseteq t_i^*$? *(enthält t_i^* die Variable s_i^* ?)*

ja ⇒ **s und t sind nicht unifizierbar** *(fertig)*

nein ⇒ $\vartheta' := \{[s,t']: [s,t] \in \vartheta, t' := t|_{s^* \to t^*}\} \cup \{[s_i^*, t_i^*]\}$

$s' := \vartheta'(s_i)$; $t' := \vartheta'(t_i)$; $i := i+1$;

$\vartheta_i := \vartheta'$; $s_i := s'$; $t_i := t'$; gehe zu ▶

Resolutionsmethode für Klauseln *(… , welche nicht HORN sein müssen)*

Sei M = {K_1, …, K_n} eine Menge von Ausdrücken in Klauselform mit K_k, $K_l \in$ M:

$$K_k \equiv \bigvee_{i=1}^{m_A} A_i \leftarrow \bigwedge_{i=1}^{m_B} B_i \qquad\qquad K_l \equiv \bigvee_{i=1}^{m_C} C_i \leftarrow \bigwedge_{i=1}^{m_D} D_i$$

(mit A_i, B_i, C_i, D_i – Atomformeln) derart, dass ein A_i (etwa A_s) und ein D_i (etwa D_t) miteinander unifizierbar sind, d.h. $\vartheta(A_s) \equiv \vartheta(D_t)$ mit $\vartheta = m.g.u.(A_s,D_t)$.

$$M' = \bigwedge_{i=1}^{n} K_i$$ ist kontradiktorisch (*kt M´*), gdw. *M´* nach dem Hinzufügen von

$$K \equiv \bigvee_{i=1}^{s-1} \vartheta(\mathrm{A}_i) \vee \bigvee_{i=s+1}^{m_A} \vartheta(\mathrm{A}_i) \vee \bigvee_{i=1}^{m_C} \vartheta(\mathrm{C}_i) \leftarrow \bigwedge_{i=1}^{m_B} \vartheta(\mathrm{B}_i) \wedge \bigwedge_{i=1}^{t-1} \vartheta(\mathrm{D}_i) \wedge \bigwedge_{i=t+1}^{m_D} \vartheta(\mathrm{D}_i)$$

noch immer kontradiktorisch ist.

K heißt **Resolvente** von K_k und K_l .

Die Resolutionsmethode für Klauseln ist offenbar unvollständig. Etwa

kt $\left(\forall X(p(a) \vee p(X) \leftarrow true) \;\wedge\; \forall Y(false \leftarrow p(a) \wedge p(Y)) \right)$

*kann durch sukzessive Resolution **nicht** gezeigt werden. Abhilfe schafft hier die*

Faktorenregel

Sind innerhalb eines Klauselkopfes oder -körpers einer Klausel K zwei Atomformeln A_i und A_j miteinander unifizierbar mit $\vartheta = m.g.u.\ (A_i, A_j)$, so ist $\vartheta(K)$ (nach Verkürzung infolge Idempotenz) eine Resolvente.

Um die Anwendbarkeit dieser Regel muss folglich auch erweitert werden:

Satz von ROBINSON

$$M' \equiv \bigwedge_{i=1}^{n} K_i$$

ist kontradiktorisch (*kt M'*), gdw. durch wiederholte Anwendung von Resolutions-methode und Faktorenregel in endlich vielen Schritten die leere Klausel $false \leftarrow true \equiv$ abgeleitet werden kann.

Dass niemand ein Deduktionstool auf der Basis der Resolution über Klauseln geschrieben hat, liegt wohl an der Komplexität der Resolutionsmöglichkeiten.
Bekannte Maßnahmen zur Begrenzung dieser „kombinatorischen Explosion" sind:

1. **Entfernen tautologischer Klauseln**

2. **Entfernen von Klauseln, welche durch andere subsumiert werden**

> K_i subsumiert K_j, falls
> 1. durch ein und dieselbe Substitution ϑ jedes Literal aus K_i einem Literal aus K_j identisch wird und
> 2. K_i höchstens genauso viele Literale wie K_j hat.

3. **Beschränkung auf Resolutionen nach Stützmengenstrategie** *(set of support)*

> Wenn $M = M_1 \cup M_2$ (bei $M_1 \cap M_2 = \emptyset$) eine Klauselmenge mit erfüllbarem M_1 ist, dann lässt die Stützmengenstrategie nur solche Resolutionen zu, bei denen eine der beteiligten Klauseln aus M_2 ist oder durch Resolution mit Beteiligung von Klauseln aus M_2 gewonnen wurde.

Die „natürliche“ Inferenz nach GENTZEN (1) (*Symbol:* $\vdash_{NI}$)

Sei M eine Menge von Aussagen; H, H_1 und H_2 Aussagen. Es gelten

1. Satz der **Einbettung** $M \vdash_{NI} H$, wenn $H \in M$
2. Regeln zur **Einführung** und **Beseitigung** von Junktoren und Quantoren
   - a) **UND – Einführung**
     $M \vdash_{NI} (H_1 \wedge H_2)$, wenn $M \vdash_{NI} H_1$ und $M \vdash_{NI} H_2$
   - b) **UND – Beseitigung**
     $M \vdash_{NI} H_1$ und $M \vdash_{NI} H_2$, wenn $M \vdash_{NI} (H_1 \wedge H_2)$
   - c) **ODER – Einführung**
     $M \vdash_{NI} (H_1 \vee H_2)$, wenn $M \vdash_{NI} H_1$ oder $M \vdash_{NI} H_2$
   - d) **ODER – Beseitigung**
     $M \vdash_{NI} H$, wenn $M \vdash_{NI} (H_1 \vee H_2)$ und
     $M \cup \{H_1\} \vdash_{NI} H$ und $M \cup \{H_2\} \vdash_{NI} H$
   - e) **IMPLIKATION – Einführung**
     $M \vdash_{NI} (H_1 \rightarrow H_2)$, wenn $M \cup \{H_1\} \vdash_{NI} H_2$
   - f) **IMPLIKATION – Beseitigung**
     $M \vdash_{NI} H_2$, wenn $M \vdash_{NI} (H_1 \rightarrow H_2)$ und $M \vdash_{NI} H_1$
   - g) **ÄQUIVALENZ – Einführung**
     $M \vdash_{NI} (H_1 \leftrightarrow H_2)$, wenn $M \vdash_{NI} (H_1 \rightarrow H_2)$ und $M \vdash_{NI} (H_2 \rightarrow H_1)$

Die „natürliche“ Inferenz nach GENTZEN (2) (*Symbol:* $\vdash_{NI}$)

h) **ÄQUIVALENZ – Beseitigung**

$M \vdash_{NI} (H_1 \rightarrow H_2)$ und $M \vdash_{NI} (H_2 \rightarrow H_1)$, wenn $M \vdash_{NI} (H_1 \leftrightarrow H_2)$

i) **NEGATION – Einführung**

$M \vdash_{NI} \neg H$, wenn $M \cup \{ H \} \vdash_{NI} H^*$ und $M \cup \{ H \} \vdash_{NI} \neg H^*$

j) **NEGATION – Beseitigung**

$M \vdash_{NI} H$, wenn $M \vdash_{NI} \neg\neg H$

k) **ALLQUANTOR – Einführung**

$M \vdash_{NI} \forall X\, A(X)$, wenn $M \vdash_{NI}^{(a)} A(a)$

l) **ALLQUANTOR – Beseitigung**

$M \vdash_{NI}^{(a)} A(a)$, wenn $M \vdash_{NI} \forall X\, A(X)$

m) **EXISTENZQUANTOR – Einführung**

$M \vdash_{NI} \exists X\, A(X)$, wenn $M \vdash_{NI} A(a)$

n) **EXISTENZQUANTOR – Beseitigung**

$M \vdash_{NI} B$, wenn $M \vdash_{NI} \exists X\, A(X)$ und $M \cup \{ A(a) \} \vdash_{NI}^{(a)} B$

3. **Korrektheit** und **Vollständigkeit**

$M \vdash_{NI} H$, gdw. dies durch eine Folge von Ableitungsschritten gemäß (1) und (2) gezeigt werden kann.

$\vdash_{NI}^{(a)}$: ***Die Konstante*** a ***ist willkürlich; es ginge auch mit jeder anderen Konstanten.***

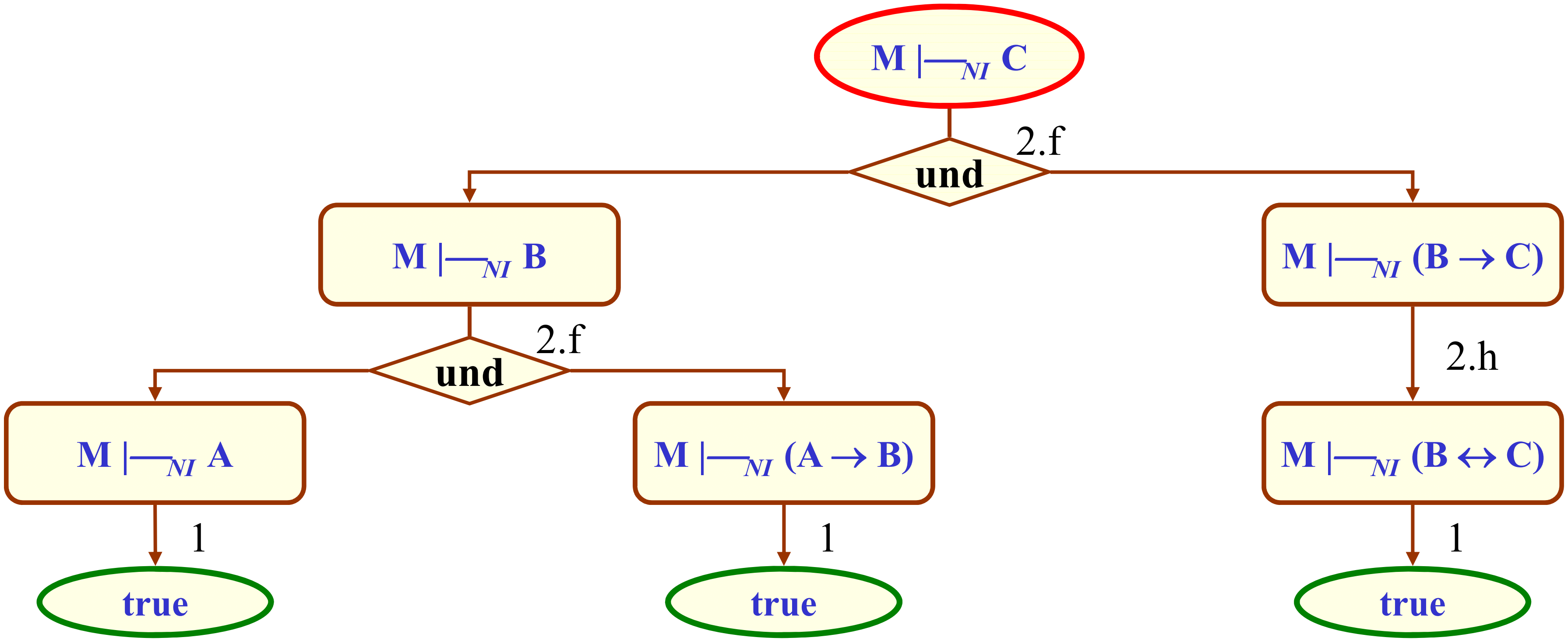
Die „natürliche“ Inferenz nach GENTZEN - *ein Beispiel aus dem Aussagenkalkül*

geg.

- $M = \{ (A \rightarrow B), (B \leftrightarrow C), A \}$
- $H = C$

ges.

- $M \mid\!\!\stackrel{?}{—}_{NI} H$



2.2.2 Induktion

2.2.2.1 Eine funktionentheoretische Interpretation des Induktionsproblems

Gegeben sei

- eine unvollständige Information in der Sprache der zulässigen Beispiele: *endliche Anfangsstücke von Folgen*
- die Klasse der zu lernenden Objekte: *die Klasse **R** der partiell-rekursiven (berechenbaren) Funktionen **f: IN → IN***
- ein Hypothesenraum: *GÖDEL-Nummer der Funktion, zugeordnet z.B. nach lexikographischer Reihenfolge der Notierungen der Hypothesen in ihrer Beschreibungssprache*

Gesucht ist

- ein Klassifikator: *eine „korrekte" Zuordnung von GÖDEL-Nummern zu einem Anfangsstück*

Einige interessante Ergebnisse

- Es gibt ***LIM ⊂ R*** : *die Klasse am endlichen Anfangsstück erkennbaren Funktionen*
- Es gibt ***FIN ⊂ LIM*** : *die Klasse von Funktionen, für welche es ein Prädikat gibt, welches entscheidet, ob die GÖDEL-Nummer der Hypothesenfolge den Endwert ihrer Konvergenz erreicht hat*
- Konsistenz der Beispiele mit den Hypothesen der Hypothesenfolge zu fordern, ist ehr schlecht für eine rasche Konvergenz

2.2.2.2 Eine aussagenlogische Interpretation des Induktionsproblems

Gegeben sei

- eine Menge von Objekten I
- ein k-dimensionaler (endlicher) Merkmalsraum, in welchem jeden Objekt $i \in I$ genau ein Punkt zugeordnet werden kann
- eine Partitionierung von I in eine endliche Menge von Objektklassen $U = \{ U_1, \ldots , U_n \}$
- eine Beispielmenge $B \subset I$: eine Menge von Objekten mit bekannter Klassenzugehörigkeit

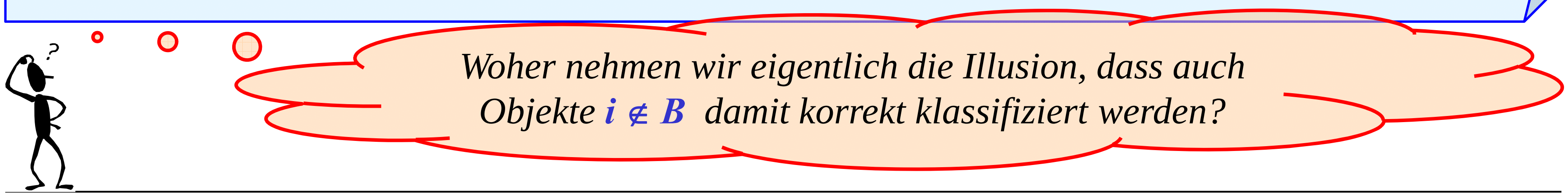
Gesucht ist

- ein Klassifikator: Eine Menge M von HORN-Klauseln

$$(Klasse = U_l) \leftarrow \bigwedge_{i=1}^{m} (Merkmal_i = Ausprägung_i^{\,j})$$

an die man folgende Anforderungen stellt:

1. Konsistenz: Alle Beispiele müssen korrekt klassifiziert werden.
2. Es sollten möglicht wenige Regeln sein: $| M | << | I |$
3. Es sollten möglichst „kurze“ Regeln sein: $m << k$



Eigenschaften von Anwendungen dieser Interpretation des Induktionsproblems

1. Der Merkmalsraum ist meist endlich.
   (*im Gegensatz zum Definitionsbereich partiell rekursiver Funktionen*)
   - Die „Repräsentativität“ der Beispielmenge kann man oftmals einschätzen.
2. Die Anzahl der Klassen ist meist endlich.
   (*im Gegensatz zur Menge der partiell rekursiven Funktionen*)
3. Es wird „nur“ Konsistenz mit der Beispielmenge gefordert.
   - Inwieweit der Anwender der Korrektheit des Klassifikators für „nicht - Beispiele“ Glauben schenkt, bleibt ihm überlassen.
   - Er minimiert sein Risiko durch möglichst hohe „Repräsentativität“ der Beispielmenge (*wie etwa bei EMNID-Umfragen*) .

Ein Beispiel

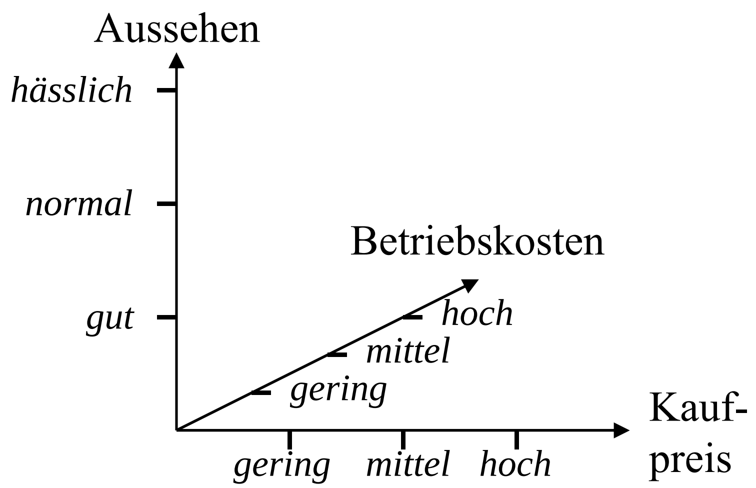
Gegeben sei

- eine Menge von Objekten: *Personenkraftwagen der Mittelklasse*
- ein dreidimensionaler (endlicher) Merkmalsraum, in welchem jeden Objekt genau ein Punkt zugeordnet werden kann: *Abbildung Merkmalsraum*
- drei disjunkte Objektklassen: *empfehlenswerte - , indifferente – und abratenswerte PKW*
- 14 Beispiele aus dem Bekanntenkreis: *Tabelle Beispielmenge*

Gesucht ist

- eine Menge von HORN-Klauseln zur Klassifikation der Objekte bzgl. ihrer Merkmale

Merkmalsraum:



Beispielmenge:

Kaufpreis	**Aussehen**	**Betriebskosten**	**Klasse**
mittel	***hässlich***	***gering***	***indifferent***
gering	***hässlich***	***gering***	***indifferent***
gering	***hässlich***	***normal***	***abratenswert***
hoch	***normal***	***normal***	***indifferent***
mittel	***gut***	***normal***	***empfehlenswert***
mittel	***normal***	***hoch***	***abratenswert***
mittel	***gut***	***hoch***	***abratenswert***
hoch	***normal***	***hoch***	***abratenswert***
gering	***normal***	***gering***	***empfehlenswert***
mittel	***normal***	***gering***	***empfehlenswert***
gering	***gut***	***gering***	***empfehlenswert***
hoch	***hässlich***	***normal***	***abratenswert***
hoch	***gut***	***normal***	***indifferent***
gering	***gut***	***normal***	***empfehlenswert***

2.2.2.2.1 Left-to-right Regelgenerierung

1. Partitioniere die Beispielmenge bezüglich des ersten (linken) Merkmals in

Kaufpreis = *hoch*

Aussehen	Betriebskosten	Klasse
normal	*normal*	*indifferent*
normal	*hoch*	*abratenswert*
hässlich	*normal*	*abratenswert*
gut	*normal*	*indifferent*

Kaufpreis = *gering*

Aussehen	Betriebskosten	Klasse
hässlich	*gering*	*indifferent*
hässlich	*normal*	*abratenswert*
normal	*gering*	*empfehlenswert*
gut	*gering*	*empfehlenswert*
gut	*normal*	*empfehlenswert*

Kaufpreis = *mittel*

Aussehen	Betriebskosten	Klasse
hässlich	*gering*	*indifferent*
gut	*normal*	*empfehlenswert*
normal	*hoch*	*abratenswert*
gut	*hoch*	*abratenswert*
normal	*gering*	*empfehlenswert*

Gehören alle Beispiele einer Partition mit $Merkmal_1 = Auspraegung_1^1$ zur gleichen Klasse K, so entsteht bilde die Regel

$$Klasse = K \leftarrow Merkmal_1 = Auspraegung_1^1$$

und setze das Verfahren mit jeder der verbleibenden Partitionen fort.

2. Partitioniere jede „nicht klassenreine“ Partition nach dem gleichen Verfahren stufenweise so lange weiter, bis Partitionen mit identischer Klassenzugehörigkeit entstehen.
   Jede neue Partitionierung auf der Basis eines Merkmals $\boldsymbol{Merkmal_i}$ in $\boldsymbol{m}$ Partitionen gemäß der auftretenden Ausprägungen $\boldsymbol{Auspraegung_i^1, \ldots, Auspraegung_i^m}$ führt zu einer konjunktiv hinzuzufügenden neuen Prämisse der entstehenden Regel $\boldsymbol{Merkmal_i = Auspraegung_i^j}$.

(Kaufpreis = *hoch*) ∧ (Aussehen = *normal*)

Betriebskosten	**Klasse**
normal	***indifferent***
hoch	***abratenswert***

⇨ weitere Partitionierung

(Kaufpreis = *hoch*) ∧ (Aussehen = *hässlich*)

Betriebskosten	**Klasse**
normal	***abratenswert***

➡ Regel:
Klasse = *abratenswert* ← (Kaufpreis = *hoch*) ∧ (Aussehen = *hässlich*)

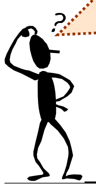
(Kaufpreis = *hoch*) ∧ (Aussehen = *gut*)

Betriebskosten	**Klasse**
normal	***indifferent***

➡ Regel:
Klasse = *indifferent* ← (Kaufpreis = *hoch*) ∧ (Aussehen = *gut*)

Für das Beispiel entsteht dieses Regelwerk zur Klassifikation:

Klasse = *abratenswert*	←	(Kaufpreis=*hoch*)	∧	(Aussehen=*hässlich*)		
Klasse = *indifferent*	←	(Kaufpreis=*hoch*)	∧	(Aussehen=*normal*)	∧	(Betriebskosten=*normal*)
Klasse = *abratenswert*	←	(Kaufpreis=*hoch*)	∧	(Aussehen=*normal*)	∧	(Betriebskosten=*hoch*)
Klasse = *indifferent*	←	(Kaufpreis=*hoch*)	∧	(Aussehen=*gut*)		
Klasse = *indifferent*	←	(Kaufpreis=*mittel*)	∧	(Aussehen=*hässlich*)		
Klasse = *empfehlenswert*	←	(Kaufpreis=*mittel*)	∧	(Aussehen=*normal*)	∧	(Betriebskosten=*gering*)
Klasse = *abratenswert*	←	(Kaufpreis=*mittel*)	∧	(Aussehen=*normal*)	∧	(Betriebskosten=*hoch*)
Klasse = *empfehlenswert*	←	(Kaufpreis=*mittel*)	∧	(Aussehen=*gut*)	∧	(Betriebskosten=*normal*)
Klasse = *abratenswert*	←	(Kaufpreis=*mittel*)	∧	(Aussehen=*gut*)	∧	(Betriebskosten=*hoch*)
Klasse = *indifferent*	←	(Kaufpreis=*gering*)	∧	(Aussehen=*hässlich*)	∧	(Betriebskosten=*gering*)
Klasse = *abratenswert*	←	(Kaufpreis=*gering*)	∧	(Aussehen=*hässlich*)	∧	(Betriebskosten=*normal*)
Klasse = *empfehlenswert*	←	(Kaufpreis=*gering*)	∧	(Aussehen=*normal*)		
Klasse = *empfehlenswert*	←	(Kaufpreis=*gering*)	∧	(Aussehen=*gut*)		



- ***13 Regeln aus 14 Beispielen zu erzeugen ist nicht wirklich gut.***
- ***Mit diesem Verfahren konnte kaum etwas durch die Induktion verallgemeinert werden!***
- ***Kann man das Verfahren vielleicht optimieren, dass es möglichst wenige und möglichst kurze Regeln erzeugt?***

2.2.2.2.2 Regelgenerierung nach dem ID3-Algorithmus

Idee

- Ein Merkmal, welches die Beispiele in gleich mächtige Teilmengen zu partitionieren vermag, gibt uns die Chance, die Anzahl der weiter „abzufragenden“ Merkmale zu minimieren.
- Eine minimale Anzahl „abzufragender“ Merkmale bedeutet eine minimale Anzahl von Regelprämissen, d.h. „kurze“ Regeln.

Grundlage

- Inwieweit die Ausprägungen eines Merkmals die Beispielmenge in gleich mächtige Teilmengen partitioniert, lässt sich durch Berechnung der durchschnittlichen **Entropie der Information** $Merkmal_i = Auspraegung_i^j$ über alle auftretenden Ausprägungen $Auspraegung_i^1, \ldots, Auspraegung_i^n$ quantifizieren.

Fußnote

Die Idee, dieses Maß über den Informationswert einer Aussage zur induktiven Inferenz optimaler Regelwerke zu nutzen, ist noch nicht alt: Sie wurde 1983 erstmals von J.R. Quinlan zur Bewertung von Endspielstellungen beim Schach vorgeschlagen.

Was ist die Entropie der Information?

- Ein Versuch ***P*** wird durchgeführt.
- Dabei können ***n*** verschiedene Versuchsergebnisse eintreten.
- Die Wahrscheinlichkeit des Auftretens des ***i***-ten Versuchsergebnisses sei p_i.

Die Entropie der Information ***H(P)*** über das Versuchsergebnis ist

$$H(P) = -\sum_{i=1}^{n} p_i \cdot ld(p_i)$$

Was ist demnach die Entropie der Information über die Klassenzugehörigkeit auf der Grundlage der Ausprägung eines partitionierenden Merkmals?

$$H(Auspr_i) = -\sum_{j=1}^{m} p(Klasse_j \mid Auspr_i) \cdot ld(p(Klasse_j \mid Auspr_i))$$

m	Anzahl der Klassen
$p(Klasse_j \mid Auspr_i)$	Wahrscheinlichkeit der Zugehörigkeit des Objekts zur Klasse $Klasse_j$ unter der Bedingung, dass das betrachtete Merkmal die Ausprägung $Auspr_i$ hat.

Was ist mit nicht auftretenden Ausprägungen?

Wegen $\lim_{p \to 0}(p \cdot ld(p)) = 0$ können solche Ausprägungen weggelassen werden.

Wenden wir dies auf unser Beispiel an!

Entropien der Ausprägungen des Merkmals **Kaufpreis**

H(Kaufpreis=gering) = - *p(empfehlenswert|gering) · ld(p(empfehlenswert|gering))*
- *p(indifferent|gering) · ld(p(indifferent|gering))*
- *p(abratenswert|gering) · ld(p(abratenswert|gering))*
= - (3/5) · ld(3/5) – (1/5) · ld(1/5) – (1/5) · ld(1/5)
= ***1.371***

H(Kaufpreis=mittel) = - (2/5) · ld(2/5) – (1/5) · ld(1/5) – (2/5) · ld(2/5)
= ***1.522***

H(Kaufpreis=hoch) = - (2/4) · ld(2/4) – (2/4) · ld(2/4)
= ***1***

Die durchschnittliche Entropie des Merkmals Kaufpreis ergibt sich nun durch Wichtung der Entropien seiner Ausprägungen mit der Wahrscheinlichkeit des Auftretens dieser Ausprägungen.

Randbemerkung

Natürlich müssen wir diese Wahrscheinlichkeit anhand der Beispielmenge schätzen; sie kann u.U. weit von der Realität abweichen, wenn die Beispielmenge „schlecht" ist.

$$
\begin{aligned}
H(Kaufpreis) = \; & p(Kaufpreis{=}gering) \cdot H(Kaufpreis{=}gering) \\
+ \; & p(Kaufpreis{=}mittel) \cdot H(Kaufpreis{=}mittel) \\
+ \; & p(Kaufpreis{=}hoch) \cdot H(Kaufpreis{=}hoch) \\
= \; & (5/14) \cdot 1.371 + (5/14) \cdot 1.522 + (4/14) \cdot 1 \\
= \; & \underline{\mathbf{1.319}}
\end{aligned}
$$

Analog kann man dies für die anderen Merkmale berechnen und kommt zu

$$
\begin{aligned}
H(Aussehen) &= \underline{\mathbf{1.319}} \\
H(Betriebskosten) &= \underline{\mathbf{1.026}}
\end{aligned}
$$

Was spricht nun für einen höheren Informationswert für die Klassenzugehörigkeit? Sollen dafür ehr große oder ehr kleine durchschnittliche Entropien sprechen?

Kommen (noch) viele Klassen in Frage, ist das schlecht für die Zugehörigkeitsaussage:

- Entropie ***H(Merkmal$_i$ = Auspraegung$_i^j$)*** bei Gleichverteilung
  - auf 2 Klassen mit je 50 % : ***1***
  - auf 4 Klassen mit je 25 % : ***2***
  - auf 8 Klassen mit je 12.5 % : ***3***
- Entropie ***H(Merkmal$_i$ = Auspraegung$_i^j$)*** bei Verteilung auf 4 Klassen
  - mit 25 %, 25 %, 25 %, 25 % : ***2***
  - mit 12.5 %, 37.5 %, 25 %, 25 % : ***1.905***
  - mit 12.5 %, 12.5 %, 37.5 %, 37.5 % : ***1.811***

$\Rightarrow$ ***Je kleiner dieser Entropiewert, desto weniger Klassen sind wahrscheinlich.***

Die kleinste durchschnittliche Entropie lieferte das Merkmal Betriebskosten.
Dieses Merkmal sollte demnach zuerst abgefragt werden.
Das Verfahren wird mit den sich dabei ergebenden Sub-Beispielmengen fortgesetzt.

Die Sub-Beispielmengen sind

Betriebskosten = *gering*

Kaufpreis	Aussehen	Klasse
mittel	***hässlich***	***indifferent***
gering	***hässlich***	***indifferent***
gering	***normal***	***empfehlenswert***
mittel	***normal***	***empfehlenswert***
gering	***gut***	***empfehlenswert***

Betriebskosten = *normal*

Kaufpreis	Aussehen	Klasse
gering	***hässlich***	***abratenswert***
hoch	***normal***	***indifferent***
mittel	***gut***	***empfehlenswert***
hoch	***hässlich***	***abratenswert***
hoch	***gut***	***indifferent***
gering	***gut***	***empfehlenswert***

Betriebskosten = *hoch*

Kaufpreis	Aussehen	Klasse
mittel	***normal***	***abratenswert***
mittel	***gut***	***abratenswert***
hoch	***normal***	***abratenswert***

Damit ist die erste Regel klar:
Klasse=*abratenswert* ← (Betriebskosten=*hoch*)
Weitere Regeln mit entsprechender Aussage über ***Betriebskosten*** in der ersten Prämisse entstehen durch Anwendung desselben Verfahrens auf o.g. Sub-Beispielmengen.

Für das Beispiel entsteht dieses Regelwerk zur Klassifikation:

Klasse = *abratenswert* ← (Betriebskosten=*hoch*)
Klasse = *indifferent* ← (Betriebskosten=*gering*) ∧ (Aussehen=*hässlich*)
Klasse = *empfehlenswert* ← (Betriebskosten=*gering*) ∧ (Aussehen=*normal*)
Klasse = *empfehlenswert* ← (Betriebskosten=*gering*) ∧ (Aussehen=*gut*)
Klasse = *abratenswert* ← (Betriebskosten=*normal*) ∧ (Aussehen=*hässlich*)
Klasse = *indifferent* ← (Betriebskosten=*normal*) ∧ (Aussehen=*normal*)
Klasse = *indifferent* ← (Betriebskosten=*normal*) ∧ (Aussehen=*gut*) ∧ (Kaufpreis=*hoch*)
Klasse = *empfehlenswert* ← (Betriebskosten=*normal*) ∧ (Aussehen=*gut*) ∧ (Kaufpreis=*mittel*)
Klasse = *empfehlenswert* ← (Betriebskosten=*normal*) ∧ (Aussehen=*gut*) ∧ (Kaufpreis=*gering*)

Vergleich der Ergebnisse

	left-to-right	***ID 3***
Anzahl entstandener Regeln	**13**	**9**
durchschnittliche Anzahl der Prämissen einer Regel	**2.6**	**2.2**

Das ID 3 - Verfahren kann leicht für die Verarbeitung unscharfer Klassenzugehörigkeiten adaptiert werden; es gibt ein ID 3 – Tool, welches dies unterstützt: ***1stClass*** ®

2.2.2.3 Eine prädikatenlogische Interpretation des Induktionsproblems

Gegeben sei

- eine Menge von Aussagen des PK1 $M = \{ A_1, ..., A_n \}$

Gesucht ist

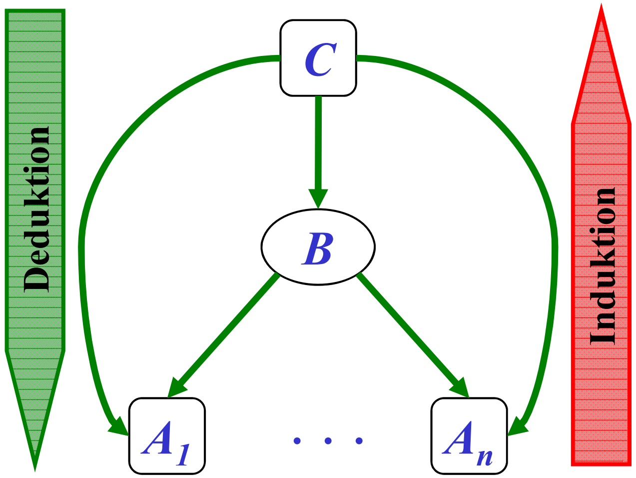
- eine („allgemeinere") Aussage B, welche alle A_i „begründet", d.h. es gilt $ag\ (B \rightarrow A_i)$ für jede „Beobachtung" A_i

Eine derartige Aussage B heißt induktiver Schluss für $M = \{ A_1, ..., A_n \}$.

I.allg. gibt es viele induktive Schlüsse B einer Menge von Aussagen $\{ A_1, ..., A_n \}$, welche unterschiedlich interessant sind. Die „beste" Begründung ist diejenige, welche die Aussagen $\{ A_1, ..., A_n \}$ am spezifischsten begründet:

B ist **bester induktiver Schluss** für $\{A_1, ..., A_n\}$, falls

1. Die Aussage B ist ein induktiver Schluss für $\{ A_1, ..., A_n \}$, d.h. $ag\ (B \rightarrow A_i)$ gilt für alle A_i und
2. Jeder andere induktive Schluss C ist auch induktiver Schluss für $\{ B \}$.



- Induktion im PK1 ist Ermittlung eine „speziellsten gemeinsamen allgemeinen Aussage“, d.h. eines speziellsten Antiunifikators.
- Ein speziellster Antiunifikator ist der (untere) Extremwert einer (zu definierenden) Relation $\sqsupseteq$ (sprich: *ist allgemeiner als*) über PK1-Ausdrücken.

1. Für quantorfreie Ausdrücke A und B gilt $B \sqsupseteq A$, falls $ag\,(B \rightarrow A)$.
2. Eine **Antiunifikation** einer Menge einfacher Ausdrücke $\{A_1, \ldots, A_n\}$ ist ein einfacher Ausdruck B, für welchen $\forall i\,(B \sqsupseteq A_i)$ gilt.
3. Die **speziellste Antiunifikation** Φ einer Menge einfacher Ausdrücke $\{A_1, \ldots, A_n\}$ ist diejenige Antiunifikation, für welche gilt:
   Für jede andere Antiunifikation C ist $C \sqsupseteq \Phi$.

- Die Termeinsetzung ϑ, welche die speziellste Antiunifikation Φ aus allen Ausdrücken $A_1, \ldots, A_n$ erzeugt, heißt **speziellster Antiunifikator** (*most specific antiunifier* = ***m.s.a.***) von $A_1, \ldots, A_n$:

  $\vartheta = \text{m.s.a.}(\{A_1, \ldots, A_n\})$, falls
  $\Phi = \vartheta(A_1) = \vartheta(A_2) = \ldots = \vartheta(A_n)$ speziellste Antiunifikation von $\{A_1, \ldots, A_n\}$ ist.

Ein Verfahren zur Ermittlung des speziellsten Antiunifikators 2er Terme

Antiunifikation versteht man am besten als ein Reduktionsprozess, der die zu lösende Aufgabe systematisch reduziert, für welchen *Termination* und *Korrektheit* zu zeigen ist.

Gegeben

- zwei Terme t_1 , t_2
- eine Markierungsmenge M , welche diejenigen Positionen in der zu konstruierenden Antiunifikation enthält, an denen noch etwas zu tun ist; initial enthält M das leere Wort: $M = \{ \varepsilon \}$

Gesucht

- die sukzessive zu konstruierende Substitution ϑ in Form einer Menge von Tripeln [*Variable*, *Ersetzung_in_*t_1 , *Ersetzung_in_*t_2] ; initial ist ϑ leer: $\vartheta = \emptyset$

- Das (potentiell nichtdeterministische) Reduktionssystem ist regelorientiert.
- Das Regelwerk wird auf ein 4-Tupel $[\ \vartheta(t_1)\ ,\ \vartheta(t_2)\ ,\ M\ ,\ \vartheta\ ]$ angewandt:
  - Ausgangssituation: $[\ t_1 , t_2 , \{ \varepsilon \} , \emptyset\ ]$
  - Zielsituation: $[\ \mathbf{t}\ ,\ \mathbf{t}\ ,\ \emptyset\ ,\ m.s.a.(t_1 , t_2)\ ]$

Notationen

- Positionen in Termen sind Wörter über der Menge der natürlichen Zahlen $I\!N$; das leere Wort ist ε .
- Wenn p eine Position im Term t ist, dann bezeichnet
  - die Notation $t.p$ das Funktionssymbol an dieser Position (zwei Funktionssymbole gelten nur dann als gleich, wenn sowohl der Funktionsname als auch die Stelligkeiten gleich sind, ansonsten sind sie ungleich)
  - die Notation $t|_p$ den an dieser Position beginnenden Teilterm.
- Die Notation $t[p \leftarrow X]$ bezeichnet denjenigen Term, welcher aus t nach der Ersetzung des an der Position p beginnenden Teilterms durch X entsteht.
- Die Projektion $\pi_i(M)$ einer Menge M von n-Tupeln ist die Menge der i-ten Komponenten eines jeden Tupels. $\pi_1(\vartheta)$ ist die Menge der Variablen, die in den Tripeln von ϑ als jeweils erste Komponente vorkommen.
- Die Notation $\alpha(f)$ bezeichnet die Stelligkeit eines Funktionssymbols f . Für Konstanten und Variablen ist die Stelligkeit 0.

Ein Regelwerk zur Ermittlung des speziellsten Antiunifikators 2er Terme (1)

R1 $p \in M\ ,\ t_1.p \neq t_2.p\ ,\ [\ X\ ,\ t_1|_p\ ,\ t_2|_p\ ] \in \vartheta$

$[\ t_1\ ,\ t_2\ ,\ M\ ,\ \vartheta\ ] \quad \Rightarrow [\ t_1[p \leftarrow X]\ ,\ t_2[p \leftarrow X]\ ,\ M\ \setminus \{\ p\ \}\ ,\ \vartheta]$

An einer zu bearbeitenden Stelle stehen verschiedene Funktionssymbole. Die an dieser Stelle beginnenden Teilterme sind bereits schon einmal durch eine Variable ersetzt worden.
Diese Teilterme werden auch an dieser Stelle durch dieselbe Variable ersetzt. Die bearbeitete Stelle wird aus M entfernt.

R2 $p \in M\ ,\ t_1.p \neq t_2.p\ ,\ \neg\exists\ X\!:\ [\ X\ ,\ t_1|_p\ ,\ t_2|_p\ ] \in \vartheta\ ,\ X \notin \pi_1(\ \vartheta\ )$

$[\ t_1\ ,\ t_2\ ,\ M\ ,\ \vartheta\ ] \quad \Rightarrow [\ t_1[p \leftarrow X]\ ,\ t_2[p \leftarrow X]\ ,\ M\ \setminus \{\ p\ \}\ ,\ \vartheta \cup \{\ [\ X\ ,\ t_1|_p\ ,\ t_2|_p\ ]\ \}\ ]$

An einer zu bearbeitenden Stelle stehen verschiedene Funktionssymbole. Die an dieser Stelle beginnenden Teilterme sind bislang nicht schon einmal durch eine Variable ersetzt worden. X ist eine bislang nicht benutzte Variablenbenennung.
Die Teilterme werden an dieser Stelle durch die neue Variable ersetzt. Die bearbeitete Stelle wird aus M entfernt. Die vorgenommene Ersetzung wird dem bisherigen Antiunifikator hinzugefügt.

Ein Regelwerk zur Ermittlung des speziellsten Antiunifikators 2er Terme (2)

R3 $p \in M\,,\ t_1.p = t_2.p\,,\ t_1|_p \neq t_2|_p\,,\ \alpha(t_1.p) = n$

$[\,t_1, t_2, M, \vartheta\,] \Rightarrow [\,t_1, t_2, (M \setminus \{p\}) \cup \{p1, \ldots, pn\}, \vartheta\,]$

An einer zu bearbeitenden Stelle stehen Terme mit demselben Funktionssymbol. Die an dieser Stelle beginnenden Teilterme sind jedoch verschieden voneinander. Die Stelligkeit der Funktionssymbole sei n.

An dieser Position ist nichts zu ändern; sie wird aus M entfernt. Die Teiltermpositionen müssen jedoch in die Menge der noch zu bearbeitenden Positionen aufgenommen werden.

R4 $p \in M\,,\ t_1|_p = t_2|_p$

$[\,t_1, t_2, M, \vartheta\,] \Rightarrow [\,t_1, t_2, M \setminus \{p\}, \vartheta\,]$

An einer zu bearbeitenden Stelle stehen identische Teilterme.

An dieser Position ist nichts zu tun; sie wird ersatzlos aus M entfernt.

Korrektheit des Regelwerks

1. *Nur R1 und R2 nehmen Termersetzungen vor. Da hierbei Terme stets durch Variablen ersetzt werden, entsteht in jedem Falle eine Antiunifikation.*
2. *R1 nimmt eine Ersetzung kompletter Terme vor. Sie wird nur dann angewandt, wenn genau diese Ersetzung an anderen Positionen notwendig war. Der Nachweis, dass dies die speziellste Ersetzung ist, verschiebt sich dorthin.*
3. *R2 behandelt verschiedener Funktionssymbole. Hier gibt es keine speziellere Ersetzung als eine Variable.*
4. *R3 behandelt gleiche Funktionssymbole. Um die Antiunifikation so speziell wie möglich zu halten, wird das Funktionssymbol in diese aufgenommen. Eventuelle Ersetzungen bleiben der Antiunifikation der Argumente mit gleichem Regelwerk vorenthalten.*
5. *R4 stellt klar, dass keine Ersetzung die speziellste Ersetzung identischer Teilterme ist.*

Demnach liefert das Regelwerk den speziellsten Antiunifikator.

Termination des Regelwerks

1. *R1, R2 und R4 verkleinern die Menge der zu bearbeitenden Positionen M jeweils echt.*
2. *R3 führt zu einer gegebenen Position aus M nur längere Positionen ein. Wegen der Endlichkeit der gegebenen Ausgangsterme ist R3 nur endlich oft anwendbar.*

Demnach wird M nach einer endlichen Anzahl von Regelanwendungen leer.

Ein Beispiel

$t_1 = p(\ f(a,b)\ ,\ f(a,b)\ ,\ X\ ,\ a\ )$ $\qquad$ $t_2 = p(\ f(a,X)\ ,\ g(a,b)\ ,\ b\ ,\ Y\ )$

- Ausgangssituation: $[p(\ f(a,b)\ ,\ f(a,b)\ ,\ X\ ,\ a\ )\ ,\ p(\ f(a,X)\ ,\ g(a,b)\ ,\ b\ ,\ Y\ )\ ,\ \{\ \varepsilon\ \}\ ,\ \emptyset\ ]$
- Zielsituation: $[\ t\ ,\ t\ ,\ \emptyset\ ,\ m.s.a.(\ t_1\ ,\ t_2\ )\ ]$

	t_1	t_2	M	ϑ
	$p(\ f(a,b)\ ,\ f(a,b)\ ,\ X\ ,\ a\ )$	$p(\ f(a,X)\ ,\ g(a,b)\ ,\ b\ ,\ Y\ )$	$\{\ \varepsilon\ \}$	$\emptyset$
R3	$p(\ f(a,b)\ ,\ f(a,b)\ ,\ X\ ,\ a\ )$	$p(\ f(a,X)\ ,\ g(a,b)\ ,\ b\ ,\ Y\ )$	{1,2,3,4}	$\emptyset$
R3	$p(\ f(a,b)\ ,\ f(a,b)\ ,\ X\ ,\ a\ )$	$p(\ f(a,X)\ ,\ g(a,b)\ ,\ b\ ,\ Y\ )$	{11,12,2,3,4}	$\emptyset$
R4	$p(\ f(a,b)\ ,\ f(a,b)\ ,\ X\ ,\ a\ )$	$p(\ f(a,X)\ ,\ g(a,b)\ ,\ b\ ,\ Y\ )$	{12,2,3,4}	$\emptyset$
R2	$p(\ f(a,X_0)\ ,\ f(a,b)\ ,\ X\ ,\ a\ )$	$p(\ f(a,X_0)\ ,\ g(a,b)\ ,\ b\ ,\ Y\ )$	{2,3,4}	$\{\ [X_0,b,X]\ \}$
R2	$p(\ f(a,X_0)\ ,\ X_1\ ,\ X\ ,\ a\ )$	$p(\ f(a,X_0)\ ,\ X_1\ ,\ b\ ,\ Y\ )$	{3,4}	$\{\ [X_0,b,X]\ ,$ $[X_1,f(a,b),g(a,b)]\ \}$
R2	$p(\ f(a,X_0)\ ,\ X_1\ ,\ X_2\ ,\ a\ )$	$p(\ f(a,X_0)\ ,\ X_1\ ,\ X_2\ ,\ Y\ )$	{4}	$\{\ [X_0,b,X]\ ,$ $[X_1,f(a,b),g(a,b)]\ ,$ $[X_2,X,b]\ \}$
R2	$p(\ f(a,X_0)\ ,\ X_1\ ,\ X_2\ ,\ X_3\ )$	$p(\ f(a,X_0)\ ,\ X_1\ ,\ X_2\ ,\ X_3\ )$	$\emptyset$	$\{\ [X_0,b,X]\ ,$ $[X_1,f(a,b),g(a,b)]\ ,$ $[X_2,X,b]\ ,$ $[X_3,a,Y]\ \}$

Einsetzen eines Antiunifikators in die Ursprungsterme

Unser Beispiel:

$t_1 = p(\,f(a,b)\,,\ f(a,b)\,,\,X\,,\,a\,)$

$t_2 = p(\,f(a,X)\,,\,g(a,b)\,,\,b\,,\,Y\,)$

$\vartheta = \{\ [X_0,b,X]\ ,\ [X_1,f(a,b),g(a,b)]\ ,\ [X_2,X,b]\ ,\ [X_3,a,Y]\ \}$

Einsetzungsregel:

Für jedes Tripel $[X,\ \vartheta_1(X),\ \vartheta_2(X)]$ aus ϑ werden diejenigen Positionen ermittelt, an denen in $t_1|_p = \vartheta_1(X)$ **und** $t_2|_p = \vartheta_2(X)$ ist und dort wird in beiden Ursprungstermen die Ersetzung durch X vorgenommen.

im Beispiel:

$[X_0,b,X]$	X_0 wird (nur) an Position 12 eingesetzt
$[X_1,f(a,b),g(a,b)]$	X_1 wird (nur) an Position 2 eingesetzt
$[X_2,X,b]$	X_2 wird (nur) an Position 3 eingesetzt
$[X_3,a,Y]$	X_3 wird (nur) an Position 4 eingesetzt

Noch ein Beispiel

$t_1 = p(f(a,b), f(a,b), X, b)$ $t_2 = p(f(a,X), g(a,b), b, X)$

- Ausgangssituation: $[p(f(a,b), f(a,b), X, b), p(f(a,X), g(a,b), b, X), \{\varepsilon\}, \varnothing]$
- Zielsituation: $[\mathbf{t}, \mathbf{t}, \varnothing, m.s.a.(t_1, t_2)]$

	t_1	t_2	M	ϑ
	$p(f(a,b), f(a,b), X, b)$	$p(f(a,X), g(a,b), b, X)$	$\{\varepsilon\}$	$\varnothing$
R3	$p(f(a,b), f(a,b), X, b)$	$p(f(a,X), g(a,b), b, X)$	{1,2,3,4}	$\varnothing$
R3	$p(f(a,b), f(a,b), X, b)$	$p(f(a,X), g(a,b), b, X)$	{11,12,2,3,4}	$\varnothing$
R4	$p(f(a,b), f(a,b), X, b)$	$p(f(a,X), g(a,b), b, X)$	{12,2,3,4}	$\varnothing$
R2	$p(f(a,X_0), f(a,b), X, b)$	$p(f(a,X_0), g(a,b), b, X)$	{2,3,4}	$\{[X_0,b,X]\}$
R2	$p(f(a,X_0), X_1, X, b)$	$p(f(a,X_0), X_1, b, X)$	{3,4}	$\{[X_0,b,X], [X_1,f(a,b),g(a,b)]\}$
R2	$p(f(a,X_0), X_1, X_2, b)$	$p(f(a,X_0), X_1, X_2, X)$	{4}	$\{[X_0,b,X], [X_1,f(a,b),g(a,b)], [X_2,X,b]\}$
R1	$p(f(a,X_0), X_1, X_2, X_0)$	$p(f(a,X_0), X_1, X_2, X_0)$	$\varnothing$	$\{[X_0,b,X], [X_1,f(a,b),g(a,b)], [X_2,X,b]\}$

Einsetzung des Antiunifikators

$t_1 = p(\, f(a,b) \;,\, f(a,b) \,,\, X \,,\, b\,)$
$t_2 = p(\, f(a,X) \,,\, g(a,b) \,,\, b \,,\, X\,)$
$\vartheta = \{\, [X_0,b,X] \,,\, [X_1,f(a,b),g(a,b)] \,,\, [X_2,X,b] \,\}$

$[X_0,b,X]$	X_0 wird an Position 12 und 4 eingesetzt
$[X_1,f(a,b),g(a,b)]$	X_1 wird (nur) an Position 2 eingesetzt
$[X_2,X,b]$	X_2 wird (nur) an Position 3 eingesetzt

Antiunifikation einer Menge von Termen

$$\boldsymbol{m.s.a.}(\{t_1, t_2\}) \quad = \quad \boldsymbol{m.s.a.}(t_1, t_2)$$

$$\boldsymbol{m.s.a.}(\{t_1, t_2, t_3, \dots, t_n\}) \quad = \quad \boldsymbol{m.s.a.}(\{\vartheta(t_1), t_3, \dots, t_n\})$$

mit $\vartheta = \boldsymbol{m.s.a.}(t_1, t_2)$

Botschaft

Wann immer Gemeinsamkeiten mehrerer Objekte identifiziert werden müssen, d.h. „Schablonen" entworfen werden sollen, kann man dieses Verfahren entsprechend adaptieren und anwenden.

2.2.2.4 Eine fallbasierte Interpretation des Induktionsproblems

Gegeben sei

- eine Menge von Objekten I
- ein k-dimensionaler (endlicher) Merkmalsraum, in welchem jeden Objekt $i \in I$ genau ein Punkt zugeordnet werden kann
- eine Partitionierung von I in eine endliche Menge von Objektklassen $U = \{ U_1, \ldots, U_n \}$
- eine Beispielmenge $B \subset I$: eine Menge von Objekten mit bekannter Klassenzugehörigkeit

Gesucht ist

- die wahrscheinlichste Klassenzugehörigkeit eines neuen Beispiels (ohne den Klassifikator explizit zu ermitteln)

Warum „fallbasierte Interpretation"?

- ➢ Fallbasiertes Schließen (CBR) beruht auf einer Fallbasis, in welcher (a) ein „ähnlichster" Fall gesucht wird und (b) dessen Lösung angepasst wird.
- ➢ Hier entartet (a) zum wahrscheinlichsten Fall und (b) zur Kopie von dessen Lösung.

Ein Beispiel

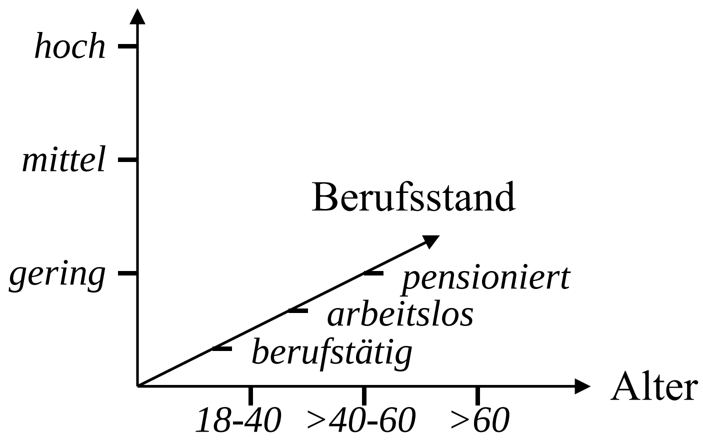
Gegeben sei

- eine Menge von Objekten: *Bewohner der Stadt Ilmenau*
- ein dreidimensionaler (endlicher) Merkmalsraum, in welchem jeden Objekt genau ein Punkt zugeordnet werden kann: *Abbildung Merkmalsraum*
- vier disjunkte Objektklassen: *Bewohner mit einem täglichen Fernsehkonsum von (1) weniger als 1 Stunde, (2) >1 – 2 Stunden, (3) >2 – 3 Stunden und (4) mehr als 3 Stunden*
- 20 Beispiele aus dem Bekanntenkreis: *Tabelle Beispielmenge*

Gesucht ist

- die wahrscheinlichste Klassenzugehörigkeit eines 25-jährigen berufstätigen Bewohners mit mittlerem Einkommen

Merkmalsraum:



Beispielmenge:

Alter	Ein-kommen	Berufs-stand	Klasse
22	*mittel*	*berufstätig*	*>1 – 2 h*
43	*hoch*	*berufstätig*	*>1 – 2 h*
45	*gering*	*berufstätig*	*>1 – 2 h*
76	*mittel*	*pensioniert*	*>3 h*
34	*gering*	*arbeitslos*	*>1 – 2 h*
34	*mittel*	*berufstätig*	*>2 – 3 h*
56	*hoch*	*berufstätig*	*>3 h*
32	*mittel*	*berufstätig*	*>1 – 2 h*
23	*gering*	*arbeitslos*	*>2 – 3 h*
65	*mittel*	*pensioniert*	*>2 – 3 h*

Alter	Ein-kommen	Berufs-stand	Klasse
33	*gering*	*arbeitslos*	*>3 h*
36	*hoch*	*berufstätig*	*>1 – 2 h*
25	*gering*	*arbeitslos*	*≤ 1 h*
47	*mittel*	*berufstätig*	*≤ 1 h*
45	*hoch*	*berufstätig*	*>2 – 3 h*
63	*mittel*	*pensioniert*	*>3 h*
51	*hoch*	*berufstätig*	*>1 - 2 h*
60	*mittel*	*pensioniert*	*>1 – 2 h*
31	*gering*	*arbeitslos*	*>1 – 2 h*
39	*mittel*	*berufstätig*	*>2 – 3 h*

BAYES'sche Formel *(hier: logisch interpretiert, Literatur: mengenalgebraisch notiert)*

1. Seien A und B Aussagen über das Eintreffen von (nicht unmöglichen) Ereignissen sowie P(A) und P(B) Wahrscheinlichkeiten des Eintreffens mit $0 < P(A), P(B) \leq 1$.
2. Die bedingte Wahrscheinlichkeit von A unter der Bedingung B ist $P(A|B) = \frac{P(A \wedge B)}{P(B)}$
3. Sei $\Omega = B_1 \vee \ldots \vee B_n$ ein sicheres Ereignis, bestehend aus paarweise unvereinbaren Ereignissen $B_1, \ldots, B_n$: $P(B_1 \vee \ldots \vee B_n) = \sum_{i=1}^{n} P(B_i) = 1$, $P(B_i \wedge B_j) = 0$ für $i \neq j$
4. Für ein zufälliges Ereignis A gilt: $A = A \wedge \Omega = A \wedge (B_1 \vee B_2 \vee \ldots \vee B_n)$
5. Lt. 2. ist $P(A) = P(A \wedge \Omega) = P(A) * \sum_{i=1}^{n} P(B_i) = \sum_{i=1}^{n} P(A \wedge B_i) = \sum_{i=1}^{n} P(B_i) * P(A|B_i)$

Fragt man nach der Wahrscheinlichkeit eines Ereignisses B_i unter der Bedingung, dass A bereits eintrat (*Wahrscheinlichkeit a posteriori*), ergibt sich

$$P(B_i | A) = \frac{P(B_i \wedge A)}{P(A)} = \frac{P(B_i) * P(A|B_i)}{P(A)} = \frac{P(B_i) * P(A|B_i)}{\sum_{i=1}^{n} P(B_i) * P(A|B_i)}$$

BAYES'sche Formel

BAYES'sches Lernen: Ermittlung der wahrscheinlichsten Klassenzugehörigkeit

geg.:

- Beispiele, d.h. Objekte, für die die Ausprägungen aller Merkmale A_i (i=1,..,n) sowie die Klassenzugehörigkeit B_j (j=1,..,k) bekannt ist

ges.:

- diejenige Klasse $K \in \{B_1,...,B_k\}$ die für ein neues Objekt mit bekannten Merkmalsausprägungen A_i am wahrscheinlichsten ist

$$K = B_j : \max\{P(B_j \mid A_1,\dots,A_n)\}$$

$$= B_j : \max\{\frac{P(A_1,\dots,A_n \mid B_j) * P(B_j)}{P(A_1,\dots,A_n)}\}$$

lt. BAYES'scher Formel

$$= B_j : \max\{P(A_1,\dots,A_n \mid B_j) * P(B_j)\}$$

Vernachlässigung der Division, da konstanter Nenner

$$= B_j : \max\{\prod_{i=1}^{n} P(A_i \mid B_j) * P(B_j)\}$$

Annahme unabhängiger Merkmale, so dass deren UND-Verknüpfung deren Wahrscheinlichkeiten multipliziert

BAYES'sches Lernen am Beispiel (Folie #89)

Merkmale
- Alter ∈ { 18-40 , >40-60 , >60 }
- Einkommen ∈ { gering , mittel , hoch }
- Berufsstand ∈ { berufstätig , pensioniert , arbeitslos }

Klassen
- B_1: ≤ 1 h tgl. Fernsehkonsum
- B_2: >1-2 h tgl. Fernsehkonsum
- B_3: >2-3 h tgl. Fernsehkonsum
- B_4: > 3 h tgl. Fernsehkonsum

Beispiele: Tabelle Folie #90

gesucht: wahrscheinlichste Klassenzugehörigkeit eines Bewohners mit den Attributwerten

A_1: *Alter = 25* | A_2: *Einkommen = mittel* | A_3: *Berufsstand = berufstätig*

Wahrscheinlichkeiten, geschätzt durch relative Häufigkeit in der Beispielmenge

				Produkt der Zeile
$P(A_1\|B_1)$: 1/2	$P(A_2\|B_1)$: 1/2	$P(A_3\|B_1)$: 1/2	$P(B_1)$: 2/20	**0,01250**
$P(A_1\|B_2)$: 5/9	$P(A_2\|B_2)$: 3/9	$P(A_3\|B_2)$: 6/9	$P(B_2)$: 9/20	**0,05556**
$P(A_1\|B_3)$: 3/5	$P(A_2\|B_3)$: 3/5	$P(A_3\|B_3)$: 3/5	$P(B_3)$: 5/20	**0,05400**
$P(A_1\|B_4)$: 1/4	$P(A_2\|B_4)$: 2/4	$P(A_3\|B_4)$: 1/4	$P(B_4)$: 4/20	**0,00625**
			Maximum der Zeilenprodukte	**0,05556 (aus B_2)**

Der Bewohner gehört am wahrscheinlichsten zur Klasse B_2 (>1-2 h tgl. Fernsehkonsum).

Verfeinerung für zu kleine Beispielmengen

- Die geschätzte Wahrscheinlichkeiten der Zugehörigkeit zu einer Klasse A_i , zu welcher n_i der n Beispiele gehören, wird um eine gewichtete Wahrscheinlichkeit p‘ aus früheren Beobachtungsreihen ergänzt.
- Als Wichtungsfaktor m verwendet man häufig 1/Anzahl der Attributwerte des zu schätzenden Attributs, d.h. im o.g. Beispiel (Fernsehkonsum) m = 1/4 .

$$p = \frac{n_i}{n} \Rightarrow p = \frac{n_i + m * p'}{n + m}$$

Anwendung im Text-Mining (z.B. Suchmaschine für Textdokumente im www)

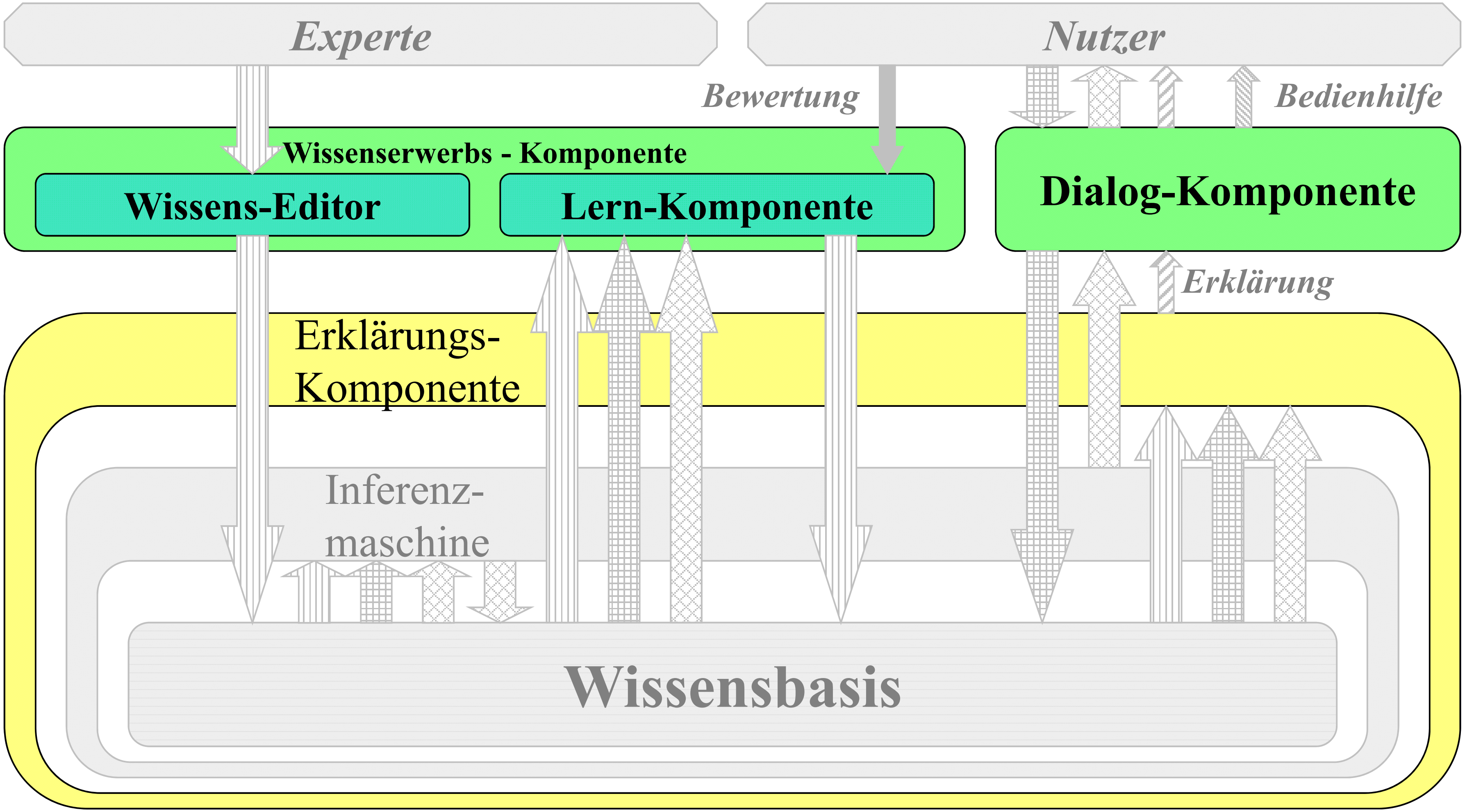
- Merkmale: mögliche Positionen im Text (1 ... n)
- Ausprägungen: Wort an der jeweiligen Position
- Klassen = { brauchbar , unbrauchbar }
- Beispielmenge: Textdokumente, welche bereits klassifiziert sind

2.2.3 Abduktion

- ... ist das Schließen von Folgen auf deren Ursachen
- ... und damit schon im Grundansatz falsch, denn $\{ A \rightarrow B \,,\, B \} \not\models A$
- ... aber bei diagnostischen Aufgaben eine übliche Praxis:

- ... wird um so legitimer,
  - je vollständiger die Ursachen in einem solchen Regelwerk erfasst werden und
  - je eindeutiger dazu Symptome zugeordnet werden können.

Abschnitte 2.3 ff. behandeln die verbleibenden Komponenten kurz:



2.3.1 Erklärungs-Komponente

Wozu dient sie?

1. Erklärung der Inferenzschritte
2. Instruktion von Nutzern (auch durch Lernen mit „unwirklichen“ Fällen)
3. „Debugger“ beim Systementwurf

Was wird erklärt?

1. das Zustandekommen von Fragen an den Nutzer (WHY-Erklärung)
2. das Zustandekommen der Zwischen- und Endergebnisse (HOW-Erklärung)
3. das bisher akquirierte fallspezifische Faktenwissen (WHAT-Erklärung)

Wie wird's erklärt?

1. in der Reihenfolge der Inferenzschritte (vorwärts)
2. von einem aktuellen Inferenzergebnis aus (rückwärts)

*Erklärungsinformationen sollen **informativ**, **anschaulich**, **instruktiv**, **nutzerangepasst** und **auf wesentliche Inhalte beschränkt** sein.*

2.3.2 Dialog-Komponente

Wozu dient sie?

1. Akquisition fallspezifischen Faktenwissens
2. Präsentation der Zwischen- und Endergebnisse
3. Präsentation der durch die Erklärungskomponente gelieferten Erklärungsinformation
4. Bedienhilfe

Welchen Anforderungen soll sie genügen?

1. Konsistenz *, d.h. gleichartige Aktionen sollen gleichartige Vorgänge auslösen.*
2. Adaptierbarkeit und Adaptivität *, d.h. nutzerangepasste ...*
   - *... Informationspräsentation und Terminologie*
   - *... Art und Detaillierungsgrad der Erklärungsinformation*
   - *... Möglichkeiten, Aktionssequenzen zusammenzufassen*
   - *... Interaktionstechnik (computer- oder nutzergeführter Dialog)*
3. Hilfekomponenten
   - *wahlfrei und/oder kontextorientiert*
   - *anforderungsorientiert und/oder selbsttätig*

2.3.3 Wissenserwerbs-Komponente

Erwerb bereichsspezifischen Expertenwissens ...

zunehmender Gegenstand der Forschung

(1) im Dialog mit dem Autor einer Wissensbasis

- Wissenseditierung in einer dem Autor vertrauten Darstellungsform
- ... mit Syntaxüberprüfung und -hilfe
- ... mit semantischen Tests: Inkonsistenzen, Redundanzen, logische Vollständigkeit, Einhaltung von Invarianzen und domänenspezifischer Gesetzmäßigkeiten usw.

(2) durch Lernen aus der Praxis

- ... mit Methoden der stochastischen Approximation
- ... mit Methoden des maschinellen Lernens aus Beispielen und Testfällen

(3) durch Mischformen (Lernen aus Testfällen mit anschließendem Refinement)

2.5 Wissenserwerbs-Komponente: Beispiele

Erwerb bereichsspezifischen Expertenwissens ...

(1) im Dialog mit dem Autor einer Wissensbasis

... mit semantischen Tests

VALENS (LibRT B.V., Amsterdam): *Eine Entwicklungsumgebung für Business-Rules mit integrierter Verifikation (Detektion verschiedenster Anomalien)*

(2) durch Lernen aus der Praxis

... mit Methoden der stochastischen Approximation

KNAUF, 1990 (Diss.): *Systematisches Erlernen von „Regelsignifikanzen" und „Relevanzen" von (Sub-) Wissensbasen*

... mit Methoden des maschinellen Lernens aus Beispielen und Testfällen

BAYES, 1761: BAYES'sche Formel

QUINLAN, 1983: *ID3*, **KNAUF**, 1993: Antiunifikation

(3) durch Mischformen (Lernen aus Testfällen mit anschließendem Refinement)

KNAUF, 2000 (Habil.): *Revision (reconstruction) of rules based on proven invalidities within a validation framework*